UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques Appliquées**

Arrêté ministériel : 7 août 2006

Présentée par

Ali HAJJ HASSAN

Thèse dirigée par **Sophie Lambert-Lacroix**
et codirigée par **François Pasqualini**

préparée au sein **du laboratoire TIMC-IMAG & STMicroelectronics**
dans **l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique (MSTII)**

Détection Multidimensionnelle au Test Paramétrique avec recherche automatique des causes

Thèse soutenue publiquement le **28 novembre 2014**,
devant le jury composé de :

M. Massih-Reza Amini
Professeur à l'Université Joseph Fourier (Grenoble 1), Président
M. Jean-Michel Poggi
Professeur à l'Université Paris Descartes, Rapporteur
M. Alain Rakotomamonjy
Professeur à l'Université de Rouen, Rapporteur
M. Vivian Viallon
Maître de conférences à l'Université Claude Bernard (Lyon 1), Examinateur
Mme. Sophie Lambert-Lacroix
Professeur à l'Université Pierre Mendès France (Grenoble 2), Directeur de thèse
M. François Pasqualini
Senior Member of Technical Staff chez STMicroelectronics Crolles, Co-Directeur de thèse

A mes chers parents et grâce à eux,

A mon coeur Abir,

A tous ceux que j'aime.

Remerciements

Tout travail réussi dans la vie nécessite en premier lieu la bénédiction de Dieu, et ensuite l'aide et le support de plusieurs personnes. Je tiens donc à remercier et à adresser ma reconnaissance à toute personne qui m'a aidé afin de réaliser ce travail.

Cette thèse Cifre a été réalisée dans le cadre d'une collaboration entre STMicroelectronics et le laboratoire TIMC-IMAG de l'Université de Grenoble. Je tiens à remercier l'ANRT (Association Nationale de la Recherche et de la Technologie) qui a partiellement financé cette thèse.

Je remercie vivement les membres du jury pour l'intérêt qu'ils ont porté à ce travail : M. Massih-Reza Amini, M. Vivian Viallon, M. Jean-Michel Poggi et M. Alain Rakotomamonjy, auxquels je dis un grand merci pour le temps qu'ils ont pris pour rapporter mon travail et pour leurs remarques constructives.

Je tiens à remercier mon direceur de thèse, Mme Sohpie Lambert-Lacroix, pour m'avoir encadré et conseillé, pour ses commentaires efficaces, et pour le temps qu'elle a consacré à diriger mes travaux.

Je remercie de tout mon coeur M. François Pasqualini, qui a co-dirigé ce travail avec une compétence et un professionnalisme exemplaires. Merci pour ses conseils, ses encouragements, sa sympathie, sa gentillesse, sa bonne humeur permanente et son remarquable sens de l'humour.

Un grand merci à tous mes amis et collègues à ST et TIMC. Je remercie tout particulièrement M. Alain Deleporte et M. Jean De-Caunes pour leur sympathie. Nos discussions pleines d'humour ont été des moments agréables.

J'adresse mes chaleureux remerciements à mes amis en France, vous étiez ma deuxième famille. Merci pour les inoubliables moments que nous avons passé ensemble.

Pour terminer, j'adresse mon grand amour à mes parents et ma femme. Mon plus grand souhait dans cette vie, c'est de les voir toujours à côté de moi, en bonne santé, heureux et que la paix soit avec eux.

Résumé

Détection Multidimensionnelle au Test Paramétrique avec Recherche Automatique des Causes

Ali Hajj Hassan

Aujourd'hui, le contrôle des procédés de fabrication est une tâche essentielle pour assurer une production de haute qualité. A la fin du processus de fabrication en semi-conducteur, un test électrique, appelé test paramétrique (PT), est effectué. PT vise à détecter les plaques dont le comportement électrique est anormal, en se basant sur un ensemble de paramètres électriques statiques mesurées sur plusieurs sites de chaque plaque. Le but de ce travail est de mettre en place un système de détection en temps réel au niveau de PT, pour détecter les plaques anormales à partir d'un historique récent de mesures électriques. Pour cela, nous développons un modèle de détection dynamique basé sur une technique de réapprentissage optimisée, où le modèle de détection est mis à jour à travers une fenêtre temporelle glissante. Notre modèle de détection est basé sur les machines à vecteurs supports à une classe (1-SVM), une variante de l'algorithme d'apprentissage statistique SVM, introduit dans le cadre des problèmes de classification à une classe pour la détection d'anomalies. Pour améliorer la performance prédictive de l'algorithme de classification 1-SVM, deux méthodes de sélection de variables ont été développées. La première méthode de type filtrage est basée sur un score calculé avec la méthode MAD_e, une approche robuste pour la détection univariée des valeurs aberrantes. La deuxième méthode de type wrapper est une adaptation à l'algorithme 1-SVM de la méthode d'élimination récursive des variables avec SVM (SVM-RFE). Pour les plaques anormales détectées, nous proposons une méthode permettant de déterminer leurs signatures multidimensionnelles afin d'identifier les paramètres électriques responsables de l'anomalie. Finalement, nous évaluons notre système proposé sur des jeux de données réels de STMicroelectronics, et nous le comparons au système de détection basé sur le test de T^2 de Hotelling, un des systèmes de détection les plus connus dans la littérature. Les résultats obtenus montrent que notre système est performant et peut fournir un moyen efficient pour la détection en temps réel.

Mots clés : détction d'anomalies, Machines à Vecteurs Supports, sélection de variables, Test Paramétrique, détection en temps réel, signature multidimensionnelle.

Abstract

Multidimensional Detection at Parametric Test with automatic diagnosis

by Ali Hajj Hassan

Nowadays, control of manufacturing process is an essential task to ensure production of high quality. At the end of the semiconductor manufacturing process, an electric test, called Parametric Test (PT), is performed. The PT aims at detecting wafers whose electrical behavior is abnormal, based on a set of static electrical parameters measured on multiple sites of each wafer. The purpose of this thesis is to develop a dynamic detection system at PT level to detect abnormal wafers from a recent history of electrical measurements. For this, we develop a real time detection system based on an optimized learning technique, where training data and detection model are updated through a moving temporal window. The detection scheme is based on one class Support Vector Machines (1-SVM), a variant of the statistical learning algorithm SVM widely used for binary classification. 1-SVM was introduced in the context of one class classification problems for anomaly detection. In order to improve the predictive performance of the 1-SVM classification algorithm, two variable selection methods are developed. The first one is a filter method based on a calculated score with MADe filter, a robust approach for univariate outlier detection. The second one is of wrapper type that adapts the SVM Recursive Feature Elimination method (SVM-RFE) to the 1-SVM algorithm. For detected abnormal wafers, we propose a method to determine their multidimensional signatures to identify the electrical parameters responsible for the anomaly. Finally, we evaluate our proposed system on real datasets of STMicroelecronics and compare it to the detection system based on Hotelling's T^2 test, one of the most known detection systems in the literature. The results show that our system yields very good performance and can provide an efficient way for real-time detection.

Keywords : anomaly detection, Support Vector Machines, variable selection, Parametric Test, real-time detection, multidimensional signature.

Table des matières

Présentation de STMicroelectronics

Présentation générale

Le groupe ST a été créé en juin 1987 à la suite du regroupement de Thomson Semiconducteurs (France) et de SGS Microelectronica (Italie). En mai 1998, SGS-THOMSON Microelectronics devient STMicroelectronics.

STMicroelectronics est une multinationale qui conçoit, développe, fabrique et commercialise une vaste gamme de circuits intégrés et de composants discrets utilisés dans de nombreuses applications. STMicroelectronics est le numéro 1 européen dans le secteur des semi-conducteurs.

Depuis sa création, la société a considérablement étendu et enrichi sa gamme de produits et de technologies, et renforcé son réseau de distribution et de fabrication en Europe, en Amérique du Nord et dans la région Asie-Pacifique. Ce processus d'expansion permanent se poursuit avec l'amélioration des sites existants et la construction de nouvelles usines de fabrication 300 mm submicroniques (taille inférieure au micron).

Activités

ST reçoit des plaques de silicium vierges qui serviront de base à la fabrication des puces microélectroniques. Ces puces interviennent dans plusieurs utilisations quotidiennes. En effet, chaque personne utilise chaque jour environ 250 circuits électroniques. Des exemples de ces utilisations sont données dans la Figure 1.

Les plaques de silicium vont suivre un long cheminement dans la salle blanche via les nombreux ateliers. La fabrication de circuits intégrés se divise en deux grandes parties comme le montre la Figure 2.

Front-End Cette activité consiste à produire des puces sur des plaques de silicium et d'en tester la fonctionnalité. Pour parvenir à une performance de pointe, les équipements

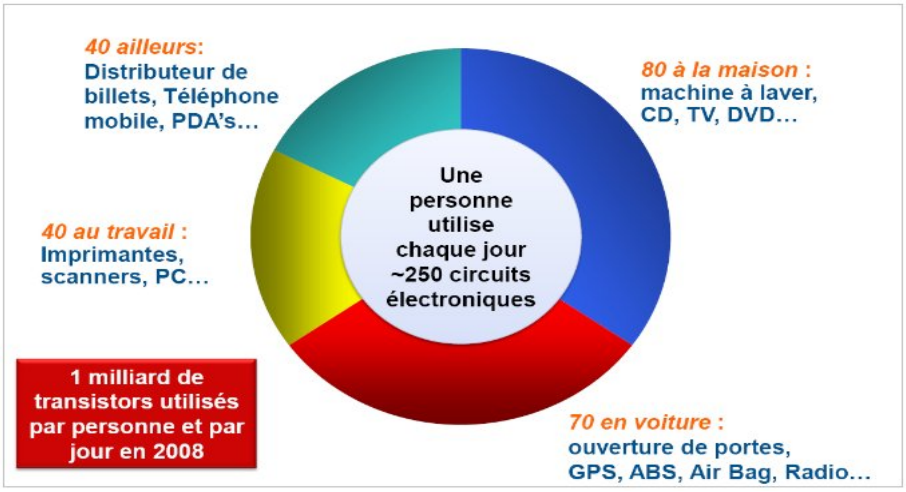

FIGURE 1 – Utilisation quotidienne des circuits électroniques.

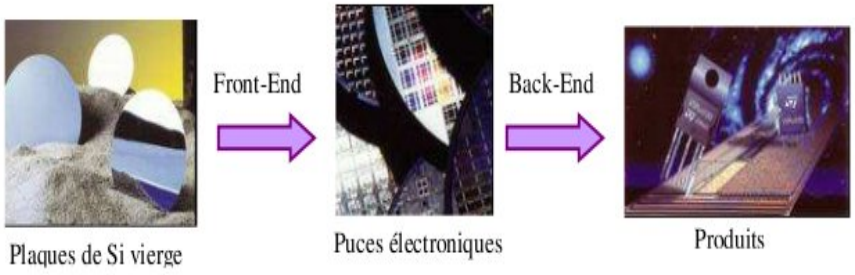

FIGURE 2 – Principales étapes de fabrication d'un Circuit Intégré

utilisés en salle blanche mettent en oeuvre les technologies les plus avancées afin de réaliser des motifs submicrométriques sur les plaquettes.

Back-End Les puces achevées dans les usines du Front-End sont envoyées dans les usines de Back-End pour être assemblées dans des boîtiers. Pour ce faire, les puces de la plaque sont découpées puis collées sur un support. Des fils sont ensuite soudés afin de relier la puce aux connections du boîtier avant de mouler le tout dans de la résine. Les circuits intégrés sont ensuite testés, marqués, emballés, puis expédiés aux clients qui les assembleront sur des circuits imprimés.

Secteurs d'application et clients

STMicroelectronics dispose de centres de recherche, de centres de conception, de sites de production et de bureaux de vente dans de nombreux pays. Son portefeuille d'applications couvre un grand nombre de secteurs :

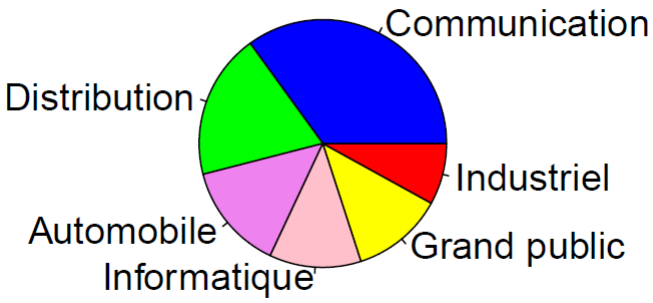

FIGURE 3 – Secteurs de vente du premier trimestre 2010

- Grand public : téléviseurs, lecteurs DVD, lecteurs MP3, LCD, ...
- Cartes à puce.
- Automobile : commandes de moteurs, injection électronique, multimédia, freinage, ABS, ...
- Communication : téléphones portables, circuits ADSL, bluetooth, modem WiFi, ...
- Périphériques informatiques : ordinateurs, disques durs, moniteurs, webcams, ...

La Figure 3 montre la distribution des secteurs de vente de STMicroelectronics pendant le premier semestre de l'année 2010. La société travaille avec plus de 1500 clients dont les principaux sont Samsung, Nokia, Delta, Philips, Thomson, Sony, Nintendo, Alcatel, Motorola, Ericsson, Siemens, Hewlett Packard, Seagate, Western Digital, IBM, Bosch, Ford, Daimler Chrysler.

Le site de Crolles

STMicroelectronics Crolles (cf. Figure 4) est situé à 15km au nord-est de Grenoble, au pied de la Dent de Crolles, dans la vallée séparant les deux chaînes de montagnes : la Chartreuse et Belledonne.

Crolles 1 (Crolles 200mm) Le site STMicroelectronics de Crolles conçu en 1989 a été inauguré en 1993. Ce nouveau site comporte alors deux divisions : la division chargée de la production de composants hautement complexes et la division Recherche et Développement (R&D). La production de puces électroniques est réalisée sur des plaques de silicium de 8 pouces (200mm de diamètre). La longueur de la grille du transistor varie de 350nm à 120nm. La capacité de production de Crolles 200mm est de 7200 plaques/semaine.

FIGURE 4 – Le site de Crolles

Crolles 2 (Crolles 300mm) En 2002, l'association entre STMicroelectronics, NXP (Philips Semiconductors) et Freescale (Motorola Semiconductors) mène à la création de l'Alliance Crolles 2 avec la construction d'une nouvelle salle blanche de $5000m^2$. Cette nouvelle unité de production plus moderne permet la fabrication de puces électroniques à partir de plaques de silicium de 12 pouces (300mm de diamètre). L'évolution technologique apportée par rapport à Crolles 1 est double : la taille des plaques sur lesquelles sont gravées les puces augmentent de 200mm à 300mm tandis que la longueur de la grille diminue (de 110nm jusqu'à 15nm). A la fin de l'année 2007, l'alliance Crolles 2 (STMicroelectronics, NXP, Freescale) prend fin et la salle blanche de Crolles 2 est rachetée entièrement par STMicroelectronics. La capacité de production de Crolles 300mm est de 3200 plaques/semaine.

Les salles blanches du site ST Crolles

Dans la salle blanche, on réalise différentes opérations technologiques sur des plaques de silicium pour fabriquer des puces. Lorsque ces puces sont mises en boîtier, on parle de circuits intégrés. Les éléments qui constituent la puce sont très petits (500 à 600 fois plus petit qu'un cheveu), les dimensions sont inférieures au micron (μm).

La présence de particules sur la plaquette de silicium au cours de la fabrication peut entraîner des problèmes de fonctionnement : puces hors service ou non fiables. Il faut donc protéger la plaquette des contaminants, c'est pourquoi on fabrique les puces dans une salle blanche.

Les contaminants sont générés par tout ce qui nous entoure. Il faut donc limiter l'accès à la salle blanche aux seuls éléments indispensables à la fabrication des puces, c'est-à-dire : les machines, les matières premières et les hommes. Mais comme tout génère des particules, il va aussi falloir limiter les apports de contaminations par les éléments entrant en salle blanches et par la salle blanche elle-même.

La salle blanche se compose de trois grandes parties distinctes (cf figure 5) :

- **le plénum :** partie supérieure de la salle blanche par laquelle se fait l'injection d'air. L'air étant prélevé à l'extérieur, il est préalablement traité pour être amené aux caractéristiques voulues ; taux d'empoussièrement (classe : nombre de particules de taille équivalente à 0.1 μm, contenu dans un volume d'1 m^3 d'air), humidité (40% $\pm$ 2%), température (21$\circ C \pm 0.5 \circ C$).
- **la salle blanche :** lieu où sont effectués les procédés. Cette salle est balayée par un flux laminaire verticale de 0,45m/s qui chasse les particules vers les bas et se trouve en surpression par rapport à l'extérieur (15 Pa) afin de maintenir un environnement ultra propre en évitant que l'air extérieur ne rentre dans la salle lors de l'ouverture d'une porte. Les cloisons de la salle blanche sont lisses et fabriquées d'une matière non contaminante : c'est de l'aluminium alvéolé revêtu d'une peinture époxy enrichie en carbone. Le carbone est conducteur et permet l'évacuation de l'électricité statique vers la terre. Cela évite que les murs attirent les particules.
- **le basement :** partie inférieure de la salle où se trouvent toutes les infrastructures qui n'ont pas besoin d'être en salle blanche et qui sont susceptibles de générer de la contamination (particulaire ou volatile). C'est là où sont regroupés les pompes (pour éviter les vibrations en salle) qui aspirent l'air de la salle ainsi que tous les circuits d'alimentations en produits chimiques, liquides ou gazeux.

Les étapes de la fabrication des puces

Les plaques de silicium vont subir environ 200 opérations pendant les 2 mois que dure la fabrication des puces. Elles vont faire de nombreux allers et retours dans les différents ateliers de la salle blanche.

Atelier Photolithographie la photolithographie permet de dessiner à la surface de la plaque la géométrie des composants. Elle définit des zones permettant les opérations technologiques suivantes.

Atelier Gravure Sèche / Etch : la gravure sèche permet d'enlever sélectivement la matière à l'aide de plasma. Elle intervient le plus souvent après la photolithographie.

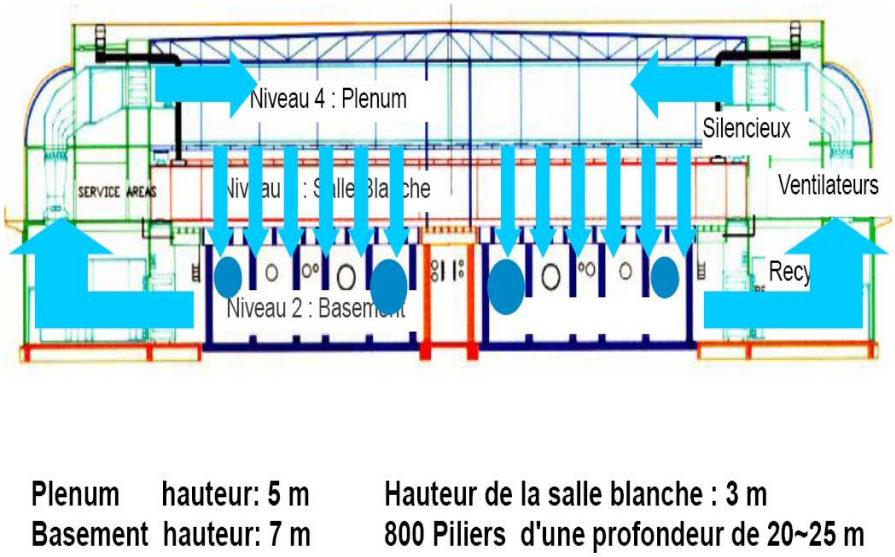

FIGURE 5 – Représentation schématique de la salle blanche

FIGURE 6 – Vue de la salle blanche

Atelier Gravures Humides / Wet : la gravure humide permet d'enlever sélectivement de la matière avec des produits chimiques liquides. Elle intervient principalement pour les nettoyages.

Atelier Implantation : réalisation d'implantations ioniques d'atomes spécifiques dans le silicium afin d'améliorer la conductivité.

Atelier CMP (Chemical Mechanical Polishing) : polissage mécano-chimique de la plaquette afin de réduire l'épaisseur des dépôts et de planariser les couches.

Traitement Thermiques : fabrication de couches isolantes sur la plaque par dépôt ou par oxydation dans des fours à très hautes températures.

Dépôt CVD (Chemical Vapor Deposition) / Diélectrique : dépôt de matières isolantes à la surface de la plaquette par méthode CVD à haute température pour l'isolation des couches métalliques et inter-métalliques.

Métal : dépôt de couches conductrices qui permettront de relier les composants entre eux pour assurer les fonctions électriques voulues.

Mise à Epaisseur : rabotage mécanique pour enlever la matière en face arrière, en fin de process, afin d'amincir la plaque avant le découpage des puces en vue de la mise en boîtier.

Test paramétrique : mesures électriques statiques permettant de valider les paramètres physiques des circuits réalisés (transistors, résistance, capacité).

Tri Electrique des Plaques (Electrical wafer Sorting) : vérification de la fonctionnalité de chaque puce. Les mauvaises puces sont encrées.

Qualité : contrôle qualité des plaques à certains niveaux de la création du circuit intégré.

Caractérisation : étude des défauts à différents niveaux du circuit par coupe ou «deprocessing» (élimination progressive des empilements de matériaux afin d'identifier les défauts générés lors de la fabrication).

Métrologie : mesure physique de différents paramètres physiques en cours de process (épaisseur, contamination...).

Introduction générale

Contexte et problématique

Aujourd'hui, le contrôle des procédés de fabrication est une tâche essentielle pour assurer le bon fonctionnement des processus et garantir par la suite une production de haute qualité. Ceci est un défi particulier lorsque les processus ont un grand nombre d'opérations et systèmes complexes, ce qui est le cas dans le processus de fabrication de dispositifs à semi-conducteurs et des circuits intégrés. En effet, les progrès permanents réalisés dans le domaine de l'intégration en microélectronique aboutissent à une complexification des circuits et à la réduction des tailles des puces. Ceci est dans le but de proposer sur le marché des composants compacts et plus performants.

La détection précoce et précise des défauts sur ces composants est alors nécessaire pour le maintien d'un processus dans sa condition optimale et réduire les coûts de fabrication. Sur les systèmes actuels (circuits comprenant des millions de transistors, cartes électroniques multicouches), les activités de test et de diagnostic prennent également de plus en plus d'importance tout en étant de plus en plus difficiles à réaliser.

Le processus de fabrication de puces électroniques regroupées en wafers (matériaux semi-conducteur utilisés dans la fabrication des puces) est un processus long et complexe. Il nécessite un contrôle de qualité répondant à des spécifications bien particulières. A la fin du processus de fabrication, un test électrique, appelé test paramétrique, est effectué. Ce test permet de faire un premier tri en sortie de production. La procédure PT vise à détecter les wafers dont le comportement électrique est anormal, en se basant sur un ensemble de paramètres électriques statiques (résistivité, conductivité, etc) mesurés avec des structures de test placées sur plusieurs sites du wafer. Par exemple, en 300mm (wafer dont le diamètre est de 300 mm), ces paramètres sont mesurés sur tous les lots et tous les wafers en neuf sites (emplacements sur chaque wafer) : un site à proximité du centre du wafer et huit sites répartis sur le wafer dont au moins cinq de ces huit sites qui doivent être à proximité du bord du wafer (cf. Figure 7). Le bon fonctionnement du wafer est vérifié en testant si les paramètres électriques mesurés ne s'écartent d'une

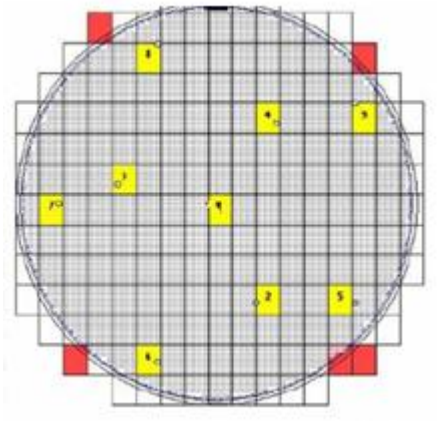

FIGURE 7 – Position des structures de test placées en 9 sites du wafer pour effectuer les mesures des paramètres électriques statiques en 300mm.

valeur cible (Target) située entre la valeur de spécification minimale (Lower Specification Limit LSL) et la valeur maximale (Upper Specification Limit USL). Ces deux valeurs s'appellent limites de spécification des paramètres électriques. La méthode de détection actuelle du test paramétrique est basée sur des règles d'arrêt consistant à l'évaluation de chaque paramètre individuel par rapport à ses propres limites de spécification pour identifier les wafers anormaux.

Pendant la procédure de test, les wafers passent par deux phases successives :

1. La phase qualifiée de "hold" consiste à écarter tous les lots (un lot est un groupement de 25 wafers qui ont parcouru ensemble toutes les étapes du processus) dont certains paramètres spécifiques mesurés sur chaque wafer composant ces lots sortent du domaine de spécifications. Ces wafers sont alors re-analysés par les ingénieurs au Test Paramétrique pour confirmer le mauvais fonctionnement du/des wafer(s) ou attribuer l'alerte à un problème de test. Ensuite, un "full test" (tests complémentaires) est réalisé en cas d'anomalie confirmée du/des wafer(s) en retirant ces derniers de la ligne de production afin d'y appliquer des tests plus spécifiques.
2. La deuxième phase vise à tester la fonctionnalité de chaque puce à l'aide du test EWS (Electrical Wafer Sort) pour garantir le bon fonctionnement selon des spécifications données.

La méthode de détection actuelle au PT est une approche univariée avec une efficience limitée : le taux d'arrêt (hold) peut aller jusqu'à 50% alors que le taux de rejet réel (scrap) est inférieur à 1%. De plus, les ingénieurs PT ne regardent que les paramètres électriques de type "Gated", car parmi tous les paramètres électriques mesurés, ils sont les seuls paramètres pour lesquels les limites de spécification sont parfaitement définies. Ces

paramètres peuvent être des paramètres de type qualité (Quality Q), ou des paramètres de type fiablité (Reliability R).

Objectifs de la thèse

Notre travail porte donc sur l'analyse des données issues du test paramétrique pour prédire l'état de fonctionnement des puces électroniques afin d'éliminer les pièces défectueuses en sortie de la chaîne de production. Nous envisageons, dans le but d'améliorer les gains de productivité, de trouver un moyen permettant de reconnaître le mauvais fonctionnement des puces dès la première phase de tests sans passer par des tests complémentaires, et sans attendre les tests finaux qui sont plus approfondis et très coûteux (comme le test EWS).

Notre but est de réduire au maximum le taux de d'arrêt et éventuellement de le ramener au taux de rejet. Autrement dit, notre travail consiste à proposer un système capable de détecter dès la première phase de mesure les wafers anormaux en minimisant conjointement le nombre de wafers incorrectement rejetés (réellement bons, ils sont appelés dans ce cas des fausses alarmes) et le nombre de wafers anormaux non détectés.

Pour cela, l'objectif principal de la thèse est de développer une approche de détection multivariée basée sur l'apprentissage statistique permettant :

1. d'améliorer la qualité et l'efficience de détection : identifier automatiquement les wafers anormaux en réduisant le temps d'analyse.
2. de pouvoir détecter des wafers atypiques hors population mais dans les limites de spécifications.
3. de fournir les signatures multidimensionnelles de ces wafers anormaux pour pouvoir remonter à la cause de l'anomalie.

Concernant le deuxième point, le but est d'identifier des wafers qui ne ressemblent pas à la population normale, mais qui ne peuvent pas être détectés avec l'approche actuelle chez ST car les paramètres électriques sont dans les limites de spécifications. La détection rapide de ce type de wafers permet d'anticiper d'éventuels problèmes de rendement rencontrés au test final EWS.

Notre approche multivariée utilise une méthode de classification pour détecter les wafers anormaux et une méthode de sélection de variables qui permet d'améliorer la performance de détection en réduisant la dimension de l'espace de variables, et de sélectionner les paramètres électriques mis en cause.

Les approches multivariées sont plus adéquates pour le suivi d'un processus complexe. Elles sont donc utilisées pour traiter les données multivariables dans le but d'en extraire les informations pertinentes nécessaires pour prendre des décisons critiques. Leur principal avantage, par rapport aux approches univariées, est la prise en compte de la corrélation qui peut exister entre les variables.

Enfin nous envisageons exploiter l'approche de détection multivariée que nous avons développée dans le cadre d'une application industrielle. D'où la nécessité d'implémenter un modèle de detection en temps réel qui détecte les wafers anormaux à partir d'une fenêtre temporelle glissante de mesures électriques.

État de l'art

Les approches statistiques multivariées ont été utilisées avec succès pour la surveillance des processus industriels [20, 50, 51, 97]. L'Analyse en Composantes Principales (ACP) est une approche multivariée qui a reçu une attention particulière et a été largement utilisée pour la détection de défauts dans les processus industriels. C'est l'une des techniques linéaires de compression les plus largement utilisées pour extraire des informations pertinentes à partir de données de grande dimension. Le but de l'ACP est de réduire la dimension des données originelles en les projetant dans un sous-espace de dimension réduite, minimisant la déformation du nuage de points initial. Après avoir réduit la dimension, la variabilité d'un processus est capturée par le suivi de la statistique T^2 de Hotelling ou de la surveillance des résidus (Q chart)[35]. Dans le cas des processus non linéaires, une analyse en composantes principales non-linéaire KPCA (Kernel PCA) a été utilisée pour traiter la non-linéarité à l'aide des fonctions noyaux [56] .

L'ACP a été utilisée pour développer respectivement un modèle statique (test hors ligne) et un modèle dynamique (détection en temps-réel) pour la détection de défauts dans les installations de traitement biologique des eaux usées [7, 35]. Elle a été aussi utilisée pour la détection de défauts dans un procédé de gravure en taille-douce sur une plaque métallique (etch process) [97]. Des modèles statiques de détection basés sur l'ACP ont aussi été construits pour la détection d'anomalies dans les processus industriels des semi-conducteurs au niveau SPC (Statistical Process Control) [61] et au Test Paramétrique (PT) [65].

Une méthode alternative basée sur l'apprentissage statistique est les Machines à Vecteurs Supports à une classe (1-SVM) [76], une variante du SVM classique [92]. 1-SVM est une des méthodes de classification à une classe les plus connues pour la détection d'anomalies. Un modèle statique de détection basé sur 1-SVM a été utilisé pour la détection de défauts

dans un procédé de gravure en semi-conducteur, et dans un problème de simulation de génie chimique [61]. Il a été démontré que la méthode 1-SVM est performante dans les deux applications considérées. De plus, cette méthode est plus performante que la technique de détection basée sur l'ACP. Même dans les cas non-linéaires, des expériences de simulation ont montré que la technique 1-SVM détecte mieux que la méthode d'ACP non-linéaire KPCA.

La méthode 1-SVM a été appliquée avec succès dans d'autres domaines comme la médecine, la sécurité, l'informatique, etc. Nous citons quelques d'exemples d'application :

- la détection des tumeurs cérébrales par segmentation des images IRM (Imagerie par Résonance Magnétique) [101].
- la détection des infections nosocomiales [21].
- La détection des accès anormaux à la base de registre utilisée par le système d'exploitation Windows [41].
- la détection des activités anormales ou suspectes sur la cible analysée (un réseau informatique pae exemple) par un système de détection d'intrusion [96].
- la classification d'images de télédétection [67].

Contributions

Dans ce travail, nous considérons le problème de la détection automatique des wafers dont le comportement électrique est atypique, un problème devenant de plus en plus important dans l'industrie des semi-conducteurs. Nous développons un modèle de détection basé sur l'algorithme de classification à une classe 1-SVM. Comme les wafers sont décrits par plusieurs centaines de paramètres électriques, une sélection de paramètres pertinents est nécessaire. Pour accomplir la tâche de sélection de variables, nous avons développé une nouvelle méthode de "ranking" de type filtrage qui utilise le score de valeurs aberrantes dans chaque variable pour obtenir une hiérarchie des variables. Ce score est calculé avec le filtre MAD_e, une technique robuste de détection univariée des valeurs aberrantes. Une deuxième méthode de "ranking" pour la sélection de variables est présentée. C'est une méthode de type wrapper basée sur un score calculé à partir de l'algorithme 1-SVM. Ce score représente la variation du vecteur de poids dans cet algorithme. Cette méthode est une adaptation à l'algorithme 1-SVM de la méthode SVM-RFE (Recursive Feature Elimination with SVM), une méthode d'élimination récursive des variables basée sur un score calculé avec les méthodes SVM.

Le modèle de détection en temps réel exploite une technique de réapprentissage dynamique optimisée consistant en une mise à jour du modèle de détection (une des méthodes de sélection de variables et l'algorithme 1-SVM) sur un historique glissant de données.

Après avoir détecté les wafers anormaux, nous souhaitons définir les signatures multidimensionnelles de ces wafers pour pouvoir remonter à la cause. Pour cela nous avons proposé une méthode permettant d'identifier les paramètres électriques qui ont déclenché l'anomalie. Ceci se fait en déterminant la distance normalisée par rapport à la population normale, des paramètres retenus par la méthode de sélection de variables utilisée.

D'où nos principales contributions réalisées pendant cette thèse sont :
- Le développement des méthodes de sélection de variables appropriées à l'algorithme 1-SVM.
- Le choix optimal des paramètres de l'algorithme 1-SVM et des méthodes de sélection de variables associées.
- Le développement d'un modèle de détection en temps réel exploitant la méthode 1-SVM avec une méthode de sélection de variables suivant une technique de réapprentissage dynamique.
- L'optimisation du scénario de réapprentissage dynamique.
- La définition de la signature multidimensionnelle des wafers anormaux.

A notre connaissance, notre travail est le premier qui implémente un système de détection en temps réel dans le domaine des semi-conducteurs, et en même temps le premier qui développe un modèle dynamique basé sur la méthode 1-SVM. Il est aussi le premier travail qui traite la sélection de variables avec la méthode 1-SVM. En effet, nos données contiennent des centaines de variables, alors que les travaux existants sur la détection de défauts dans les procédés industriels ont traité des données avec quelques dizaines de variables.

Organisation de la thèse

Ce manuscrit est constitué d'une introduction générale, de cinq chapitres principaux et une conclusion générale.

Chapitre 1 : Détection d'anomalies.

Nous commençons ce chapitre par une présentation générale du contexte de la détection d'anomalies introduite dans le cadre d'un problème de classification à une classe. Puis nous parlons de différents aspects ainsi que les difficultés rencontrées dans un problème de détection d'anomalies. Ensuite nous décrivons les techniques

basées sur la classification et nous nous intéressons en particulier aux méthodes de machines à vecteurs support (SVM), considérées parmi les méthodes les plus répandues en apprentissage automatique introduites initialement pour la classification binaire. Les méthodes SVM ont été appliquées pour la détection d'anomalies en les adaptant aux techniques de classification à une classe. Nous argumentons ainsi l'utilisation de ces méthodes comme un choix principal dans notre travail consistant à détecter les wafers anormaux dans le domaine des semi-conducteurs. Nous présentons aussi des techniques basées sur la distance, notamment le clustering, les voisins les plus proches et le T^2 de Hotelling. Nous choisissons cette dernière technique, fréquemment utilisée en milieu industriel, comme une méthode de détection alternative aux méthodes SVM. A la fin de ce chapitre sont présentées les techniques statistiques et leurs limitations.

Chapitre 2 : Machines à Vecteurs Supports (SVM).

Dans la première partie de ce chapitre, nous présentons en détail les machines à vecteurs supports dans le cadre de la classification binaire. L'objectif des SVM est de déterminer un hyperplan linéaire qui donne une séparation otpimale des données d'apprentissage provenant de deux classes distinctes. Nous formulons alors deux algorithmes SVM : le premier déterminant un hyperplan à marge maximale, et le deuxième déterminant un hyperplan à marge souple (C-SVM) afin de permettre à quelques observations d'être mal classées dans le cas où les données ne sont pas séparables. Ensuite nous montrons comment les méthodes SVM peuvent traiter le problème de séparation non-linéaire grâce aux fonctions noyaux. Afin d'obtenir un modèle peformant, nous considérons le problème de sélection des paramètres du modèle ainsi que le choix du noyau. Ensuite, nous donnons une autre reformulation du problème de l'hyperplan à marge souple (ν-SVM) en substituant le paramètre C dans la fonction objective d'optimisation par un paramètre ν ayant une meilleure interprétation en terme de vecteurs de support et des erreurs d'apprentissage. Dans la deuxième partie de ce chapitre, nous présentons les machines à vecteurs supports à une classe (1-SVM), la méthode de détection principale dans notre travail. La méthode 1-SVM, qui est une variante de l'algorithme ν-SVM, est parmi les méthodes de classification à une classe les plus connues pour la détection d'anomalies. L'idée de base est de créer un hyperplan séparant la majorité des données d'apprentissage de l'origine avec une marge maximale. Des démonstrations théoriques relatives à l'algorithme 1-SVM sont données à la fin du chapitre.

Chapitre 3 : Réduction de dimension.

Dans le cas où la dimension de l'espace des variables est élevée, une réduction de la dimensionalité est indispensable pour améliorer les performances prédictives des méthodes de détection utilisées. Cette réduction peut être réalisée avec une

compression ou une sélection de variables. Les approches de type compression de variables transforment les données d'un espace de grande dimension dans un espace de dimension plus petite. Nous présentons dans ce cadre l'analyse en composantes principales (ACP). Puis nous considérons les approches de sélection de variables qui essaient de trouver un sous-ensemble optimal des variables originelles. Nous nous intéressons aux approches appropriées aux méthodes de classification à une classe, et à l'algorithme 1-SVM en particulier. Nous décrivons tout d'abord notre méthode de filtrage basée sur le filtre MAD_e, puis une autre méthode de type wrapper que nous avons adaptée à l'algorithme 1-SVM de la méthode SVM-RFE.

Chapitre 4 : Quelques résultats de la détection avec 1-SVM en mode statique.

Nous commençons tout d'abord par introduire les modèles qui vont servir à la détection d'anomalies dans notre étude. Nous présentons les mesures de performance utilisées pour évaluer ces modèles. Puis nous parlons de la phase de prétraitement des données nécessaires pour appliquer le modèle de détection comprenant essentiellement la transposition des données et le traitement des mesures hors limites de validité. Ensuite nous donnons une description détaillée de deux jeux de données réels de STMicroelectronics exploités dans notre étude, avant de présenter quelques résultats de détection en mode statique. L'influence de l'hyperparamètre γ du noyau gaussien sur la performance de l'algorithme 1-SVM est étudiée, ainsi que l'importance de la sélection de variables pour améliorer la pertinence de détection avec cet algorithme. Enfin, nous étudions l'influence de la présence d'anomalies dans l'étape d'apprentissage de 1-SVM.

Chapitre 5 : Détection en temps réel basée sur 1-SVM.

Ce dernier chapitre constitue une application industrielle des systèmes de détection développés. Nous commençons par décrire notre système de détection en temps réel basé sur un modèle dynamique de 1-SVM avec l'une des deux méthodes de sélection de variables spécifiquement développées. Nous décrivons également les deux scénarios de mise à jour de la fenêtre mobile de ce système, et nous expliquons le choix des hyperparamètres optimaux. Ce système est ensuite appliqué aux deux jeux de données réels de STMicroelectronics. Ce système est comparé à un système alternatif basé sur le test de T^2 de Hotelling qui est très connu dans le domaine de détection de défauts en industrie. Enfin, les signatures multidimensionnelles des wafers anormaux détectés par notre système sont déterminées afin d'identifier le ou les paramètre(s) électrique(s) responsable(s) de l'anomalie.

Nous concluons cette thèse en exposant les points forts de nos contributions et les perspectives de recherche dans ce domaine.

Publications effectuées pendant la thèse

Articles dans des revues :

- A. HAJJ HASSAN, S. Lambert-Lacroix and F. Pasqualini. Real-time fault detection in semiconductor using one-class support vector machines. *International Journal of Computer Theory and Engineering*, 7(3) : 191-196, 2015.

Conferences internationales :

- A. HAJJ HASSAN, S. Lambert-Lacroix and F. Pasqualini. A new approach of One Class Support Vector Machines for Detecting Abnormal Wafers in Semi-conductor. *In Proceedings of MSDM, Meeting on statistics and Data Mining*, pages 35–41, Hammamet, Tunisie, March 2013.
- A. HAJJ HASSAN, S. Lambert-Lacroix and F. Pasqualini. A new multivariate statistical approach for fault detection in semiconductor. *Abstract for the 13th European Advanced Process Control and Manufacturing Conference*, Dresden, Germany, April 2013.
- A. HAJJ HASSAN, S. Lambert-Lacroix et F. Pasqualini. Une nouvelle approche multivariée pour la détection de défauts dans le domaine de semi-conducteur. *In Proceedings of the 45e Journées de Statistique*, Toulouse, France, Mai 2013.

Chapitre 1

Détection d'anomalies

1.1 Classification à une classe

La classification multiclasse est un concept très important en apprentissage automatique et en fouille de données (data mining). En général, l'objectif des algorithmes de classification multi-classe consiste à classifier un objet inconnu parmi plusieurs classes prédéfinies. Cependant, parfois il n'est pas nécessaire de classer les objets dans deux ou plusieurs classes. Dans de nombreuses applications, les modèles d'apprentissage construits pour la classification visent à détecter des observations peu fréquentes considérées comme atypiques parmi une grande majorité des observations appartenant à une classe cible. Dans ce cas, Une classification binaire (à deux classes) peut conduire à une situation où l'apprentissage est déséquilibré : les observations de la classe cible sont en sur-apprentissage et les observations atypiques en sous-apprentissage. Dans de telles situations, le problème de classification à une classe (ou monoclasse) peut être considéré afin de construire un classifieur approprié.

Le problème de classification à une classe se caractérise donc par la présence d'une classe cible. on suppose que seules les données de cette classe sont disponibles pour l'apprentissage du classifieur, alors que l'ensemble de test comprend des exemples positifs (classe cible) et négatifs (classes d'anomalies). Un classifieur monoclasse cherche à décrire la distribution des exemples positifs et de traiter les exemples négatifs comme des anomalies qui peuvent être détectées sans avoir à apprendre explicitement leur apparition.

Une classification à une classe est aussi connue comme une détection des objets aberrants (outlier detection [8, 98]), une détection de la nouveauté (novelty detection [59]) ou une détection d'anomalies (anomaly detection [70]). La détection d'anomalies se réfère au problème de trouver les observations dans les données qui ne sont pas conformes

au comportement normal prévu. Une anomalie est donc une observation qui est considérablement différente, divergente, dissemblable ou distincte, du reste des données. Le problème est alors de définir cette dissimilarité entre objets. Typiquement, celle-ci est estimée par une fonction calculant la distance entre objets, la tâche suivante consiste à déterminer les objets les plus éloignés de la masse. Ces objets anormaux sont souvent appelés des anomalies, des observations aberrantes (outliers), des observations discordantes, des exceptions, des défauts, des erreurs ou des nouveautés dans différents domaines d'application.

La détection d'anomalies a été largement étudiée et utilisée dans une grande variété de domaines d'application tels que la détection des défauts dans les processus industriels [30, 61, 80, 84] qui est notre domaine d'intérêt, la détection de la fraude à la carte bancaire [11, 68] et dans l'assurance [11, 94], la détection d'intrusion [54], la santé publique [58, 85], la reconnaissance des paroles [4] et de nombreux autres domaines.

L'importance de la détection d'anomalies est due au fait que les anomalies traduisent des informations importantes (et souvent critiques) dans une grande variété de domaines d'application. Par exemple, les anomalies dans les données de transaction de cartes de crédit pourraient indiquer le vol de carte de crédit. De même, les anomalies dans les données de santé publique sont largement utilisés pour détecter les tendances anormales dans les dossiers médicaux des patients qui pourraient être des symptômes d'une nouvelle maladie. Les anomalies dans notre étude correspopndent à des éventuels wafers défectueux, et leur détection nous permet de gagner en terme de temps et de coût de production.

L'application de la détection d'anomalies dans un grand nombre de domaines a donné lieu à une grande diversité de techniques de détection d'anomalies. Un grand nombre de ces techniques a été développé pour résoudre des problèmes ciblés se rapportant à un domaine d'application particulier, tandis que d'autres ont été développé de façon plus générique.

Dans la suite de ce chapitre, nous parlerons des défis rencontrés dans un problème de détection d'anomalies ainsi que des différents aspects de ce problème. Puis nous présenterons les différentes techniques utilisées pour détecter les anomalies.

1.2 Défis

Comme nous avons mentionné précédemment, une anomalie peut être définie comme une observation qui n'est pas conforme à un comportement normal attendu. Une approche directe sera de caractériser la classe cible (comportement normal) et de déclarer toute

observation dans les données n'appartenant pas à cette classe comme une anomalie. Mais plusieurs facteurs rendent cette approche apparemment simple très difficile :

- La définition d'une région normale, comprenant tous les comportements normaux possibles, est très difficile. Souvent, un comportement normal ne cesse d'évoluer et une notion existante d'un comportement normal pourrait ne pas être suffisamment représentatif dans l'avenir.
- La limite entre un comportement normal et un autre, anormal, n'est souvent pas précise. Ainsi, une anomalie qui se trouve près de la frontière peut être une observation normale et vice-versa.
- La disponibilité des étiquettes des données pour l'apprentissage et la validation est souvent un problème majeur pour le développement d'une technique de détection des anomalies.

En présence des défis énumérés ci-dessus, une formulation généralisée du problème de la détection d'anomalies basée sur la définition abstraite d'anomalies n'est pas facile à résoudre. En fait, la plupart des techniques existantes simplifie le problème en se concentrant sur une formulation spécifique. La formulation est induite par des facteurs différents tels que la nature des données, la nature des anomalies à détecter, etc. Dans plusieurs cas, ces facteurs dépendent du domaine d'application dans lequel la technique est à appliquer. Ainsi, il y a de nombreuses formulations différentes du problème de la détection d'anomalies qui ont été explorées dans diverses disciplines telles que les statistiques, l'apprentissage automatique, la fouille de données, la théorie de l'information. Comme l'illustre la Figure 1.1, une technique de détection des anomalies possède les éléments suivants :

1. Nature des données, nature des anomalies, les contraintes et les hypothèses qui constituent le problème de reformulation.
2. Le domaine d'application dans lequel la technique est à appliquer.
3. Le concept et les idées utilisées à partir d'une ou plusieurs disciplines.

Pour une meilleure compréhension de ce qui suit, nous introduisons la terminologie suivante : un problème de détection d'anomalies se réfère à la tâche d'identifier les observations anormales dans des données selon une définition particulière du comportement anormal. Les anomalies se réfèrent à ces observations anormales. Une technique de détection d'anomalies est une solution spécifique à un problème de détection d'anomalies. Une observation normale se réfère à une observation dans les données qui n'est pas une

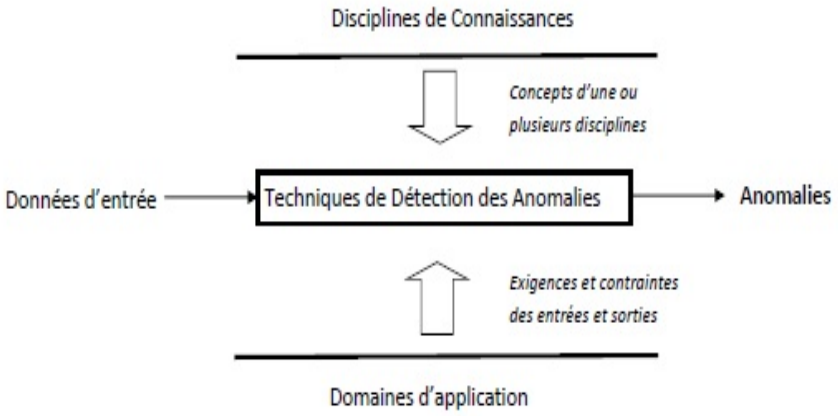

FIGURE 1.1 – Une conception générale d'une technique de détection d'anomalies (Source [17]).

anomalie. Les sorties d'une technique de détection d'anomalies pourraient être des observations étiquettées (anormale ou normale). Certaines techniques attribuent également un score à chaque observation basé sur la mesure dans laquelle l'obsrvation est considérée comme anomalie.

1.3 Les différents aspects d'un problème de détection d'anomalies

Les données

Les éléments clés de toute technique de détection d'anomalies sont les données d'entrée dans lesquelles les anomalies résident. Il s'agit d'une collection des instances de données (également appelés vecteurs, points, obesrvations, entités). Chaque instance peut être décrite par un ensemble des attributs (également désignés sous le nom de variables, caractéristiques). Les instances de données peuvent être de différentes types tels que binaire, catégorique ou continue. Chaque instance peut consister en un seul attribut (univariée) ou un ensemble d'attributs (multivariée).
Une observation importante est que les variables utilisées par toute technique de détection d'anomalies ne se réfèrent pas nécessairement aux variables observées dans les données d'entrée. Plusieurs techniques utilisent par exemple la compression des variables [2], ou construisent des variables plus complexes à partir des variables initiales [28]. Ces méthodes donc consistent à travailler avec un ensemble de variables qui sont les plus

susceptibles de discriminer entre les observations normales et les anomalies dans les données. Un défi majeur pour toute technique de détection d'anomalies consiste à identifier un ensemble optimal de variables qui peuvent permettre à l'algorithme de donner les meilleurs résultats en terme de précision ainsi que d'efficacité de calcul [99].

Types de supervision

Outre les données d'entrée (ou observations), un algorithme de détection d'anomalies peut aussi avoir quelques informations supplémentaires à sa disposition. La présence des étiquettes pour les données d'apprentissage est une telle information qui a été largement utilisée (surtout par les techniques basées sur des concepts de l'apprentissage automatique et la théorie de l'apprentissage statistique [93]). Les étiquettes de données indiquent la classe des instances de données, par exemple, les instances normales peuvent être étiquetées par $+1$ et les instances anormales par -1. Selon la disponibilité des étiquettes, les techniques de détection d'anomalies peuvent être divisées en trois catégories : supervisées, semi-supervisées et non supervisées [17].

Techniques supervisées

Dans ces techniques, nous disposons des étiquettes des données d'apprentissage. L'échantillon d'apprentissage est alors constitué des instances appartenant à deux classes clairement identifiées. Le but est de modéliser à la fois la normalité et l'anormalité à partir de cet échantillon étiqueté. Les techniques supervisées de détection d'anomalies ont une notion explicite des comportements normal et anormal. D'où des modèles précis peuvent être construits. L'inconvénient de cette méthode est que l'obtention d'une grande quantité de données d'apprentissage étiquetées avec précision est un processus lent, puisque l'étiquetage est normalement effectué par un expert humain. Il peut également être très difficile d'obtenir un ensemble de données d'apprentissage qui couvre tous les types d'anomalie possibles. Un autre inconvénient des techniques supervisées est que, souvent, les instances de données anormales sont beaucoup moins nombreuses que les cas de données normales. Certaines techniques injectent artificiellement des anomalies dans un ensemble d'observations normales pour obtenir un échantillon d'apprentissage étiqueté et appliquer par suite des techniques supervisées de détection d'anomalies [1].

Techniques semi-supervisées

Dans ces techniques, nous disposons des étiquettes des instances normales seulement. L'échantillon d'apprentissage est alors constitué uniquement des observations de la classe

cible. L'approche typique de ces techniques est de modéliser la classe des instances normales et d'attribuer toute observation qui ne correspond pas à ce modéle à la classe d'anomalies. Les techniques qui modélisent les observations normales durant la phase d'apprentissage sont très connues. Les étiquettes des observations normales sont relativement faciles à obtenir. En outre, un comportement normal est bien défini et par conséquent il est plus facile de construire un modèle caractérisant un comportement normal à partir des données d'apprentissage.

Techniques non supervisées

La troisième catégorie de techniques ne fait aucune hypothèse quant à la disponibilité des étiquettes des données d'apprentissage. Donc le but est de déterminer les anomalies sans aucune connaissance a priori sur les données. Ainsi, ces techniques sont largement les plus appliquées. Les techniques de cette catégorie font d'autres hypothèses sur les données. Par exemple, les techniques statistiques paramétriques supposent une distribution paramétrique d'une ou de deux classes des observations. Un inconvénient majeur des techniques non supervisées est le taux élevé de fausses alarmes, à cause du manque de connaissances a priori sur les données, ce qui ne permet pas de définir explicitement le comportement normal et/ou le comportement anormal.

La disponibilité des étiquettes détermine le choix d'un des modes de fonctionnement ci-dessus. Généralement, les techniques semi-supervisées et non supervisées sont plus utilisées que les techniques supervisées supposant la disponibilité des anomalies dans les données d'apprentissage. Une des raisons est que l'obtention des observations considérées comme anomalies et couvrant tous les types possibles du comportement anormal est difficile. En effet, de nouveaux types d'anomalies pourraient surgir, pour lesquels il n'existe pas de données d'apprentissage étiquetées.

Types d'anomalies

Il est important, pour une technique de détection des anomalies de définir l'anomalie qu'on souhaite détecter par cette technique. Les anomalies peuvent être classées en trois catégories en fonction de sa composition et sa relation avec les autres données. Les exemples des types d'anomalies cités dans la suite sont adaptés de l'étude de Chandola et al. [17].

Anomalies individuelles

C'est le type le plus simple d'anomalie, et fait objet de la majorité des techniques existantes pour la détection d'anomalies. Une observation est une anomalie individuelle lorsque une ou plusieurs valeurs de ses attributs ne ressemblent pas aux valeurs prises par les observations considérées comme normales. Les techniques qui détectent les anomalies individuelles analysent la relation de chaque exemple individuellement avec le reste des exemples.

Par exemple, dans la détection de la fraude de carte de crédit, chacune des données représente généralement une transaction par carte de crédit. Nous supposons, pour une raison de simplicité, que les données sont définies en utilisant seulement deux attributs : heure de la journée et le montant dépensé. La Figure 1.2 montre une représentation de données en deux dimensions. La sphère représente la région normale pour les instances de données. Les deux transactions A et B se situent en dehors des limites de la région normale et sont donc considérées comme des anomalies individuelles. A est une transaction frauduleuse qui est une anomalie car elle se produit à un moment anormal et le montant dépensé est anormalement élevé. B est une anomalie qui a un montant dépensé exceptionnellement élevé, même si le temps de transaction est normal.

Anomalies contextuelles

Elles sont également appelées des anomalies conditionnelles [83]. Ces anomalies sont dues à la survenue d'un exemple de données dans un contexte spécifique dans les données fournies. Elles sont aussi des anomalies individuelles mais la différence avec le premier type d'anomalies est que les anomalies contextuelles pourraient ne pas être des anomalies dans un contexte différent. Donc ces anomalies sont définies dans un contexte. La notion du contexte est induite par la structure dans l'ensemble des données et doit être spécifiée comme une partie de la reformulation du problème.

Les anomalies contextuelles satisfont deux propriétés :

1. Les données ont un caractère spatial/séquentiel. Chaque instance de données est définie avec deux ensembles d'attributs : attributs contextuels et attributs comportementaux. Les attributs contextuels définissent la position d'une instance et sont utilisés pour déterminer le contexte pour cette instance. Par exemple, dans les séries chronologiques, le temps est un attribut contextuel qui détermine la position de l'instance dans la séquence entière. Les attributs comportementaux sont les attributs non contextuels d'une instance.

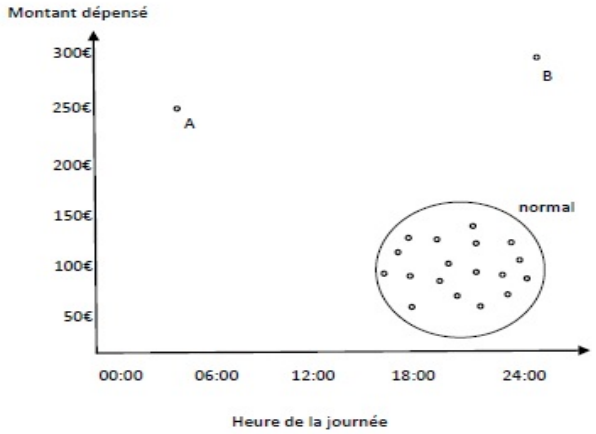

FIGURE 1.2 – Anomalies individuelles dans un jeu de données de transactions de carte de crédit à deux dimensions.

2. Le comportement anormal est déterminé en utilisant les valeurs des attributs comportementaux dans un contexte spécifique.

Les anomalies contextuelles ont été les plus couramment étudiées dans les données temporelles [74] et les données spatiales [24]. La Figure 1.3 montre un exemple d'une série temporelle de la température qui indique la température menseulle d'un domaine au cours des dernières années. Une température de 2 degrés Celsius peut être normal pendant l'hiver (au temps t_1) à cet endroit, mais la même valeur au cours de l'été (au temps t_2) serait une anomalie.

Anomalies collectives

Ces anomalies se produisent car un sous-ensemble des données est divergent du reste des données. Les exemples de ce type des anomalies ne sont pas des anomalies par eux-mêmes, mais leur présence ensemble comme une sous-structure est anormale. Ce type d'anomalies est significatif seulement quand les données ont un caractère spatial ou séquentiel. Ces anomalies sont des sous-graphes ou sous-séquences anormales se produisant dans les données.

La Figure 1.4 illustre une anomalie collective dans les données d'une série chronologique. Le signal est une onde sinusoîdale mais il y a au milieu des fréquences anormales par rapport au reste du signal.

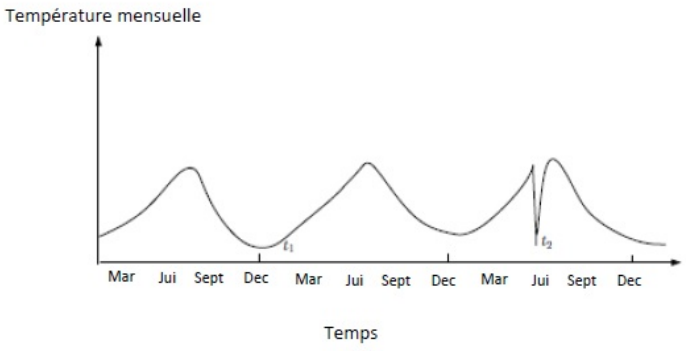

FIGURE 1.3 – Une anomalie contextuelle dans une série temporelle des températures (Source [17]).

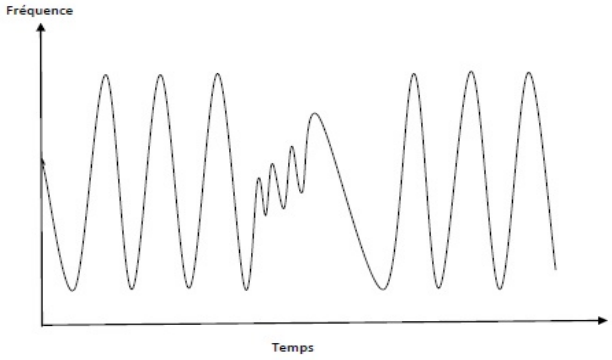

FIGURE 1.4 – Anomalies collectives dans une onde sinusoîdale.

Sorties d'une technique de détection d'anomalies

La nature des anomalies mentionnées ci-dessus impose une exigence sur la structure des anomalies détectées par la technique. Une autre exigence pour toute technique de détection d'anomalies est la manière dont les anomalies sont signalées. D'une façon générale, ces techniques tombent dans l'une des deux catégories suivantes

Etiquettes

Les techniques de cette catégorie attribuent une étiquette à chaque exemple testé. Ainsi, elles se comportent comme un algorithme de classification. L'avantage de ces techniques est qu'elles fournissent un ensemble exacte d'anomalies pour les analystes. L'incovénient de ces techniques est qu'elles ne différencient pas entre les anomalies ; le classement entre les anomalies n'est pas fourni. Souvent, un niveau de confiance est associé pour décider

de la nature d'un exemple. Dans ce cas, une décision binaire n'est pas possible, ce qui motive la nécessité des techniques de type Scores décrites ci-dessous.

Scores

Ces techniques attribuent à chaque exemple un score en fonction de la mesure dans laquelle cet exemple est considéré comme une anomalie. Ainsi, la sortie de ces techniques est une liste de classement d'anomalies. Un analyste peut choisir soit d'analyser les anomalies les plus "anormales", ou d'utiliser un seuil pour sélectionner un ensemble d'anomalies. Souvent, le choix de ce seuil n'est pas simple et doit être fait arbitrairement.

Outre la définition de la nature des données et des anomalies, le domaine d'application peut également imposer certaines contraintes, telles que le niveau souhaité de précision et l'efficacité de calcul. Par exemple dans notre étude, il faut faire un compromis entre la qualité de détection des wafers anormaux et le taux de fausses alarmes suivant des contraintes définies a priori. Plus précisement, nous pouvons nous décider de choisir une technique de détection avec un taux de détection optimal en permettant au taux de fausses alarmes d'aller jusqu'à 30%, ou d'obtenir un taux de détection moins élevé afin de réduire le taux de fausses alarmes à une valeur plus petite (inférieur à 15% par exemple).

Techniques de détection d'anomalies

Le problème de détection d'anomalies peut être reformulé de différentes façons selon les entrées, les exigences et les contraintes. Les techniques décrites dans les sections suivantes traitent principalement les anomalies individuelles. Plusieurs de ces techniques peuvent être étendues pour gérer les autres types d'anomalies.

Les techniques de détection d'anomalies peuvent être divisées en quatre catégories : les techniques basées sur la classification, les techniques basées sur la distance, les techniques statistiques et les autres techniques.

1.4 Techniques basées sur la classification

L'objectif principal de la classification est de construire un modèle de classification en se servant des étiquettes des données d'apprentissage et par suite classer les observations à venir selon leurs classes d'appartencance en utilisant le modèle appris. Le problème

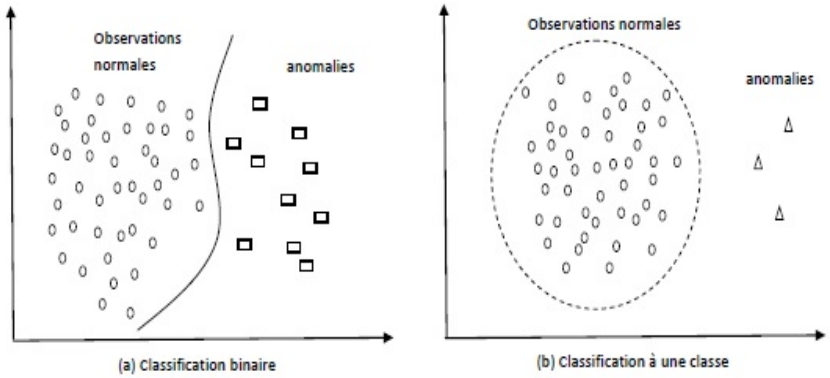

FIGURE 1.5 – Approche basée sur la classification pour la détection d'anomalies.

de détection d'anomalies est parfois traité comme un problème de classification à deux classes :

– une classe cible dont les données sont disponibles.
– une autre classe (classe d'anomalie) dont les données sont difficiles à obtenir.

Le principe général des méthodes proposées dans ce cadre consistent à générer artificiellement [1, 88] les données de la classe négative puis à employer des algortithmes traditionnels d'apprentissage à deux classes. Ces techniques se situent dans le cadre des techniques de détection d'anomalies supervisées.

D'autre part, les méthodes de classification à une classe visent à caractériser une classe des objets normaux et identifier tous les objets qui ne ressemblent pas à cette classe. Ces techniques se situent dans le cadre de techniques de détection d'anomalies semi-supervisées, puisqu'elles exigent la connaissance de la classe normale seulement. Le problème de classification est modélisé comme un problème binaire où les exemples qui n'appartiennent pas à la classe normale sont des anomalies.

La Figure 1.5 montre comment les techniques de classification peuvent être utilisées pour la détection d'anomalies. En mode supervisé, le classifieur est construit pour séparer les deux classes. La phase de test consiste à attribuer un exemple de test à une des deux classes. En mode semi-supervisé, on construit une région autour des exemples normaux et tout exemple en dehors de cette région est considéré comme une anomalie. Les avantages et les inconvénients des techniques de classification supervisée et semi-supervisée sont présentés dans le Tableau 1.1.

Les machines à vecteurs supports (SVM) [22] sont parmi les techniques de détection d'anomalies basées sur la classification les plus connues. Le choix des SVM comme méthode de détection dans notre travail sera justifié dans la section suivante.

Classification	Avantages	Inconvénients
Supervisée	Ces techniques ont une très bonne qualité de détection de plusieurs types d'anomalies déjà vues	Besoin des étiquettes des exemples de deux classes Ces techniques ne peuvent pas détecter les anomalies différentes de celles vues en apprentissage
Semi-supervisée	Le comportement normal peut être appris avec précision	Besoin des étiquettes des exemples de la classe normale Des nouveaux exemples normaux inédits en apprentissage peuvent être reconnus comme des anomalies, ce qui donne un taux de fausses alarmes élevé

TABLE 1.1 – Avantages et inconvénients des techniques de classification supervisée et semi-supervisée.

1.4.1 Machines à Vecteurs Support (SVM)

Les Machines à vecteurs support (Support Vector Machines SVM) sont des algorithmes d'apprentissage statistique utilisés principalement pour la classification binaire. Un algorithme SVM sépare les données appartenant à deux classes différentes en déterminant un hyperplan qui donne une séparation maximale.

Les SVM ont été appliquées pour la détection d'anomalies en les adaptant aux techniques de classification à une classe (apprentissage semi-supervisé)[72]. Ainsi, une technique intuitive serait de déterminer la plus petite hypersphère [87] contenant tous les exemples appartenant à la classe normale. La phase de test consisterait à déterminer de quel côté de cet hypersphère se situe l'exemple de test. Une autre variante de l'approche ci-dessus [75], appelée machines à vecteurs support à une classe (1-SVM), consiste à déterminer une région de volume minimum contenant la plupart des données et ceci en séparant les observations de la classe normale de l'origine avec une marge maximale.

1.4.2 Choix des SVM pour la détection d'anomalies

Nous avons choisi 1-SVM comme une technique de classification à une classe pour la détection d'anomalies. La raison du choix d'une classification à une classe est que les données n'ont pas suffisamment d'anomalies par rapport aux observations normales. En

effet, au test paramétrique le pourcentage des wafers anormaux n'est qu'une fraction de pourcent.

D'autre part, l'avantage principal de 1-SVM par rapport aux autres méthodes de classification mono-classe est qu'il se concentre uniquement sur l'estimation de l'enveloppe d'une région contenant les observations de la classe cible plutôt que sur l'estimation de sa densité de probabilité. D'autres avantages menant au choix de 1-SVM pour la détection des wafers anormaux dans le domaine des semi-conducteurs sont les suivants :

1. Les SVM fournissent un temps d'apprentissage satisfaisant et des résultats de classification de bonne précision [44].
2. Les SVM sont des algorithmes d'apprentissage automatique qui, avec l'utilisation des fonctions noyaux, permettent de capturer et modéliser la non-linéarité dans les données.
3. Ces algorithmes ne supposent aucune hypothèse sur la distribution des données (la normalité par exemple).
4. L'existance de LIBSVM [18], une bibliothèque de programmation implémentant les machines à vecteurs support et d'utilisation simple.
5. Les résultats obtenus avec 1-SVM ont été très prometteurs [6, 36, 102].

Le Chapitre 2 sera consacré à présenter en détails les machines à vecteurs support biclasse, et la variante développée dans le cadre de la classification à une classe pour la détection d'anomalies.

1.5 Clustering

Le clustering est une technique d'apprentissage non-supervisée utilisée pour la classification de données. Les données sont divisées en groupes appelés "clusters" en se basant sur une distance ou une fonction de dissimilarité. Les données du même cluster ont un comportement similaire. Des techniques de détection d'anomalies basées sur le clustering ont été développées. L'idée principale est que les instances normales appartiennent aux clusters denses. Dans ces techniques, les anomalies vérfifient une des trois hypothèses suivantes :

1. elles n'appartiennent à aucun cluster car elles sont très peu nombreuses et différentes des instances normales.
2. elles appartiennent à un petit cluster où ils sont très différentes des autres instances.
3. elles appartiennent à des clusters peu denses où elles sont loins des autres instances situées dans le même cluster (anomalies locales).

Nous citons par exemple les algorithmes *FindOut* [100] et *CBLOF* (Cluster Based Local Outlier Factor) [40].

Ces techniques sont par contre coûteuses en temps de calcul avec une complexité de l'ordre $O(n^2)$. De plus, si les données de la classe normale ne forment aucun cluster ou forment des clusters non significatifs, ces techniques peuvent échouer. Finalement, dans le cas où on dispose de données de grande dimension, les individus ont plus de chance d'être eloignés les uns des autres au moins sur un sous-ensemble d'attributs, d'où la distance devient une mesure inappropriée.

1.6 Plus proches voisins

Les plus proches voisins est une approche largement utilisée en apprentissage automatique et fouille de données. Dans ces méthodes, un objet est analysé par rapport à ses voisins les plus proches. Cette aproche a été appliquée dans la classification, le clustering et la détection d'anomalies. En détection d'anomalies, les méthodes des plus proches voisins sont caractérisées par le fait qu'elles ont une notion explicite de proximité, définie sous la forme d'une mesure de distance ou de similarité pour toute paire des instances de données. Alors que le clustering a une vision globale des données, les méthodes des plus proches voisins analysent chaque objet par rapport à son voisinage local. L'idée de base est que les observations normales ont des voisins proches, tandis que les anomalies sont situées loin des autres observations.

Ces techniques sont divisées en deux catégories. Des méthodes sont basées sur la distance où les anomalies sont les observations les plus distantes des autres observations. Nous citons dans ce cadre la méthode des k plus proches voisins (k-Nearest Neighbors k-NN) [48]. Les autres méthodes sont basées sur la densité où les anomalies sont des observations situées dans des régions moins denses que les autres. Les algorithmes *LOF* (Local Outlier Factor) [14] et *COF* (Connectivity Outlier Factor) [86] sont des exemples de ces méthodes.

L'avantage des techniques des plus proches voisins est qu'elles peuvent être utilisées dans un mode non-supervisé ou semi-supervisé. De plus, elles sont des méthodes non-paramétriques qui ne font aucune hypothèse sur la distribution des données. Les incovénients sont les suivants :

- Si les observations normales n'ont pas le nombre suffisant de voisins, ces techniques peuvent échouer.
- La complexité de calcul des distances entre toutes les observations est de l'ordre $O(n^2)$. La plupart de ces techniques requiert le calcul de la distance aux plus proches voisins

de chaque observation. D'où ces techniques, comme en Clustering, sont coûteuses du point de vue computationnelle.
– En grande dimension, et comme en clustering, la distance devient aussi une mesure inappropriée. Par suite, la notion de voisinage perd de sa signification.

1.7 Techniques statistiques

Les techniques statistiques peuvent être considérées comme la détermination du modèle probabiliste génératif (ou l'estimation de la fonction de distribution de probabilité des données), puis de tester si une instance est engendrée par ce modèle ou non. Ces techniques estiment donc un modèle statistique qui capte la distribution des données, et les instances sont évaluées par rapport à la façon dont elles s'adaptent au modèle. Si la probabilité qu'une instance devant être générée par ce modèle est très faible, l'instance est considérée comme une anomalie.

Comme les techniques basées sur la classification, ces techniques fonctionnent généralement en deux phases : la phase d'apprentissage qui comprend l'estimation du modèle statistique (estimation des paramètres de la distribution), et la phase de test où une instance de test est comparée au modèle afin de déterminer si elle est une anomalie ou non. La technique permet d'estimer la densité de probabilité soit pour les cas normaux, soit pour les cas anormaux (techniques semi-supervisées), en fonction des étiquettes disponibles. Une technique non supervisée détermine un modèle statistique qui correspond à la majorité des observations, et toute observation qui se situe dans une région de faible probabilité est déclarée comme une anomalie.

1.7.1 Techniques paramétriques vs techniques non-paramétriques

Les techniques statistiques comprennent les techniques paramétriques et les techniques non-paramétriques. Les techniques paramétriques supposent que les données de la classe normale (et peut-être anormale) sont générées à partir d'une distribution paramétrique sous-jacente. Ainsi, la phase d'apprentissage consiste à estimer les paramètres de la distribution de l'échantillon donné. Ensuite, la probabilité d'occurence qu'une instance de test soit générée à partir de cette distribution est déterminée. Si cette probabilité est très faible, l'instance de test est considérée comme une anomalie. Concernant la distribution sous-jacente des données, plusieurs tests statistiques supposent une distribution gaussienne. Pour les données avec des variables catégorielles, une distribution multinomiale pourrait être supposée. De même, plusieurs techniques supposent une distribution d'une

chaîne de Markov lors de la modélisation des données séquentielles [63]. Dans des scénarios réels, une seule distribution ne capture pas effectivement la distribution réelle des données. D'où plusieurs techniques supposent que les données proviennent d'un mélange de distributions [29], et par conséquent la phase d'apprentissage consiste à estimer les paramètres du modèle de mélange (modèle de mélanges gaussiens par exemple).

Les techniques non-paramétriqus ne supposent pas la connaissance de la distribution des données. Ces techniques utilisent les méthodes non-paramétriques pour apprendre une distribution. Une des techniques les plus connues pour estimer une densité de probabilité multidimensionnelle est *l'estimateur à noyau de Parzen* ou encore appelée méthode de Parzen-Rozenblatt (Parzen windows) [25].

1.7.2 T^2 de Hotelling

La technique de détection basée sur la statistique de T^2 de Hotelling est une des techniques statistiques paramétriques les plus utilisées. Cette statistique, proposée en 1947 par Hotelling [42], permet d'établir une carte de contrôle multivariée afin d'analyser un système de variables corrélées. Cette statistique permet de résumer en une seule variable l'état de contrôle du procédé. Elle tient compte de la valeur cible des différentes variables, de leurs variances respectives et également des corrélations qui les lient.

Les données sont stockées dans une matrice $X_{n\times p}$, où chacune des n lignes représente un wafer décrit par p paramètres électriques. X s'écrit comme suit :

$$X = [x_{1.} \dots x_{i.} \dots x_{n.}]^T = [x_{.1} \dots x_{.j} \dots x_{.p}]$$

où $x_{i.} = [x_{i1} \dots x_{ij} \dots x_{ip}]^T$, $x_{.j} = [x_{1j} \dots x_{ij} \dots x_{nj}]^T$ et A^T désigne le transposé d'une matrice A.

Le principe de cette méthode est simple : une carte de contrôle est construite en se basant sur la distance (élevée au carré) entre une observation définie par les valeurs prises par les p grandeurs observées et la valeur centrale de ces p grandeurs lorsque le processus est en fonctionnement normal. Puis cette distance est comparée à une limite de contrôle supérieure notée UCL. Si la statistique T_i^2 de la ième observation est inférieur à UCL, on considérera que l'observation (le wafer dans notre cas) est sous-contrôle statistique et donc elle est normale (le wafer est sain). Par contre, si T_i^2 dépasse la limite de contrôle UCL, on dira que l'observation n'est plus sous-contrôle (le wafer est anormal).

Une hypothèse fondamentale dans la construction de l'indice T^2 de Hotelling est que les vecteurs d'observations sont issues d'une distribution normale mutlidimensionnelle. Pour

calculer la statistique de Hotelling d'une observation quelconque $x_{i.}$, deux cas peuvent se présenter :

1. Si le vecteur $(x_{.1}, x_{.2}, \ldots, x_{.p})^T$ suit une distribution normale $N(\mu^p, \Sigma^2)$, où $\mu^p = [\mu_1, \ldots, \mu_p]^T$ représente le vecteur des espérances des p variables, et Σ^2 est la matrice de covariance de dimension $p \times p$, on peut montrer que la statistique

$$T_i^2 = (x_{i.} - \mu^p)^T (\Sigma^2)^{-1} (x_{i.} - \mu^p) \tag{1.1}$$

   suit une loi Chi-deux $\mathcal{X}^2$ à p degrés de liberté. La limite de contrôle supérieure UCL est déterminée alors en se basant sur cette distribution. Plus précisement, UCL est le quantile d'ordre $1 - \alpha$ de la loi Chi-deux $\mathcal{X}^2$ à p degrés de liberté tel que

$$P(\mathcal{X}^2(p) > UCL) = \alpha$$

2. Si μ^p et Σ^2 ne sont pas connus, la matrice de covariance Σ^2 est estimée par la matrice de covariance empirique $\hat{\Sigma^2}$, et le vecteur μ^p est estimé par le vecteur des moyennes empiriques $\hat{\mu^p} = [\hat{\mu}_1, \ldots \hat{\mu}_j \ldots \hat{\mu}_p]^T$. L'expression de la statistique T^2 est donnée par :

$$T_i^2 = (x_{i.} - \hat{\mu^p})^T (\hat{\Sigma^2})^{-1} (x_{i.} - \hat{\mu^p}) \tag{1.2}$$

   La limite de contrôle supérieure est donnée par la distribution de Fisher comme suit :

$$UCL = \frac{p(n-1)(n+1)}{n(n-p)} F(p, n-p) \tag{1.3}$$

   où $F(p, n-p)$ est le quantile de la distribution de Fisher à p et $n-p$ degrés de liberté tel que :

$$P(F(p, n-p) > UCL) = \alpha$$

1.8 Techniques et domaines d'application

Comme nous avons vu, les techniques statistiques pour la détection d'anomalies utilisent des méthodes de modélisation statistique existantes pour modéliser les différentes types de distribution. Cependant, quand la dimension de l'espace des variables est élevée, l'estimation des distributions devient un problème NP-difficile. De plus, les hypothèses paramètriques ne sont souvent pas validées pour des jeux de données réels.

D'autres techniques de détection d'anomalies sont basées sur, par exemple, la théorie de l'information ou de la théorie spectrale [17]. Le tableau 1.2 illustre quelques domaines d'application où différentes techniques de détection d'anomalies peuvent être utilisées.

	Méthodes de classification	Méthodes basées sur la distance	Méthodes statistiques	Autres méthodes
Détection d'intrusion	*	*	*	*
Détection des fraudes	*	*	*	
Détection des défauts dans les unités mécaniques	*		*	*
Détection de dommages structurels	*		*	
Domaine de traitement d'image	*	*	*	*
Les réseaux de capteurs	*	*	*	*

TABLE 1.2 – Techniques et domaines d'application de la détection d'anomalies.

On peut constater que la classification et les méthodes statistiques peuvent être utilisées dans de nombreux domaines différents. Les informations contenues dans le Tableau 1.2 sont adaptées de [17].

Toutes les techniques de détection d'anomalies ont leurs points forts et leurs points faibles et aucune d'entre elles ne peut être utilisée dans toutes les situations. Par exemple, les techniques basées sur la distance ne fonctionnent pas bien avec des données en grande dimension, et les méthodes de classification supervisée ont besoin des étiquettes de l'échantillon d'apprentissage. Certains problèmes peuvent être résolus avec des techniques de classification à une classe tandis que d'autres nécessitent des méthodes de classification multi-classes.

Chapitre 2

Machines à Vecteurs Supports (SVM)

2.1 Apprentissage statitique et SVM

Depuis quelques années, de nouvelles méthodes d'apprentissage se développent sur la base de la Théorie de l'Apprentissage Statistique (Statistical Learning Theory) de Vapnick et Chervonenkis [93]. L'une de ces méthodes, appelée Machine à Vecteur de Support ou SVM (Support Vector Machine) [22], permet de réaliser des estimations en classification (à deux classes ou plus) [15] ou en régression [82].

Les machines à vecteurs supports sont une classe d'algorithmes basés sur le principe de minimisation du "risque structurel", décrit par la théorie de l'apprentisage statistique de Vapnick et Chervonenkis [93]. Ce principe représente la dépendance entre l'erreur de généralisation du modèle d'apprentissage et la famille de fonctions utilisée pour apprendre le modèle. Une courte description de ce principe sera donnée dans la Section 2.3.1.

Les méthodes SVM sont largement répandues en apprentissage statistique et ont eu beaucoup de succès dans quasiment tous les domaines où elles ont été appliquées. Ces méthodes ont rapidement été adoptées pour leur capacité à travailler avec des données de grandes dimensions, le faible nombre d'hyperparamètres, leurs garanties théoriques, et leurs bons résultats en pratique.

Les machines à vecteurs supports exploitent les concepts relatifs à la théorie de l'apprentissage statistique pour aborder d'une façon nouvelle la question du dilemme biais-variance. Le compromis entre la capacité d'apprentissage (adéquation aux données) et la capacité de généralisation (complexité du modèle et capacité à donner des prédictions

avec une précision élevée pour les futurs observations) de ces méthodes est respectivement accompli en minimisant l'erreur empirique et dans le même temps, en essayant de maximiser la distance entre les deux classes. La justification intuitive de cette méthode d'apprentissage est la suivante : si l'échantillon d'apprentissage est linéairement séparable, il semble naturel de séparer parfaitement les éléments des deux classes de telle sorte qu'ils soient le plus loin possible de la frontière choisie.

L'idée de base des SVM est de trouver un hyperplan qui donne une séparation optimale entre des observations appartenant à deux classes distinctes. La première version de SVM était capable de classifier seulement les données séparables, ce qu'on appelle SVM à "marge maximale" (hard margin SVM). Puis une variante de cette version, appelée SVM à "marge souple" (soft margin SVM), a été introduite pour classifier les données dans le cas non séparable [22].

Dans ce chapitre nous parlons tout d'abord du problème de la classification linéaire. Puis nous présentons la théorie des méthodes SVM et comment séparer linéairement les données dans les cas séparable et non séparable. Nous expliquons ensuite comment les méthodes SVM peuvent être utilisées pour une séparation non linéaire des données . A la fin du chapitre, nous donnons une description détaillée de la méthode SVM à une classe, une variante des SVM classiques, introduite dans le cadre d'une classification à une classe pour la détection d'anomalies.

2.2 Problème de la classification linéaire

Le problème de la discrimination linéaire remonte aux années 1930 quand Fisher a proposé la première procédure de classification binaire. Entre les années 1960 et 1980 les statistiques ont connu une révolution menée par l'apparition de la théorie de l'apprentissage statistique. Et depuis 1992, les machines à vecteurs supports en sont le plus grand succès.

2.2.1 Formalisation du problème

Le problème de classification rentre dans le cadre de l'apprentissage statistique supervisé. Le but est de prévoir la classe d'appartenance notée y d'un vecteur p-dimensionnel x en se basant sur les mesures des variables qui l'expliquent avec pour seule information celle contenue dans l'echantillon d'apprentissage $\mathcal{S}$.

Dans le cadre la discrimination biclasse, on considère un couple (M, Y) de variables aléatoires à valeurs dans $\mathcal{X} \times \mathcal{Y}$, où $\mathcal{X}$ désigne l'espace des variables explicatives souvent

pris dans $\mathbb{R}^p$, $\mathcal{Y} = \{-1, +1\}$. L'échantillon d'apprentissage $\mathcal{S}$ est ainsi une collection de n réalisations indépendantes et identiquement distribuées (*i.i.d.*) du couple aléatoire (M, Y) dont sa distribution jointe est fixe mais inconnue. Cet ensemble est souvent noté par :

$$\mathcal{S} = \{(M_1, Y_1), (M_2, Y_2), \ldots, (M_n, Y_n)\}.$$

Le but est de construire une fonction $h : \mathcal{X} \longrightarrow \mathcal{Y}$ telle que $P\big(h(M) \neq Y\big)$ soit minimale. h est appelée *classifieur*. L'appartenance d'une observation $m_i \in \mathbb{R}^p$ (une réalisation de la variable aléatoire M_i) à une classe ou à une autre dépend de la valeur -1 ou 1 prise par $y_i = h(m_i)$. Dans la suite, et pour raison de conformité avec les notations introduites dans le Chapitre 1, nous désignons par $x_i{=}x_{i.} \in \mathbb{R}^p$ l'observation m_i.

2.2.2 Approche générale

Généralement, la classification binaire est accomplie au moyen d'une fonction à valeurs réelles $f : \mathcal{X} \subseteq \mathbb{R}^p \to \mathbb{R}$. La fonction h introduite dans le paragraphe précédent est telle que $h(x) = sgn\big(f(x)\big)$, i.e. toute observation x_i est affectée à la classe qui correspond au signe de $f(x_i)$: si $f(x_i) \geq 0$, $x_{i.}$ est affecté à la classe positive ($h(x_i) = +1$) sinon elle sera dans la classe négative ($h(x_i) = -1$). En classification linéaire la fonction f est linéaire en x_i et elle prend la forme générale suivante,

$$f(x_i) = \langle w, x_i \rangle + b,$$

où $(w, b) \in \mathbb{R}^p \times \mathbb{R}$ sont les paramètres à estimer de la fonction f et l'opérateur $\langle\,\rangle$ désigne le produit scalaire usuel dans $\mathbb{R}^p$. La règle de décision est donc donnée par $sgn\big(f(x_i)\big)$. Comme nous sommes dans un cadre de classification par apprentissage, l'estimation des paramètres de f s'effectue en se basant sur l'échantillon d'apprentissage $\mathcal{S}$.

Géométriquement, le classifieur h divise l'espace des variables explicatives $\mathcal{X}$ en deux demi-espaces correspondant chacun à une classe. Cette séparation est réalisée par l'hyperplan $H_{(w,b)}$ défini par l'équation $\langle w, x \rangle + b = 0$ où w est un vecteur normal à l'hyperplan de séparation qu'on appelle vecteur de poids, tandis que la variation du paramètre b, appelé biais, engendre une simple translation de l'hyperplan séparateur. Il est donc nécessaire d'employer une représentation à $(p + 1)$ paramètres de liberté pour parcourir l'ensemble de tous les hyperplans de $\mathbb{R}^p$.

L'objectif de la discrimination linéaire est de trouver la bonne (En terme de capacité de généralisation) fonction f dans la classe de toutes les fonctions linéaires de $\mathcal{X}$ dans $\mathbb{R}$ que nous noterons $\mathcal{F}_{\mathcal{L}}$. La classe de tous les hyperplans qui en decoule sera notée $\mathcal{H}$. Cette forme simple de classifieur a été énormement utilisée dans la classification binaire. Les

termes "vecteur de poids" pour le vecteur w et "biais" pour le paramètre b sont empruntés de la litterature des reseaux de neurones.

Plusieurs algorithmes itératifs simples motivés par l'optimisation de différentes fonctions de coûts ont été introduits dans les années 1960 pour séparer linéairement des observations issues de deux populations différentes. Les machines à vecteurs supports cherchent plutôt à séparer parfaitement les deux classes de telle sorte que leurs élements soient le plus loin possible de l'hyperplan trouvé. C'est pour définir cette notion de "plus loin" que l'on introduit la "marge".

2.2.3 Définitions de base

Dans ce paragraphe nous introduisons la notion de marge pour la classe des fonctions linéaires $\mathcal{F}_{\mathcal{L}}$. Cette grandeur est au coeur des SVM et elle jouera un rôle important tout au long de la suite de cette partie.

Définition 2.2.1 (Marge biclasse)**.** Soit f une fonction de $\mathcal{F}_{\mathcal{L}}$. La marge d'une observation $(x_i, y_i) \in \mathcal{S}$ relativement à la fonction f est définie par $\Delta_i = y_i f(x_i)$.

Cette marge peut prendre une valeur négative. Elle dépend de la fonction f et non du classifieur $sgn\big(f(.)\big)$. Si g est un multiple de f , les classifieurs pour ces deux fonctions sont les mêmes mais leurs marges sont différentes.

La valeur absolue de γ_i est proportionnelle à la distance euclidienne séparant le point x_i de l'hyperplan $H_{(w,b)}$ associé à f :

$$d(x_i, H_{(w,b)}) = \frac{|\langle w,\, x_i\rangle + b|}{\|w\|} = \frac{|f(x_i|}{\|w\|} = \frac{|y_i f(x_i)|}{\|w\|} = \frac{|\Delta_i|}{\|w\|}. \tag{2.1}$$

Ces deux quantités ne coïncident que lorsque $\|w\| = 1$, dans ce cas nous parlons de la *marge euclidienne*. Enfin, l'observation x_i est bien classée par f si et seulement si $\Delta_i > 0$. Après avoir introduit la notion de la marge, nous définissons ensuite :

- La *distribution de marges* d'un hyperplan $H_{(w,b)}$ par rapport à l'échantillon d'apprentissage $\mathcal{S}$ est définie par :

$$\mathcal{M}_{\mathcal{S}}(H_{(w,b)}) = \{\Delta_i = y_i(\langle w,\, x_i\rangle + b); i = 1,\, 2, \ldots,\, n\}$$

- La *marge de l'hyperplan* $H_{(w,b)}$ par rapport à l'échantillon d'apprentissage $\mathcal{S}$ est définie par :

$$m_{\mathcal{S}}(H_{(w,b)}) = \min_{1 \le i \le n} \mathcal{M}_{\mathcal{S}}(H_{(w,b)})$$

Si nous normalisons w, notre hyperplan aura comme équation :

$$H_{\left(\frac{w}{\|w\|},\frac{b}{\|w\|}\right)} : \left\langle \frac{w}{\|w\|}, x \right\rangle + \frac{b}{\|w\|} = 0.$$

Les hyperplans $H_{(w,b)}$ et $H_{\left(\frac{w}{\|w\|},\frac{b}{\|w\|}\right)}$ sont les mêmes mais ils donnent lieu à des marges différentes. Dans ce cas, la valeur absolue de la marge d'un point x_i est égale à la distance euclidienne entre ce point et l'hyperplan $H_{\left(\frac{w}{\|w\|},\frac{b}{\|w\|}\right)}$ (cf. équation 2.1 avec $\|w\| = 1$). Ainsi, c'est la métrique euclidienne que nous utilisons en calculant les marges plus tard. Ce changement d'échelle va jouer un rôle important dans la formulation des problèmes de maximisation de la marge, au sens euclidien, pour les machines à vecteurs supports.

– Finalement, on définit sur $\mathcal{S}$ la marge de l'échantillon d'apprentissage $m_{\mathcal{S}}$ comme étant le maximum de la marge euclidienne sur l'ensemble des hyperplans normalisés $\mathcal{H}_{\mathcal{N}}$ où :

$$\mathcal{H}_{\mathcal{N}} = \left\{ H_{\left(\frac{w}{\|w\|},\frac{b}{\|w\|}\right)} : (w,b) \in \mathbb{R}^{\prime} \times \mathbb{R} \right\},$$

et

$$m_{\mathcal{S}} = \max_{\mathcal{H}_{\mathcal{N}}} \left\{ \min_{1 \leq i \leq n} \mathcal{M}_{\mathcal{S}} \left(H_{\left(\frac{w}{\|w\|},\frac{b}{\|w\|}\right)} \right) \right\}.$$

L'hyperplan qui réalise ce maximum est appelé l'hyperplan à marge maximale et la valeur de sa marge sera positive dans le cas où $\mathcal{S}$ est linéairement séparable. Dans le cas où l'échantillon d'apprentissage est non-linéairement séparable, un tel hyperplan n'existe pas. En pratique, l'hypothèse que l'échantillon $\mathcal{S}$ est linéairement séparable est assez forte. De ce fait, d'autres mesures plus flexibles ont été utilisées afin d'autoriser quelques violations de la marge et pour prendre en compte des propriétés plus générales de l'échantillon d'apprentissage (cf. Section 2.3.3).

2.3 Machines à Vecteurs Supports biclasses

Au cours de cette section nous allons présenter les principaux problèmes et résultats de l'apprentissage des machines à vecteurs supports dans le cadre de la classification binaire. Tous les algorithmes qui seront exposés sont motivés par la théorie des bornes developpée par Vapnik et Chervonenkis [93]. Cette théorie propose différents majorants de l'erreur de généralisation. Par exemple nous pouvons nous intéresser à l'optimisation de la marge, la marge relaxée par l'introduction des variables d'écarts ξ_i ou le nombre des vecteurs supports,...etc. Dans ce paragraphe nous allons mettre en lumière la formulation

la plus utilisée qui transforme le problème de recherche de l'hyperplan séparateur en la minimisation de la norme de son vecteur de poids w.

2.3.1 Minimisation du risque structurel

Dans les méthodes d'apprentissage, deux échantillons de données principaux sont généralement utilisés : l'échantillon d'apprentisage et l'échantillon de test. L'échantillon d'apprentisage représente la part des données utilisée pour apprendre le modèle, et l'échantillon de test est l'autre partie, non exploitée dans la phase d'apprentissage, utilisée pour évaluer les performances de généralisation du modèle. La qualité de ce modèle est alors jugée à sa capacité à réduire l'erreur de test ou de "généralisation". Cependant, comme le modèle n'est pas construit en utilisant l'échantillon de test, l'erreur de généralisation ne peut pas être évaluée exactement car elle dépend de la distribution de probabilité des données :

$$R[f] = \int Q(x) d\Omega(x, y) \tag{2.2}$$

où Q est la fonction d'erreur, x est le vecteur d'entrée, et Ω est la distribution des données (qui nous est inconnue).

La seule information dont nous disposons comme évaluation de l'erreur est l'erreur d'apprentissage (ou le risque empirique) :

$$R_{emp}[f] = \frac{1}{n} \sum_{i=1}^{n} Q(x_i). \tag{2.3}$$

Cette information n'est pas suffisante. En effet, on peut facilement trouver un modèle minimisant l'erreur d'apprentissage mais pour lequel l'erreur de généralisation sera très grande. Un exemple simple est la régression de données linéaires bruitées au moyen d'une fonction polynomiale : plus le degré du polynôme sera grand, plus l'erreur d'apprentissage sera faible, mais plus l'erreur de généralisation sera élevée. On peut donc comprendre que cette dernière est aussi liée à la famille de fonction utilisée comme modèle. Cette dépendance est nommée "risque structurel".

Dans leur Théorie de l'Apprentissage Statistique, Vapnik et Chervonenkis ont prouvé qu'il est possible de définir une majoration du risque structurel en fonction de la famille de fonction utilisée pour le modèle. L'une de ces majorations peut être calculée en utilisant la dimension de Vapnik-Chervonenkis (dimension VC) qui représente le plus grand nombre d'observations pouvant être séparées de toutes les façons possibles par une fonction appartenant à l'ensemble de fonctions linéaires $\mathcal{F}_{\mathcal{L}}$. La borne VC est alors définie ainsi : si la dimension VC, notée v, de la famille de fonctions utilisées est inférieure à la

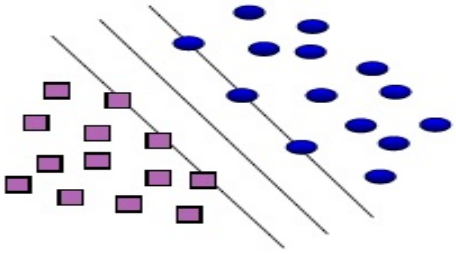

FIGURE 2.1 – Hyperplan linéaire pour des données séparables.

taille de l'échantillon d'apprentissage n, alors avec une probabilité d'au moins $1-\zeta$, on a :

$$R[f] \leq R_{emp}[f] + \sqrt{\frac{\upsilon\left(log(\frac{2n}{\upsilon})+1\right) - log(\frac{\zeta}{4})}{n}}. \tag{2.4}$$

Cette dernière équation nous donne une borne sur le risque structurel de f après calcul de son risque empirique. L'utilité pratique de la borne donnée par l'Equation (2.4) dépend de la taille n de l'échantillon d'apprentissage et de la dim VC de la classe de fonctions $\mathcal{F}_{\mathcal{L}}$. Avec une confiance $1-\zeta$ donnée, le risque de l'une des fonctions qui minimisent le risque empirique se rapproche du minimum sur $\mathcal{F}_{\mathcal{L}}$ du risque structurel lorsque la taille de l'échantillon d'apprentissage augmente, et ce, d'autant plus vite que la dimension VC de $\mathcal{F}_{\mathcal{L}}$ est faible. Ainsi, Vapnik considère que la taille n d'un échantillon d'apprentissage S est faible pour estimer une fonction dans une classe de dimension υ si le rapport n/υ est petit, soit $n/\upsilon < 20$.

2.3.2 Hyperplan à "marge maximale"

L'hyperplan à marge maximale est le modèle le plus simple des machines à vecteurs supports et il constitue le point de départ pour d'autres algorithmes plus complexes. C'est le cas de classification le plus simple où il y a seulement deux classes et cet hyperplan ne peut être construit que si les exemples d'apprentissage sont séparables (cf. Fgure 2.1). L'idée des SVM est de déterminer un hyperplan séparateur qui sépare les deux classes et de maximiser la distance entre l'hyperplan séparateur et les observations les plus proches de chaque classe.

Il est évident qu'il existe une multitude d'hyperplan valide mais la propriété remarquable des SVM est que cet hyperplan est optimal. Nous allons chercher parmi les hyperplans valides, celui qui passe "au milieu" des points de deux classes d'exemples. Formellement,

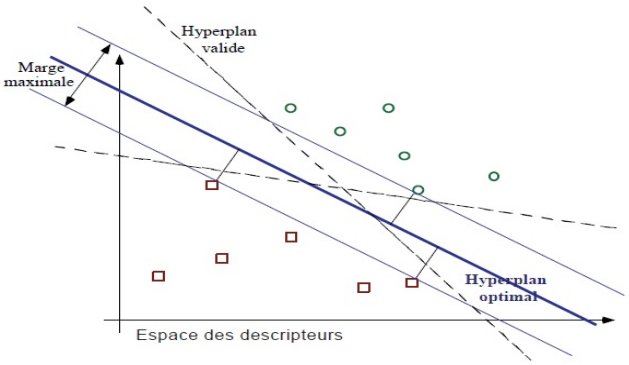

FIGURE 2.2 – Hyperplan optimal à marge maximale pour les données séparables.

cela revient à chercher un hyperplan dont la distance minimale aux données d'apprentissage est maximale. On appelle cette distance la marge entre l'hyperplan et les exemples. L'hyperplan séparateur optimal est celui qui maximise cette marge. La Figure 2.2 illustre les notions "hyperplan optimal" et "marge maximale" présentées dans ce paragraphe.

L'estimation des paramètres (w^*, b^*) de cet hyperplan se fait donc en résolvant le problème d'optimisation suivant :

$$(w^*, b^*) = \underset{(w,b)}{arg\,max} \left\{ \underset{i}{min} \left[y_i(\langle w, x_i \rangle + b) \right], \|w\| = 1 \right\}. \tag{2.5}$$

Par conséquent ce que nous optimisons dans le problème (2.5) est la marge au sens de la norme euclidienne vu que nous imposons la contrainte $\|w\| = 1$. Cette contrainte n'implique aucune restriction sur l'ensemble des hyperplans $\mathcal{H}$ de $\mathbb{R}^p$. En effet :

$$\begin{aligned} \mathcal{H} &= \left\{ h/h(x) = sgn\big(\langle w, x \rangle + b\big) \right\} \\ &= \left\{ h/h(x) = sgn\left(\left\langle \frac{w}{\|w\|}, x \right\rangle + \frac{b}{\|w\|} \right) \right\} \\ &= \left\{ h/h(x) = sgn\left(\langle w, x \rangle + b \right); \|w\| = 1 \right\}. \end{aligned} \tag{2.6}$$

Dire que les deux classes de l'échantillon d'apprentissage $\mathcal{S}$ sont linéairement séparables est équivalent à dire qu'il existe des paramètres $(w^*, b^*) \in \mathbb{R}^p \times \mathbb{R}$ tels que l'on a pour tout $i \in \{1, \ldots, n\}$:

$$\begin{aligned} \langle w^*, x_i \rangle + b^* &> 0 \; si \; y_i = +1 \\ \langle w^*, x_i \rangle + b^* &< 0 \; si \; y_i = -1 \end{aligned}$$

ce qui est équivalent à :

$$y_i(\langle w^*, x_i \rangle + b^*) > 0 \quad \forall i \in \{1, \ldots, n\}.$$

Les paramètres (w^*, b^*) sont normalisés par $min_i\,[y_i(\langle w^*, x_i\rangle + b^*)]$, pour obtenir $(\tilde{w}, \tilde{b})$ tels que :

$$y_i(\langle \tilde{w}, x_i\rangle + \tilde{b}) > 1 \quad \forall i \in \{1, \ldots, n\}. \tag{2.7}$$

Ainsi la marge de l'hyperplan $H_{(\tilde{w},\tilde{b})}$ notée Δ_H est la distance entre les hyperplans d'équations $\langle \tilde{w}, x\rangle + \tilde{b} = 1$ et $\langle \tilde{w}, x\rangle + \tilde{b} = -1$ qui sont parallèles à $H_{(\tilde{w},\tilde{b})}$ et lui sont équidistants. Soient x^+ et x^- deux points vérifiant respectivement les deux équations précédentes, nous écrivons alors :

$$\begin{aligned} \Delta_H &= \langle \tfrac{\tilde{w}}{\|\tilde{w}\|}, x^+\rangle - \langle \tfrac{\tilde{w}}{\|\tilde{w}\|}, x^-\rangle \\ &= \tfrac{2}{\|\tilde{w}\|}. \end{aligned} \tag{2.8}$$

Comme notre but est de maximiser cette marge, cela revient alors à minimiser la norme du vecteur de poids $\|\tilde{w}\|$. L'hyperplan à marge maximale est la solution du problème primal d'optimisation suivant portant sur les paramètres w et b :

$$\begin{array}{ll} \text{Minimiser}_{\text{w,b}} & \frac{1}{2}\|w\|^2\ , \\ \text{sous les contraintes} & y_i\,(\langle w, x_i\rangle + b) \geq 1, \quad \text{i=1, 2,. . . , n.} \end{array} \tag{2.9}$$

Nous nous retrouvons ainsi face à un problème d'optimisation quadratique convexe sous contraintes linéaires.

Pour résoudre le problème (2.9), on construit le *lagrangien* L qui est la somme de la fonction objective et d'une combinaison linéaire des contraintes du problème (2.9). Ce lagrangien possède un unique *point-selle* pour ce genre de problèmes. Soit alors :

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i\,[y_i(\langle w, x_i\rangle + b) - 1], \tag{2.10}$$

où les α_i, $i = 1, \ldots, n$, sont des réels positifs désignant les multiplicateurs de Lagrange ou variables duales associées aux contraintes du problème (2.9). Le coefficient $\frac{1}{2}$ qui apparaît ici est rajouté pour simplifier les calculs de dérivée qui vont suivre.

Le problème primal et sa formulation duale ont la même solution qui correspond à un point-selle du lagrangien. Pour trouver ce point selle, on est appelé à minimiser L par rapport aux variables primaires w et b et le maximiser par rapport aux variables duales α_i. Le point selle doit donc satisfaire les conditions nécessaires dites de stationnarité, i.e la dérivée du Lagrangien par rapport aux variables primaires doit s'annuler :

$$\frac{\partial \mathrm{L}(w, b, \alpha)}{\partial w} = 0 \iff w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{2.11}$$

$$\frac{\partial \mathrm{L}(w, b, \alpha)}{\partial b} = 0 \iff \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.12}$$

En substituant (2.11) et (2.12) dans (2.10), on élimine les variables primaires et l'on obtient la forme duale du problème d'optimisation. C'est un problème quadratique consistant à trouver les multiplicateurs de Lagrange $\alpha_i \geq 0$ tels que :

$$\begin{array}{ll} \text{Maximiser}_{\alpha} & W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle, \\ \text{sous les contraintes} & \sum_{i=1}^{n} y_i \alpha_i = 0, \\ & \alpha_i \geq 0, \quad i = 1, 2, \ldots, n. \end{array} \tag{2.13}$$

Ce dernier problème peut être résolu en utilisant des méthodes standards de programmation quadratique [34]. Une fois la solution optimale $\alpha^* = (\alpha_1^*, \ldots, \alpha_n^*)$ du problème (2.13) obtenue, le vecteur de poids de l'hyperplan à marge maximale recherché s'écrit :

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i. \tag{2.14}$$

Il est à noter que les conditions de Karush-Kuhn-Tucker (KKT) [53],

$$\alpha_i^* [y_i (\langle w^*, x_i \rangle + b^*) - 1] = 0, \quad i = 1, 2, \ldots, n \tag{2.15}$$

qui expriment le fait qu'à l'optimum le produit des variables duales et des contraintes associées doit être nul, nous donnent une information très utile sur la structure de la solution. Ces conditions (2.15) impliquent que seuls les points qui sont sur les hyperplans frontière $\langle x_i, w \rangle + b = \pm 1$ jouent un rôle, car les multiplicateurs de Lagrange α_i^* sont non nuls pour ces seuls points. Ces points sont appelés *vecteurs supports* (SVs). L'ensemble des indices des vecteurs supports est noté sv :

$$sv = \{i \in \{1, 2, \ldots, n\} \, ; \, \alpha_i^* \neq 0\}$$

Compte tenu des conditions de KKT, ces vecteurs définissent à eux seuls la solution du problème (2.9). Ils constituent donc la partie active de l'échantillon d'apprentissage. Si un vecteur non support est supprimé de l'echantillon d'apprentissage, nous retrouvons toujours le même hyperplan séparateur,i.e. la solution du problème (2.9) reste la même. Cette proprieté rend les machines à vecteurs supports très attractives car elles permettent d'extraire les élements representatifs de l'échantillon d'apprentissage.

Enfin, la fonction linéaire correspondante à l'hyperplan solution peut alors être écrite :

$$f(x) = \langle w^*, x \rangle + b^* = \sum_{i \in sv} \alpha_i^* y_i \langle x_i, x \rangle + b^*, \tag{2.16}$$

où b^* est obtenue en utilisant n'importe quel vecteur support $(x_i, y_i)_{i \in sv}$ dans l'équation

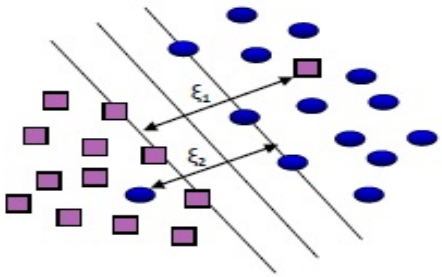

FIGURE 2.3 – Hyperplan optimal à marge souple pour des données non-séparables.

(2.15). On remarque que l'hyperplan solution ne requiert que le calcul des produits scalaires $\langle x, x_i \rangle$ entre des vecteurs de l'espace d'entrée $\mathcal{X}$. On remarque aussi que la solution ne dépend plus de la dimension p de l'espace d'entrée, mais de la taille n de l'échantillon d'apprentissage et même du nombre de vecteurs supports qui est bien inférieur à n.

La règle de classification d'une nouvelle observation $\mathbf{x}$ basée sur l'hyperplan à marge maximale est donnée par :

$$h(x) = sgn\left(f(x)\right) = sgn\left(\sum_{i \in sv} \alpha_i^* y_i \langle x_i, x \rangle + b^*\right). \tag{2.17}$$

2.3.3 Hyperplan à "marge souple" : C-SVM

L'hypothèse que les données soient séparables conditionne beaucoup la résolution du problème (2.9). En effet, il suffit qu'une observation des deux classes viole la contrainte (2.3) pour que ce problème n'ait plus de solution.

Comme indiqué précédemment, la première version de SVM (SVM à marge maximale) peut seulement classifier les données séparables. Alors, comment pouvons-nous généraliser cette version pour gérer les données non-séparables ?

Cette généralisation de l'hyperplan à marge maximale à été proposée par [22] en introduisant les variables d'écart à la marge $(\xi_i)_{1 \leq i \leq n}$ (cf. Figure 2.3). Les SVM qui sont capables de séparer les donnés non-séparables sont appelés SVM à marge souple. Les SVM à marge souple cherchent un hyperplan séparateur comme dans les SVM à marge maximale, tout en permettant à quelques observations d'être mal classées. L'idée consiste à relâcher les contraintes (2.7) dans le but d'autoriser quelques erreurs de classification. Les SVM à marge souple essayent de minimiser la somme des variables d'écart à la marge ξ_i des erreurs d'apprentissage et de maximiser la marge entre les classes. Notons que la valeur de ξ_i est égale à zéro si x_i est classée correctement.

Une première réalisation pour déterminer un hyperplan à marge souple est appelée C-SVM (C-Support Vectors Machines). Cette approche utilise le paramètre de pénalisation C qui contrôle le compromis entre la maximisation de la marge et la permission à quelques observations d'être mal classés. Le problème (2.9) devient alors :

$$\begin{array}{ll} \text{Minimiser}_{\text{w,b},\xi} & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i, \\ \text{sous les contraintes} & y_i(\langle w,\, x_i\rangle + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \qquad\qquad \text{i=1, 2,..., n.} \end{array} \tag{2.18}$$

Autrement dit, on cherche à maximiser la marge en s'autorisant pour chaque contrainte une erreur positive ξ_i la plus petite possible. Le paramètre supplémentaire C qui apparaît ici est un hyperparamètre fixé par l'utilisateur et doit être calibré en fonction de données. Il permet de contrôler l'importance de l'erreur que l'on s'autorise par rapport à la taille de la marge. Plus C est important, moins d'erreurs sont autorisées.

En suivant la même démarche du Lagrangien que précédemment, nous aboutissons à la forme duale

$$\begin{array}{ll} \text{Maximiser}_{\alpha} & W(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i,\, x_j\rangle, \\ \text{sous les contraintes} & \sum_{i=1}^{n} y_i\alpha_i = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1, 2, \ldots, n. \end{array} \tag{2.19}$$

La seule différence par rapport au problème (2.13) est la majoration des α_i par C. On peut montrer que si $\mathcal{S}$ est séparable et quand C est suffisamment grand, les problèmes (2.13) et (2.19) deviennent équivalents. Le choix de C sera discuté dans la Section 2.3.5.

2.3.4 SVM non linéaires

Comment les algorithmes de SVM ci-dessus peuvent être généralisés au cas où la fonction de décision n'est pas une fonction linéaire de données ? Est-il possible de construire un hyperplan linéaire qui sépare les données non linéaires ?

L'extension de SVM pour traiter les données non linéaires est basée sur une transformation de ces données dans un espace de dimension plus grande (un espace de Hilbert de dimension finie ou infinie). Puis une classification linéaire est effectuée dans cet espace transformé (cf. Figure 2.4).

En remarquant que dans la résolution des problèmes (2.13) et (2.19), seuls les produits scalaires $\langle x_i,\, x_j\rangle$ sont nécessaires, les SVM peuvent être étendues pour traiter le cas non-linéaire. L'astuce qui fait la force des SVM repose sur les noyaux autoreproduisants

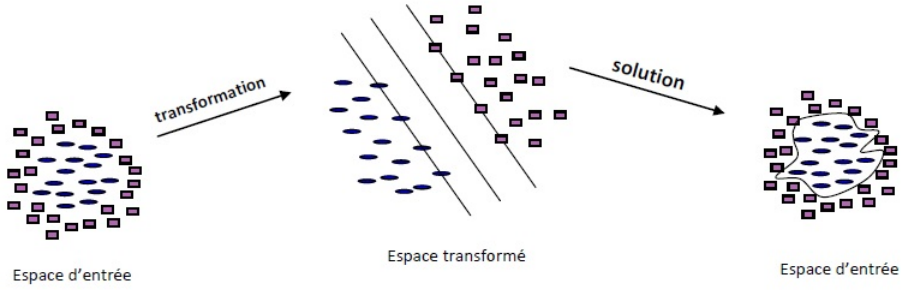

FIGURE 2.4 – Séparation non linéaire des données.

[5]. L'idée de Boser et al. [12] est de transformer les observations x_i dans un espace de Hilbert $\mathcal{H}$ de dimension plus élevé, à l'aide d'une fonction non-linéaire $\phi : \mathbb{R}^p \longrightarrow \mathcal{H}$.

L'espace $\mathcal{H}$ ainsi obtenu est appelé espace de redescription ou aussi espace transformé. Tout ce qu'il nous reste à faire c'est de résoudre le problème (2.13) ou (2.19) dans l'espace $\mathcal{H}$, en remplacant $\langle x_i, x_j \rangle$ par $\langle \phi(x_i), \phi(x_j) \rangle$. Cependant, il peut être parfois possible de ne pas avoir à effectuer explicitement ces produits scalaires dans $\mathcal{H}$ grâce à l'utilisation de fonctions noyaux. C'est ce qu'on appelle l'*astuce de noyaux* (kernel trick [3]). En effet, si on peut trouver une "fonction noyau" K telle que

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle, \tag{2.20}$$

nous aurons alors seulement besoin d'utiliser K dans l'algorithme d'apprentissage, sans jamais avoir besoin de connaître explicitement ϕ. Un exemple de cette fonction est le noyau gaussien

$$K(x, z) = exp\big(-\gamma \|x - z\|^2 \big). \tag{2.21}$$

Nous donnons maintenant un exemple simple pour un noyau, pour lequel nous pouvons construire la fonction ϕ. Supposons que les données sont des vecteurs dans $\mathbb{R}^2$, et que nous choisissons $K(x_i, x_j) = (\langle x_i, x_j \rangle)^2$. Il est facile de trouver un espace $\mathcal{H}$, et une fonction ϕ de $\mathbb{R}^2$ dans $\mathcal{H}$, telle que $(\langle x, y \rangle)^2 = \langle \phi(x), \phi(y) \rangle$: nous choisissons $\mathcal{H} = \mathbb{R}^3$ et

$$\phi(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2} x_1 x_2 \\ x_2^2. \end{pmatrix}$$

Notons que ni la fonction ϕ ni l'espace $\mathcal{H}$ sont uniques pour un noyau donné. Nous pourrions également trouver une deuxième fonction $\phi : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$

$$\phi(x) = \frac{1}{2} \begin{pmatrix} x_1^2 - x_2^2 \\ 2x_1 x_2 \\ x_1^2 + x_2^2 \end{pmatrix}$$

ou aussi une troixième fonction $\phi : \mathbb{R}^2 \longrightarrow \mathbb{R}^4$

$$\phi(x) = \begin{pmatrix} x_1^2 \\ x_1 x_2 \\ x_1 x_2 \\ x_2^2. \end{pmatrix}$$

telles que $(\langle x, y \rangle)^2 = \langle \phi(x), \phi(y) \rangle$.

Pour quels noyaux existe-t-il une paire $\{\mathcal{H}, \phi\}$, avec les propriétés décrites ci-dessus ? Jusqu'à présent, le seul critère que nous avons énoncé pour savoir si une fonction symétrique $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ est une fonction noyau était celui d'exhiber un espace de redescritpion $\mathcal{H}$ et de vérifier que K correspond à un produit scalaire dans $\mathcal{H}$. Nous cherchons maintenant à caractériser les fonctions noyaux sans passer explicitement par l'espace de redescription.

Etant donné l'ensemble d'apprentissage $\mathcal{S}$, nous pouvons calculer la matrice de Gram (ou matrice noyau) G dont les éléments sont : $G_{ij} = K(x_i, x_j)_{1 \leq i,j \leq n}$. Une matrice réelle G de dimension $n \times n$ vérifiant la forme quadratique $Q(v) = v^T G v \geq 0$ pour tout vecteur $v \in \mathbb{R}^n$ est dite semi-définie positive. Si $Q(v) = 0$ seulement quand $v = 0$, alors la matrice G est dite définie positive. De manière équivalente, une matrice symétrique est semi-définie positive si et seulement si toutes ses valeurs propres sont positives.

Dans la pratique on choisit un noyau K qui satisfait la condition de Mercer ([23, 64]) afin de garantir la décomposition (2.20). Une fonction $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ est une fonction noyau si et seulement si elle est symétrique et semi-définie positive. Ceci est équivaut à dire que, pour toute fonction u telle que

$$\int u(x)^2 \mathrm{d}x \tag{2.22}$$

est fini (i.e. $u \in L_2(\mathcal{X})$, on a

$$\int K(x, y) u(x) u(y) \mathrm{d}x \, \mathrm{d}y \geq 0. \tag{2.23}$$

De manière équivalente, une fonction noyau définissant une matrice de Gram semi-définie positive pour tout ensemble d'apprentissage $\mathcal{S}$ est semi-définie postive.

Notons que dans certains cas, il est difficile de vérifier si les conditions de Mercer sont satisfaites, puisque la relation (2.23) doit être valable pour toute fonction u satisfaisante l'équation (2.22).

Les premiers noyaux basiques utilisés dans les problèmes de reconnaissance de forme, et en SVM notamment, ont été les suivants :

- linéaire : $K(x, z) = \langle x, z \rangle$.
- polynomial : $K(x, z) = \big(\gamma \langle x, z \rangle + \tau\big)^{\eta}$, $\gamma > 0$.
- gaussien : $k(K, z) = exp\big(- \gamma \left\| x - z \right\|^2 \big)$, $\gamma > 0$.
- sigmoîde : $K(x, z) = tanh\big(\gamma \langle x, z \rangle + \tau\big)$.

Pour plus de détails sur les méthodes de noyaux, le lecteur pourrait se référer à [79]

2.3.5 Choix du noyau et des hyperparamètres

L'implémentation d'un algorithme de SVM biclasse exige les spécifications suivantes : la fonction noyau, les hyperparamètres du noyau et le paramètre C dans C-SVM ou ν dans ν-SVM. Le réglage de ces paramètres est une étape cruciale pour obtenir un modèle performant. Il n'existe pas une méthode automatique pour spécifier les paramètres du modèle construit.

Choix du noyau

Le fait d'utiliser différents types de noyau avec différents réglages de leurs hyperparamètres aboutit à des configurations géométriques qui correspondent à une variété d'estimateurs non linéaires dans l'espace de redescription.

En général, le noyau gaussien est un choix raisonnable. Empiriquement, il fournit les meilleures performances en classification [62]. Contrairement au noyau linéaire, ce noyau peut gérer la séparation des données non linéaires. En outre, le noyau linéaire est un cas particulier du noyau gaussien, puisque le noyau linéaire avec un paramètre de pénalisation $\tilde{C}$ a la même performance qu'un noyau gaussien avec certains paramètres (C, γ) [47]. De plus, le noyau sigmoîde se comporte comme le noyau gaussien pour certains hyperparamètres [57].

Une deuxième raison justifiant le choix du noyau gaussien est le nombre des hyperparamètres à régler. Ce noyau présente l'avantage de n'avoir qu'un hyperparamètre à fixer (γ) contre trois pour les noyaux polynomiaux (γ, τ et η) et deux pour les noyaux sigmoïdes (γ et τ).

Enfin, le noyau gaussien a moins de difficultés numériques. Un point clé pour ces noyaux est que $0 < K(x, z) \leq 1$ contrairement aux noyaux polynomiaux avec lequels les valeurs peuvent aller à l'infini $\left(\gamma\langle x, z\rangle + \tau > 1\right)$ ou zéro $\left(\gamma\langle x, z\rangle + \tau < 1\right)$ quand le degré η est large. Par ailleurs, il faut noter que le noyau sigmoïde n'est pas valide (i.e. le produit scalaire de deux vecteurs) sous certains paramètres [93].

Vu les multiples avantages du noyau gaussien par rapport aux autres noyaux, nous avons décidé de choisir ce noyau lors d'une construction d'un modèle SVM. L'hyperparamètre γ contrôle la largeur du noyau gaussien. Selon Vapnick [92], le choix de la valeur de γ dépend de la dimension de l'espace de description auquel les SVM sont appliqués, et celui-ci préconise de choisir $\gamma = \frac{1}{p}$.

Choix de C et γ

Pour le choix de C, comme pour le choix de γ, on ne sait pas à l'avance quels sont les meilleurs réglages pour un problème donné. Par conséquent une sorte de sélection de modèle (choix de paramètres) doit être effectuée. L'objectif est d'identifier les valeurs idéales de (C, γ) de sorte que le classifieur donne la meilleure prédiction des futurs observations. Une stratégie commune consiste à séparer l'échantillon dont on dispose en deux parties : un ensemble d'apprentissage pour construire le modèle et un ensemble de validation afin d'obtenir le meilleur réglage de paramètres. La qualité de prédiction obtenue sur l'ensemble de validation reflète plus précisement la performance de classification sur un ensemble indépendant (i.e. qui n'a pas servi dans la phase d'apprentissage). En pratique, les jeux de données sont rarement suffisamment grands pour que l'erreur de test calculée sur l'ensemble de validation estime correctement l'erreur de prédiction du modèle. Une version améliorée de cette procédure est la validation croisée, qui est une alternative très utilisée pour gérer la parcimonie des données.

Dans une validation croisée, nous divisons tout d'abord l'échantillon d'apprentissage en V sous-ensembles de même taille. Séquentiellement, un sous-ensemble est testé en utilisant le classifieur appris sur les $V - 1$ sous-ensembles restants. Ainsi, chaque instance de données a été testé une seule fois, par suite la *précision* de la validation croisée est le pourcentage des données qui sont correctement classées.

Nous expliquons maintenant comment trouver les meilleurs paramètres C et γ à utiliser. La méthode consiste à mettre en place une "grille de recherche" (grid-search) pour C et

γ qui va utiliser la validation croisée ou l'ensemble de validation. Différentes paires de (C, γ) sont constituées, et celle avec la meilleure précision obtenue à partir d'une de deux procédures est retenue. Il a été constaté dans [43] qu'essayer des séquences de croissances exponentielles de C et γ est une méthode pratique pour identifier les bons paramètres (par exemple, $C = 2^{-5}, 2^{-3}, \ldots, 2^{15}$, $\gamma = 2^{-15}, 2^{-13}, \ldots, 2^{3}$). Après avoir identifié une meilleure région sur la grille, une recherche de grille plus fine sur cette région peut être effectuée pour trouver un couple (C, γ) qui donne une meilleure précision avec la procédure utilisée. Après avoir trouvé le meilleur couple, nous apprenons le classifieur final à partir de l'échantillon d'apprentissage tout entier.

2.3.6 Hyperplan à "marge souple" : ν-SVM

Dans plusieurs méthodes d'apprentissage statistique, un reparamétrage de l'algorithme original est utile afin de simplifier son utilisation. Dans l'algorithme C-SVM, le seul paramètre dont on dispose est la constante de régularisation C. Rappelons que C est une constante positive contrôlant le compromis entre la maximisation de la marge et la permission à quelques données d'apprentissage d'être mal classées. Comme nous avons vu dans la Section 2.3.5 (choix de C et γ), il n'est pas facile de sélectionner une valeur appropriée pour C. Pour le substituer par un paramètre avec une interprétation intéressante, une autre réalisation pour déterminer un hyperplan optimal à marge souple, appelée ν-SVM, a été introduite par Schölkopf *et al.* [78].

Le nouveau paramètre introduit $\nu \in [0, 1]$ permet de contrôler le nombre de vecteurs supports et le nombre des erreurs de la marge (observations mal classés ou situés dans la marge). Plus précisement, Schölkopf *et al.* ont démontré que ν est une borne inférieure de la fraction de vecteurs supports, et une borne supérieure de la fraction des erreurs de la marge (2.38).

Comme un problème primal pour cette approche, nous considérons

$$\begin{array}{ll} \text{Minimiser}_{\text{w,b},\rho,\xi} & \frac{1}{2}\|w\|^2 - \nu\rho + \frac{1}{n}\sum_{i=1}^{n}\xi_i, \\ \text{sous les contraintes} & y_i(\langle w,\, \phi(x_i)\rangle + b) \geq \rho - \xi_i \\ & \xi_i \geq 0,\ \text{i=1, 2,. . . , n},\ \rho \geq 0. \end{array} \tag{2.24}$$

Notons que la constante C n'apparait plus dans cette formulation. Elle a été substituée par un paramètre ν, et également par une variable supplémentaire ρ à optimiser. Pour comprendre le rôle de ρ, notons que pour $\xi = 0$, la contrainte dans (2.24) indique simplement que les deux classes sont séparées par la marge $2\rho/\|w\|$.

Pour former le dual de l'algorithme ν-SVM, nous considérons le lagrangien

$$\begin{aligned} L(w,\xi,b,\rho,\alpha,\beta,\delta) = \; & \tfrac{1}{2}\|w\|^2 - \nu\rho + \tfrac{1}{n}\textstyle\sum_{i=1}^n \xi_i \\ & - \textstyle\sum_{i=1}^n \big(\alpha_i(y_i(\langle w,\, \phi(x_i)\rangle + b) - \rho + \xi_i) + \beta_i\xi_i\big) \\ & -\delta\rho, \end{aligned} \tag{2.25}$$

en utilisant les multiplicateurs α_i, β_i, $\delta \geq 0$. Cette fonction doit être minimisée par rapport aux variables primales w, ξ, b, ρ, et maximisée par rapport aux variables duales α, β, δ. Nous calculons les dérivées partielles correspondantes, et en les posant égales à 0 nous obtenons les conditions suivantes :

$$\frac{\partial L(w,\xi,b,\rho,\alpha,\beta,\delta)}{\partial w} = 0 \iff w = \sum_{i=1}^n \alpha_i y_i \phi(x_i), \tag{2.26}$$

$$\frac{\partial L(w,\xi,b,\rho,\alpha,\beta,\delta)}{\partial \xi} = 0 \iff \alpha_i + \beta_i = \frac{1}{n}, \tag{2.27}$$

$$\frac{\partial L(w,\xi,b,\rho,\alpha,\beta,\delta)}{\partial b} = 0 \iff \sum_{i=1}^n \alpha_i y_i = 0, \tag{2.28}$$

$$\frac{\partial L(w,\xi,b,\rho,\alpha,\beta,\delta)}{\partial \rho} = 0 \iff \sum_{i=1}^n \alpha_i - \delta = \nu. \tag{2.29}$$

En substituant (2.26) et (2.27) dans L, sachant que α_i, β_i, $\delta \geq 0$, cela nous amène au problème d'optimisation quadratique suivant :

$$\text{Maximiser}_\alpha \quad W(\alpha) = -\tfrac{1}{2}\textstyle\sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j \, \langle \phi(x_i), \phi(x_j)\rangle \tag{2.30}$$

$$\text{sous les contraintes} \quad 0 \leq \alpha_i \leq \tfrac{1}{n}, \tag{2.31}$$

$$\textstyle\sum_{i=1}^n y_i \alpha_i = 0, \tag{2.32}$$

$$\textstyle\sum_{i=1}^n \alpha_i \geq \nu. \tag{2.33}$$

La fonction de décision résultante est la suivante :

$$h(x) = sgn\left(\sum_{i\in sv} y_i \alpha_i K(x_i, x) + b\right). \tag{2.34}$$

Comparée avec le dual de C-SVM (2.19), il y a deux différences. Premièrement, il existe une contrainte additionnelle (2.33). Deuxièment, le terme linéaire $\sum_{i=1}^n \alpha_i$ n'apparaît plus dans la fonction objective (2.30). Ceci a une conséquence intéressante : la fonction objective dans (2.30) est quadratique homogène en α. Il est facile de vérifier que la même fonction de décision est obtenue si nous commençons avec la fonction primale

$$\frac{1}{2}\|w\|^2 + C\big(-\nu\rho + \frac{1}{n}\sum_{i=1}^n \xi_i\big), \tag{2.35}$$

(i.e., si nous utilisons C). La seule différence étant que les contraintes (2.31) et (2.33) auraient un facteur supplémentaire C sur le côté droit. Dans ce cas, en raison de l'homogénité, la solution du dual serait normalisée par C, mais il est facile de voir que la fonction de décision correspondante ne sera pas changée. Nous pouvons donc poser $C = 1$.

Pour déterminer b et ρ, nous considérons deux ensembles $S_{\pm}$, de même taille $s > 0$, contenant des vecteurs de suport x_i avec $0 < \alpha_i < 1$ et $y_i = \pm 1$ respectivement. Puis suite aux conditions KKT, la contrainte dans (2.24) devient une égalité avec $\xi_i = 0$. Par conséquent,

$$b = -\frac{1}{2s} \sum_{x \in S_+ \cup S_-} \sum_j \alpha_j y_j K(x, x_j), \tag{2.36}$$

$$\rho = \frac{1}{2s} \Big(\sum_{x \in S_+} \sum_j \alpha_j y_j K(x, x_j) - \sum_{x \in S_-} \sum j \alpha_j y_j \langle x, x_j \rangle \Big). \tag{2.37}$$

Nous présentons maintenant la Proposition (2.3.1) dans laquelle nous donnons la démonstration établie par Schölkopf *et al.* de leur interprétation du paramètre ν. En effet, définissons tout d'abord le terme *erreurs de la marge*. Les erreurs de la marge représentent toutes les observations d'apprentissage qui sont soit mal classées ou situées dans la marge(i.e. $\xi_i > 0$). La fraction des erreurs de la marge est alors

$$R^{\rho}_{emp}[f] := \frac{1}{n} |i : y_i . f(x_i) < \rho|, \tag{2.38}$$

où f est utilisé pour dénoter l'argument du signe de la fonction décisive (2.34), tel que $h = sign \circ f$.

Proposition 2.3.1. [78][propriétés de ν]

(i) ν est une borne supérieure de la fraction des erreurs de la marge.

(ii) ν est une borne inférieure de la fraction des vecteurs supports.

Preuve. (i). D'après les conditions KKT, $\rho > 0$ implique $\delta = 0$. Par suite l'inégalité (2.33) devient une égalité (cf. Equation 2.29). Ainsi, au plus une fraction ν des observations justifient $\alpha_i = \frac{1}{n}$. (ii). Les vecteurs supports (SVs) pour lesquels les α_i sont strictement positifs peuvent chacun contribuer au plus avec $\frac{1}{n}$ (cf. Equation 2.31) au terme à gauche de l'inéquation (2.33). Donc il faut au moins νn de ces vecteurs pour satisfaire cette inéquation.

□

2.4 Machines à Vecteurs Supports à une classe (1-SVM)

Dans le contexte de la détection d'anomalies, le problème de déséquilibre de classes observé dans de nombreuses applications réelles est aussi présent dans le domaine des semi-conducteurs. En effet, la part des wafers anormaux ne constituent qu'une fraction de pourcent de l'ensemble de wafers. Dans ce cas, une classification à deux classes peut conduire à une situation où l'apprentissage est déséquilibré : les données normales peuvent être en sur-apprentissage et les données anormales en sous-apprentissage.

Afin de résoudre ce genre de problème, les méthodes de classification à une classe peuvent être utilisées. Elles possèdent la particularité de caractériser une classe cible plutôt que de discriminer deux classes. Dans un sens probabiliste, une classification à une classe est équivalente à déterminer si une observation de test est produite par la distribution sous-jacente qui correspond à l'échantillon d'apprentissage des données normales.

Une des méthodes de classification à une classe les plus connues pour la détection d'anomalies est les machines à vecteurs supports à une classe [76]. Cette méthode, notée 1-SVM (One Class Support Vectors Machines), est une extension des SVM biclasses permettant l'apprentissage d'un classifieur en absence des observations de la classe négative (les anomalies) dans les données. L'approche 1-SVM est implémentée dans La librairie LIBSVM, disponible dans les logiciels R et MATLAB.

La principale différence entre 1-SVM et le SVM classique est que, pour construire un classifieur, le premier ne nécessite que les données normales dans la phase d'apprentissage tandis que le deuxième nécessite des données issues des classes normale et anormale .

Etant donné un échantillon d'apprentissage constitué des observations normales, 1-SVM détermine le support ou la région contenant la plupart des données d'apprentissage. Ceci constitue la phase d'apprentissage. Si un point de test se situe dans cette région, il est classée comme normal, sinon il est considéré comme anormal.

1-SVM estime une région $\mathcal{R}$ dans $\mathcal{X}$ de volume minimum contenant au moins $(1-\nu)n$ données, ν déterminant asymptotiquement la proportion de données hors volume. Comme dans le cas du SVM biclasse, la frontière de $\mathcal{R}$ est recherchée dans un espace transformé $\mathcal{H}$ de plus grande dimension. Puis un classifeur linéaire est construit dans le nouvel espace. Ceci est équivalent à construire un classifeur non linéaire dans l'espace d'entrée des données $\mathcal{X}$.

Dans 1-SVM, une hypothèse inhérente est que l'origine de l'espace de redescription appartient à la classe d'anomalies. Par conséquent l'objectif est de faire la séparation entre l'origine et les observations de la classe cible dans un espace de dimension plus grande

avec une marge maximale. Pour résumer, 1-SVM développe un hyperplan dans un espace transformé qui retourne une valeur positive pour les observations normales et une valeur négative pour les anomalies, en se basant sur la maximisation de la distance perpendiculaire de cet hyperplan à l'origine.

2.4.1 Algorithmes

Nous considérons toujours l'échantillon d'apprentissage $\mathcal{S}$ et supposons que les observations sont distribuées suivant une distribution de probabilité sous-jacente inconnue Ω. Nous souhaitons savoir si une nouvelle observation est issue de Ω ou non. Cela peut être fait en déterminant une région $\mathcal{R}$ dans l'espace d'entrées $\mathcal{X}$ telle que la probabilité qu'une nouvelle observation générée suivant Ω et n'appartenant pas à $\mathcal{R}$ soit majorée par une valeur $\nu \in (0, 1)$ spécifiée a priori. Ce problème est résolu en estimant une fonction de décision f qui est positive sur $\mathcal{R}$ et négative ailleurs.

Une fonction non linéaire $\phi : \mathcal{X} \to \mathcal{H}$ transforme les observations d'apprentissage de l'espace d'entrées $\mathcal{X}$ dans un espace de Hilbert $\mathcal{H}$. Dans ce nouvel espace, les données d'apprentissage suivent une distribution sous-jacente Ω' et le problème sera de déterminer une région $\mathcal{R}'$ de $\mathcal{H}$ qui contient la majorité des observations de cette distribution. Un hyperplan $H(w, \rho)$ est construit dans l'espace transformé et est défini par

$$H(w, \rho) = \langle w, \phi(x) \rangle - \rho \tag{2.39}$$

où w est le vecteur de poids qui est un vecteur normal à H, ρ est le biais et $\rho / \|w\|$ spécifie la distance de l'hyperplan à l'origine (la marge). Notons que nous avons remplacé le biais b des hyperplans considérés dans les SVM biclasses par le biais ρ afin de permettre la comparaison avec ν-SVM. En effet, la marge dans 1-SVM est égale à $\rho / \|w\|$, alors que celle de ν-SVM est égale à $2\rho / \|w\|$. Le biais ρ, ajouté dans la fonction objective (2.41), permet d'obtenir également comme pour le paramètre supplémentaire ρ introduit dans ν-SVM les mêmes propriétés de ν (Proposition 2.3.1).

Pour obtenir une séparation maximale entre l'hyperplan et l'origine, on cherche à maximiser la marge donc à minimiser $\|w\|$. Ceci revient à résoudre le problème d'optimisation quadratique suivant

$$\begin{array}{ll} Minimiser_{w,\rho} & \frac{1}{2} \|w\|^2 \\ \text{sous les contraintes} & \langle w, \phi(x_i) \rangle \geq \rho, \; i = 1, \ldots, n. \end{array} \tag{2.40}$$

Cependant, il n'est pas toujours possible de séparer parfaitement les observations de la classe cible de l'origine dans l'espace de redescription. Pour de tels cas, l'algorithme résolvant le problème d'optimisation (2.40) ne converge pas vers une solution finie. Pour tenir

compte de ce problème, on définit un paramètre $\nu \in (0, 1)$ et des variables d'écart ξ_i pour relâcher les contraintes. ν correspond à la fraction maximale des erreurs d'apprentissage, i.e. les observations situées entre l'hyperplan et l'origine. Les variables ξ_i sont utilisées pour pénaliser la fonction objective en permettant à certaines observations d'être mal classées. Le problème d'optimisation modifié peut être écrit :

$$\begin{array}{ll} Minimiser_{w,\xi_i,\rho} & \frac{1}{2}\|w\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho \\ \text{sous les contraintes} & \langle w, \phi(x_i)\rangle \geq \rho - \xi_i, \quad \xi_i \geq 0,\ i = 1, \ldots, n. \end{array} \tag{2.41}$$

D'où si w et ρ sont solutions du problème (2.41), nous nous attendons alors à ce que la fonction de décision

$$h(x) = sgn\left(\langle w.\phi(x)\rangle - \rho\right) \tag{2.42}$$

soit positive pour la plupart des données d'apprentissage x_i alors que le terme $\|w\|$ reste petit. Le compromis entre ces deux objectifs est controlé par ν. Plus précisement, ν contrôle le compromis entre maximiser la distance à l'origine et contenir la plupart des observations dans la région construite par l'hyperplan. Scholkopf et al. [75] ont démontré que ν est une borne supérieure de la fraction des erreurs d'apprentissage (données d'apprentissage situées entre l'hyperplan et l'origine), et une borne inférieure de la fraction des vecteurs supports (cf. Proposition 2.4.3 de la section 2.4.3).

La Figure 2.5 représente un schéma 2D d'un classifieur 1-SVM. Le noyau gaussien transforme les données sur une hypersphère, et 1-SVM vise à trouver un hyperplan dans cet espace transformé séparant la majorité des données de l'origine avec une marge maximale, tout en permettant à quelques observations d'être mal classées.

En utilisant les multiplicateurs de Lagrange α_i, $\beta_i \geq 0$, nous introduisons le Lagrangien (2.43), puis les dérivées partielles du Lagrangien par rapport aux variables primaires w, ξ, b sont annulées (2.44–2.46).

$$\begin{array}{rl} L(w, \xi, \rho, \alpha, \beta) = & \frac{1}{2}||w||^2 + \frac{1}{\nu l}\sum_i \xi_i - \rho \\ & - \sum_i \alpha_i \left(\langle w, \phi(x_i\rangle - \rho + \xi_i\right) \\ & - \sum_i \beta_i \xi_i, \end{array} \tag{2.43}$$

$$\frac{\partial L(w, \xi, \rho, \alpha, \beta)}{\partial w} = 0 \iff w = \sum_{i=1}^{n} \alpha_i \phi(x_i), \tag{2.44}$$

$$\frac{\partial L(w, \xi, \rho, \alpha, \beta)}{\partial \xi} = 0 \iff \alpha_i = \frac{1}{\nu n} - \beta_i \leq \frac{1}{\nu n} \tag{2.45}$$

$$\frac{\partial L(w, \xi, \rho, \alpha, \beta)}{\partial \rho} = 0 \iff \sum_{i=1}^{n} \alpha_i = 1. \tag{2.46}$$

En substituant (2.44) et (2.45) dans (2.43), et en utilisant un noyau approprié (2.20), la solution du problème est équivalente à la solution de la forme duale

$$\begin{array}{ll} Minimiser_{\alpha} & \frac{1}{2}\sum_{ij}\alpha_i\alpha_j K(x_i,x_j) \\ \text{sous les contraintes} & 0 \leq \alpha_i \leq \frac{1}{\nu n} \\ & \sum_i \alpha_i = 1. \end{array} \tag{2.47}$$

Dans (2.44), toutes les observations x_i telles que $\alpha_i > 0$ sont appelées vecteurs supports. En effet parmi les données d'apprentissage, les vecteurs supports determinent uniquement l'hyperplan séparateur. En utilisant (2.20) et (2.44) la fonction de décision (2.42) devient

$$h(x) = sgn\left(\sum_{i\in sv}\alpha_i K(x_i,x) - \rho\right). \tag{2.48}$$

On peut montrer que, à l'otpimum, les deux contraintes d'inégalité dans (2.41) deviennent des égalités si α_i et β_i sont non nuls, i.e. si $0 < \alpha_i < \frac{1}{\nu n}$. Par suite, nous pouvons déterminer ρ en exploitant que, pour de tels α_i, les observations x_i correspondants satisfont

$$\rho = \langle w, \phi(x_i)\rangle = \sum_j \alpha_i K(x_j, x_i). \tag{2.49}$$

Les propriétés de base du SVM à une classe ont été présentées et démontrées dans [75]. Le résultat le plus important est l'interprétation de ν à la fois comme la fraction asymptotique des outliers, et la fraction des vecteurs support retournés par l'algorithme.

Comme nous l'avons vu, les SVM à une classe et les SVM biclasses sont très similaires. Les SVM biclasses déterminent un hyperplan de séparation en maximisant la distance entre les deux classes, alors que les SVM à une classe séparent les données d'apprentissage de l'origine avec une marge maximale.

2.4.2 Choix du noyau et des hyperparamètres

Pour les raisons citées dans la Section 2.3.5, nous avons décidé de choisir le noyau gaussien pour apprendre le modèle 1-SVM. Par conséquent, deux hyperparamètres de ce modèle sont à régler : ν et γ. Dans la Section 4.3 du Chapitre 4, nous montrons que l'algorithme 1-SVM n'est pas sensible au hyperparamètre γ du noyau gaussien. Ainsi une petite plage de valeurs est suffisante pour déterminer une valeur optimale pour cet hyperparamètre. La méthode basée sur la "grille de recherche" (grid-search) pour C et γ introduite dans la Section 2.3.5 peut être également appliquée pour choisir la meilleure

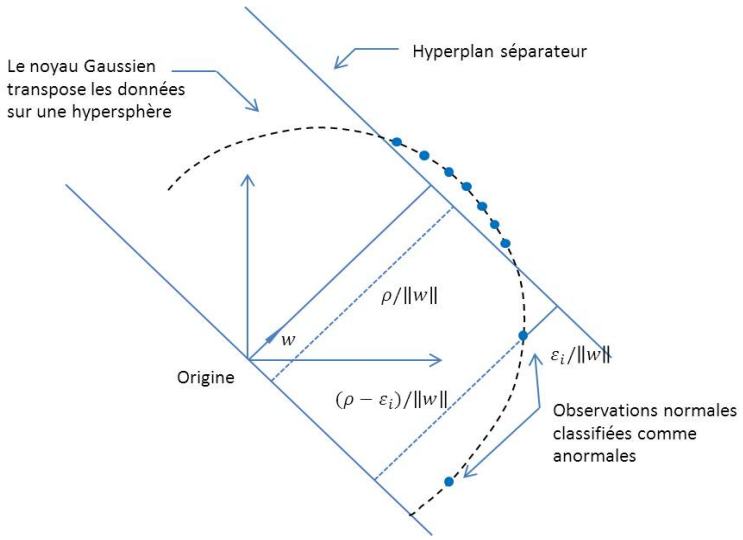

FIGURE 2.5 – Schéma d'un classifieur 1-SVM dans $\mathbb{R}^2$.

paire (ν, γ). Différentes paires de (ν, γ) sont constituées, et celle avec la meilleure précision obtenue à partir de la procédure basée sur la validation croisée ou la procédure basée sur l'ensemble de validation est retenue. On peut par exemple prendre $\gamma = 1/mp$ avec $m \in \{1,\ 2,\ 3,\ 4,\ 5\}$, et $\nu \in \{0.01,\ 0.02, \ldots,\ 0.15\}$.

2.4.3 Résultats théoriques

Dans cette section (extraite des résultats théoriques de [76]), l'algorithme 1-SVM est analysé théoriquement. On commence par l'unicité de l'hyperplan (Proposition 2.4.1). Ensuite une liaison avec la classification binaire sera établie (Proposition 2.4.2). Enfin, on démontre que le paramètre ν caractérise les fractions des vecteurs support et des erreurs de la marge (Proposition 2.4.3).

Définition 2.4.1. Un jeu de données

$$x_1, \ldots,\ x_n \tag{2.50}$$

est dit séparable de l'origine s'il existe un vecteur $w \in \mathcal{H}$ tel que $\langle w,\ \phi(x_i)\rangle > 0$ pour $i \in \{1, \ldots,\ n\}$.

Si on utilise un noyau gaussien, alors n'importe quel jeu de données est séparable après avoir transformé les données dans l'espace de Hilbert $\mathcal{H}$. En effet, notons tout d'abord que $K(x_i,\ x_j) > 0$ pour tout $i,\ j$, donc le produit scalaire entre toutes les données dans l'espace transformé est positif, ce qui implique que toutes les données se trouvent dans le même orthant. De plus, comme $K(x_i,\ x_i) = 1$ pour tout i, les données ont toutes une norme égale à 1. D'où elles sont séparables de l'origine.

Proposition 2.4.1. [76] [hyperplan support]

Si le jeu de données (2.50) est séparable, alors il existe un unique hyperlan support avec les propriétés suivantes : (i) il sépare toutes les observations de l'origine, et (ii) sa distance à l'origine est maximale parmi tous ces hyperplans. Pour tout $\rho > 0$, il est défini par

$$\underset{w\in\mathcal{H}}{min}\ \frac{1}{2}\|w\|^2 \quad sous\ \ \langle w,\ \phi(x_i)\rangle \geq \rho,\ i \in \{1,\ldots,\ n\} \tag{2.51}$$

Preuve. Pour une raison de séparabilité, l'enveloppe convexe des données ne contient pas l'origine. L'existence et l'unicité de l'hyperplan résulte alors du théorème de l'hyperplan support (e.g. [9]). En outre, la séparabilité implique qu'il existe effectivement un $\rho > 0$ et $w \in \mathcal{H}$ tels que $\langle w,\ \phi(x_i)\rangle \geq \rho$ pour $i \in \{1,\ldots,\ n\}$. La distance de l'hyperplan $\{z \in \mathcal{H}\ :\ \langle w,\ z\rangle = \rho\}$ à l'origine est égale à $\rho/\|w\|$. Par conséquent l'hyperplan optimal est obtenu en minimisant $\|w\|$ sous ces contraintes, i.e. par la solution de (2.51).

□

Le résultat suivant illustre la liaison entre la classification à une classe et la classification binaire.

Proposition 2.4.2. [76][liaison avec reconnaissance de formes]

(i) Supposons que (w, ρ) soient les paramètres de l'hyperplan support des données (2.50). Alors $(w, 0)$ paramétrise l'hyperplan séparateur optimal des données étiquetées

$$\{(x_1,\ 1),\ldots,\ (x_n,\ 1),\ (-x_1,\ -1),\ldots,\ (-x_n,\ -1)\} \tag{2.52}$$

(ii) Supposons que $(w, 0)$ soient les paramètres de l'hyperplan séparateur otpimal passant par l'origine, des données étiquettées

$$\{(x_1,\ y_1),\ldots,\ (x_n,\ y_n)\},\quad (y_i \in \{\pm 1\} pour\ i \in \{1,\ldots,\ n\}),$$

telles que $\langle w,\ \phi(x_i)\rangle$ est positif si $y_i = 1$. Supposons de plus que $\rho/\|w\|$ est la marge de l'hyperplan optimal. Alors (w, ρ) constituent les paramètres de l'hyperplan séparateur otpimal des données non étiquettées

$$\{y_1x_1,\ldots,\ y_nx_n\}. \tag{2.53}$$

Preuve. (i). Par construction, la séparation de (2.52) est un problème de points symétriques. D'où l'hyperplan séparateur optimal passe par l'origine, car sinon nous pouvons obtenir un autre hyperplan séparateur optimal en construisant l'hyperplan symétrique

du premier hyperplan par rapport à l'origine. Ce serait en contradiction avec l'unicité de l'hyperplan séparateur optimal de Vapnick [93].
Ensuite, observons que $(-w, \rho)$ paramétrise l'hyperplan support des données symétriques par rapport à l'origine, et qu'il est parallèle à l'hyperplan paramétré par (w, ρ).Ceci fournit une séparation optimale de deux ensembles de données, avec une distance 2ρ, et un hyperplan séparateur paramétré par $(w, 0)$.
(ii). Par hypothèse, w est le vecteur minimal (minimisant la norme euclidienne) satisfaisant $y_i\langle w,\, \phi(x_i)\rangle \geq \rho$ (notons que le biais est égal à 0). Ainsi, de manière équivalente, il est le vecteur minimal satisfaisant $\langle w,\, y_i\phi(x_i)\rangle \geq \rho$ pour $i \in \{1, \ldots,\, n\}$).

□

Notons que la relation est similaire dans le cas des données non séparables. Dans ce cas, les observations mal classées en classification binaire (i.e. observations qui sont soit sur le mauvais coté de l'hyperplan séparateur ou qui se situent dans la marge) s'interprétent comme des anomalies dans la classification à une classe , i.e. en des observations qui se situent entre l'hyperplan séparateur et l'origine.
L'utilité de la Proposition 2.4.2 réside dans le fait qu'elle nous permet de réutiliser certains résultats démontrés en classification binaire ([78]). La proposition suivante expliquant l'importance du paramètre ν, est un tel cas.

Proposition 2.4.3. [76][propriétés de ν]

(i) ν est une borne supérieure de la fraction des erreurs d'apprentissage.

(ii) ν est une borne inférieure de la fraction des vecteurs supports.

Preuve. Les parties (i) et (ii) résultent directement de la Proposition 2.4.2 et le fait que les erreurs d'apprentissage sont traitées de la même manière que dans le problème d'optimisation pour la classification binaire [78]. L'idée de base est que l'inégalité en (2.45) impose des contraintes sur la fraction des données qui pourrait avoir $\alpha_i = 1/(\nu l)$, i.e. la borne supérieure de la fraction des erreurs d'apprentissage, et sur la fraction des données qui devraient avoir $\alpha_i > 0$, i.e. les vecteurs supports. □

Chapitre 3

Réduction de dimension

3.1 Compression et sélection de variables

Dans notre travail, nous disposons de données de dimension élevée, puisque les wafers sont décrits par plusieurs centaines de paramètres électriques. Par conséquent une réduction de dimension nous paraît indispensable afin d'améliorer la performance prédictive de l'algorithme de classification utilisé.

En apprentissage automatique et en statistique, la réduction de dimension est le processus de réduction du nombre de variables de l'étude considérée [73]. Il existe deux types d'approches : la compression de variables et la sélection de variables [69]. Les approches de sélection de variables essaient de trouver un sous-ensemble optimal des variables afin d'améliorer la qualité de prédiction du modèle d'apprentissage. Les approches de compression de variables transforment les données d'un espace de grande dimension dans un espace de dimension plus petite. La transformation de données peut être linéaire, comme l'analyse en composantes principales (ACP), mais de nombreuses techniques de réduction de dimensionnalité non linéaires existent également [26]. Par exemple, l'ACP à noyau (kernel PCA) [77] est une généralisation de l'ACP linéaire permettant une réduction de dimensionnalité non lineaire. Comme dans la méthode SVM (cf. Section 2.3.4), la fonction noyau permet de projeter les données dans un espace de plus grande dimension de sorte que la varieté devienne linéaire et d'effectuer l'ACP dans cet espace.

Dans ce chapitre, nous présenterons une des techniques les plus connues pour la compression de variables : l'analyse en composantes principales (ACP). L'ACP réalise une transformation linéaire des données dans un nouvel espace de dimension inférieure, de telle façon que la variance des données dans cet espace soit maximisée. Nous montrons comment une ACP contribue à la détection d'anomalies. Enuite nous parlerons des différentes catégories de sélection de variables, et nous présenterons en particulier deux

méthodes que nous avons spécifiquement dévelopées pour une meilleure performance de détection avec l'algorithme 1-SVM. La première méthode de type filtrage est basée sur un score calculé avec le filtre MADe, une approche robuste pour la détection univariée des valeurs aberrantes. La deuxième méthode de type wrapper est une adaptation à l'algorithme 1-SVM de la méthode d'élimination récursive des variables basée sur la variation du vecteur de poids de l'algorithme SVM.

3.2 Compression de variables : Analyse en Composantes Principales (ACP)

L'analyse en composantes principales est une méthode de statistique exploratoire permettant de décrire un grand tableau de données de type individus / variables. L'intérêt majeur d'une ACP est d'offrir une meilleure visualisation possible des données multivariées, en identifiant les hyperplans dans lesquels la dispersion est maximale, mettant en évidence avec le maximum de précision les relations de proximité et d'éloignement entre les variables [55].Une ACP permet donc de fournir une représentation graphique "optimale" des observations projetées dans un sous-espace de dimension réduite ($d < p$), minimisant la déformation du nuage de points initial.

Mathématiquement, l'ACP correspond à l'approximation de la matrice de données X par une matrice T de même dimension $(n,\ p)$ mais de rang $d < p$. Ceci est réalisé à travers une projection orthogonale linéaire qui projette les observations multidimensionnelles représentées dans un sous-espace de dimension p dans un sous-espace de dimension inférieure d en maximisant la variance des projections. La solution de ce problème de maximisation est basée sur le calcul des valeurs et vecteurs propres de la matrice de covariance des données, puisqu'on analyse essentiellement la dispersion des données considérées. Cette solution définit à la fois la projection du sous-espace de dimension p dans le sous-espace de dimension d et la projection inverse permettant de reconstruire les données observées. L'ACP peut être donc considérée comme une technique de minimisation de l'erreur quadratique de reconstruction ou une technique de maximisation de la variance des projections (il faut noter que ces deux critères sont équivalents).

3.2.1 Minimisation d'erreur/maximisation variance

L'objectif d'une ACP étant de trouver un sous-espace de dimension $d < p$ qui permet d'avoir une représentation réduite de X, cette méthode consiste à trouver une nouvelle base dans laquelle la projection du nuage de points initial est la plus fidèle possible. Cette nouvelle base est obtenue par combinaison linéaire de la base originelle. C'est donc tout

simplement un problème de changement de base. Soit P une matrice de changement de base de dimension $p \times p$ telle que $P = [p_{.1} \dots p_{.j} \dots p_{.p}]$, où $p_{.j} = (p_{1j}, \dots, p_{pj})^T$. Un nombre d des colonnes de P vont former les vecteurs de base orthonormés d'un sous-espace $\mathbb{R}^d$ de représentation réduite des données. Ces vecteurs sont orthogonaux 2 à 2 (i.e. $p_{.j}^T p_{.k} = 0$ si $j \neq k$) et de norme égale à 1 (i.e. $p_{.j}^T p_{.j} = 1$). La matrice $T = [t_{1.} \dots t_{i.} \dots t_{n.}]^T = [t_{.1} \dots t_{.j} \dots t_{.p}]$ des scores des observations dans la nouvelle base est telle que $T = XP$.

Au sens de l'ACP, on construit P de sorte que la représentation réduite minimise l'erreur de reconstruction de X. Autrement dit, la projection à travers P est optimale si le critère d'erreur quadratique moyenne MSE(P) (Mean Square Error) d'approximation des $\{x_{i.}\}_{(i=1,\dots,n)}$ à l'aide des d premières colonnes de cette matrice est minimale.

Sous la contrainte d'orthogonalité de P, et sous l'hypothèse de la nullité de la moyenne d'une observation x_i, le critère MSE(P) est égal à :

$$MSE(P) = trace(\hat{\Sigma^2}) - trace(P^T \hat{\Sigma^2} P), \tag{3.1}$$

où $\hat{\Sigma^2}$ est la matrice de covariance empirique des variables de terme général $\hat{\Sigma_{jk}^2} = cov\,(x_{.j}, x_{.k})$, $cov\,(x_{.j}, x_{.k})$ étant la covariance entre les variables $x_{.j}$ et $x_{.k}$. Quand X est centrée, cette matrice carré symétrique d'ordre p est égale à

$$\hat{\Sigma^2} = \frac{1}{n} X^T X. \tag{3.2}$$

Comme la matrice de covariance $\hat{\Sigma^2}$ est indépendante de P, minimiser $MSE(P)$ revient à maximiser le second terme de son expression dans l'Equation (3.1) que nous notons $J_v(P)$. Ainsi, la minimisation de l'erreur quadratique d'approximation est équivalente à la maximisation de la variance des projections des données :

$$P_{opt} = arg\,\min_P MSE(P) = arg\,\max_P J_v(P). \tag{3.3}$$

3.2.2 Axes factoriels et optimisation

Avant de résoudre ce problème d'optimisation, nous introduisons les notions "axes factoriels" et "composantes principales". On appelle premier axe factoriel de X le vecteur $p_{.1}$ (première colonne de la matrice P) sur lequel le nuage se déforme le moins possible en projection, donc c'est le vecteur $p_{.1}$ tel que la variance de $Xp_{.1}$ est maximale. Le vecteur $t_{.1} = Xp_{.1}$ est appelé première composante principale. D'une façon plus générale, Le kième axe factoriel de X est le vecteur $p_{.k}$ unitaire tel que la variance de la kième

composante principale $t_{.k} = Xp_{.k}$ soit maximale et que $p_{.k}$ soit orthogonal aux $k-1$ premiers axes factoriels.

Notre objectif maintenant est de chercher un sous-espace de dimension 1 engendré par $p_{.1}$ avec comme contrainte $p_{.1}^T p_{.1} = 1$. La maximisation de la variance de projection sur $p_{.1}$, sous condition de norme unité du vecteur $p_{.1}$, est donc un problème d'optimisation sous contraintes d'égalité qui peut être résolu avec la méthode des multiplicateurs de Lagrange. Le problème d'optimisation est le suivant :

$$\begin{array}{ll} \text{Maximiser}_{\mathrm{P}_1} & J_v(P_1) = P_1^T \hat{\Sigma^2} P_1, \\ \text{sous la contrainte} & P_1^T P_1 = 1. \end{array} \tag{3.4}$$

La résolution du problème (3.4) montre que le premier axe factoriel $p_{.1}$ est le vecteur propre associé à λ_1, la plus grande valeur propre de $\hat{\Sigma^2}$. De façon plus générale, le $k$ième axe factoriel est le vecteur propre $p_{.k}$ associé à λ_k, la $k$ième plus grande valeur propre de $\hat{\Sigma^2}$. Rappelons que les valeurs propres de $\hat{\Sigma^2}$ sont positives car $\hat{\Sigma^2}$ est une matrice semi-définie positive.

En outre, les valeurs propres de la matrice de covariance $\hat{\Sigma^2}$ représentent les variances des projections des données sur les directions représentées par les vecteurs propres $p_{.j}$ ($j = 1, \ldots, p$). Les p vecteurs propres unitaires $p_{.j}$ représentent les p directions orthogonales de l'espace des données suivant lesquelles les variances des projections des données sont maximales.

Le sous-espace vectoriel de dimension d qui assure une dispersion maximale des observations est donc défini par une base orthonormée formée des d vecteurs propres correspondant aux d plus grandes valeurs propres de la matrice $\hat{\Sigma^2}$. Ce sous-espace constitue donc un espace de représentation optimale des données.

3.2.3 Réduction de la dimensionalité

Nous avons vu qu'une ACP permet de caractériser les directions orthogonales d'un espace de données porteuses du maximum d'information au sens de la maximisation des variances de projections. L'amplitude des valeurs propres de la matrice de covariance $\hat{\Sigma^2}$ des données quantifie pour chacune de ces directions la quantité d'information encodée.

Une approximation $\tilde{x_{i.}}$ de l'observation $x_{i.}$ par l'intermédiaire des d composantes principales $t_{i1}, \ldots, t_{id}$ présentant les plus fortes variances est donnée par

$$\tilde{x_{i.}} = \sum_{j=1}^{d} t_{ij} p_{.j}. \tag{3.5}$$

La perte d'information induite par la réduction de dimension de représentation de chaque $x_{i.}$ est mesurée par la différence e entre ses représentations exacte et approchée :

$$e = x_{i.} - \tilde{x_{i.}} = \sum_{j=d+1}^{p} t_{ij} p_{.j} \tag{3.6}$$

L'erreur d'approximation sur un sous-espace vectoriel de dimension d est :

$$E_d = \frac{1}{n} \sum_{i=1}^{n} \|x_{.i} - \tilde{x_{i.}}\|^2 = \sum_{k=d+1}^{p} \lambda_k.$$

Les $p - d$ composantes principales $t_{.j}$ $(j = d+1, \ldots, p))$ à partir desquelles l'erreur d'estimation e est évaluée, sont associées aux plus faibles valeurs propres $\lambda_{q+1}, \ldots, \lambda_p$. Il est par conséquent bien évident que la compression de données préserve d'autant mieux d'information que ces valeurs propres sont faibles.

3.2.4 Choix de dimension

Pour une reconstruction optimale de données, le choix du nombre de composantes principales à retenir d est crucial dans une méthode ACP. Toutefois dans le cadre de l'application de l'ACP à la détection d'anomalies (cf. Chapitre 4, Section 3.3), le nombre de composantes a un impact significatif sur la qualité de détection. Si trop peu de composantes sont utilisées, on risque de perdre des informations contenues dans les données de départ. Si par contre trop de composantes sont utilisées, il y a le risque d'avoir des composantes retenues (les composantes correspondantes aux valeurs propres les plus faibles parmi celles retenues dans le modèle) qui sont porteuses de bruit ce qui est indésirable.

De nombreuses règles ont été proposées dans la littérature [33, 91]. Dans la suite nous allons présenter quelques critères utilisés pour le choix optimal de d.

Pourcentage cumulé de la variance totale (PCV) Le pourcentage de variance (ou la part d'inertie) expliquée par un sous-espace d'ordre d engendré par les d premiers axes est donné par :

$$PCV(d) = \frac{\sum_{k=1}^{d} \lambda_k}{\sum_{k=1}^{p} \lambda_k}. \tag{3.7}$$

Le nombre de composantes à retenir est le plus petit nombre d pour lequel au moins une fraction ψ (prédéfinie par l'utilisateur) de la variance totale soit expliquée.

$$d = arg\,\min_{u} \{PCV(u) \geq \psi\}$$

Par exemple, si nous prenons $\psi = 0.8$, nous retenons le nombre minimal de composantes qui préserve 80% de l'information contenue dans les données originelles.

Règle de Kaiser Cette règle consiste à ne prendre en considération que les composantes pour lesquelles l'inertie est supérieure à l'inertie moyenne. En ACP normée, l'inertie totale est $trace(\hat{\Sigma^2}) = p$. On ne retiendra donc que les composantes associées à des variances supérieures à l'unité. Ce critère, utilisé implicitement par *SAS*, a tendance à surestimer le nombre de composantes pertinentes.

Eboulis L'éboulis est un graphique présentant la décroissance des valeurs propres. Le principe consiste à rechercher, s'il existe, un "coude" suivi d'une décroissance régulière. On sélectionne les axes se situant avant le coude. Les axes qui suivent ce coude semblent former une ligne droite horizontale, i.e. la variation des valeurs propres correspondant à n'importe quel deux axes consécutifs situés après le coude est trop faible par rapport aux variations observées avant le coude. L'information ajoutée par les axes qui suivent le coude est donc peu pertinente.

3.3 ACP et détection d'anomalies

Les méthodes de détection d'anomalies reposant sur l'analyse en composantes principales linéaires (ACP) ont reçu une attention particulière et ont été largement utilisées pour la surveillance des processus industriels ([60, 66, 81, 90]). Le principe de cette approche est d'utiliser l'analyse en composantes principales pour une compression de variables, et puis de déterminer le T^2 de Hotteling à partir des scores de données obtenues et le comparer à un seuil prédéfini.

Après avoir établi le modèle ACP, une observation x_i est projetée sur le modèle et la statistique T^2 de Hotteling (cf. Section 1.7.2) peut être déterminée à partir des d premières composantes principales $\tilde{t_{i.}} = (t_{i1}, \ldots, t_{id})$:

$$T_i^2 = \tilde{t_{i.}}^T (\hat{\Sigma}_d^2)^{-1} \tilde{t_{i.}} \tag{3.8}$$

où $\hat{\Sigma}_d^2 = diag(\lambda_1, \lambda_2, \ldots, \lambda_d)$ est une matrice diagonale contenant les d plus grandes valeurs propres de la matrice de covariance des données. La statitique de Hotteling peut alors s'écrire sous la forme suivante :

$$T_i^2 = \sum_{j=1}^{d} \frac{t_{ij}^2}{\lambda_j}$$

Le seuil approprié pour l'indice T^2 pour un risque α peut être déterminé comme dans l'Equation (1.3) de la Section 1.7.2, en remplaçant p par d. Une observation est considérée anormale si son T^2 de Hotteling dépasse le seuil, et normale sinon.

Puisque l'indice T^2 n'est pas affectée par le bruit, qui est représenté par les dernières valeurs propres, théoriquement elle est capable de représenter le comportement normal du processus. L'indice T^2 peut être interprété comme la mesure des variations normales du processus, et la violation du seuil de détection de cette statistique indique que ces variations sont en dehors des limites de contrôle et correspondent à un fonctionnement anormal.

Pour le choix de nombre de composantes d à retenir, nous utilisons la méthode de Pourcentage cumulé de la variance totale (PCV) (cf. Section 3.2.4).

3.4 Sélection de variables

Le problème de sélection de variables est crucial dans le domaine de l'apprentissage statistique et plus particulièrement dans le cadre de la classification supervisée. La complexité de nombreuses techniques de classification dépend fortement du nombre des variables décrivant les instances de données. La sélection de variables joue un rôle important en classification lorsqu'un grand nombre p de variables sont disponibles. Certaines variables pouvent être peu significatives, redondantes ou non pertinentes au regard de l'application considérée. Les méthodes de sélection de variables consistent à retenir un sous-ensemble optimal de m variables parmi les $p >> m$ variables disponibles qui améliore les performances de l'algorithme de classification .

Généralement, une méthode de sélection de variables repose principalement sur les trois composantes suivantes :

- Un algorithme de recherche permettant d'explorer l'espace des combinaisons de variables.
- Un critère d'évaluation pour mesurer la pertinence des sous-ensembles potentiels de variables.
- Une condition d'arrêt pour arrêter la procédure de recherche.

Dans la sélection de variables, nous nous intéressons à la réduction de la dimension de l'espace des variables explicatives, sans pour autant transformer ses composantes comme dans les méthodes d'analyse factorielle (compression de variables).

D'un point de vue industriel, la sélection des variables (les paramètres électriques pertinents) devrait permettre d'effectuer un diagnostic des wafers anormaux. Ce diagnostic

a pour but d'apporter plus de connaissances sur la raison de l'anomalie en définissant des signatures de ces wafers basées sur les paramètres électriques les plus pertinents. La détermination de la source ou raison d'anomalie (localisation de défauts) nous permet de remonter à la cause (root cause) et ensuite remettre le processus dans un état de fonctionnement normal après avoir effectué un correctif adapté.

D'un point de vue statistique, la sélection d'un sous-ensemble de variables pertinentes permettrait d'améliorer la performance de détection des méthodes de classification, d'accélérer le temps de calcul de ces méthodes et enfin de comprendre le processus sous-jacent ayant généré ces données [37]. Nous améliorons ainsi notre connaissance du phénomène de causalité entre les descripteurs (les paramètres électriques) et la variable à prédire (l'état de fonctionnement des wafers), ce qui est fondamental si nous voulons interpréter les résultats pour en assurer la reproductibilité.

Dans la littérature du Machine Learning, trois catégories de méthodes existantes pour la sélection de variables sont considérées et présentées dans les revues bibliographiques de [10, 37, 49] :

1. les méthodes de *filtrage*,
2. les méthodes de type *wrapper*,
3. les méthodes de type *embedded*.

Chaque méthode de sélection de variables est répartie dans une de ces trois catégories selon le type du critère de sélection et la façon dont il est pris en compte dans la procédure de classification.

Les méthodes de filtrage

Les méthodes de filtrage utilisent les propriétés statistiques des variables afin de filtrer les variables non informatives. Ces méthodes sélectionnent des sous-ensemble de variables comme une étape de pré-traitement, indépendamment du prédicteur choisi. Dans ce type de méthodes, les critères d'évaluation sont fondés uniquement sur les données et sont donc totalement indépendants du discriminateur utilisé. Les variables alors sont filtrées avant le processus d'apprentissage et de classification. Différentes fonctions sont utilisées pour définir les critères d'évaluation. Certaines sont basées sur des mesures de distance probabilistes (ex : distance de Mahalanobis), d'autres sont fondées sur les mesures d'information (ex : entropie) ou de dépendance (ex : coefficient de corrélation de Pearson, information mutuelle, score de Fisher).

Le principal avantage des méthodes de filtrage est leur efficacité calculatoire et leur robustesse face au surapprentissage (ou surajustement). Malheureusement, ces méthodes

ne tiennent pas compte des interactions qui peuvent exister entre les variables et tendent à sélectionner des variables comportant de l'information redondante plutôt que complémentaire [37]. De plus, elles ne tiennent pas compte de la performance des méthodes de classification appliquées dans une deuxième étape.

Les méthodes de type wrapper

Les méthodes de type wrapper consistent en l'évaluation de la performance de sous-ensembles de variables de manière successive, prenant ainsi en compte les interactions entre variables. Ainsi, l'algorithme de sélection "entoure" (wrapp) la méthode de classification qui évalue la performance. En fait, la machine d'apprentissage est considérée comme une boîte noire et la méthode de sélection se prête à l'utilisation de cette machine. Plus précisement, Ces méthodes consistent à utiliser les résultats de prédiction de la méthode d'apprentissage afin d'évaluer l'utilité relative des sous-ensembles de variables. Par exemple, la méthode SVM-RFE (Recursive Feature Elimination with SVM) [38] est une méthode wrapper basée sur un score calculé à partir de l'algorithme SVM.

La recherche d'un tel sous-ensemble de variables optimal requiert certaines définitions au préalable [37] : comment rechercher dans l'espace des variables tous les sous-ensembles possibles, comment évaluer la performance de prédiction d'une méthode d'apprentissage pour guider la recherche, quand arrêter l'algorithme. Bien entendu, une recherche exhaustive est un problème NP-difficile et incalculable lorsque p est grand ; il nécessite des approximations des calculs d'optimisation. Le risque de surapprentissage est grand si le nombre d'observations n est insuffisant et le nombre de variables à sélectionner doit être choisi par l'utilisateur. Enfin, le plus grand désavantage de ces méthodes est le temps de calcul qui devient vite important dès que p est grand.

John et al. [46] prétendent que la stratégie wrapper est supérieure à la stratégie filtre en terme de performance de classification, puisqu'elle utilise les performances prédictives de l'algorithme de classification comme critère d'évaluation de la pertinence du sous-ensemble de variables.

Les méthodes de type embedded

Les méthodes de type embedded incorporent la sélection de variables lors du processus d'apprentissage, sans étape de validation, pour maximiser la qualité de l'ajustement et minimiser le nombre de variables. Fan and Li [31, 32] ont démontré qu'on pourrait garantir une meilleur performance de l'algorithme d'apprentissage en faisant la sélection

des variables et l'estimation du modèle simultanément. Ces méthodes consistent généralement à pénaliser la fonction objective du problème d'optimisation de l'algorithme d'apprentissage. On cite par exemple les méthodes de minimisation avec une pénalité de type L_1 qui ont attiré beaucoup d'attention. Tibshirani [89] a proposé le lasso (least absolute shrinkage and selection operator), une méthode de moindres carrés pénalisée par la norme L_1, pour la sélection des variables dans les modèles linéaires et les modèles linéaires généralisés. Les SVM pénalisés par la norme L_1 (1-norm SVM) ont été considérés pour accomplir la tache de sélection des variables dans SVM [13, 103] .

En utilisant la méthode d'apprentissage comme une boîte noire, les méthodes de type wrapper sont remarquablement universelles et simples. Mais les méthodes de type embedded peuvent être plus efficaces à plusieurs égards. Tout d'abord ils feraient un meilleur usage des données disponibles en n'ayant pas besoin de partager les données d'apprentissage en un échantillon d'apprentissage et un échantillon de validation. Ensuite ces méthodes seraient bien plus avantageuses en terme de temps de calcul que les méthodes de type wrapper car elles évitent le réapprentissage du prédicteur pour tout sous-ensemble de variables sélectionné. Finalement, elles seraient robustes face au problème de surajustement.

3.5 Sélection de variables et 1-SVM

Dans cette section, nous considérons le problème de la sélection de variables dans le cadre de la détection d'anomalies basée sur l'algorithme de classification semi-supervisée 1-SVM. A notre connaissance, et contrairement à l'algorithme de classification supervisée SVM, il n'existe pas des méthodes de sélection de variables pouvant améliorer la performance de détection pour l'algorithme 1-SVM. Dans les travaux existants, la méthode 1-SVM a été appliquée sans sélection de variables sur des données dont l'espace de variables est relativement petit (quelques dizaines de variables) [21, 41, 95, 101], où avec une compression de variables en utilisant une ACP [61].

Dans la littérature, on peut trouver plusieurs méthodes de sélection de variables qui ont été développées pour améliorer la performance prédictive de l'algorithme de classification supervisée SVM. On peut citer par exemple la méthode de filtrage basée sur le score de Fisher [19], la méthode wrapper d'élimination récursive des variables SVM-RFE , ou aussi la méthode embedded des SVM pénalisés par la norme L_1 (1-norm SVM) [103].

Le score de Fisher et la méthode SVM-RFE sont des méthodes de ranking permettant d'établir une hiérarchie des variables. Ces méthodes consistent à attribuer une "valeur

d'importance" à chaque variable puis les classer par ordre décroissant en fonction de leur importance.

Le score de Fisher d'une certaine variable utilise les moyennes et les variances respectives de cette variable dans chacune des deux classes (classe positive et classe négative). Donc ce score mesure la corrélation entre une variable et les étiquettes à prédire. Le score de Fisher pour un SVM linéaire est calculé de la façon suivante :

$$S_j = \frac{|\bar{x}_{j,+1} - \bar{x}_{j,-1}|}{\sqrt{s^2_{j,+1} + s^2_{j,-1}}}$$

où $\bar{x}_{j,+1}$ (respectivement $\bar{x}_{j,-1}$) est la moyenne empirique de la j^{ieme} variable dans la classe positive (respectivement la classe négative), $s^2_{j,+1}$ et $s^2_{j,-1}$ sont les variances empiriques de cette variable dans chacune des deux classes.

Donc cette méthode de filtrage est spécifique à un problème de classification binaire supervisé, où on nécessite la connaissance des étiquettes des observations appartenant à l'échantillon d'apprentissage. Ce qui n'est pas le cas dans notre travail, où on utilise la méthode de classification semi-supervisée 1-SVM qui apprend le modèle de détection à partir d'un échantillon d'apprentissage disposant seulement des observations normales (une seule classe).

Pour cela, nous avons développé une méthode de filtrage appropriée au classifieur 1-SVM, et plus généralement aux méthodes de classification à une classe. En effet cette méthode ne nécessite pas la connaissance des étiquettes des données d'apprentissage. C'est une méthode de ranking basée sur un score indépendant de l'algorithme 1-SVM, calculé en se basant sur une approche robuste de détection univariée des valeurs aberrantes. Ce score est le pourcentage de valeurs aberrantes dans chaque variable déterminée avec le filtre MAD_e. Ce score est donc indépendant des étiquettes à prédire. Une description détaillée de cette méthode, appelée MAD_e.FS (MAD_e for Feature Selection), est donnée dans la Section 3.5.1.

Une deuxième méthode est une adaptation à l'algorithme 1-SVM de la méthode SVM-RFE [38] initialement introduite avec l'algorithme SVM. C'est une méthode de ranking basée sur un score calculé à partir de l'algorithme 1-SVM. Elle évalue l'importance d'une variable en calculant la variation du vecteur de poids $\|w\|^2$ de l'algorithme 1-SVM après l'élimination de cette variable de l'échantillon d'apprentissage. Cette méthode sera décrite dans la Section 3.5.2. On s'attend à ce que cette méthode soit plus performante que la méthode MAD_e.FS puisque la première intègre les performances prédictives du classifier 1-SVM alors que la deuxième est basée sur un score indépendant du classifieur 1-SVM.

3.5.1 Notre méthode de filtrage MAD$_e$.FS

Dans cette section, nous présentons notre méthode de filtrage que nous avons développée pour sélectionner les variables les plus pertinentes afin d'améliorer la performance de détection de l'algorithme 1-SVM. Le choix d'une méthode de filtrage est justifié par le temps de calcul rapide de ce type de méthodes. Une autre justification convaincante est que les méthodes de filtrage permettent de réduire la dimension de l'espace de variables avant l'apprentissage du modèle, ce qui permet d'éviter le sur-ajustement.

L'idée principale est d'utiliser une méthode de détection univariée pour déterminer le pourcentage de valeurs aberrantes dans chaque variable. Puis les variables avec un pourcentage des valeurs aberrantes supérieur à un certain seuil prédéfini seront de potentielles variables discriminantes, tandis que les variables ayant un pourcentage non significatif des valeurs aberrantes seront considérées comme non pertinentes. Un avantage important de cette méthode est qu'elle ne nécessite pas les étiquettes des observations pour faire la sélection.

Comme approche univariée robuste de détection des valeurs aberrantes, nous avons utilisé la méthode MAD_e [16] basée sur des estimateurs robustes de la position et la dispersion . Pour mieux comprendre le mot robuste, nous introduisons la notion *point de rupture* (breakdown point)[27] d'un estimateur qui correspond à la proportion maximale admissible de valeurs abérrantes dans les données avant que cette estimation soit complètement corrompue. Par exemple, l'estimateur classique de la moyenne et de l'écart-type possèdent un point de rupture de 0%, ce qui signifie que la présence d'une seule valeur abérrante est suffisante pour que leur estimation soit complètement erronée. Alors que la médiane et l'écart absolu à la médiane (Median Absolute Deviation MAD) [39] ont un point de rupture égale à 50% [45].

Donc la robustesse de la méthode MAD_e provient du fait qu'elle utilise la médiane et l'écart absolu à la médiane comme estimateurs respectifs de la position et la dispersion. En effet ces deux estimateurs ont un point de rupture élevé, et contrairement à la moyenne et l'ecart-type, les écarts d'un petit nombre (pouvant aller jusqu'à la moitié de la taille de l'échantillon) de valeurs aberrantes ne sont pas pertinents dans le calcul de ces deux estimateurs.

3.5.1.1 Ecart absolu à la médiane

En statistique, l'écart absolu à la médiane MAD est une statistique robuste qui mesure la dispersion d'un échantillon univarié de données. Elle est plus résistante aux valeurs aberrantes que l'écart-type. Cette statistique est définie étant la médiane des écarts

absolus à la médiane des observations d'une variable $x_{.j}$:

$$MAD(j) = mediane_i\left(|x_{ij} - mediane_j(x_{.j})|\right). \tag{3.9}$$

Dans le but d'utiliser le MAD comme un estimateur consistant $\hat{\sigma}$ pour l'estimation de l'écart-type σ (i.e. $E(\hat{\sigma}) = \sigma$, où $E(\hat{\sigma})$ est l'espérance de $\hat{\sigma}$), on prend

$$\hat{\sigma} = k \times MAD, \tag{3.10}$$

où k est un facteur de normalisation constant qui dépend de la distribution des données.

Pour des données suivant une loi normale, k est pris approximativement égal à 1.4826. Expliquons maintenant le choix de cette valeur. Soit $T = (T_1, \ldots, T_n)^T$ une variable aléatoire suivant une loi normale $N(\mu, \sigma^2)$ d'espérance μ et de variance σ^2, alors la variable aléatoire $Z = \frac{T-\mu}{\sigma}$ suit la loi normale $N(0, 1)$. La médiane de T converge vers μ quand n tend vers l'infini, et ensuite la probabilité $P\left(|T - \mu| \leq MAD\right)$ converge vers $P\left(|T - \mu| \leq mediane(|T - \mu|)\right) = \frac{1}{2}$. D'où :

$$\frac{1}{2} = P\left(|T - \mu| \leq MAD\right) = P\left(|Z| \leq \frac{MAD}{\sigma}\right) \tag{3.11}$$

$$= 2F(\frac{MAD}{\sigma}) - 1 \tag{3.12}$$

où F est la fonction de répartition d'une distribution normale centrée réduite $N(0, 1)$. Soit Q l'inverse de F, i.e. Q est la fonction quantile de la loi $N(0, 1)$. L'égalité établie dans (3.11) implique :

$$F(\frac{MAD}{\sigma}) = \frac{3}{4} \Longrightarrow \frac{MAD}{\sigma} = F^{-1}(\frac{3}{4}) = Q(\frac{3}{4}). \tag{3.13}$$

Finalement, l'Equation (3.10) et l'Equation (3.13) permettent d'établir :

$$k = \frac{\sigma}{MAD} = \frac{1}{Q(\frac{3}{4})} \approx 1.4826 \tag{3.14}$$

L'équation (3.14) montre que MAD converge vers $\sigma/1.4826$ lorsque n tend vers l'infini, ce qui est équivaut à dire que $\hat{\sigma} = 1.4826 \times MAD$ converge vers σ. Autrement dit, l'espérance de $\hat{\sigma}$ pour les grands échantillons distribués suivant une loi normale est approximativement égale à l'écart-type σ de la population. D'où le facteur $k = 1.4826$ calibre $\hat{\sigma}$ dans l'équation (3.10) de sorte qu'il soit un estimateur consistant de σ.
Dans la suite, nous remplaçons $\hat{\sigma}$ par MAD_e pour désigner l'estimateur de σ obtenu en utilisant le MAD. On a donc :

$$MAD_e = 1.4826 \times MAD.$$

3.5.1.2 Description de la méthode MAD$_e$.FS

La méthode MAD$_e$ définit pour un paramètre électrique $x_{.j}$ une limite inférieure LL_j (Lower Limit) et une limite supérieure UL_j (Upper Limit) telles que toutes les valeurs situées à l'extérieur de l'intervalle $[LL_j; UL_j]$ sont considérées comme aberrantes. Les limites sont calculées de la façon suivante :

$$LL_j = mediane(x_{.j}) - 3 \times MAD_e(j)$$
$$UL_j = mediane(x_{.j}) + 3 \times MAD_e(j)$$

où $MAD_e(j) = 1.483 \times MAD(j)$.

Cette approche est similaire à la méthode SD (Standard Deviation) qui considère les valeurs situées à l'extérieur de l'intervalle $[\bar{x} \pm 3\,SD]$ comme aberrantes, où $\bar{x}$ et SD sont respectivement la moyenne et l'ecart-type empiriques de l'échantillon univarié (le paramètre $x_{.j}$). Cependant, la moyenne et l'écart-type sont remplacés respectivement par la médiane et le MAD_e. Puisque cette approche utilise deux estimateurs robustes ayant un point de rupture élevé, elle ne sera pas affectée par les valeurs aberrantes contrairement à la méthode SD.

Le pourcentage de valeurs aberrantes OOL_j (Out Of Limit) du paramètre électrique $x_{.j}$ est donné par :

$$OOL(j) = 100 \times \frac{card\left(\{i;\, x_{ij} \notin [LL_j; UL_j]\}\right)}{n}\%,$$

où card() représente le cardinal d'un ensemble.

Avec notre méthode MAD$_e$.FS, le sous-ensemble S des paramètres électriques pertinents sélectionnés est constitué des paramètres électriques dont le pourcentage de valeurs aberrantes dépasse un seuil θ_q, où θ_q est le quantile du vecteur $OOL = (OOL(1), \ldots, OOL(p))$ d'ordre q,

$$S = \{x_{.j};\, OOL(j) > \theta_q\}.$$

Par exemple si nous posons $q = 0.75$ (i.e. θ_q est le troixième quartile), nous décidons alors de retenir le quart des paramètres électriques qui ont le plus grand pourcentage de valeurs aberrantes. D'une façon plus générale, et pour une valeur $q \in]0, 1[$, les $100(1-q)\%$ des paramètres les plus aberrants sont considérés commme pertinents et discriminatifs pour notre méthode de filtrage.

3.5.2 Elimination récursive des variables avec 1-SVM (1-SVM-RFE)

Dans cette section, nous présentons une méthode de sélection de variables de type wrapper intégrant les pouvoirs prédictifs du classifieur 1-SVM. C'est une méthode de ranking exploitant le principe d'élimination récursive des variables et basée sur un score calculé à partir de la méthode 1-SVM. Cette méthode, appelée 1-SVM-RFE est une adaptation de la méthode SVM-RFE à l'algorithme 1-SVM.

Nous rappelons tout d'abord la méthode SVM-RFE puis nous présentons notre algorithme 1-SVM-RFE.

3.5.2.1 Elimination récursive des variables avec SVM (SVM-RFE)

L'algorithme RFE-SVM a été proposé par Guyon et al. [38] pour sélectionner des gènes qui sont pertinents pour un problème de classification du cancer. La méthode est basée sur une sélection séquentielle descendante (backward sequential selection) et exploitant les SVM, de façon récursive pour sélectionner un sous-ensemble de variables optimal. Ce sous-ensemble optimal est constitué par les r parmi p variables ($r < p$) qui maximisent la performance du prédicteur. Cet algorithme intègre le filtrage dans le processus d'apprentissage SVM dans le but d'évaluer chaque sous-ensemble grâce à un classifieur SVM mais aussi pour avoir des informations sur la contribution de chaque variable sur la construction de l'hyperplan séparateur.

On part de l'ensemble complet des variables, on élimine progressivement la variable la moins pertinente jusqu'à avoir laissé r variables. La variable éliminée est celle dont son élimination minimise la variation de $\|w\|^2$. Ainsi le critère de classement (ranking criterion) R_c pour une variable j est :

$$\left| \|w\|^2 - \left\|w^{(j)}\right\|^2 \right| = \frac{1}{2} \left| \sum_{i,k} \alpha_i^* \alpha_k^* y_i y_k K(x_i, x_k) - \sum_{i,k} \alpha_i^{*(j)} \alpha_k^{*(j)} y_i y_k K^{(j)}(x_i, x_k) \right| \quad (3.15)$$

où $K^{(j)}(x_i, x_k)$ est l'élément correspondant à la $i$ème ligne, $k$ième colonne de la matrice de Gram G (cf. Section 2.3.4) des données d'apprentissage obtenue après l'élimination de la variable j

$$K^{(j)}(x_i, x_k) = G_{ik}^{(j)} = \langle \phi(x_i^j), \phi(x_j^j) \rangle$$

et $\alpha_k^{*(j)}$ est la solution de la duale de l'algorithme SVM. Pour une raison de simplicité et afin de réduire la complexité de calcul, les auteurs supposent que $\alpha_k^{*(j)}$ est égal à α_k^* même si une variable a été éliminée. Les auteurs ont également indiqué qu'une normalisation des données est nécessaire pour le bon fonctionnement de RFE.

De l'Equation (3.15), on peut considérer que la variable éliminée est celle qui a le moins d'influence sur la norme du vecteur de poids. En effet, la mesure $\|w\|^2$ est une mesure de pouvoir prédictif. L'idée est que les variables, qui correspondent à des directions de l'espace selon lesquelles le vecteur w admet une faible énergie, ne sont pas aussi utiles au problème que les autres variables (puisqu'ils contribuent faiblement à la définition de l'hyperplan optimal). Donc, à chaque récursion de l'algorithme SVM-RFE, la variable possédant le score R_c le plus faible est éliminée. Le processus est arrêté lorsque le critère d'arrêt est atteint. Ce critère correspond au nombre r de variables à retenir. Autrement dit, le processus est arrêté quand le sous-ensemble sélectionné récursivement par SVM-RFE ne contient que r variables à classer. Le choix optimal de r sera discuté à la fin de cette section.

L'algorithme SVM-RFE de base est donné dans l'Algorithme 1.

Algorithm 1 SVM-RFE

1. Initialisation :
$Var = [1, \ldots, p]$
r : nombre de variables fixé
2. Tant que $card(Var) > r$ **Faire**
(a) Apprentissage du SVM sur toutes les données d'apprentissage et les variables Var

(b) **Pour** toutes les variables dans Var, **Faire** évaluer le critère $R_c(j)$ de la variable j
Fin Pour
(c) Déterminer la variable minimisant le critère R_c : $f = \underset{j \in Var}{min}\, R_c(j)$
(d) Eliminer la variable f de l'ensemble de variables Var : $Var = \{Var\} \backslash f$
Fin Tant que
3. S=Var
4. Retourner S : l'ensemble solution

Avec la version originale de RFE où une variable est éliminée par itération, le temps d'exécution de l'algorithme SVM-RFE est extrêmement coûteux. Pour surmonter cet inconvénient, les auteurs de [38], et en raison du grand nombre de gènes, proposent d'éliminer plusieurs variables simultanément en une itération ; il s'agit dans ce cas de ceux ayant le R_c le plus faible. Pour accélérer le temps de calcul de la procédure de sélection de variables, Rakotomamonjy [71] propose d'éliminer la moitié des variables pour les itérations initiales où le nombre de variables utilisées est grand, jusqu'à ce que 100 variables restent à être classées. Ensuite une variable est retirée à chaque itération.

Notons qu'éliminer plusieurs variables à la fois peut affecter la performance de classification. Dans ce cas-là, on n'obtient pas un critère de rang sur des variables, mais un critère de rang sur des sous-ensembles de variables qui sont imbriqués les uns dans les autres. Si les variables sont éliminées une à une comme le propose l'algorithme initial, les auteurs dans [38] mettent en garde sur la pertinence des variables du plus haut rang : seul le

sous-ensemble de variables sélectionné est optimal, et pas les variables de plus haut rang considérées individuellement. En effet, RFE est une méthode de type "wrapper" qui va avoir tendance à sélectionner des variables contenant de l'information complémentaire, améliorant ainsi la tâche de classification. Les variables considérées une à une dans la sélection ne contiennent que peu d'information pertinente.

Un autre inconvénient de l'algorithme SVM-RFE est que la variable ou le sous-ensemble de variables éliminé ne peut plus jamais revenir dans le sous-ensemble sélectionné, ce qui pourrait biaiser la recherche.

Il est important de noter que RFE ne s'intéresse pas à la recherche du sous-ensemble de taille optimale, mais donne une mesure d'importance sur chaque variable ou groupe de variables. Rakotomamonjy [71] a proposé plusieurs solutions possibles afin de répondre à la question suivante : *combien de variables sélectionnées par l'algorithme SVM-RFE doivent être fournies au prédicteur ?*

Ces solutions ont été testées dans [71] sur un jeu de données non-linéaire. La méthode basée sur l'erreur de validation était la meilleure. Cette méthode consiste à utiliser une procédure de leave-one-out ou un ensemble de validation pour estimer l'erreur de généralisation en fonction du nombre de variables sélectionnées et de choisir le nombre de variables qui minimise l'erreur de test. Cette solution est simple, cependant elle est coûteuse en terme de calcul.

3.5.2.2 L'algorithme 1-SVM-RFE

Similairement à la méthode SVM-RFE, la méthode 1-SVM-RFE est basée sur une sélection séquentielle descendante (backward sequential selection) et exploitant 1-SVM de façon récursive pour sélectionner un sous-ensemble de variables optimal maximisant la performance du prédicteur.

La mesure du pouvoir prédictif est toujours $\|w\|^2$. A chaque itération, la variable éliminée est celle dont son élimination minimise la variation de $\|w\|^2$ de la méthode 1-SVM. Ainsi le critère de classement (ranking criterion) R_c pour une variable j est :

$$\left| \|w\|^2 - \left\|w^{(j)}\right\|^2 \right| = \frac{1}{2} \left| \sum_{i,k} \alpha_i^* \alpha_k^* K(x_i, x_k) - \sum_{i,k} \alpha_i^{*(j)} \alpha_k^{*(j)} K^{(j)}(x_i, x_k) \right| \tag{3.16}$$

où $\alpha_k^{*(j)}$ est la solution de la duale de l'algorithme 1-SVM.

L'algorithme 1-SVM-RFE de base est donné dans l'Algorithme 2.

Algorithm 2 1-SVM-RFE

1. Initialisation :
$Var = [1, \ldots, p]$
r : nombre de variables fixé
2. Tant que $card(Var) > r$ **Faire**
(a) Apprentissage du 1-SVM sur toutes les données d'apprentissage et les variables Var
(b) **Pour** toutes les variables dans Var, **Faire** évaluer le critère $R_c(j)$ de la variable j
Fin Pour
(c) Déterminer la variable minimisant le critère R_c : $f = \underset{j \in Var}{min} R_c(j)$
(d) Eliminer la variable f de l'ensemble de variables Var : $Var = \{Var\} \backslash f$
Fin Tant que
3. S=Var
4. Retourner S : l'ensemble solution

Pour accélérer le temps de calcul, nous proposons d'éliminer la moitié des variables à la première itération, puis 5% des variables aux itérations suivantes jusqu'à ce que la taille du sous-ensemble à sélectionner sera égale à r. Afin de gagner encore en temps de calcul, nous définissons un hyperparamètre q qui permet de déterminer le nombre r de variables à retenir par l'algorithme 1-SVM-RFE. Identiquement à notre méthode de filtrage, et pour une valeur de $q \in]0, 1[$, le nombre de variables à retenir r correspond à la fraction $(1 - q)$ du nombre de variables initiales p. Le choix optimal de q peut être accompli selon la solution proposée par Rakotomamonji [71], déjà mentionnée à la fin de la section précédente. Par exemple, une plage de valeurs de q dans $[0.5; 1[$ et avec un pas de 0.05 nous permet d'éviter de parcourir de nombreuses valeurs de r, et par suite réduire le coût de calcul.

Chapitre 4

Quelques résultats de la détection avec 1-SVM en mode statique

Dans ce chapitre, nous introduisons tout d'abord deux modèles de détection que nous avons développés pendant notre étude. Ils sont basés sur les deux méthodes de sélection de variables que nous venons de présenter dans le Chapitre 3, et sur l'algorithme de détection 1-SVM. Les deux modèles peuvent être appliqués en mode statique ou en mode dynamique selon qu'on prend en compte ou non l'ordre chronologique des wafers.

Nous présentons quelques résultats de la détection avec 1-SVM en mode statique. Nous commençons par étudier l'influence de l'hyperparamètre γ du noyau gaussien sur la performance de 1-SVM, et nous allons voir qu'il n'est pas nécessaire de faire un réglage extrêmement fin pour cet hyperparamètre. Ensuite nous montrons l'importance de l'application de nos méthodes de sélection de variables avec l'algorithme 1-SVM, révélée par des améliorations significatives de la performance de détection de cet algorithme. Puis nous étudions l'influence de la présence d'anomalies dans la phase d'apprentissage du modèle 1-SVM. Ceci va servir dans l'interprétation des résultats de comparaison de deux scénarios de mise à jour de notre système de détection en temps réel (cf. Section 5.4 du Chapitre 5).

Avant de présenter ces résultats, nous introduisons les mesures de performances permettant d'évaluer les capacités de détection des modèles utilisés, ainsi que les différentes étapes de prétraitement de données, et nous décrivons les jeux de données réels de ST qui vont servir pour évaluer les différents modèles de détection.

4.1 Modèles de détection

Le but de notre travail était de développer une approche statistique multivariée pour détecter les anomalies dans le domaine des semi-conducteurs. Ces anomalies représentent des wafers dont le comportement électrique est anormal. Cette approche multivariée vise donc à détecter les wafers anormaux en regardant les mesures des paramètres électriques statiques obtenues à l'issu du Test Paramétrique (PT).

Dans les chapitres précédents, nous avons présenté la méthode 1-SVM pour la détection d'anomalies. Nous avons présenté également deux techniques de réduction de dimension : la méthode de type filtrage MAD_e.FS, et la méthode de type wrapper 1-SVM-RFE.

Ainsi nous introduisons deux modèles de détection :

1. 1-SVM.MAD_e : ce modèle utilise tout d'abord notre méthode MAD_e.FS pour sélectionner les paramètres électriques pertinents, puis l'algorithme 1-SVM est appliqué sur le sous-ensemble de paramètres sélectionnés pour détecter les wafers anormaux.
2. 1-SVM.RFE : dans ce modèle, la sélection de variables est effectuée en utilisant la méthode de type wrapper 1-SVM-RFE, puis un classifieur 1-SVM est construit à partir des paramètres retenus.

Ces deux modèles ont été testés suivant deux modes de fonctionnement : un mode statique et un mode dynamique.

- Mode statique : les modèles sont figés. Ils ne prennent pas en considération l'ordre chronologique des wafers et l'évolution temporelle des mesures de paramètres électriques.
- Mode dynamique : Le modèle dynamique exploite les modèles de détection développés tout en considérant le facteur "temps". Le but étant de réaliser une détection en temps réel dans le contexte d'une application industrielle. L'idée est d'appliquer les modèles de détection dans un cadre temporel où l'ordre chronologique des wafers est respecté. En mode dynamique, les wafers mavericks sont détectés à partir d'un historique récent des mesures électriques à l'aide d'une fenêtre temporelle glissante. Pour ce faire, deux conditions sont nécessaires :
  1. Les mesures doivent etre triées par ordre chronologique.
  2. L'historique disponible doit être suffisament long pour pouvoir construire le modèle sur un échantillon d'apprentissage représentatif contenant une grande variété des mesures électriques.

		Modèle	
		anomalie	normal
PT	anomalie	TN	FN
	normal	FP	TP

TABLE 4.1 – Résultats possibles lors d'une application d'un modèle de détection.

4.2 Evaluation de Performance et jeux de données

Puisque l'hypothèse construite par un algorithme d'apprentissage peut être comprise comme une approximation de la fonction cible, la qualité de cette approximation, appelée performance, est d'un grand intérêt. Il s'agit d'une mesure de la capacité de généralisation des hypothèses.

D'une part, l'évaluation de performance est importante pour faire le réglage des paramètres de l'algorithme d'apprentissage ou pour trouver un sous-ensemble optimal des variables dans le cadre d'une sélection de variables. D'autre part, elle est utilisée pour comparer et évaluer les différents algorithmes d'apprentissage utilisés. De cette façon, l'algorithme optimal pour un problème spécifique peut être étudié.

4.2.1 Mesures de Performance

Le terme "mesure de performance" spécifie la manière dont la performance de l'algorithme est déterminée. Le tableau 4.1 montre les différents résultats possibles lors d'une application d'un modèle de détection :

1. TN : le nombre de vrais négatifs(True Negative), i.e. le nombre de wafers considérés comme anormaux au PT et détectés par le modèle.
2. FN : le nombre de faux négatifs (False Negative), i.e. le nombre de wafers considérés comme anormaux au PT et non détectés par le modèle.
3. FP : le nobmre de faux positifs (False Positive), i.e. le nombre de wafers considérés comme normaux au PT et détectés par le modèle.
4. TP : le nombre de vrais positifs (True Positive), i.e. le nombre de wafers considérés comme normaux au PT et non détectés par le modèle.

Par suite nous pouvons calculer la sensibilité et la spécificité. La sensibilité (4.1) représente le taux de vrais négatifs, i.e la précision de détection des observations anormales. La spécificité (4.2) représente le taux de vrais positifs, i.e la précision de détection des observations normales.

$$sensibilité = 100\ \frac{TN}{TN + FN}\ \% \tag{4.1}$$

$$spécificité = 100\ \frac{TP}{TP + FP}\ \% \tag{4.2}$$

En se basant sur ces deux mesures, nous définissons deux mesures de performance : le taux de détection (Detection Rate DR) et le taux de fausses alarmes (False Alarms Rate FAR). Le taux de détection (sensibilité) est le pourcentage de wafers anormaux détectés par notre modèle tandis que le taux de fausses alarmes (100-spécificité) est le pourcentage des wafers normaux considérés comme des anomalies par notre modèle.

Une mesure de performance largement utilisée dans la classification est la précision (accuracy), qui représente la fraction d'observations correctement classées dans l'ensemble de test. Nos jeux de données sont constitués presque entièrement d'observations normales alors que les anomalies sont rares. Alors les probabilités a priori des classes sont très différentes, d'où cette mesure n'est pas adaptée à notre étude. Par exemple, sur un ensemble de données avec une distribution de $98\% - 2\%$ des classes, il est facile d'atteindre 98% de précision en attribuant simplement chaque nouveau cas à la classe cible (wafers normaux). Une telle mesure est inappropriée pour la détection des wafers anormaux, comme le classificateur aurait échoué de reconnaître un wafer anormal.

Pour surmonter ce problème, nous avons choisi la métrique G-means (4.3) présentée dans [52].

$$G - means = \sqrt{sensitivité \times spécificité} = \sqrt{DR \times (100 - FAR)} \tag{4.3}$$

L'idée de base de G-means est de maximiser les valeurs de la sensibilité et de la spécificité et de garder le taux de détection et le taux de fausses alarmes en équilibre. Des valeurs élevées de cette mesure indique une bonne qualité du modèle de classification.

4.2.2 Courbes FAR-DR

Une façon plus avancée et fréquemment utilisée pour l'évaluation de la performance des modèles d'apprentissage automatique est fournie par ce qu'on appelle la courbe FAR-DR. Cette courbe fournit un moyen approprié pour l'évaluation de classificateurs en intégrant leur performance sur une gamme de seuils de décision. Par exemple, cette gamme peut être de différentes valeurs d'un paramètre libre du modèle d'apprentissage comme le paramètre ν dans les SVM à une classe. La courbe FAR-DR montre donc la relation ou le compromis entre le taux de détection et le taux de fausses alarmes en faisant varier une condition ou un paramètre libre du modèle utilisé. Pour chaque élément de cette gamme, un point de la courbe FAR-DR est déterminé. Un point parfait dans cette

courbe présenterait un taux de détection de 100% avec un taux de fausses alarmes de 0%. Par conséquent, les chercheurs essaient de pousser la courbe vers ce point idéal, i.e vers la partie haute gauche.

La courbe FAR-DR est inspirée de la courbe ROC. Cette dernière montre aussi le compromis entre le taux de détection et le taux de fausses alarmes en faisant varier un paramètre libre du modèle utilisé. La seule différence entre les deux courbes est la gamme de valeurs de ce paramètre exploitée par ces courbes. Avec la courbe ROC, on fait varier le paramètre ν du 1-SVM pour différentes valeurs appartenant à $]0,\ 1[$, alors qu'avec la courbe FAR-DR on peut se contenter de varier ce paramètre sur un sous-ensemble de $]0,\ 1[$. Dans notre étude, les courbes FAR-DR sont tracées en faisant varier ν sur $[0.01,\ 0.2]$ avec un pas de 0.01.

4.2.3 Prétraitement des données

Les données collectées sur Crolles 300 (mesures des paramètres, limites de validité) sont extraites à partir de la base de données avec le logiciel PT_ANALYSIS_II. Cet outil a été développé à STMicroelectronics, sur le site de Crolles, pour visualiser et analyser toutes les données du PT (boites à moustaches, histogrammes, indicateurs statistiques, mapping des wafers ...).

Les paramètres PT

Les paramètres collectés sont les paramètres de type "Gated". Ce sont tous les paramètres électriques définis dans le programme du test paramétrique pour l'acceptation des lots et des wafers. Ces paramètres comprennent les paramètres "clés", un sous-ensemble de paramètres de fiabilité et de qualité considérés comme majeurs pour la technologie et doivent être garantis aux clients.

Les paramètres Gated sont classés en deux catégories :

- Les paramètres de fiabilité : paramètres potentiellement liés à des problèmes de fiabilité, en raison du processus de fabrication, et qui peuvent avoir un impact sur la durée de vie du circuit. Nous citons par exemple les paramètres "VBD_AREA_PN_1" et "VBD_AREA_PN_2" qui mesurent la tension de claquage de deux structures de test différentes.
- Les paramètres de qualité : paramètres qui caractérisent le rendement de fabrication du produit. Nous citons par exemple le paramètre "NIOFF03028TG14" qui mesure le courant résiduel du transistor fermé, le paramètre clé "NVT02044RST14" qui mesure la tension seuil d'un transistor caractérisé par une longuer et une largeur bien spécifique,

et le paramètre “NIDS0205RD14” mesurant le courant de saturation d'un transistor caractérisé par une longuer et une largeur bien spécifique.

Comme nous avons déjà vu dans la Figure 7, ces paramètres sont mesurés sur tous les lots et tous les wafers en neuf sites (emplacements) de chaque wafer.

Transposition des données

Les fichiers contenant les données sont sous forme d'une matrice : chaque colonne représente un indicateur du wafer (identifiant du lot, numéro du wafer dans le lot, nom du produit, position (location), date de prise des mesures, ...) ou un paramètre PT, et chaque ligne, une observation (un site d'un wafer d'un lot) (Tableau 4.2).

Au début, nous avons résumé les neuf sites de mesures de chaque paramètre électrique par leur moyenne et leur écart-type. Ceci nous amené à une perte d'information importante et a réduit la pertinence de la détection. Pour cela, un code a été réalisé dans le logiciel pour garder toutes les valeurs correspondantes aux neuf sites de mesure, mais dans un format exploitable. Ce code transpose les “raw data” d'un wafer (9 lignes correspondant aux 9 sites du wafer) en colonnes, donc chaque wafer d'un lot sera représenté par une ligne et décrit par 9 mesures de chaque paramètre en colonnes (Tableau 4.3). Donc la dimension de l'espace de variables est égale à neuf fois le nombre de paramètres électriques mesurés, puisque chaque site de mesure d'un paramètre va à son tour constituer un nouveau paramètre.

Le Tableau 4.2 montre un exemple de fichier de données extrait de la base de données de ST, où chaque wafer est décrit par neuf lignes, chaque ligne correspond aux mesures d'un site de mesures de l'ensemble de paramètres électriques. Avec le code de transposition développé, nous obtenons dans le Tableau 4.3 le format exploitable en analyse de données, où chaque wafer est décrit par une seule ligne, et le facteur site figure en colonnes et pas en ligne (BVNWPW11_1,..., BVNWPW11_9).

Traitement des mesures hors limites de validité

Outre les limites de spécifications, les ingénieurs PT définissent aussi des limites de validité pour les paramètres électriques, qui sont bien éloignées des limites de spécification. Une mesure hors validité peut être dûe à un défaut de fabrication ou un problème de mesure. Les mesures hors validité qui s'écartent trop des autres mesures pourraient affecter le bon fonctionnement des méthodes statistiques utilisées. Par exemple, on peut trouver une mesure égale à 10^{21} pour un paramètre électrique causée par un problème

lot	lotwaf	site	...	BVNWPW11	...	W_MAX
Q102YAK	Q102YAK_1	1	STD	8.806	...	1.08
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Q102YAK	Q102YAK_1	9	STD	8.784	...	1.35
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Q102YAK	Q102YAK_25	1	STD	8.885	...	1.17
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Q102YAK	Q102YAK_25	9	STD	8.915	...	0.96
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Q131MWH	131MWH_25	1	STD	9.768	...	0.82
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Q131MWH	131MWH_25	9	STD	9.879	...	0.75

TABLE 4.2 – Exemple de fichier de données extrait à partir de la base de données

lot	lotwafer	...	BVNWPW11_1	...	BVNWPW11_9	...	W_MAX_9
Q102YAK	Q102YAK_1	...	8.806	...	8.784	...	1.35
Q102YAK	Q102YAK_25	...	8.885	...	8.915	...	0.96
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Q131MWH	Q131MWH_25	...	9.768	...	9.879	...	0.75

TABLE 4.3 – Transposition en colonnes des “Raw data” du Tableau 4.2

de mesure. Par conséquent, il est nécessaire de filtrer ce type de mesures non représentatives du procédé de fabrication. Pour cela, nous classifions suite aux avis des ingénieurs PT les paramètres électriques en deux catégories définissant si les mesures hors validité ont une signification physique (un défaut de fabrication) ou non (un problème de mesure). Nous construisons alors un fichier contenant les paramètres électriques et leur catégorie constituée de deux modalités :

– Categorie C_1 : une meure hors validité a une signification physique.
– Categorie C_2 : une meure hors validité n'a pas une signification physique.

Le logiciel PT_ANALYSIS_II permet avec l'option “Exclude Out of Validity” (exclure les mesures hors validité) de remplacer automatiquement dans les fichiers extraits les mesures hors validité par des données manquantes.

Ensuite, pour traiter les données manquantes qui correspondent à des valeurs hors validité, un programme a été créé avec le logiciel R. Ce programme prend en entrée un fichier de données contenant les mesures des paramètres et un autre contenant les catégories de

ces paramètres. Le programme parcourt tout le fichier de données et dès qu'une valeur manquante est détectée, il procède à une de deux actions suivantes :

- si la donnée manquante correspond à un paramètre de "Catégorie C_1", nous la remplaçons par la limite de validité la plus proche (supérieure ou inférieure) du paramètre électrique correspondant.
- si la donnée manquante correspond à un paramètre de "Catégorie C_2", nous la remplaçons par la moyenne des mesures valides du même paramètre électrique et même site de mesure des wafers du même lot.

4.2.4 Description des jeux de données

Nous avons décidé de travailler sur la technologie "IMAGER 140" (I140). Elle fait partie des technologies les plus avancées à Crolles. De plus, elle présente un volume de production important ce qui est un avantage pour la collecte des données et la pertinence des analyses.

Cette technologie consiste à fabriquer des capteurs d'images. Ces capteurs sont des dispositifs qui permettent de transcrire une scène observée en un signal électronique à travers une optique de focalisation. Autrement dit, ils permettent de transformer une image réelle en signal numérique. Nous en utilisons au quotidien à travers des photocopieurs, appareils photo, caméras numériques, lecteurs code barres et téléphones portables.

Les différentes techniques de détection ont été testées sur deux jeux de données réelles de ST : dataset 1 et dataset 2. Ces deux jeux de données sont décrits dans le Tableau 4.4. Chaque jeu de données consiste en un produit de la technologie I140 avec plusieurs mois de production.

	dataset 1	dataset 2
Produit	P_1	P_2
Période de production	4 mois	2 mois
Nb de wafers	**2550**	**1600**
Nb de wafers scrapés au PT	**12**	**23**
% de wafers scrapés au PT	0.47	1.44
Nb de paramètres électriques	118	84
Dimension de l'espace de variables	**1062**	**756**

TABLE 4.4 – Description des jeux de donnéees réelles de ST utilisées dans notre étude.

Le dataset 1 est composé de 2550 wafers décrits chacun par 118 paramètres électriques mesurés en neuf sites ce qui fait un espace de variables de dimension égale à 1062 (9×118).

Il contient 12 wafers considérés comme anormaux au PT. Le dataset 2 est composé de 1600 wafers dont 23 sont considérés comme anormaux au PT. Chaque wafer est décrit par 84 paramètres électriques mesurés également en neuf sites, d'où la dimension de l'espace de variables est égale à 756 (9×84). Les wafers anormaux du dataset 1 ont des profils très atypiques et sont donc faciles à détecter par les trois modèles de détection que nous allons utilisés contrairement aux wafers anormaux du dataset 2. Des taux de détection très élevés peuvent être obtenus avec les trois modèles utilisés et sur les deux jeux de données, mais avec des taux de fausses alarmes dans le dataset 1 beaucoup plus petits que dans le dataset 2. Alors nous pouvons dire que le dataset 1 est un jeu de données idéal où tous les modèles sont capables de réaliser une détection performante, tandis que le dataset 2 représente un défi pour les modèles de détection et va permettre de choisir le modèle le plus performant.

4.3 Influence de l'hyperparamètre γ sur la performance de 1-SVM

Dans la Section 2.4.2, et pour les raisons citées dans la Section 2.3.5, nous avons vu que le noyau gaussien représente le meilleur choix en tant qu'une fonction noyau pour les algorithmes SVM et 1-SVM. Pour cela nous avons commencé par étudier l'influence de l'hyperparamètre γ du noyau gaussien sur la performance de l'algorithme 1-SVM appliqué sans sélection de variables.

Selon Vapnik [92], le choix de la valeur de γ dépend de la dimension de l'espace de variables auquel les SVM sont appliqués, et celui-ci préconise de choisir $\gamma = 1/p$.

Toutefois, comme on peut le constater sur la Figure 4.3 cette valeur n'est pas toujours optimale (ici la valeur préconisée par Vapnik correspond à l'abscisse 1, pour laquelle la performance de 1-SVM n'est pas optimale). C'est pourquoi nous avons utilisé un parcours d'un ensemble de valeurs de γ discrètes réelles définies par

$$\gamma = \frac{1}{m \times p}$$

avec $m \in \{\frac{1}{3},\ \frac{1}{2},\ 1,\ 1.5,\ 2,\ 2.5,\ 3,\ 3.5,\ 4,\ 4.5, 5\}$.

Les bornes supérieures et inférieures de cet intervalle ont été définies de façon empirique afin de couvrir une plage de valeur suffisamment large autour de la valeur recommandée $\gamma = 1/p$.

L'influence de l'hyperparamètre γ a été étudiée sur les deux jeux de données en mode statique. Nous avons partitionné le jeu de données en un ensemble d'apprentissage et

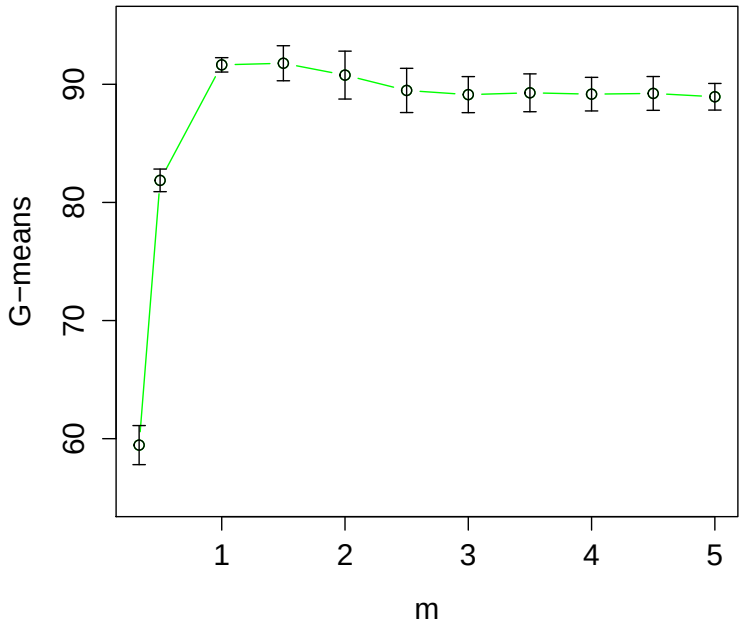

FIGURE 4.1 – Dataset 1 : variation de la mesure de performance G-means de l'algorithme 1-SVM en fonction de différentes valeurs de m, pour $\nu = 0.05$.

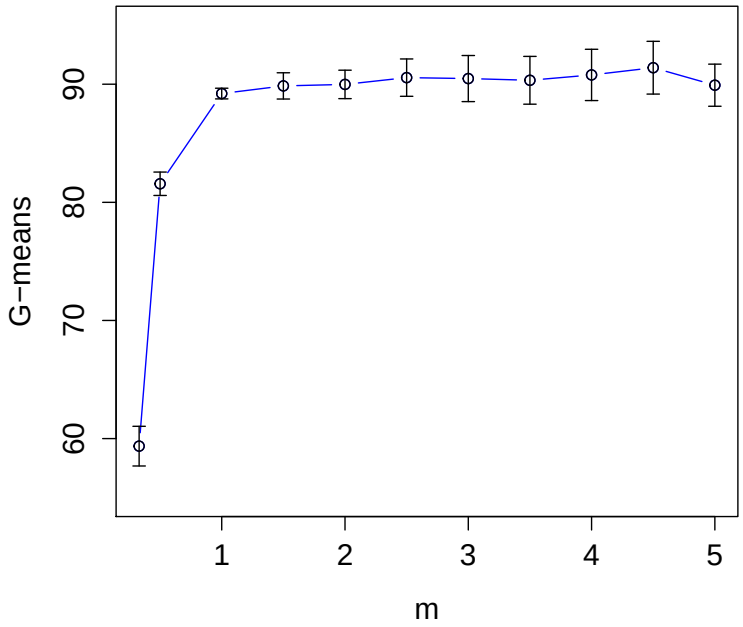

FIGURE 4.2 – Dataset 1 : variation de la mesure de performance G-means de l'algorithme 1-SVM en fonction de différentes valeurs de m, pour $\nu = 0.1$.

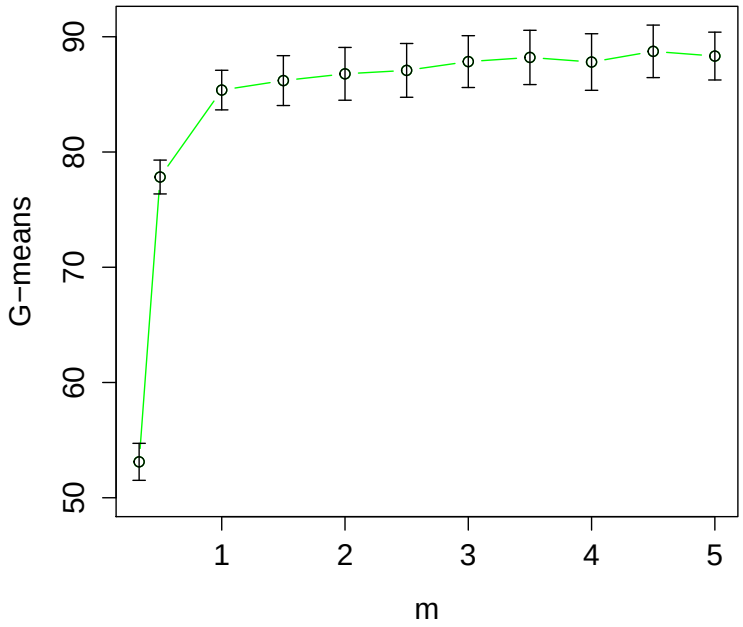

FIGURE 4.3 – Dataset 2 : variation de la mesure de performance G-means de l'algorithme 1-SVM en fonction de différentes valeurs de m, pour $\nu = 0.1$.

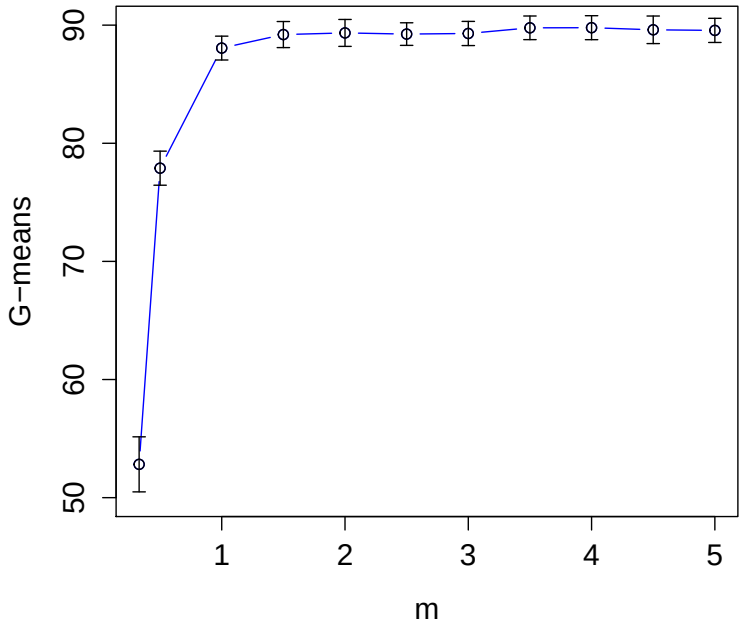

FIGURE 4.4 – Dataset 2 : variation de la mesure de performance G-means de l'algorithme 1-SVM en fonction de différentes valeurs de m, pour $\nu = 0.15$.

un ensemble test avec des proportions respectives de 2/3 et 1/3. Ensuite nous avons appris le modèle 1-SVM sur l'échantillon d'apprentissage ne contenant que des wafers considérés comme normaux au PT, et nous avons déterminé les taux de détection et de fausses alarmes obtenus suite à l'application du modèle sur l'échantillon de test. Ce scénario est répété 100 fois (100 partitionnements aléatoires du jeu de données suivant les proportions 2/3 ;1/3). Pour chaque valeur de m, la moyenne et l'écart-type de la mesure de performance G-means sont calculés sur les 100 répétitions aléatoires.

Les Figures 4.1 et 4.2 (resp. Figures 4.3 et 4.4) montrent les variations de la mesure de performance G-means en fonction de différentes valeurs de m dans le dataset 1 (resp. dataset 2), pour $\nu = 0.05$ et $\nu = 0.1$ (resp. $\nu = 0.1$ et $\nu = 0.15$) . Des barres d'erreurs ont été ajoutées pour représenter les écarts-type de G-means. Comme on peut le constater, les quatre courbes présentent une assez large plage ($m \geq 1$) pour laquelle la mesure G-means est élevée et peu variable, avec aussi une légère variabilité observée au niveau de chaque valeur de m (des petites barres d'erreur). L'algorithme 1-SVM montre alors une performance quasiment stable, et Le choix de m dans cette plage n'est donc pas critique.

Ces résultats montrent alors que 1-SVM n'est pas sensible au hyperparamètre γ du noyau gaussien. Donc il n'est pas nécessaire de faire un réglage précis pour cet hyperparamètre. Pour la suite, nous avons décidé de retenir la valeur $\gamma = 1/3p$ ($m = 3$). Toutefois, nous pouvons, en présence d'un échantillon de validation, choisir une valeur optimale de γ dans la plage de valeurs proposée dans cette section en se basant sur l'erreur de validation.

4.4 Importance de la sélection de variables dans 1-SVM

Dans cette section, nous montrons l'importance de l'utilisation de deux méthodes de sélection de variables dans l'amélioration de la performance de détection de l'algorithme 1-SVM. Nous avons considéré les deux jeux de données et nous avons testé les modèles 1-SVM, 1-SVM.MAD$_e$ et 1-SVM.RFE en mode statique selon le même scénario d'apprentissage et de test de la section précédente. L'hyperparamètre γ est posé égal à $1/3p$. Dans les modèles 1-SVM.MAD$_e$ et 1-SVM.RFE, p représente le nombre de paramètres électriques retenus après la sélection de variables. Nous avons considéré 20 valeurs de ν telles que $\nu \in \{0.01,\ 0.02, \ldots,\ 0.2\}$. L'hyperparamètre q des méthodes de sélection de variables est pris égal à 0.75 (i.e. nous avons retenu le quart des paramètres électriques les plus pertinents avec chaque méthode).

Nous donnons dans les Tableaux 4.5 et 4.6 des illustrations numériques de l'amélioration de la performance de détection après avoir utilisé chacune des méthodes de sélection de variables respectivement dans le dataset 1 et le dataset 2. Nous avons choisi 6 valeurs

ν	1-SVM		1-SVM.MAD$_e$		1-SVM.RFE	
	DR	FAR	DR	FAR	DR	FAR
0.03	77.50 (5.85)	4.19 (0.71)	80.15 (3.24)	3.65 (1.15)	81.12 (2.05)	3.37 (0.98)
0.04	83.61 (1.52)	5.40 (0.91)	84.35 (2.45)	4.87 (0.82)	84.73 (3.21)	4.41 (0.85)
0.05	84.72 (3.16)	6.13 (0.98)	86.07 (2.15)	5.89 (0.71)	87.11 (2.66)	4.99 (0.71)
0.06	89.44 (3.75)	7.06 (0.96)	90.55 (1.49)	6.77 (1.21)	90.72 (1.26)	6.29 (1.15)
0.07	91.39 (1.52)	8.30 (1.02)	91.66 (0)	8.23 (1.02)	91.66 (0)	8.46 (0.82)
0.08	91.66 (0)	9.11 (1.66)	91.66 (0)	8.71 (1.16)	91.66 (0)	8.55 (1.35)

TABLE 4.5 – Dataset 1 : ilustration numérique de l'amélioration de la performance de détection avec les méthodes de sélection de variables pour $q = 0.75$.

ν	1-SVM		1-SVM.MAD$_e$		1-SVM.RFE	
	DR	FAR	DR	FAR	DR	FAR
0.05	52.52 (6.93)	5.78 (1.28)	79.45 (8.79)	5.28 (1.13)	93.57 (5.29)	5.96 (1.22)
0.06	62.22 (8.44)	6.89 (1.52)	89.30 (5.43)	7.22 (1.33)	95.65 (0)	6.43 (1.19)
0.07	71.22 (7.79)	8.12 (1.57)	93.30 (3.54)	7.84 (1.23)	95.65 (0)	7.78 (1.29)
0.10	88.65 (5.13)	10.96 (1.76)	95.65 (0)	10.63 (1.42)	95.65 (0)	10.55 (1.35)
0.12	94 (2.88)	12.92 (1.75)	95.65 (0)	12.29 (1.64)	95.65 (0)	12.93 (1.93)
0.15	95.65 (0)	15.86 (2.03)	95.65 (0)	15.57 (2.26)	95.65 (0)	15.53 (1.79)

TABLE 4.6 – Dataset 2 : ilustration numérique de l'amélioration de la performance de détection avec les méthodes de sélection de variables pour $q = 0.75$.

de ν représentatives de cette amélioration. Pour chaque valeur de ν, nous donnons la moyenne et l'écart-type (entre paranthèse) de deux mesures de performance DR et FAR obtenus sur les 100 répétitions aléatoires avec chacun de trois modèles de détection. Dans le Tableau 4.5, une amélioration de deux mesures de performance a été observée en appliquant les méthodes de sélection de variables, surtout pour les premières valeurs de ν ($\nu \in \{0.03,\ 0.04,\ 0.05\}$). Les méthodes 1-SVM-RFE et MAD$_e$.FS ont montré des performances similaires, avec des légères augmentations de DR et réductions de FAR apportées par la méthode 1-SVM-RFE par rapport à la méthode MAD$_e$.FS. Cette légère amélioration de performance de détection est dûe à la facilité de détection des wafers anormaux dans ce jeu de données (cf. Section 4.2.4). Il suffit de voir que le modèle 1-SVM fournit une bonne performance de détection pour toutes les valeurs de ν, et ceci sans appliquer une sélection de variables. Nous nous attendons à une amélioration plus significative de la performance de détection dans le dataset 2 après avoir sélectionné les paramètres électriques pertinents. Dans le Tableau 4.6, nous voyons clairement les améliorations significatives dans la performance de détection, apportées par chacune des deux méthodes de sélection de variables. En effet, pour une même valeur de ν, nous observons des FAR similaires et peu variables (petits écarts-type) mais des améliorations importantes et moins variables des DR avec les modèles 1-SVM.MAD$_e$ et 1-SVM.RFE par rappot au modèle 1-SVM. Par exemple, pour $\nu = 0.05$, nous notons en moyenne

ν	1-SVM		1-SVM.MAD$_e$		1-SVM.RFE	
	DR	FAR	DR	FAR	DR	FAR
0.05	52.52 (6.93)	5.78 (1.28)	66.17 (14.23)	5.42 (1.14)	88.43 (4.36)	5.96 (1.25)
0.06	62.22 (8.44)	6.89 (1.52)	72.70 (14.11)	6.55 (1.35)	91.04 (1.36)	6.41 (1.28)
0.07	71.22 (7.79)	8.12 (1.57)	77.30 (13.76)	7.10 (1.23)	91.30 (0)	7.91 (1.41)
0.10	88.65 (5.13)	10.96 (1.76)	88.52 (7.28)	10.19 (1.52)	92.87 (2.11)	10.53 (1.65)
0.12	94 (2.88)	12.92 (1.75)	91.04 (6.24)	11.85 (1.84)	95.57 (0.61)	12.41 (1.71)
0.15	95.65 (0)	15.86 (2.03)	94.35 (3.07)	14.96 (1.94)	95.65 (0)	15.73 (1.64)

TABLE 4.7 – Dataset 2 : ilustration numérique de l'amélioration de la performance de détection avec les méthodes de sélection de variables pour $q = 0.9$.

des augmentations respectives de DR d'environ 27% et 41%, puis d'environ 22% et 24% pour $\nu = 0.07$. Ensuite, le taux de détection a atteint 94% avec le modèle 1-SVM (pour $\nu = 0.12$). Des taux de détection très proches ont été obtenus en appliquant les modèles 1-SVM.MAD$_e$ (93.3%, $\nu = 0.07$) et 1-SVM.RFE (93.57%, $\nu = 0.05$), mais avec des réductions significatives des fausses alarmes respectivement égales à 5 et 7% environ. Pour un taux de détection égal à 95.65% obtenu sans sélection de variables ($\nu = 0.15$), les méthodes de filtrage ($\nu = 0.10$) et de wrapper ($\nu = 0.06$) ont abouti à ce même taux mais avec des réductions respectives de fausses alarmes de 5 et 9% environ. De plus, ces résultats montrent que le modèle 1-SVM.RFE est plus performant que le modèle 1-SVM.MAD$_e$. Ce résultat était attendu vu que le premier modèle utilise une méthode wrapper qui intègre les performances prédictives du classifieur 1-SVM alors que le deuxième utilise une méthode de filtrage basée sur un score indépendant du classifieur 1-SVM.

Le Tableau 4.7 montre aussi l'amélioration de la performance de détection apportée par les méthodes 1-SVM.MAD$_e$ et 1-SVM.RFE par rapport au 1-SVM, avec q pris égal à 0.9. Le modèle 1-SVM.RFE est toujours plus performant que le modèle 1-SVM.MAD$_e$. Cependant, et en comparant pour les mêmes valeurs de ν, ce tableau au Tableau 4.6, nous remarquons une diminution respective des DR et FAR des modèles 1-SVM.MAD$_e$ et 1-SVM.RFE. Avec $q = 0.90$, nous observons une dégradation significative de la performance du modèle 1-SVM.MAD$_e$ par rapport à $q = 0.75$, et ceci pour les 6 valeurs de ν. De plus, pour $\nu \in \{0.10,\ 0.12,\ 0.15\}$, le modèle 1-SVM a donné des DR plus élevés et moins variables que le modèle 1-SVM.MAD$_e$. Ceci peut être dû à l'élimination d'une ou plusieurs variables discriminatives par la méthode de filtage. Concernant le modèle 1-SVM.RFE, une légère dégradation de performance est soulignée pour quelques valeurs de ν ($\nu \in \{0.05,\ 0.06,\ 0.07,\ 0.1\}$). 1-SVM.RFE a donc été moins affecté par l'augmentation de q que 1-SVM.MAD$_e$. Ceci peut s'expliquer par le fait que la méthode de sélection de type wrapper a pu garder les paramètres électriques discriminants dans les 10% paramètres sélectionnés ($q = 0.9$) contrairement à la méthode de type filtrage.

L'augmentation de q de 0.75 à 0.90 a abouti à une baisse de performance des méthodes de sélection de variables.

Pour résumer, les résultats des Tableaux 4.5 et 4.6 ont montré l'importance d'application d'une méthode de sélection de variables pour une détection plus performante avec la méthode 1-SVM. Ces deux jeux de données ont montré aussi la supériorité du modèle 1-SVM.RFE par rapport au modèle 1-SVM.MAD$_e$. D'autre part, la comparaison des Tableaux 4.7 et 4.6 montre que le choix de l'hyperparamètre q est sensible. Il est donc important de faire un choix optimal de cet hyperparamètre afin d'optimiser la performance de détection de la méthode 1-SVM.

4.5 Influence de la présence d'anomalies dans l'étape d'apprentissage de 1-SVM

1-SVM est une méthode d'apprentissage semi-supervisée où le modèle de classification est construit sur un échantillon d'apprentissage constitué des observations normales. Donc l'idée d'avoir des anomalies dans l'échantillon d'apprentissage n'est pas tout à fait réaliste. Cependant, il est important d'étudier ce cas afin de mieux interpréter les résultats de comparaisons de deux scénarios de mise à jour de notre système de détection en temps réel dans le Chapitre 5. De plus, il se peut qu'il existe un ou plusiers wafers dans l'échantillon d'apprentissage considérés comme normaux par les ingénieurs PT et qu'ils soient réellement anormaux.

Ceci nous amène à se poser la question suivante : que se passe-t-il si l'échantillon d'apprentissage contient une très petite fraction d'anomalies ? Rappelons tout d'abord que le paramètre ν de cette méthode est une borne supérieure de la fraction des erreurs d'apprentissage et permet alors de maximiser la marge de séparation en classant quelques observations normales comme anomalies. Intuitivement, nous pouvons nous attendre à ce que ces anomalies, avec une valeur appropriée de ν, vont être situées entre l'hyperplan séparateur et l'origine et par suite leur présence ne va pas affecter fortement la performance prédictive du classifieur 1-SVM. Cependant, et pour une même valeur de ν, la marge du classifieur 1-SVM est plus large en absence d'anomalies dans l'échantillon d'apprentissage et par conséquent la présence d'anomalies dans la phase d'apprentissage peut réduire la capacité de généralisation du classifieur 1-SVM.

Nous étudions donc, dans cette section, l'influence de la présence d'anomalies dans la phase d'apprentissage du modèle 1-SVM sur sa performance de détection. Cette condition est testée sur les deux jeux de données en mode statique. Nous avons considéré deux types d'échantillons d'apprentissage : un échantillon non contaminé ne contenant que

ν	Sans anomalies		Avec anomalies	
	DR	FAR	DR	FAR
0.06	67.92 (14.19)	6.04 (0.84)	64.58 (14.34)	5.94 (0.83)
0,08	72.08 (13.80)	8.28 (1.22)	70.83 (14.06)	8.12 (1.23)
0,10	75.42 (15.91)	10.43 (1.48)	72.08 (16.64)	10.22 (1.46)
0,12	75.83 (14.28)	12.35 (1.41)	73.33 (14.58)	12.25 (1.44)

TABLE 4.8 – Dataset 1 : influence de la présence d'anomalies dans l'échantillon d'apprentissage sur la performance du modèle de détection 1-SVM.MAD$_e$, avec $q = 0.75$.

ν	Sans anomalies		Avec anomalies	
	DR	FAR	DR	FAR
0.06	89.50 (8.39)	7.23 (1.53)	68.50 (12.74)	6.53 (1.48)
0,08	93.33 (5.32)	9.24 (1.56)	85.33 (9.31)	8.39 (1.68)
0,10	95.33 (4.18)	10.87 (1.82)	93.83 (5.27)	9.89 (1.65)
0,12	96.83 (4.08)	12.96 (1.69)	96.67 (4.12)	12.01 (1.63)

TABLE 4.9 – Dataset 2 : influence de la présence d'anomalies dans l'échantillon d'apprentissage sur la performance du modèle de détection 1-SVM.MAD$_e$, avec $q = 0.75$.

des wafers normaux, et un échantillon contaminé contenant quelques wafers anormaux. Les échantillons d'apprentissage du dataset 1 sont constitués de 1700 wafers (2/3) dont 4 wafers sont anormaux dans les échantillons contaminés ce qui représente une fraction de 0.24% environ. Les échantillons d'apprentissage du dataset 2 sont constitués de 1077 wafers (2/3) dont 11 wafers sont anormaux dans les échantillons contaminés ce qui représente une fraction de 1% environ. L'écart significatif des fractions d'anomalies injectées dans les échantillons d'apprentissage est dû au faible nombre des wafers anormaux dans le dataset 1. Les échantillons de test correspondant à chacun des deux types des échantillons d'apprentissage sont les mêmes, et ils contiennent les 12 (resp. 8) wafers anormaux non injectés dans les échantillons d'apprentissage contaminés du dataset 1 (resp. dataset 2). Nous avons procédé à 100 répétitions aléatoires des wafers normaux et anormaux dans les échantillons d'apprentissage.

Dans cette section, nous avons uniquement utilisé le modèle 1-SVM.MAD$_e$ qui est beaucoup plus rapide en terme de temps de calcul que le modèle 1-SVM.RFE, l'objectif essentiel étant d'étudier le comportement du modèle de détection 1-SVM avec la présence d'anomalies dans l'échantillon d'apprentissage et non pas comparer les performances des modèles de détection. Nous aurions pu choisir 1-SVM sans sélection de variables dans cette section, mais nous avons préféré séléctionner les paramètres électriques pertinents avec la méthode de filtrage pour améliorer la performance prédictive de ce modèle et réduire le temps de calcul. L'hyperparamètre q est pris égal à 0.75 dans les deux jeux de données.

Les Tableaux 4.8 et 4.9 montrent respectivement les résultats de comparaison de la performance de deux modèles de détection dans le dataset 1 et le dataset 2, construits respectivement sur des échantillons non contaminés (Sans anomalies) et sur des échantillons contaminés (Avec anomalies), et ceci pour quatre valeurs illustratives de ν choisies parmi les 20 valeurs introduites dans la section précédente. Nous remarquons que pour toutes les valeurs de ν, les taux de détection sont plus élevés et moins variables en absence d'anomalies dans les échantillons d'apprentissage, tandis que les taux de fausses alarmes sont légèrement inférieurs dans les échantillons contaminés. Le modèle 1-SVM construit à partir d'un échantillon non contaminé tend alors à détecter plus rapidement les anomalies qu'un modèle construit à partir d'un échantillon contaminé. Notons finalement que la présence d'anomalies dans l'échantillon d'apprentissage du modèle 1-SVM n'affecte pas fortement sa performance de détection. Regardons par exemple dans le dataset 1 (resp. dataset 2) les résultats du cas "Avec anomalies" pour $\nu \geq 0.08$, où les taux de détection dépassent 70% (resp. 85%) avec une réduction moyenne d'environ 0.15% (resp. 1%) de fausses alarmes par rapport au cas "Sans anomalies". La réduction de fausses alarmes peut être expliquée par le fait que la région des observations normales devient plus grande en présence d'anomalies puisque la marge du classifieur 1-SVM est moins large dans ce cas.

Nous constatons que la différence de performance de détection entre le cas "Sans anomalies" et le cas "Avec anomalies" est moins significative dans le dataset 1 que dans le dataset 2. Ceci est dû à la faible fraction de contamination dans les échantillons d'apprentissage du dataset 1, qui est cinq fois inférieur à la fraction de contamination dans les échantillons d'apprentissage du dataset 2.

Notons que l'écart observé au niveau des taux de détection entre les deux jeux de données ne veut absolument pas dire que le modèle 1-SVM.MAD$_e$ est moins performant dans le dataset 1 que dans le dataset 2. Nous rappelons qu'il y a 8 wafers anormaux dans les échantillons de test du dataset 1. Ainsi, un taux de détection égal à 75% représente 6 wafers anormaux détectés parmi 8. De plus, il y a un wafer dans ce jeu de données impossible à détecter, le profil de ce wafer sera présenté dans la Section 5.5. Le fait que ce wafer figure dans l'échantillon de test ou non, explique les écart-type élevés des taux de détection obtenus avec ce jeu de données et dans les deux cas d'étude, puisque un wafer sur huit représente un pourcentage de 12.5 %. Finalement, il se peut que la valeur choisie de q (0.75) est plus appropriée au dataset 2 qu'au dataset 1, puisque la dimension de ce dernier est bien supérieure à celle du premier et par suite il faut peut être augmenter la valeur de q dans le dataset 2 pour obtenir une meilleure performance.

Chapitre 5

Détection en temps réel basée sur 1-SVM

Ce dernier chapitre constitue une application industrielle des systèmes de détection développés. Nous commençons par décrire notre système de détection en temps réel basé sur un modèle dynamique de 1-SVM avec une des deux méthodes de sélection de variables spécifiquement développées. Ce système est ensuite appliqué sur les deux jeux de données réels de STMicroelectronics. Ce système est comparé à un système alternatif basé sur le test de T^2 de Hotelling qui est très connu dans le domaine de détection de défauts en industrie. Ce système, que nous le dénotons par T^2.PCA, consiste en une compression de variables effectuée avec une ACP, puis la statistique T^2 de Hotelling est déterminée sur les scores de données obtenues et comparée à un seuil prédéfini.

A la fin de ce chapitre, les signatures multidimensionnelles des wafers anormaux détectés par notre système sont déterminées afin d'identifier le ou les paramètre(s) électrique(s) responsable(s) de l'anomalie.

5.1 Notre système de détection en temps réel

L'idée derrière le développement d'un système de détection en temps réel était d'exploiter nos deux modèles de détection développés 1-SVM.MAD$_e$ et 1-SVM.RFE en mode dynamique afin de tester en ligne de production (in-line testing) l'état électrique des wafers (normal ou anormal). Ce système vise donc à détecter en temps réel les wafers anormaux en se basant sur un historique récent de mesures électriques. Dans ce qui suit, et afin de simplifier la lecture, nous notons notre modèle de sélection de variables et de

classification par 1-SVM.FS (one-class SVM with Feature Selection). Ce modèle peut représenter 1-SVM.MAD$_e$ ou 1-SVM.RFE.

5.1.1 Description du système

Notre système de détection est basé sur trois étapes principales :

1. La sélection d'un jeu de données de référence, représentant le comportement de fonctionnement normal.
2. Une mise à jour des données d'apprentissage par une fenêtre mobile, pour obtenir une procédure en temps réel.
3. Application du modèle 1-SVM.FS aux données d'apprentissage mises à jour.

Donc nous définisssons d'abord un jeu de données de référence, représentant un état de fonctionnement normal. Pour cela, nous sélectionnons dans la base de données historique, un ensemble de wafers normaux correspondant à un état nominal de processus. En ce qui concerne la taille des données de référence, un grand ensemble des données augmente la fiabilité de détection. Ainsi la taille des données de référence doit être suffisamment grande, nous permettant de définir une région normale qui englobe une grande variété des wafers normaux.

Le jeu de données de référence va servir en tant qu'échantillon d'apprentissage pour construire un modèle caractérisant le comportement normal du processus. Quand un nouveau lot (un groupement de 25 wafers qui parcourent ensemble toutes les étapes du processus) arrive, le modèle 1-SVM.FS construit sur le jeu de données de référence est utilisé pour tester si chacun des 25 wafers est normal ou anormal. Le lot testé joint l'échantillon d'apprentissage initial, tandis que le lot le plus ancien dans cet échantillon sera retiré ou entretenu selon les scénarios choisis que nous allons décrire ci-dessous. Ainsi, un nouvel échantillon d'apprentissage est formé. Le modèle 1-SVM.FS est reconstruit sur ce nouvel échantillon et sera utilisé pour prédire l'état de fonctionnement des 25 wafers suivants. Cette procédure est répétée avec l'arrivée de chaque nouveau lot. Une vue générale de notre système de détection est présentée dans la Figure 5.1.

5.1.2 Deux scénarios de mise à jour

Comme nous avons vu, la dynamicité du modèle 1-SVM.FS est assurée au moyen d'une fenêtre temporelle mobile. Nous considérons 2 scénarios reflétant deux modes de mise à jour de cette fenêtre :

– Scénario 1 (taille croissante) : dans ce scénario, le lot testé à chaque itération est ajouté à l'échantillon d'apprentissage existant sans éliminer les anciennces données.

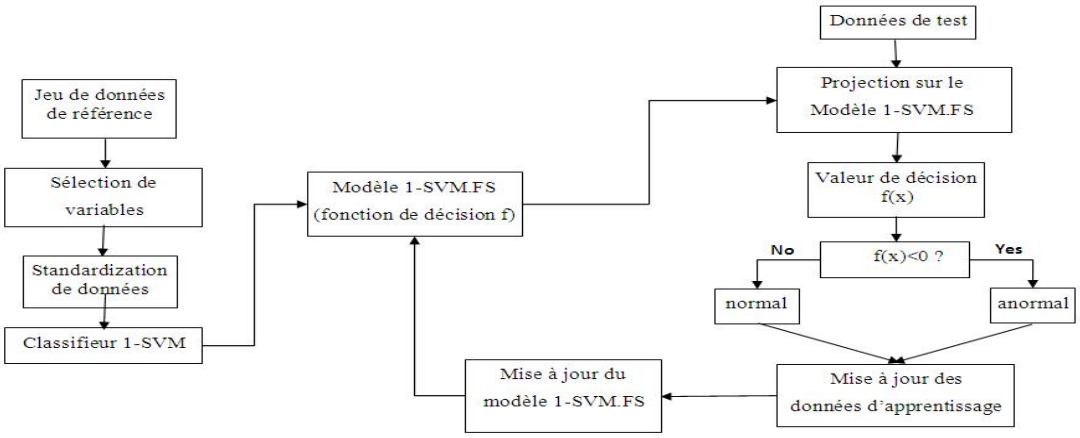

FIGURE 5.1 – Schéma fonctionnel de notre système de détection en temps réel basé sur le modèle dynamique 1-SVM.FS.

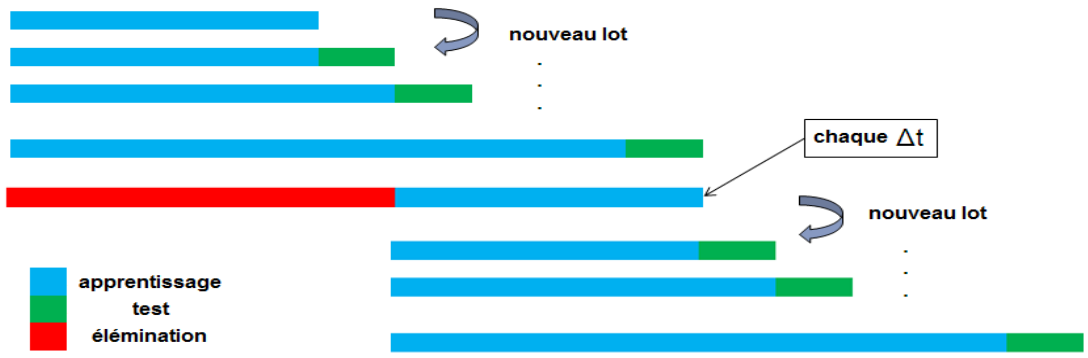

FIGURE 5.2 – Fenêtre mobile du système de détection en temps réel suivant le scénario 1.

Alors le modèle 1-SVM.FS est mis à jour suivant une fenêtre mobile de taille croissante. Comme le comportement normal ne cesse pas d'évoluer, nous avons décidé d'éliminer à la fois des anciennes données de l'ensemble d'apprentissage existant après une période prédéfinie Δt. Δt dépend du volume de production du produit considéré.

– Scénario 2 (taille fixe) : pendant l'opération de détection en temps réel, la fenêtre mobile conserve la même taille que celle de l'échantillon d'apprentissage initial (ou le jeu de données de référence), et fonctionne selon la procédure "First-IN-First-Out", en ajoutant le nouveau lot déjà testé dans la base d'apprentissage et éliminant le lot le plus ancien de cette base.

Les deux scénarios sont illustrés respectivement dans les Figures 5.2 et 5.3. Finalement, notons que les méthodes de sélection de variables sont appliquées à chaque mise à jour des données d'apprentissage, ce qui permet le suivi en temps réel d'éventuels changements des paramètres électriques discriminants responsables de l'anomalie des wafers.

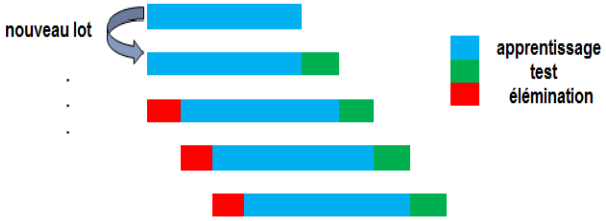

FIGURE 5.3 – Fenêtre mobile du système de détection en temps réel suivant le scénario 2.

5.1.3 Choix optimal de q et ν

Le modèle 1-SVM.FS nécessite le réglage du paramètre ν (le seuil de détection dans l'algorithme 1-SVM) et de l'hyperparamètre q des méthodes de sélection de variables. Rappelons que l'hyperparamètre q représente la fraction des paramètres retenus pour chacune de ces deux méthodes.

Par conséquent, un choix optimal de ν et q doit être fait. Pour accomplir cette tâche, nous utilisons la procédure basée sur l'ensemble de validation, expliquée dans la Section 2.3.5 du Chapitre 2. Pour cela, nous constituons à partir de la base de données un ensemble de validation contenant des wafers normaux contaminé par la présence de quelques wafers anormaux. Cet ensemble est utilisé pour identifier la meilleure paire (q, ν) afin que le classifieur 1-SVM puisse prédire avec la meilleure précision les données de test (les nouveaux wafers qui arrivent). Une recherche sur une grille de valeurs de q et ν est effectuée. Le modèle 1-SVM.FS est construit sur le jeu de données de référence en utilisant les différentes paires $(q,\ \nu)$. Pour chaque paire, les données de l'ensemble de validation sont testées avec le modèle 1-SVM.FS construit. Le taux de détection (DR) et le taux de fausses alarmes (FAR) sont ensuite calculés. La paire qui optimise ces deux mesures de performance est choisie. Plus précisément, la meilleure paire $(q,\ \nu)$ est celle qui donne le meilleur compromis entre la maximisation du taux de détection et la minimisation du taux de fausses alarmes. La paire choisie est utilisée à chaque mise à jour du modèle 1-SVM.FS. L'optimum selon les deux mesures de performance n'est pas toujours unique. Il se peut que nous obtenons par exemple un DR de 100% et un FAR de 15% avec une paire, et un DR de 90% et un FAR de 5% avec une autre paire, dans ce cas laquelle des deux paires choisissons-nous ? Nous décidons alors de choisir la paire qui maximise la mesure G-means.

Dans notre étude, le calibrage de ν et q pour notre système de détection en temps réel est fait seulement une fois au début sur le jeu de données de référence et l'ensemble de validation. Plus précisément, ce calibrage n'est pas fait dans chaque fenêtre mobile

où l'échantillon d'apprentissage et le modèle 1-SVM.FS sont mis à jour. Ceci n'est pas possible car nous ne pouvons pas nous disposer d'un nouvel ensemble de validation pour chaque fenêtre mobile. Une seule possibilité est alors de considérer chaque nouveau lot testé comme un ensemble de validation avec les wafers détectés par notre système comme wafers anormaux, et nous pouvons ensuite appliquer la procédure basée sur cet ensemble de validation pour choisir la meilleure paire (q, ν). Cependant, les wafers considérés comme anormaux par notre système peuvent réellement être des fausses alarmes.

Nous pourrons d'ailleurs mettre à jour ces hyperparamètres au retour des résultats du test électrique final où un nouveau ensemble de validation contenant quelques wafers anormaux est constitué. Cette mise à jour pourra être faite tous les deux mois, une période qui correspond au temps de cycle de production.

5.2 Taille du jeu de données de référence

Nous commençons par étudier l'influence de la taille du jeu de données de référence, que nous dénotons par n_{ref}, sur la performance de nos modèles dynamiques. Pour cela nous avons testé en temps réel les modèles 1-SVM.MAD$_e$ et 1-SVM.RFE sur les deux jeux de données, suivant les deux scénarios et pour des valeurs de n_{ref} égales à 400 et 600 (resp. 300 et 400) pour le dataset 1 (resp. dataset 2).

Le Tableau 5.1 représente une illustration numérique pour certaines valeurs de ν de la comparaison de performance de détection réalisée sur le dataset 1 avec le modèle 1-SVM.MAD$_e$ suivant le scénario 1. En augmentant n_{ref} de 400 à 600, les taux de détection restent les mêmes alors que les taux de fausses alarmes sont légérement inférieurs.

Dans la Figure 5.4, des courbes FAR-DR de la détection en temps réel sur le dataset 2 avec le modèle 1-SVM.MAD$_e$ suivant le scénario 1 et pour les deux valeurs de n_{ref} sont obtenues en faisant varier ν ($\nu \in \{0.01, 0.02, \ldots, 0.19, 0.2\}$). Malgré les légères améliorations en terme de taux de détection et de fausses alarmes obtenues pour quelques valeurs de ν, les deux courbes sont similaires.

Ensuite, des illustrations numériques pour certaines valeurs de ν de la comparaison de performance de détection réalisée sur le dataset 1 et le dataset 2 avec le modèle 1-SVM.RFE suivant le scénario 1 sont données respectivement dans les Tableaux 5.2 et 5.3 . D'après ces tableaux, l'augmentation de n_{ref} aboutit aux mêmes résultats observés avec le modèle 1-SVM.MAD$_e$ appliqué avec le même scénario (scénario 1). En effet, les taux de détection sont constants, avec des petites réductions en fausses alarmes (sauf pour $\nu = 0.04$ dans le dataset 2, le taux de fausses alarmes était supérieur). Nous pouvons donc constater que l'augmentation de la taille du jeu de données de référence n'affecte pas

ν	$n_{ref} = 400$		$n_{ref} = 600$	
	DR	FAR	DR	FAR
0.03	70	5.66	70	4.40
0.04	80	6.79	80	6.50
0.05	80	8.54	80	6.61
0.09	90	15.38	90	14.21

TABLE 5.1 – Dataset 1 : illustration numérique comparant la détection en temps réel avec le modèle 1-SVM.MAD$_e$ suivant le scénario 1 pour $n_{ref} = 400$ et $n_{ref} = 600$, en faisant variant ν et pour $q = 0.75$.

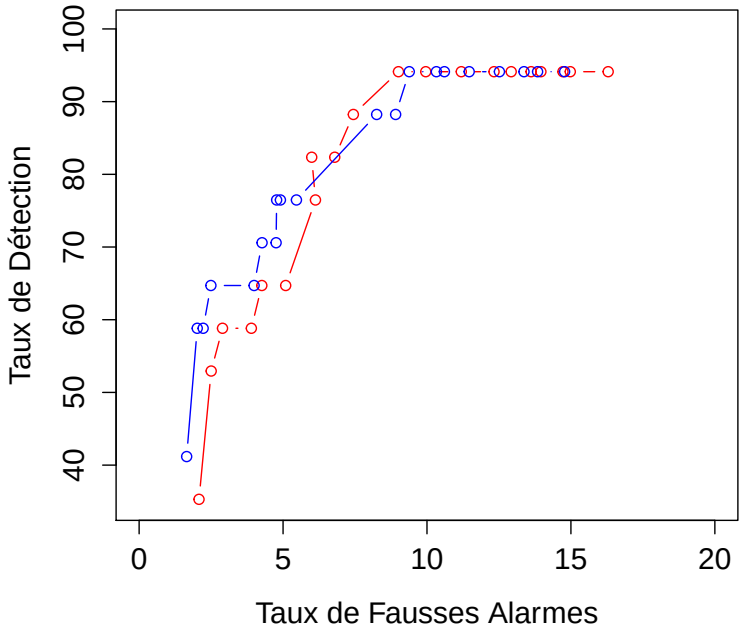

FIGURE 5.4 – Dataset 2 : courbes FAR-DR de la détection en temps réel avec le modèle 1-SVM.MAD$_e$ suivant le scénario 1 pour $n_{ref} = 300$ et $n_{ref} = 400$, en faisant variant ν et pour $q = 0.75$.

significativement la performance de détection de 1-SVM.MAD$_e$ et 1-SVM.RFE appliqués suivant le scénario 1. Ceci est dû à la taille croissante de la fenêtre mobile du scénario 1 qui permet l'augmentation au fur et à mesure de la taille des données d'apprentissage puisque chaque lot testé rejoint l'échantillon d'apprentissage mis à jour à chaque fenêtre sans éliminer le lot le plus ancien de cet échantillon. La taille croissante de la fenêtre mobile du scénario 1 aboutit donc à une détection pertinente sans nécessiter une taille n_{ref} assez élevée. Cependant, la taille du jeu de données de référence joue un rôle important en effectuant une détection suivant le scénario 2 avec le modèle 1-SVM.MAD$_e$. Ceci est illustré dans le Tableau 5.4 et la Figure 5.5. Les résultats obtenus dans le Tableau 5.4 sur le dataset 1 montrent que l'augmentation de la taille du jeu de données de

ν	$n_{ref} = 300$		$n_{ref} = 400$	
	DR	FAR	DR	FAR
0.03	80	4.12	80	4.03
0.04	80	6.32	80	6.21
0.05	80	8.40	90	6.53
0.09	90	15.91	90	12.04

TABLE 5.2 – Dataset 1 : illustration numérique comparant la détection en temps réel avec le modèle 1-SVM.RFE suivant le scénario 1 pour $n_{ref} = 400$ et $n_{ref} = 600$, en faisant variant ν et pour $q = 0.75$.

ν	$n_{ref} = 300$		$n_{ref} = 400$	
	DR	FAR	DR	FAR
0.04	70.59	4.12	70.59	6.03
0.05	82.35	6.32	82.35	6.21
0.06	88.24	8.40	88.24	6.53
0.12	94.12	15.91	94.12	12.04

TABLE 5.3 – Dataset 2 : illustration numérique comparant la détection en temps réel avec le modèle 1-SVM.RFE suivant le scénario 1 pour $n_{ref} = 300$ et $n_{ref} = 400$, en faisant variant ν et pour $q = 0.75$.

ν	$n_{ref} = 400$		$n_{ref} = 600$	
	DR	FAR	DR	FAR
0.03	70	9.66	80	5.65
0.04	70	11.03	80	6.31
0.05	80	13.81	90	9.29
0.09	90	22.56	90	15.22

TABLE 5.4 – Dataset 1 : illustration numérique comparant la détection en temps réel avec le modèle 1-SVM.MAD$_e$ suivant le scénario 2 pour $n_{ref} = 400$ et $n_{ref} = 600$, en faisant variant ν et pour $q = 0.75$.

référence aboutit à une amélioration significative dans la détection : les taux de détection augmentent (sauf pour $\nu = 0.09$ où ce taux est le même dans les deux cas, sachant que c'est le taux maximum qu'on peut obtenir sur le dataset 1), suivis de réductions importantes en terme de fausses alarmes. Les améliorations significatives des deux mesures de performance sont observées également pour toutes les valeurs de ν dans la Figure 5.5, où nous avons tracé les courbes FAR-DR de la détection en temps réel sur le dataset 2 avec le modèle 1-SVM.MAD$_e$ suivant le scénario 2 et avec les deux valeurs de n_{ref}. Comme la taille de la fenêtre mobile dans le scénario 2 est constante, augmenter la taille du jeu de données de référence aboutit à l'amélioration de la performance de détection du modèle 1-SVM.MAD$_e$. En effet, la méthode 1-SVM nécessite de nombreuses données d'apprentissage pour donner une frontière de décision de bonne précision parce que ses vecteurs supports ne proviennent essentiellement que des données de la classe positive

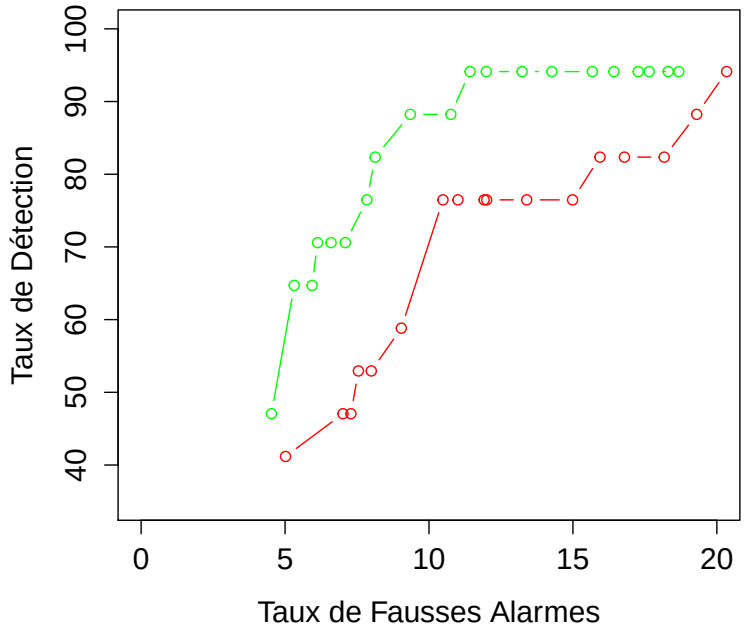

FIGURE 5.5 – Dataset 2 : courbes FAR-DR de la détection en temps réel avec le modèle 1-SVM.MAD$_e$ suivant le scénario 2 pour $n_{ref} = 300$ et $n_{ref} = 400$, en faisant variant ν et pour $q = 0.75$.

(wafers normaux).

L'influence de la taille du jeu de données de référence sur la performance du modèle 1-SVM.RFE appliqué suivant le scénario 2 a été aussi étudié sur les deux jeux de données. Des illustrations numériques de cette étude sont présentées respectivement dans les Tableaux 5.5 et 5.6, où sont présentés les résultats de comparaison pour certaines valeurs de ν. Des améliorations importantes des deux mesures de performance sont observées en augmentant la taille n_{ref} dans le dataset 1 (cf Tableau 5.5). Dans le Tableau 5.6, nous remarquons que les taux de détection n'ont pas évolué avec l'augmentation de n_{ref}, vu que ces taux sont déjà trés élevés dès les premières valeurs de ν, et pour les deux valeurs de n_{ref}. Concernant les fausses alarmes, nous avons observé de légères réductions au niveau des quatre premières valeurs de ν, puis des réductions significatives pour ν égal à 0.12 et 0.16.

Pour conclure, la taille du jeu de données de référence est un facteur important pour une détection pertinente avec le scénario 2. Nous avons vu que généralement les taux de détection augmentent et les taux de fausses alarmes diminuent en augmentant cette taille. Contrairement au scénario 2, le scénario 1 ne dépend pas visiblement de ce facteur.

ν	$n_{ref} = 300$		$n_{ref} = 400$	
	DR	FAR	DR	FAR
0.03	80	6.89	80	4.91
0.04	80	8.14	90	5.62
0.05	80	12.71	90	8.64
0.09	90	21.16	90	14.05

TABLE 5.5 – Dataset 1 : illustration numérique comparant la détection en temps réel avec le modèle 1-SVM.RFE suivant le scénario 2 pour $n_{ref} = 400$ et $n_{ref} = 600$, en faisant variant ν et pour $q = 0.75$.

ν	$n_{ref} = 300$		$n_{ref} = 400$	
	DR	FAR	DR	FAR
0.04	88.24	7.98	88.24	7.34
0.05	94.12	9.14	94.12	8.48
0.06	94.12	10.90	94.12	10.42
0.12	94.12	20.66	94.12	16.47

TABLE 5.6 – Dataset 2 : illustration numérique comparant la détection en temps réel avec le modèle 1-SVM.RFE suivant le scénario 2 pour $n_{ref} = 300$ et $n_{ref} = 400$, en faisant variant ν et pour $q = 0.75$.

5.3 Exemples de performance des systèmes de détection en temps réel

Notre objectif expérimental était d'évaluer la capacité de nos systèmes de détection pour détecter automatiquement les wafers anormaux. Il est également important de réduire au minimum le taux de fausses alarmes car ils provoquent l'interruption injustifiée dans le cycle de production.

Les systèmes de détection basés sur les modèles dynamiques 1-SVM.MAD$_e$, 1-SVM.RFE et T^2.PCA sont testés suivant les deux scénarios de mise à jour sur les deux jeux de données réels de STMicroelectronics, afin de prouver l'efficacité et la supériorité de nos systèmes de détection. Idéalement, nous voulons un haut DR (pour détecter la majorité des wafers anormaux) et un faible taux de fausses alarmes (pour éviter de classer les wafers normaux comme anormaux).

5.3.1 Dataset 1

Le dataset 1 est composé de 2550 wafers du produit P_1 de la technologie I140, et correspond à une période de production de 4 mois. 12 wafers sont considérés comme anormaux au PT dans ce jeu de données. Dans cette expérience, nous avons défini un jeu de données

de référence contenant 600 wafers normaux. L'ensemble de validation consiste en 100 wafers dont 2 sont anormaux. Par conséquent, il reste 1850 wafers à tester (74 lots) dont 10 sont anormaux. Nous avons construit les modèles de détection sur le jeu de données de référence avec plusieurs paires de l'hyperparamètre q et le seuil de détection ν. Nous considérons respectivement 9 et 20 valeurs de q et ν :

$$q \in \{0.25,\ 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.75,\ 0.8,\ 0.85,\ 0.9\},$$
$$\nu \in \{0.01,\ 0.02, \ldots,\ 0.19,\ 0.2\}.$$

Les étiquettes des wafers de l'ensemble de validation sont ensuite prédites en utilisant chacun de 180 (9×20) modèles construits. Les taux de détection et de fausses alarmes sont calculés pour chaque modèle de prédiction. Nous sélectionnons la paire de valeurs qui optimise simultanément ces deux mesures de performance (i.e qui optimise la mesure de performance G-means). Pour le modèle 1-SVM.MAD$_e$, nous avons retenu $q = 0.8$ et $\nu = 0.04$ où nous avons détecté tous les wafers anormaux de l'ensemble de validation (DR= 100%) pour un FAR égal à 4.35% (G-means= 97.8). Pour le modèle 1-SVM.RFE, nous avons retenu $q = 0.85$ et $\nu = 0.03$ où nous avons aussi détecté tous les wafers anormaux de l'ensemble de validation pour un FAR égal à 3.25% (G-means= 98.36).

D'une façon similaire, nous avons sélectionné la meilleure paire (ψ, α) (cf. Section 3.3) du modèle T^2.PCA en prenant $\psi \in \{0.75,\ 0.8,\ 0.85,\ 0.9\}$ et considérant la même plage de valeurs de ν pour α. La performance optimale est obtenue pour $\psi = 0.85$ and $\alpha = 0.01$, où DR et FAR sont respectivement égaux à 100% et 2.75% (G-means= 98.62).

Pour le scénario 1, Δt corespond à la période pour laquelle nous aurions testé la moitié des lots. Plus précisément, la taille de la base d'apprentissage croît à chaque itération correspondante à un test d'un lot, jusqu'à avoir testé la moitié des lots. Ensuite les anciens wafers sont éliminés de la base de sorte que nous gardons seulement un nombre de wafers égal à celui du jeu de données de référence.

Après avoir défini le jeu de données de référence et sélectionné les hyperparamètres otpimaux pour les trois modèles de détection, nous procédons maintenant à la détection en temps réel en appliquant chacun des trois modèles sur les données. 74 fenêtres mobiles sont considérées pour mettre à jour les modèles de détection, puisque nous nous disposons de 74 lots à tester. Les mises à jour suivant une fenêtre mobile des données d'apprentissage et des modèles de détection à l'arrivée de chaque nouveau lot testé permettent ainsi d'obtenir une procédure en temps réel. Ces mises à jour suivent un des deux scénarios déjà définis : scénario 1 (taille croissante) et scénario 2 (taille fixe).

Le Tableau 5.7 montre une comparaison de performance des trois systèmes de détection en temps réel suivant les deux scénarios de mise à jour des fenêtres mobiles. Notons que

Fenêtre mobile	Système	DR	FAR
Scénario 1	1-SVM.MAD$_e$	80	5.64
	1-SVM.RFE	90	3.05
	T^2.PCA	80	6.21
Scénario 2	1-SVM.MAD$_e$	90	6.11
	1-SVM.RFE	90	5.75
	T^2.PCA	90	7.36

TABLE 5.7 – Dataset 1 : comparaison de la performance de détection des systèmes 1-SVM.MAD$_e$, 1-SVM.RFE et T^2.PCA.

le taux de détection DR représente le pourcentage de wafers anormaux détectés par le système utilisé parmi tous les wafers anormaux présents dans les 74 fenêtres (ici il y a 10), et le taux de fausses alarmes FAR représente la moyenne des fausses alarmes obtenues sur l'ensemble de 74 fenêtres. Les résultats révèlent une similarité de performance entre les trois systèmes. Une haute performance a été obtenue avec les trois systèmes. Ils ont détecté 9 parmi 10 wafers anormaux en utilisant le scénario 2 et 8 parmi 10 avec le scénario 1 (sauf pour 1-SVM.RFE qui a détecté 9 parmi 10). Avec les deux scénarios, 1-SVM.RFE a réduit les fausses alarmes par rappot aux deux autres systèmes, qui à leur tour avaient des taux très proches.

5.3.2 Dataset 2

Le dataset 2 est composé de 1600 wafers du produit P_2 de la technologie I140, dont 23 sont considérés comme anormaux au PT. Dans cette expérience, nous avons construit un jeu de données de référence contenant 400 wafers normaux. L'ensemble de validation consiste en 100 wafers dont 6 sont anormaux. Par conséquent, il reste 1100 wafers à tester (44 lots) dont 17 sont anormaux. Comparé au dataset 1, ce jeu de données a une dimension d'espace de variables plus petite et un pourcentage de wafers anormaux plus grand.

En suivant la même procédure utilisée dans le dataset 1 pour sélectionner les hyperparamètres optimaux, nous avons retenu $q = 0.75$ et $\nu = 0.13$ pour le modèle 1-SVM.MAD$_e$ où nous avons détecté tous les wafers anormaux de l'ensemble de validation (DR= 100%) pour un FAR égal à 11.37% (G-means= 94.14). Pour le modèle 1-SVM.RFE, nous avons retenu $q = 0.8$ et $\nu = 0.06$ où nous avons aussi détecté tous les wafers anormaux de l'ensemble de validation pour un FAR égal à 6.19% (G-means= 96.86). Pour le modèle T^2.PCA, la performance optimale sur l'ensemble de validation est obtenue pour $\psi = 0.75$ and $\alpha = 0.18$, où DR et FAR sont respectivement égaux à 66.67% (4 sur 6 wafers anormaux) et 15.89% (G-means= 74.88).

Fenêtre mobile	Système	DR	FAR
Scénario 1	1-SVM.MAD$_e$	94.12	9.38
	1-SVM.RFE	88.24	6.53
	T^2.PCA	64.71	11.56
Scénario 2	1-SVM.MAD$_e$	94.12	13.23
	1-SVM.RFE	94.12	10.42
	T^2.PCA	70.59	15.89

TABLE 5.8 – Dataset 2 : comparaison de la performance de détection des systèmes 1-SVM.MAD$_e$, 1-SVM.RFE et T^2.PCA.

44 fenêtres mobiles sont considérées pour mettre à jour les modèles de détection. Les mises à jour des modèles de détection se font selon le scénario 1 (taille croissante) et le scénario 2 (taille fixe). Comme dans le dataset 1, le taux de détection DR représente le pourcentage de wafers anormaux détectés par le système utilisé parmi tous les wafers anormaux présents dans les 44 fenêtres (ici il y a 17), et le taux de fausses alarmes FAR représente la moyenne des fausses alarmes obtenues sur l'ensemble de 44 fenêtres.

Les résultats de comparaison sont donnés dans le Tableau 5.8. Avec les deux scénarios, le système de T^2 de Hotelling basé sur le modèle dynamique T^2.PCA montre une faible performance dans la détection des wafers anormaux (DR respectivement égal à 64.71 et70.59%). Tandis que le système basé sur le modèle 1-SVM.MAD$_e$ a été capable de détecter 94.12% des wafers anormaux avec les deux scénarios. De plus, avec ce système nous avons obtenu des taux de fausses alarmes inférieurs par rapport au système basé sur T^2.PCA, mais supérieurs par rapport au système basé sur 1-SVM.RFE avec lequel les taux de détection selon les deux scénarios ont été respectivement égaux à 88.24 et 94.12%. Avec les trois systèmes de détection, le scénario 1 a réduit le taux de fausses alarmes par rapport au scénario 2. Concernant nos deux systèmes, une réduction de fausses alarmes de 2.85 et 2.81 sont obtenues avec 1-SVM.RFE par rapport à 1-SVM.MAD$_e$ en utilisant respectivement les scénarios 1 et 2. Notons que dans ce jeu de données considérant seulement deux mois de production, nous n'avons pas éliminé les anciennes données de la base d'apprentissage existante après une période Δt pour le scénario 1, comme il a été recommandé dans la Section 5.1.2. Cette action a eu lieu seulement dans la première expérience considérant quatre mois de production, où le nombre de wafers est grand.

Temps de calcul

Nous nous intéressons maintenant à la comparaison de temps de calcul des trois systèmes de détection en temps réel. Pour cela, nous considérons le critère de performance "Temps". Ce critère représente le temps moyen, en secondes, émis par un système pour prédire l'état

Système	Temps (sec)	
	Dataset 1	Dataset 2
1-SVM.MAD$_e$	1.44	1.05
1-SVM.RFE	1412.71	912.62
T^2.PCA	7.49	4.40

TABLE 5.9 – Temps moyen émis par un système de détection pour tester un nouveau lot selon le scénario 2.

des wafers d'un nouveau lot à tester selon le scénario 2 caractérisé par une fenêtre mobile de taille fixe. A une itération quelconque, ce critère comprend le temps nécessaire pour :

1. la mise à jour de la méthode de sélection de variables.
2. la mise à jour du modèle 1-SVM.
3. la phase de prédiction du nouveau lot.

Le Tableau 5.9 nous montre, sur les deux jeux de données, que 1-SVM.MAD$_e$ et T^2.PCA sont très rapides en temps de calcul, avec une supériorité du premier système sur le deuxième. Ces deux systèmes sont nettement meilleurs que 1-SVM.RFE, qui est à son tour extrêmement coûteux en temps de calcul (cf. Section 3.5.2). Notons que le critère calculé avec les trois systèmes est plus grand dans le dataset 1, puisque la dimension de son espace de variables est plus élevée que celle du dataset 2.

5.3.3 Conclusion

Pour conclure, le système basé sur le modèle 1-SVM.RFE est le plus efficient. Les taux de détection avec ce système ont été très élevés avec des réductions significatives de fausses alarmes par rapport aux autres systèmes de détection utilisés.

Similairement au système 1-SVM.RFE, notre système basé sur 1-SVM.MAD$_e$ a montré une bonne performance de détection sur les deux jeux de données réels. De plus il est beaucoup plus rapide en temps de calcul par rapport au système basé sur 1-SVM.RFE.

Sur le dataset 1, le système basé sur T^2.PCA a montré une performance compététive à celles des deux autres systèmes, avec un peu plus de fausses alarmes. La performance de ce système s'est dégradée sur le dataset 2. Ce système exige quelques contraintes pour une bonne détection qui ne peuvent pas être vérifiées par les données comme la normalité des variables (T^2 de Hotelling), ou/et la linéarité ainsi que la corrélation entre les variables (ACP).

La mise à jour de la fenêtre mobile des modèles dynamiques aboutit à une réduction de fausses alarmes selon le scénario 1, et à une détection plus rapide des wafers anormaux selon le scénario 2. Une interprétation expliquant ces résultats sera donnée dans la section

suivante. Par conséquent, nous pouvons décider d'appliquer nos systèmes de détection selon le scénario 2, mais en enrichissant encore le jeu de données de référence avec des observations normales. Cela va amener l'algorithme 1-SVM à reconnaître plus facilement les wafers normaux et par suite réduire les fausses alarmes.

Pour une application industrielle, nous recommandons donc d'utiliser le système 1-SVM.RFE selon le scénario 2. Pour une technologie volumineuse, nous pourrons utiliser 1-SVM.MAD$_e$ qui est un système simple, efficace et beaucoup plus rapide que 1-SVM.RFE. Avec un réglage fin des paramètres de ce système, il aboutit à une performance compétitive à celle du système 1-SVM.RFE.

5.4 Comparaison de deux scénarios de mise à jour

Dans cette section, nous avons pour objectif de confirmer les conclusions tirées dans la section précédente concernant les deux scénarios utilisés pour la mise à jour de la fenêtre mobile de nos techniques de détection.

Le Tableau 5.10 représente une illustration numérique de la comparaison des scénarios 1 et 2 suivant nos modèles dynamiques de détection appliqués sur le dataset 1. Pour les quatre valeurs de ν et avec les deux modèles de détection, le scénario 1 donne des taux de fausses alarmes plus petits par rapport au scénario 2. Avec le modèle 1-SVM.MAD$_e$, les taux de détection avec le scénario 2 sont plus élevés que ceux du scénario 1, sauf pour $\nu = 0.09$ où un taux de détection maximal est obtenu avec les deux modèles. Avec le modèle 1-SVM.RFE et selon les deux scénarios, les taux de détection atteignent tous la valeur maximale égale à 90%.

Pour le dataset 2, des illustrations numériques pour six valeurs de ν de la comparaison des scénarios 1 et 2 suivant les modèles dynamiques 1-SVM.MAD$_e$ et 1-SVM.RFE sont données respectivement dans les Tableaux 5.11 et 5.12. Le scénario 1 fournit une réduction significative en terme de fausses alarmes par rapport au scénario 2, et ceci pour les deux modèles de détection. Par exemple, dans le Tableau 5.11 (resp. Tableau 5.12), nous notons une réduction entre 3 (resp. 1.3) et 5% (resp. 4.5%) environ de fausses alarmes pour les différentes valeurs de ν. Pour les quatre premières valeurs de ν et avec les deux modèles, les taux de détection ont augmenté avec le scénario 2. Pour les deux dernières valeurs de ν dans chaque tableau, les taux de détection pour les deux modèles sont maximaux et sont égaux à 94.12%. Donc pour toute valeur de ν des Tableaux 5.11 et 5.12, le scénario 2 a un taux de détection supérieur ou égal à celui du scénario 1.

Ces résultats confirment donc nos conclusions de la section précédente concernant les deux scénarios. D'une part, la réduction de fausses alarmes avec le scénario 1 est dûe à

ν	1-SVM.MAD$_e$				1-SVM.RFE			
	scénario 1		scénario 2		scénario 1		scénario 2	
	DR	FAR	DR	FAR	DR	FAR	DR	FAR
0.03	70	4.17	80	5.36	90	2.79	90	4.81
0.04	80	5.64	90	6.11	90	3.05	90	5.75
0.05	80	6.27	90	8.57	90	4.41	90	8.11
0.09	90	12.92	90	14.15	90	8.63	90	13.71

TABLE 5.10 – Dataset 1 : illustration numérique comparant les scénario 1 et 2 avec les modèles dynamiques 1-SVM.MAD$_e$ et 1-SVM.RFE.

ν	scénario 1		scénario 2	
	DR	FAR	DR	FAR
0.07	70.59	4.76	76.47	7.84
0.08	76.47	4.78	82.35	8.13
0.09	76.47	4.91	88.23	9.35
0.11	88.24	8.25	94.12	11.43
0.13	94.12	9.38	94.12	13.23
0.15	94.12	10.61	94.12	15.68

TABLE 5.11 – Dataset 2 : illustration numérique comparant la détection en temps réel avec le modèle 1-SVM.MAD$_e$ suivant les scénario 1 et 2, pour $q = 0.75$.

ν	scénario 1		scénario 2	
	DR	FAR	DR	FAR
0.04	70.59	6.03	88.24	7.34
0.05	82.35	6.21	94.12	8.48
0.06	88.24	6.53	94.12	10.42
0.11	88.24	13.24	94.12	16.46
0.12	94.12	12.04	94.12	16.47
0.13	94.12	14.92	94.12	19.41

TABLE 5.12 – Dataset 2 : illustration numérique comparant la détection en temps réel avec le modèle 1-SVM.RFE suivant les scénario 1 et 2, pour $q = 0.75$.

la taille croissante de la fenêtre mobile où à chaque mise à jour le nouveau lot testé est ajouté à la base d'apprentissage. Ceci permet à l'algorithme 1-SVM d'enrichir sa base d'apprentissage avec plus d'observations normales ce qui facilite la reconnaissance de futures observations normales et donc on obtient moins de fausses alarmes. D'autre part, la détection rapide des wafers anormaux avec le scénario 2 est dûe à une stratégie de mise à jour efficace de la base d'apprentissage avec sa fenêtre de taille fixe et contenant moins de wafers anormaux que celle du scénario 1, et cela améliore la performance de détection de l'algorithme 1-SVM. Ceci a été démontré dans la Section 4.5.

5.5 Signatures multidimensionnelles des wafers anormaux

Quand notre système de détection déclenche une alarme (wafer anormal), une tâche essentielle sera d'identifier les paramètres électriques responsables de cette anomalie. Ceci est fait en définissant la signature multidimensionnelle du wafer anormal pour pouvoir remonter à la cause et ensuite effectuer les modifications opérationelles nécessaires pour faire retourner le processus aux conditions normales. Nous avons donc proposé une méthode basée sur la distance normalisée par rapport à la population normale, des paramètres retenus par la méthode utilisée pour la sélection de variables .

A une itération donnée, la base d'apprentissage est mise à jour et le modèle de détection (1-SVM.MAD$_e$ ou 1-SVM.RFE) est construit. Un nouveau lot à tester arrive, un wafer (ou plusieurs) appartenant à ce lot est considéré comme anormal par notre modèle. Nous considérons l'ensemble des paramètres électriques pertinents retenus par la méthode de sélection de variables. Pour chaque variable (un des paramètres électriques mesuré sur un des neuf sites de mesure), nous déterminons la valeur normalisée de la façon suivante :

$$D(x_{ij},\ B) = \frac{x_{ij} - \bar{x}_{j,B}}{s_{j,B}} \tag{5.1}$$

où x_{ij} représente la mesure de la $j$ème variable pour le wafer détecté d'indice i, B constitue l'échantillon d'apprentissage du modèle utilisé pour détecter ce wafer, $\bar{x}_{j,B}$ et $s_{j,B}$ sont respectivement la moyenne et l'écart-type de la variable j sur l'ensemble de wafers appartenant à B. Pour deux wafers détectés appartenant à un même lot, B est le même, et il est différent pour deux wafers détectés qui appartiennent à deux lots différents, puisque la fenêtre mobile ne va pas être la même.

Cette distance normalisée permet de détecter des éventuelles déviations d'un ou plusieurs paramètres sélectionnés par rapport à la base d'apprentissage B. Notre méthode de définition de signature permet donc de visualiser les valeurs normalisées des paramètres électriques pertinents, et de déterminer en particulier les paramètres responsables de l'anomalie (i.e. les paramètres qui ont des valeurs normalisées très élevées ou très basses).

Dans ce qui suit, nous reprenons les résultats de détection en temps réel de la Section 5.3 avec les systèmes 1-SVM.MAD$_e$ et 1-SVM.RFE selon le scénario 2 afin d'étudier les signatures des wafers anormaux. Le scénario 2 a été choisi pour comparer les signatures des différents wafers anormaux par rapport à une base d'apprentissage de même taille. Les noms des wafers étudiés ne seront pas affichés pour des raisons de confidentialité. Nous ne montrons pas les noms des paramètres électriques sur les axes des abscisses des figures pour des raisons de lisibilité. Nous encerclons en rouge le groupement de paramètres discriminants, chaque groupement étant constitué d'un ou plusieurs paramètres,

avec un ou plusieurs sites qui peuvent déclencher pour chaque paramètre. Toutefois, les sites de mesure des paramètres électriques qui déclenchent peuvent être différents d'un wafer anormal à un autre. Nous désignons par "wafer anormal" un wafer considéré comme anormal par les ingénieurs PT et détecté par nos systèmes de détection, et par "signatures" d'un wafer les deux signatures obtenues avec chacune des méthodes de sélection de variables.

Dans le dataset 1 et dataset 2, les paramètres électriques identifiés par notre méthode comme responsables de l'anomalie d'un wafer ont été confirmés par les ingénieurs PT. Le nombre de paramètres électriques qui apparaissent dans les deux signatures d'un wafer n'est pas le même puisque la valeur de nombre de variables retenues q n'est pas la même pour les deux méthodes de sélection de variables. Rappelons que le dataset 1 et dataset 2 correspondent à deux produits d'une même technologie, d'où la majorité des paramètres électriques sont communs pour ces deux produits.

Dataset 1

Nous donnons un exemple de signature d'un wafer anormal dans le dataset 1. Les wafers anormaux dans ce dataset ont tous des signatures similaires. Les Figures 5.6 et 5.7 montrent les signatures du wafer anormal QXXXXXX_15. Trois paramètres ont communément déclenché selon les deux méthodes de sélection de variables. Les paramètres de fiabilité "VBD_AREA_NP_2" et "VBD_AREA_PN_2" (cercles en bas) ont des valeurs extrêmement basses en plusieurs sites de mesure par rapport à la population normale, tandis que le paramètre de fiabilité "VBD_AREA_PN_1" (cercle en haut à droite) avait une valeur d'un site de mesure extrêmement élevée. Les paramètres "VBD_AREA_PN_1" et "VBD_AREA_PN_2", comme le paramètre "VBD_AREA_NP_2", mesurent la tension de claquage d'une structure de test. De plus, avec la méthode MAD_e.FS, le paramètre de qualité "NIDS0205RD14" était parmi l'ensemble de paramètres responsables de l'anomalie de ce wafer. En effet, il avait des valeurs trés élevées pour deux sites de mesure. Alors que ce paramètre n'a pas été retenu par la méthode 1-SVM-RFE. Cet exemple a montré que nos méthodes de sélection de variables ont réussi à identifier simultanément les paramètres électriques qui ont déclenché la détection. De plus, la majorité des paramètres électriques sélectionnés par ces méthodes sont communs.

L'identification efficace des paramètres électriques discriminants ("VBD_AREA_NP_2", "VBD_AREA_PN_1" et "VBD_AREA_PN_2") avec leur écart extrême par rapport à la population d'apprentissage ont permis de détecter facilement tous les wafers anormaux (sauf le wafer QXXXXXX_03). Ce qui rend le dataset 1 un jeu de données facile pour la détection d'anomalies.

Les Figures 5.16 et 5.18 montrent les signatures du wafer QXXXXXX_03, indétectable par les systèmes 1-SVM.MAD$_e$ et 1-SVM.RFE. Bien que ce wafer possède quelques mesures électriques qui sont un peu loin par rapport à l'échantillon d'apprentissage qui a servi pour tester ce wafer, son profil reste différent des profils des wafers anormaux dans ce jeu de données (comme le profil du wafer QXXXXXX_15 par exemple). De plus, ces mesures un peu écartées n'étaient pas la cause pour laquelle les ingénieurs PT l'ont considéré comme anormal. Selon eux, ce wafer a été arrêté car ses trois paramètres électriques de fiabilité "VBD_AREA_NP_2", "VBD_AREA_PN_1" et "VBD_AREA_PN_2" ont chacun un ou deux sites de mesure qui s'écartent un peu des limites de spécification. De plus, les limites de spécification de ces paramètres sont serrées, et par conséquent nous avons des difficultés à identifier avec notre distance normalisée ces trois paramètres comme responsables de l'anomalie du wafer.

Dataset 2

Dans cette section, nous donnons les signatures multidimensionnelles des trois wafers anormaux détectés par nos systèmes.

Les Figures 5.10 et 5.11 montrent les signatures multidimensionnelles du wafer anormal QXXXXXX_09, obtenues respectivement par la méthode MAD$_e$-FS et la méthode 1-SVM-RFE . Avec les deux méthodes de sélection de variables, trois groupements de paramètres ont déclenché l'anomalie de ce wafer. Un premier groupement (le cercle en bas à gauche) caractérise ce wafer par des valeurs très basses du paramètre de qualité "NGA0205SF175" par rapport à l'échantillon d'apprentissage, ce paramètre qui a déclenché en plusieurs sites représente le gain du transistor. Nous soulignons les mêmes observations sur le deuxième groupement (le cercle en bas à droite) avec le paramètre de qualité "NIOFF03028TG14" mesurant le courant résiduel du transistor fermé. Le troixième groupement (cercle en haut) concerne les deux paramètre qualité clé "NVT02044RST14" et le paramètre de qualité "NVT0205RD14" mesurant la tension de seuil de deux transistors, caractérisés chacun par une longueur et une largeur bien spécifique. Les valeurs de ces deux paramètres, chacun sur plusieurs sites, ont été extrêmement élevées par rapport à l'échantillon d'apprentissage.

Un paramètre électrique supplémentaire par rapport au trois que nous venons de voir a été responsable de l'anomalie du wafer QXXXXXX_20. Les signatures de ce wafer sont tracées dans les Figures 5.12 et 5.13. Ce paramètre supplémentaire (le cercle en haut à droite) est appelé "NIDS0205RD14", c'est un paramètre de qualité qui mesure le courant de saturation d'un transistor d'une longueur et une largeur bien spécifique. Les sites de

mesure de ce paramètre qui ont déclenché avaient des valeurs plus élevées (deux sites avec MAD_e.FS et un site avec 1-SVM-RFE) par rapport à la population normale.

Concernant le wafer anormal QXXXXXX_02, nous retrouvons toujours les trois groupements de paramètres identifiés dans les trois wafers précédents, mais avec moins de sites de mesure qui ont déclenché pour chaque paramètre. D'ailleurs, les signatures de ce wafer tracées dans les Figures 5.14 et 5.15 montrent deux nouveaux paramètres qui ont contribué à la détection de ce wafer. Le premier (en haut à droite) est appelé "NVT0205SF175" et ce paramètre de qualité, comme les paramètres "NVT02044RST14" et "NVT0205RD14", mesure la tension du seuil d'un transistor possédant une longueur et une largeur bien spécifique. Le deuxième (en bas à droite) est le paramètre de fiabilité "VBD_AREA_NP_2" mesurant la tension de claquage d'une structure de test qui sert à effectuer les mesures électriques sur le wafer. Ces deux paramètres ont déclenché en un seul site de mesure en utilisant la méthode MAD_e.FS, et en plusieurs sites avec la méthode 1-SVM-RFE. Le paramètre "NVT0205SF175" (resp. "VBD_AREA_NP_2") avait des valeurs très élevées (resp. très basses) par rapport à la base d'apprentissage. Le paramètre "VBD_AREA_NP_2" a déclenché également dans le dataset 1 pour des valeurs très basses (cf. Figures 5.14 et 5.15).

Les Figures 5.16 et 5.18 montrent les signatures du wafer QXXXXXX_07. Aucun de nos systèmes de détection n'a pu détecté ce wafer puisque le taux de détection maximale obtenu dans le Tableau 5.7 a été égal à 94.12% (l'équivalent de 16 wafers anormaux détectés parmi 17). Les signatures de ce wafer ont été comparées à celles du wafer normal QXXXXXX_15 (cf. Figures 5.17 et 5.19) appartenant à la base d'apprentissage qui a servi pour tester le wafer QXXXXXX_07. Pour le wafer QXXXXXX_07, et avec les deux méthodes de sélection de variables, les paramètres électriques identifiés comme responsables de la détection ont été un peu plus éloignés par rapport à ceux du wafer QXXXXXX_15. Cependant, le profil du wafer non détecté reste loin des profils des wafers anormaux et il ressemble donc à un wafer normal. Les raisons d'arrêt de ce wafer par les ingénieurs PT reviennent, comme pour le wafer QXXXXXX_07 du dataset 1, à deux paramètres de fiabilité ("VBD_PF_NP_2" et "VBD_PF_PN_1") avec respectivement un et deux sites de mesure qui ne sont pas dans les limites de spécification.

Conclusion

Les résultats de cette section ont montré une bonne efficacité de notre méthode à identifier les paramètres électriques responsables de l'anomalie.

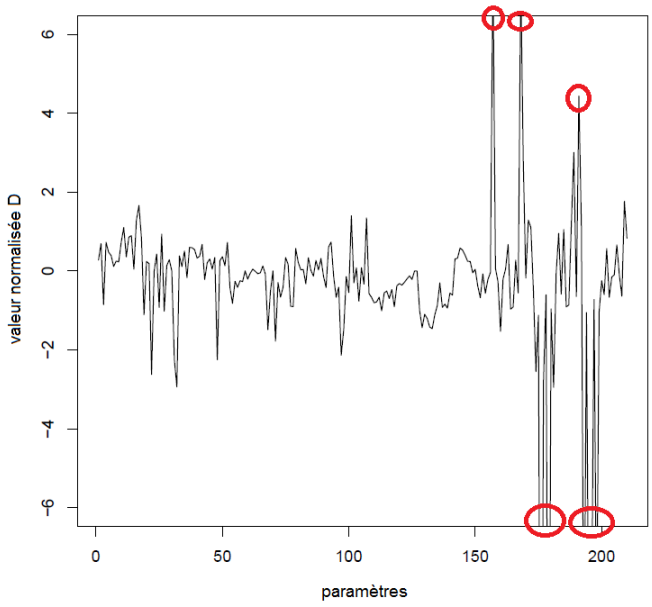

FIGURE 5.6 – Dataset 1 : signature multidimensionnelle du wafer anormal QXXXXXX_15 déterminée avec la méthode MAD_e-FS.

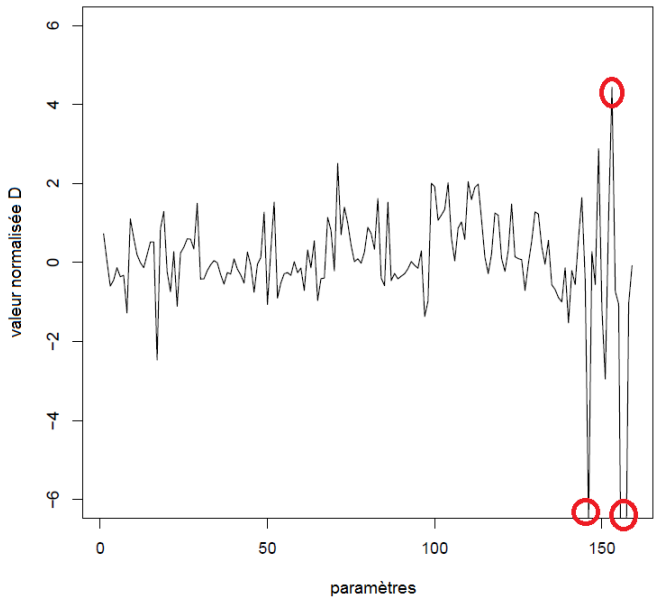

FIGURE 5.7 – Dataset 1 : signature multidimensionnelle du wafer anormal QXXXXXX_15 déterminée avec la méthode 1-SVM-RFE.

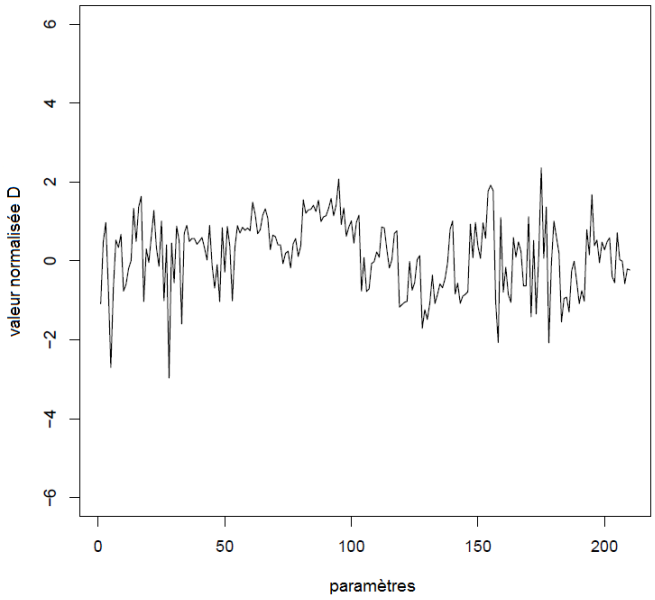

FIGURE 5.8 – Dataset 1 : signature multidimensionnelle du wafer anormal QXXXXXX_03 non détecté par nos systèmes, déterminée avec la méthode MAD_e-FS.

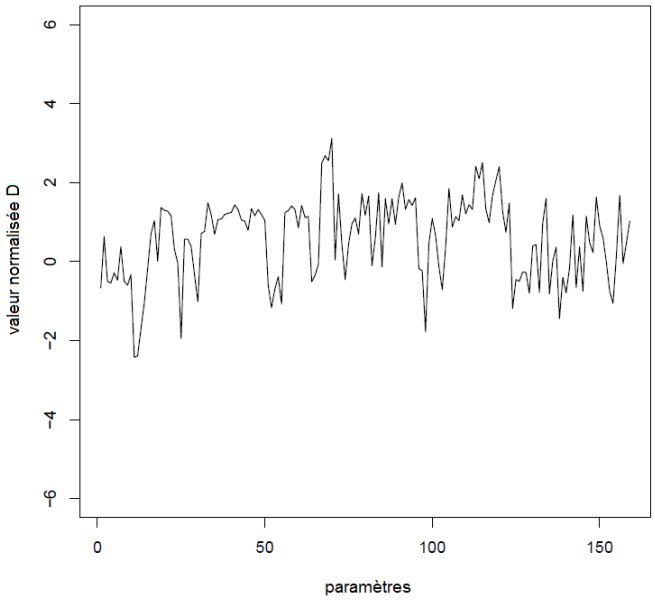

FIGURE 5.9 – Dataset 1 : signature multidimensionnelle du wafer anormal QXXXXXX_03 non détecté par nos systèmes, déterminée avec la méthode 1-SVM-RFE.

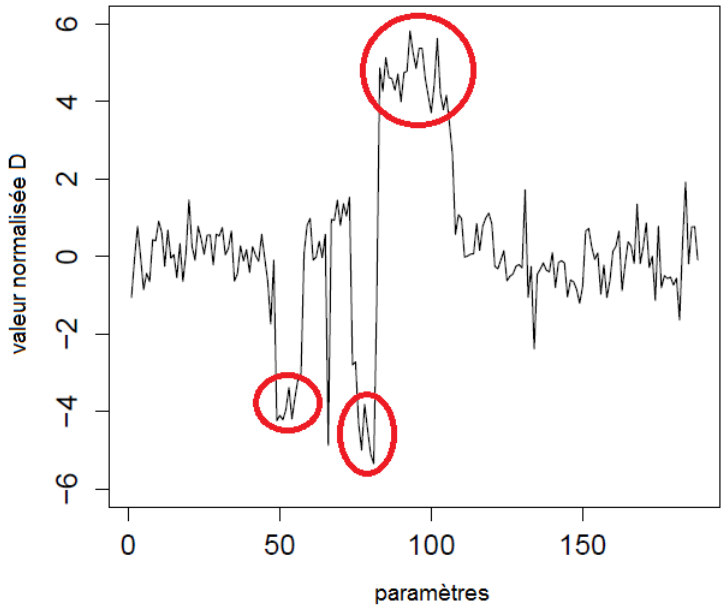

FIGURE 5.10 – Dataset 2 : signature multidimensionnelle du wafer anormal QXXXXXX_09 déterminée avec la méthode MAD_e-FS.

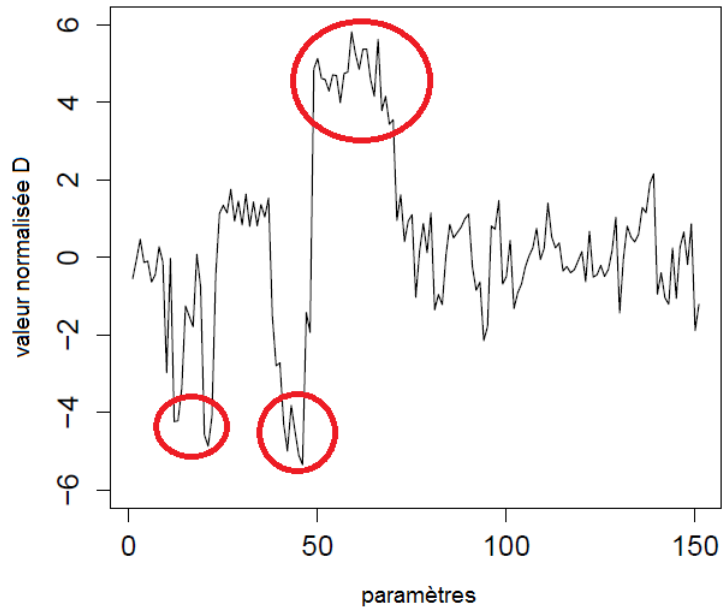

FIGURE 5.11 – Dataset 2 : signature multidimensionnelle du wafer anormal QXXXXXX_09 déterminée avec la méthode 1-SVM-RFE.

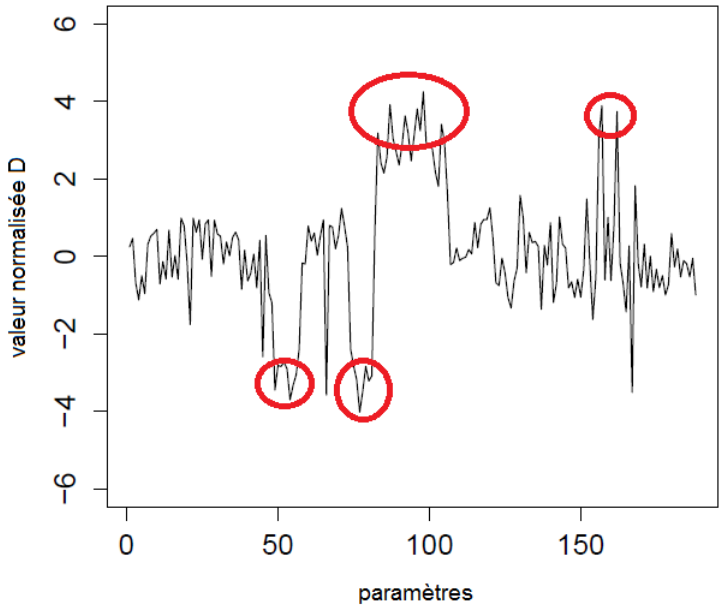

FIGURE 5.12 – Dataset 2 : signature multidimensionnelle du wafer anormal QXXXXXX_20 déterminée avec la méthode MAD_e-FS.

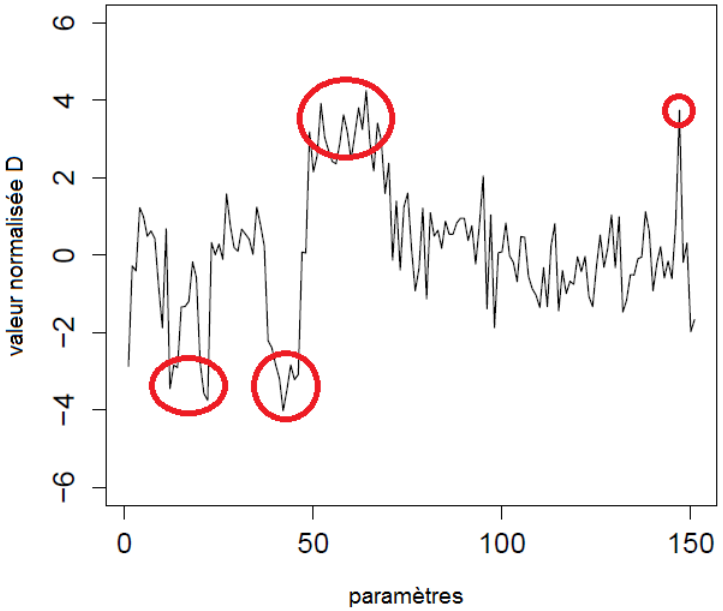

FIGURE 5.13 – Dataset 2 : signature multidimensionnelle du wafer anormal QXXXXXX_20 déterminée avec la méthode 1-SVM-RFE.

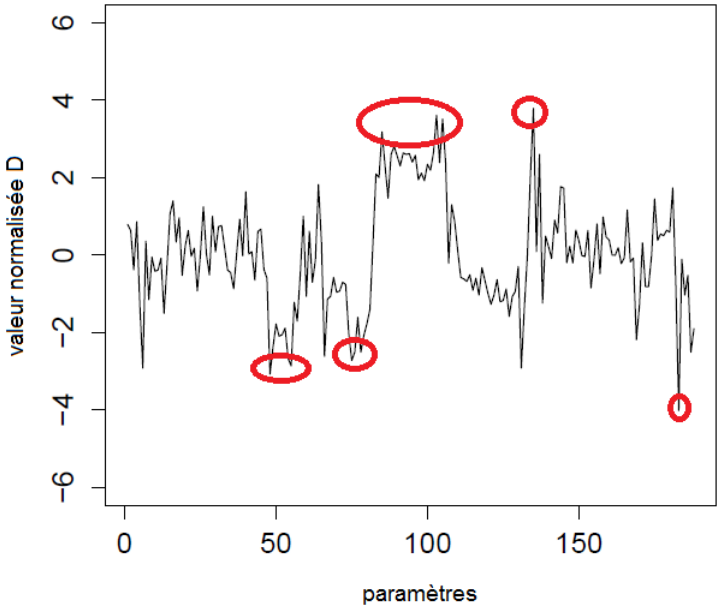

FIGURE 5.14 – Dataset 2 : signature multidimensionnelle du wafer anormal QXXXXXX_02 déterminée avec la méthode MAD_e-FS.

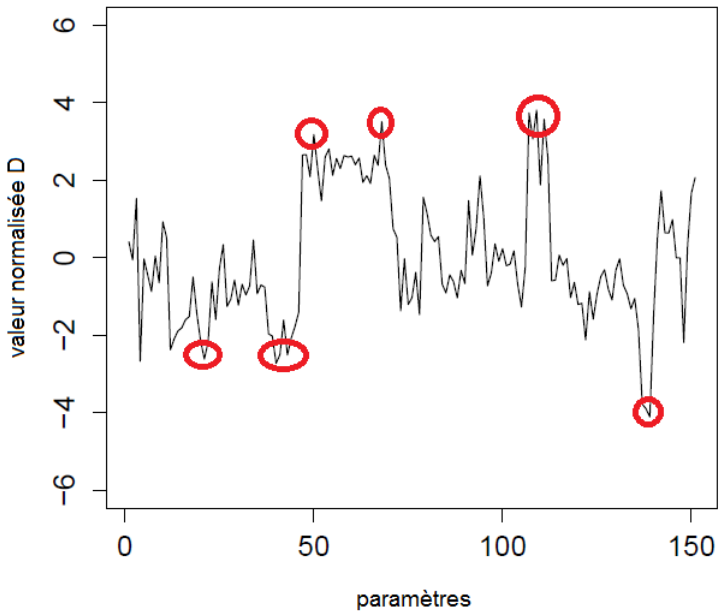

FIGURE 5.15 – Dataset 2 : signature multidimensionnelle du wafer anormal QXXXXXX_02 déterminée avec la méthode 1-SVM-RFE.

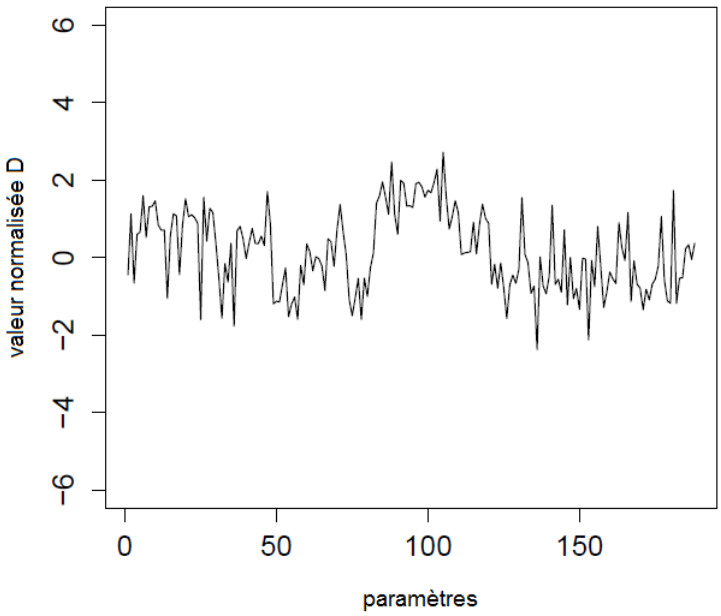

FIGURE 5.16 – Dataset 2 : signature multidimensionnelle du wafer anormal QXXXXXX_07 non détecté par nos systèmes, déterminée avec la méthode MAD_e-FS.

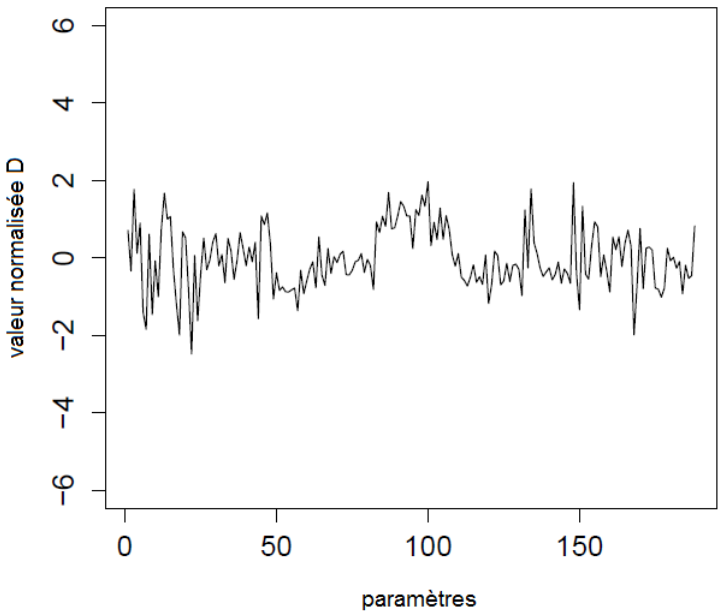

FIGURE 5.17 – Dataset 2 : signature multidimensionnelle du wafer normal QXXXXXX_15, déterminée avec la méthode MAD_e-FS.

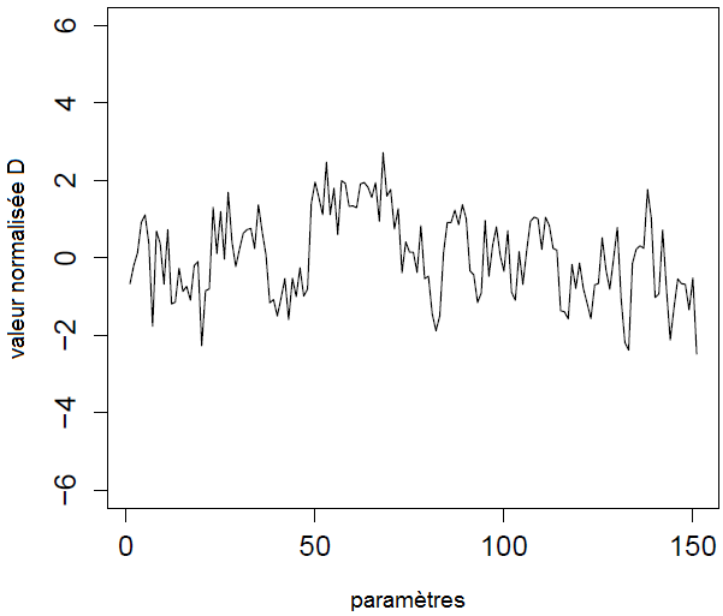

FIGURE 5.18 – Dataset 2 : signature multidimensionnelle du wafer anormal QXXXXXX_07 non détecté par nos systèmes, déterminée avec la méthode 1-SVM-RFE.

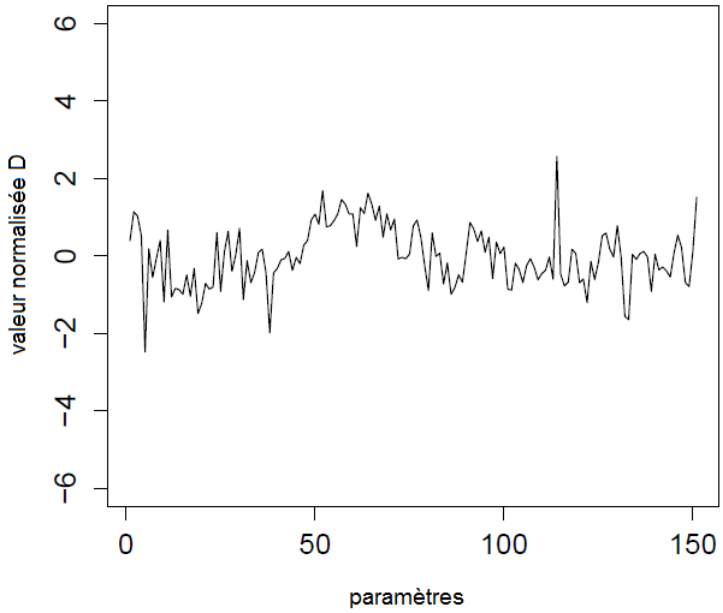

FIGURE 5.19 – Dataset 2 : signature multidimensionnelle du wafer normal QXXXXXX_15, déterminée avec la méthode 1-SVM-RFE.

Conclusion générale et perspectives

Dans ce travail, nous avons considéré le problème de la détection automatique des wafers anormaux dans le domaine de semi-conducteur. Nous avons développé un système de détection en temps réel basé sur une technique de réapprentissage dynamique. Cette technique exploite l'algorithme de classification à une classe 1-SVM avec l'une de deux méthodes de sélection de variables spécifiquement développées : une méthode de type filtrage basée sur un score indépendant de 1-SVM et calculé avec le filtre univarié MAD_e, et une autre méthode de type wrapper appropriée à 1-SVM et basée sur un score calculé à partir de cet algorithme. Le réapprentissage dynamique est assuré par une mise á jour des modèles de détection à travers une fenêtre mobile suivant deux scénarios proposés.

L'efficience de notre système de détection a été démontrée sur des données réels de STMicroelectronics. Il a été comparé également à un système de détection alternatif basé sur une compression de variables avec la méthode ACP et la statistique de Hotelling. Notre systèmes a montré une haute performance avec des taux de détection très élevés et des taux de fausses alarmes extrêmement réduits par rapport à l'approche univariée existante. De plus, notre système était plus performants que le système alternatif.

Après avoir détecté les wafers anormaux, nous avons développé une méthode permettant de visualiser les signatures multidimensionnelles de ces wafers. Cette méthode exploitant nos méthodes de sélection de variables ont permis aussi d'identifier efficacement les paramètres électriques pertinents et responsables de l'anomalie.

Une collaboration est actuellement en cours avec l'équipe EDA (Engineering Data Analysis) du site de Crolles pour développer un software industriel permettant de mettre en oeuvre notre système de détection en temps réel.

Perspectives

Techniquement, nous pourrons étudier la possibilité d'exploiter une méthode de sélection de variables de type "embedded" appropriée à la méthode 1-SVM. Ceci pourra être

réalisé en ajoutant une pénalité à la fonction objective du problème d'optimisation de l'algorithme 1-SVM. Cette approche, combinant à la fois l'apprentissage du modèle 1-SVM et la sélection de variables, pourra éventuellement améliorer encore la performance de notre système de détection en temps réel.

Pratiquement, nous pourrons valider les fausses alarmes de notre système par l'analyse au test final EWS, du rendement des wafers considérés comme anormaux par notre système. Il se peut que certains wafers soient atypiques qui ne ressemblent pas à la population étudiée, mais qui sont dans les limites de spécification. Ce qui expliquerait le fait que ces wafers soient considérés comme anormaux par notre système et comme normaux par les ingénieurs PT. Le test EWS est un test approfondi qui s'applique sur l'ensemble des puces, et le rendement d'un wafer (pourcentage de puces en bon état de fonctionnement) pourrait nous aider à vérifier s'il s'agissait vraiment de fausses alarmes ou non.

Contrairement à l'approche de détection univariée existante basée sur les limites de spécification n'étant définis que pour les paramètres "Gated", nous pouvons étendre notre approche de détection à tous les autres paramètres électriques mesurés.

Par ailleurs, la détection multidimensionnelle peut avoir beaucoup d'autres applications possibles en semi-conducteur. Parmi elles, deux applications majeures pourront être étudiées respectivement en *SPC* classique (Statistical Process Control ou MSP pour Maîtrise Statistique du Procédé) et en *FDC* (Faul Detection and Classification).

En effet, notre approche de détection peut être utilisée en MSP classique pour la détection basée sur les mesures de paramètres physiques en ligne de production. La MSP classique est basée sur un suivi de résumé statistique (moyenne, écart-type, étendu) venant de données individuelles mesurées sur les composants pouvant aller de 17 valeurs individuelles à plusieurs centaines. Le résumé statistique réduit la dimension de p données individuelles à 2-3 dimensions et nous amène à un perte d'information importante et réduit la pertinence de la détection.

Notre approche de détection peut aussi être utilisée en FDC pour la détection basée sur les paramètres machines. Pour la FDC, on dispose de plusieurs dizaines voire centaines de paramètres machine collectés pendant chaque process d'un wafer avec une problématique de détection similaire au test paramétrique. Cette détection multidimensionnelle permettra de simplifier l'approche en réduisant le nombre d'indicateurs de suivi à un seul, d'améliorer la pertinence de la détection et de réduire les fausses alarmes statistiques comparée à une approche unidimensionnelle.

International Journal of Computer Theory and Engineering, Vol. 7, No. 3, June 2015

Real-Time Fault Detection in Semiconductor Using One-Class Support Vector Machines

Ali Hajj Hassan, Sophie Lambert-Lacroix, and Francois Pasqualini

***Abstract*—In this paper, we propose a real-time fault detection system for the semiconductor domain, which aims to detect abnormal wafers from a recent history of electrical measurements. It is based on a dynamic model which uses our filter method as feature selection approach, and one-class support vector machines algorithm for classification task. The dynamicity of the model is ensured by updating the database through a temporal moving window. Two scenarios for updating the moving window are proposed. In order to prove the efficiency of our system, we compare it to an alternative detection system based on the Hotelling's T^2 test. Experiments are conducted on two real-world semiconductor datasets. Results show that our system outperforms the alternative system, and can provide an efficient way for real-time fault detection.**

***Index Terms*—Real-time detection, feature selection, one-class support vector machines, semiconductor.**

I. Introduction

Nowadays, the control of manufacturing processes is an essential task to ensure consistently safe operation and high quality production. This is challenging particularly when processes have a large number of operations and complex systems, which is the case in manufacturing process of semiconductor devices and integrated circuits. Early and accurate detection of faults is then required for maintaining a process at its optimal condition, and reducing manufacturing costs.

Once the manufacturing process of semiconductor ends, an electrical test, called Parametric Test (PT), is performed. PT aims to detect within shortest possible time the abnormal wafers (semiconductor material used in manufacturing of semiconductor devices) by looking at a set of static electrical parameters measured on multiple sites of each wafer.

The purpose of this work is to implement an automatic real-time detection system at PT level. Based on a multivariate statistical approach, this system aims to detect abnormal wafers through a moving temporal window of electrical measurements.

Multivariate statistical approaches have been successfully used for monitoring industrial processes [1]–[3]. Principal Component Analysis (PCA) was considered to develop respectively a static (off-line testing) and dynamic (in-line testing) models for fault detection in biological Wastewater Treatment Plant (WWTP) [4], [5]. PCA was also considered in [3] to detect faults in a semiconductor etch process. PCA is one of the most widely multivariate techniques used for extracting relevant information from high dimensional data. The goal of PCA is to reduce the dimensionality of the original data by projecting them into a lower dimensionality space without a significant loss of information. This can be done by identifying the directions that explain the maximum variation of the data. The PCA method captures the variability of a process by monitoring the T^2 metric on the new PCA components or by monitoring the residuals (Q chart) of the PCA model [4]. In case of non-linear processes, kernel principal component analysis (KPCA) was used to handle non-linearity with the help of kernel functions [6].

Another multivariate method based on statistical learning approaches is the one-class Support Vector Machines (1-SVM) [7], which is a variant of the original Support Vector Machine (SVM) algorithm [8]. 1-SVM is a useful and popular tool used for anomaly detection. A static model based on 1-SVM method and the SVM-recursive feature elimination algorithm (RFE-SVM) [9] was used in [10] for fault detection in a semiconductor etch process, and in chemical engineering simulation problem. It has been shown that 1-SVM method is an efficient method for fault detection in both domains. Moreover, the 1-SVM technique performed better than PCA. Even in non-linear cases, simulation experiments showed that 1-SVM technique outperformed the KPCA method.

However the SVM-RFE algorithm requires a huge computational time since the number of SVM models to be trained is $O(p^2)$, where p is the dimension of variable space. In our study, the variable space is characterized by several electrical parameters (hundreds or thousands). High dimensional variable space restricts the use of the SVM-RFE algorithm. Moreover, as part of the training stage at each iteration of a real-time application, this algorithm would not be computationally useful, especially when we use a short temporal moving window to update the detection algorithm.

To overcome this problem, we have developed in [11] a new filter technique selecting the most relevant features (electrical parameters). This technique is based on the Median Absolute Deviation method denoted by MADe [12], a robust approach for detecting univariate outliers. The key idea is to use the MADe method to determine the percentage of outlier in each parameter. Then parameters with a percentage of outliers exceeding a predefined threshold will be potential discriminative features. We denote this method by MADe-FS (MADe for Feature Selection).

The remainder of the paper is structured as follows. First, our main contributions in this work are mentioned in the Section II. In Section III we recall the one-class support vector machine method. Then, our filter method MADe-FS which selects the most informative parameters is also

Manuscript received February 24, 2014; revised May 15, 2014.

Hajj Hassan and S. Lambert-Lacroix are with the TIMC laboratory, Grenoble University, Grenoble, 38041 France (e-mail: {Ali.Hajj-Hassan, Sophie.Lambert}@imag.fr).

F. Pasqualini is with Process Control team, STMicroelectronics, Crolles, 38926 France (e-mail: Francois.Pasqualini@st.com).

DOI: 10.7763/IJCTE.2015.V7.955

International Journal of Computer Theory and Engineering, Vol. 7, No. 3, June 2015

recalled in Section IV. Section V describes our real-time detection system according to two proposed scenarios for updating moving window. A short description of Hotelling's T^2 test which is the basis of an alternative detection system is given in Section VI. Before concluding, Section VII serves as an application of our system on a two real-world semiconductor datasets.

II. Main Contributions

In our work [11], we have considered the problem of detecting abnormal wafers in semiconductor using electrical measurements. We have developed a static model for fault detection based on 1-SVM method for anomaly detection and our filter method MADe-FS for selecting the most relevant electrical parameters.

In this work, we consider the problem of real-time fault detection, becoming increasingly important in semiconductor domain. We develop a dynamic model which shares the same approaches of classification and feature selection as in our static model. Our dynamic model consists of updating the MADe-FS method and the 1-SVM algorithm at each update of the moving temporal window. We propose two scenarios for updating this window, and we explain our technique used to optimize the initial choice of model parameters and their updating strategy.

As an alternative system of real-time detection, we implement a similar dynamic model based on PCA method to reduce dimension and model the normal behavior, and Hotelling's T^2 statistic as multivariate control chart. Parameters of this model is selected and updated under the same strategy used in our developed dynamic model.

At our knowledge, this work is the first one to implement a real-time fault detection system in semiconductor domain, and at the same time the first one to develop a dynamic model based on the 1-SVM method. This model is applied on high dimensional data consisting of hundreds of variables while previous works on fault detection in industrial processes considered data with tens of variables.

III. One-Class Support Vectors Machines

Support Vector Machine (SVM) [13] is as an effective learning algorithm for binary classification. This algorithm aims to find an optimal hyperplane to separate the two classes of training data.

An extension of SVM, called one-class SVM (1-SVM), was subsequently proposed in [7] to handle one-class classification problem. The 1-SVM strategy is to find an optimal hyperplane in a feature space separating the training data (positive samples) from the origin (considered as negative samples) with maximum margin (the distance from the hyperplane to the origin).

Given a training dataset of n positive samples (normal wafers) $\{x_1,\ldots,x_n\}$ where each $x_i \in R^p$ is described by a vector of p features (electrical parameters). Each x_i is first transformed via a feature map φ: R^p --> F where F is a high (possibly infinite) dimensional Hilbert space generated by a positive-definite kernel K. The kernel function corresponds to an inner product in the feature space F through $K(x, x') = \varphi(x) \cdot \varphi(x')$.

The 1-SVM algorithm finds in the feature space a hyperplane H $\{z \in F;\ w \cdot z = \rho\}$ that separates the cluster of normal samples from the origin. $w \in F$ is the normal vector defining H. The margin is equal to $\rho/\|w\|$. The one-class SVM requires solving the following quadratic optimization problem:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + \frac{1}{vn}\sum_{i=1}^{n}\xi_i - \rho$$

$$st \quad w \cdot \varphi(xi) \geq \rho - \xi_i, \xi_i \geq 0, i = 1, \ldots, n. \quad (1)$$

ξ_i 's are slack variables introduced to allow misclassification for some points, and $v \in [0, 1]$ is a free parameter controlling the impact of the slack variables, i.e. the fraction of training data which are allowed to fall wihtin the margin. In fact, it can be shown that v is an upper bound on the fraction of training errors [7].

The dual problem, to be maximized, is given by:

$$\min_{\alpha} \frac{1}{2}\sum_{ij} \alpha_i\, \alpha_j\, K(x_i, x_j)$$

$$st\ 0 \leq \alpha_i \leq \frac{1}{vn}, \sum_i \alpha_i = 1. \quad (2)$$

The data x_i with non-zero α_i are the so-called support vectors. They are the training data that determine the separating hyperplane. It can also be shown that v lower bounds the fraction of support vectors [7].Once the optimal values of the parameters are found, one can classify the new data (new wafers) according to the decision function

$$g(x) = sgn\left(\sum_{i \in sv} \alpha_i\, K(x_i, x) - \rho\right), \quad (3)$$

where sv is the set of the support vectors' indices.

In practice, the 1-SVM has been successfully applied with the RBF kernel $K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|2)$ where γ is a parameter that controls the width of the kernel function. After many experiments in which we have tested many values for γ ($\gamma = 1/mp$, with $m \in \{0.5, 1, 2, 3, 4, 5\}$), results have showed that best performance of 1-SVM algorithm is obtained for $m > 1$, and this performance is not very sensitive to the kernel parameter. Hence fine-tuning of the parameter is not required. We set $\gamma = 1/5p$.

IV. Our Filter Method

In machine learning and statistics, feature selection is the process of selecting an optimal subset of relevant features in order to obtain good classification performances.

To achieve the task of feature selection, we use our filter approach based on MADe method, which is a robust univariate outlier detection method. Before presenting this method, we first introduce the Maximum Absolute Deviation (MAD) [14] of a variable $x_j \in R^n$ ($j = 1,\ldots, p$):

$$\text{MAD}(j) = \text{median}_{i \in [1,n]}\left(\left|x_{ij} - \text{median}\left(x^j\right)\right|\right). \quad (4)$$

MAD is a robust estimator of the spread in a data, similar to the standard deviation. When the MAD value is scaled by a factor of 1.483, it represents a consistent estimator of the standard deviation in a normal distribution [12]. This scaled

MAD value is the MAD_e:

$$\mathrm{MAD}_e = 1.483 \times \mathrm{MAD}. \quad (5)$$

The MAD_e method is defined as follows:

$$LL_j = \mathrm{Median}(j) - 3 \times \mathrm{MAD}_e(j)$$
$$UL_j = \mathrm{Median}(j) + 3 \times \mathrm{MAD}_e(j), \quad (6)$$

where LL_j and UL_j are respectively the lower and upper limits for the variable j.

The MADe approach is similar to the Standard Deviation (SD) method that considers the observations outside the interval $[\bar{x}\pm 3\sigma]$ as outliers, where $\bar{x}$ and σ are respectively the empirical mean and standard deviation for a univariate samples. However, the median and MAD_e are employed instead of the mean and the standard deviation. Since this approach uses two robust estimators, it is largely unaffected by the presence of extreme values in the data set.

Thus the percentage of outliers OOL_j (Out Of Limit) of the variable x_j represents the proportion of data outside the interval determined by the lower and upper limits of the MADe method. Therefore we have:

$$OOL_j = \frac{\#\, i \in [LL_j, UL_j]}{n} \quad (7)$$

Finally, the Subset of Relevant Variables (SRV) contains variables for which the percentage of outliers exceeds the threshold θ_q (cf. Eq. 8). θ_q is defined as the quantile of order q of the values in the vector OOL=(OOL_1,…, OOLp).

$$\mathrm{SRV} = \{x^j, \mathrm{OOL}_j > \theta_q\} \quad (8)$$

To conclude, our filter method consider the top $100(1-q)$ % outlying variables as the most relevant electrical parameters for the classification task.

V. Real Time Detection System

The motivation behind the development of a real-time detection system is to use the MADe-FS and 1-SVM approaches for in-line testing in the context of industrial application. This system aims to detect in real-time abnormal wafers using a recent history of electrical measurements. In the following, we denote our model of feature selection and classification by 1-SVM.FS (one-class SVM with Feature selection). This model consists of determining firstly the most relevant features in the training data using our filter method MADe-FS, and secondly applying the 1-SVM algorithm on the subset of relevant features.

Our detection system is based on three major steps:

1) Selection of a correct performance reference data set, representing the normal operating behavior
2) Real-time data updating through a moving window, to obtain a real-time procedure.
3) 1-SVM.FS application to the updated real-time data.

So we first define the reference correct performance dataset, representing a well-behaved operating condition. For this, we select from the historical database of considered products, a set of operational positive samples (normal wafers) corresponding to a nominal condition of processes. Concerning the reference data size, a large data set increases the detection reliability. Hence reference data must be large enough allowing us to define a normal region which encompasses a wide variety of positive samples.

The correct performance dataset is used as a training set to build a model describing the normal behavior of the process. When a new lot (group of 25 wafers that run together all processing steps) arrives, the 1-SVM.FS model trained on the correct performance dataset is used to test whether each of 25 wafers is normal or abnormal. The tested lot will join the initial training set while oldest lot in this set will be removed or maintained depending on the used scenarios explained below. Thus a new training set is formed. 1-SVM.FS model is retrained on the updated training set and will be used to predict the operating state of the next 25 new wafers. 1-SVM.FS model is retrained on the updated training set and will be used to predict the operating state of the next 25 new wafers. 1-SVM.FS model is retrained on the updated training set and will be used to predict the operating state of the next 25 new wafers. This procedure is repeated with the arrival of each new lot. A general view of our detection system is presented in the Fig. 1.

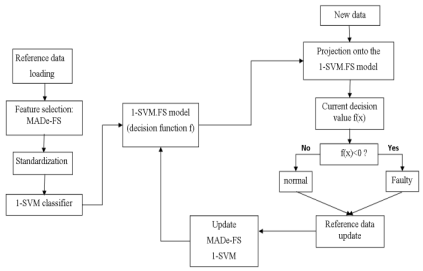

Fig. 1. Schema of our real-time detection system based on 1-SVM.FS dynamic model.

As we described before, the basic 1-SVM.FS model was made dynamic by updating the database through a moving window. We consider 2 scenarios reflecting two updating modes of the moving window:

- Scenario 1 (increased length): with this scenario, the tested lot at each iteration is added to the existing training set without removing old data. So 1-SVM.FS model is updated according to a moving window of increased length. Since normal behavior keeps evolving, we have decided to remove at once some old data from the increased training set after a defined period Δt. Δt depends on the volume production of the considered product(s).
- Scenario 2 (fixed length): during the real-time operation, the window still maintains the length of the correct performance dataset and operates as a First-In-First-Out (FIFO) shift-register, discarding old data and including new ones.

The two scenarios are illustrated respectively in the Fig. 2 and the Fig. 3.

Recall that 1-SVM.FS model requires setting the parameter ν (the threshold in 1-SVM algorithm) and two hyperparameters: the order q of the threshold θ_q in feature selection method, and the kernel parameter γ. Consequently some kind of model selection (parameter search) must be done.

To accomplish the model selection task, a validation set containing normal data contaminated by some abnormal wafers is needed. It is used to identify good (q, v) so that the classifier can accurately predict unknown data (i.e. testing data). A "grid search" on q and v is performed. 1-SVM.FS model is built on training set using various pairs of (q, v) values. For each pair, samples from validation set are projected onto the trained 1-SVM.FS model. Then Detection Rate and the False Alarms Rate (cf. Section VII-A) are computed. The pair that optimizes these two performance measures is picked. More precisely, the best pair (q, v) is the one giving the optimal compromise between maximizing the Detection Rate and minimizing the False Alarms Rate. The selected pair is used at each update of the 1-SVM.FS model.

Fig. 2. Real-time moving window using scenario 1.

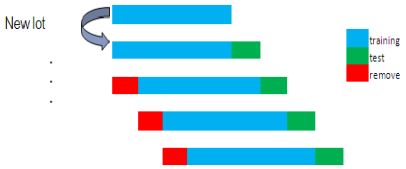

Fig. 3. Real-time moving window using scenario 2.

VI. Hotelling's T^2 Test

To make our study comparable to previous studies, we have investigated the Hotelling's T^2 test. Hotelling's T^2 statistic provides an indication of novel variability within the model space.

The principle of this test is to use PCA method to model the behavior of the normal samples. Anomalies are then detected by comparing the behavior observed with that given by the PCA model. Having established a PCA model of the positive training data, testing data are projected onto this model, and Hotelling's T^2 statistic can be computed based on the first k principal components of the model. The T^2 statistic for a sample x_i is:

$$T_i^2 = t_i^T \wedge^{-1} t_i = x_i^T p_k \wedge^{-1} p_k^T x_i \quad (9)$$

where $t_i = P_k^T xi$ is the orthogonal projection of the data xi into the model subspace defined by the k first principal components, and Λ is a diagonal matrix containing the first k eigenvalues of the covariance matrix of the positive training data.

A threshold T^2_α can be obtained using the Fisher distribution. If $T^2_i > T^2_\alpha$, the sample is categorized as abnormal, and normal otherwise. For further details on fault detection based on PCA readers are advised to read the literature [4].

To choose k, we use the Cumulative Proportion of Variance (PCV):

$$\mathrm{PCV}(k) = 100 \times \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{p} \lambda_j},$$

where $\lambda 1, \ldots, \lambda p$ are the eigenvalues sorted in descending order. Thus we retain the first k components that account for a predefined percentage of the variance in the data:

$$k = \arg\min_u \{\mathrm{PCV}(u) \geq \beta\}.$$

For example, if we set β=0.8 we retain the minimal number of components that preserve 80% of the information in the data.

Detection system based Hotelling's T^2 test is dynamically the same as our system. The data and model update is performed at the level of 25 wafers (each new lot) following the proposed two scenarios.

VII. Application

Our experimental goal was to assess the ability of our detection system to detect abnormal wafers. It is also important to minimize false alarms rate as they cause unwarranted interruption in plant operation. Let us first introduce the performance measures used in our study.

TABLE I: Confusion Matrix of Metrics Used in Performance Measures

True class vs Decision	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

A. *Performance Measures and Data*

In order to evaluate and compare the results obtained from the different methods, we used two performance criteria: Detection Rate (DR) and False Alarms Rate (FAR). Detection Rate quantifies the percentage of data predicted to be negative by the classifier that are actually negative; False Alarms Rate quantifies the percentage of data predicted to be negative by the classifier that are actually positive. These two measures are computed using the four metrics described in the Table I as follows:

$$DR = \frac{TN}{TN + FN}$$

$$FAR = \frac{FP}{FP + TP}$$

We notice that the resulting false alarms rate in the context of application of real-time detection system over a production period represents the average of false alarms rates obtained when testing separately each of all lots that have to be tested. Furthermore, the FAR-DR curve is suitable for evaluating classifiers by integrating their performance over a range of decision thresholds. It depicts the relation between DR (x-axis) and FAR (y-axis) varying the range of thresholds. The lower the misclassification error of a classifier, the closer the corresponding point is to the upper right-hand corner of the ROC curve.

International Journal of Computer Theory and Engineering, Vol. 7, No. 3, June 2015

The real-time detection system proposed in this paper has been tested on two real-world industrial datasets. Each dataset consists of wafers corresponding to one or more products of a certain technology over months of production. Each wafer is described by a certain number of electrical parameters. We give the percentage of abnormal wafers in each dataset. The description of these two datasets is given in Table II. 1-SVM.FS and Hotelling's T^2 detection systems are investigated under the two scenarios in both datasets, in order to prove again the efficacy and superiority of our detection system. Ideally, we want high DR (to detect most of the abnormal wafers) and a low false alarms rate (to avoid mistakenly classifying normal wafers as abnormal).

TABLE II: DESCRIPTION OF THE REAL WORLD INDUSTRIAL DATA USED IN OUR STUDY

Data	Production time	Nb of parameters	% of abnormal wafers
dataset 1	2 months	756	1.75
dataset 2	4 months	1062	0.5

1) Dataset 1

In this experiment, the correct performance data is formed using 300 normal wafers. The validation set consists of 100 wafers of which 6 wafers are abnormal. We have trained our 1-SVM.FS model on the correct performance data using various pairs of values for the feature selection hyperparameter q and the threshold v. We consider respectively 6 and 20 values for q and v, as follows:

$$q\epsilon\{0.25, 0.4, 0.5, 0.6, 0.75, 0.9\},$$

$$v\epsilon\{0.01, 0.02,\ldots, 0.19, 0.2\}.$$

Samples from the validation set are then predicted using each of 120 learned models. The Detection Rate and the False Alarms Rate are computed for each prediction. We have selected the pair that optimizes simultaneously these two performance measures. Here we have retained q=0.75 and v=0.16 and we have obtained a DR equal to 100% and FAR equal to 14.21%.

Similarly, we have selected for the Hotelling's T^2 test the best pair (β, α) (cf. Section VI) by taking β∈ {0.75, 0.8, 0.85, 0.9} and considering the same range of values of ν for α. The optimal performance is obtained for β=0.75 and α=0.2, where DR and FAR are respectively equal to 66.67% (4 among 6 abnormal wafers) and 17.36%.

After defining the correct performance data set and selecting the optimal parameters for 1-SVM.FS and Hotelling's T^2 models, we now proceed to the real-time detection by applying both of models to the real-time data. The real-time data are updated at each arrival of a new lot through a moving window in order to obtain a real-time procedure. The two models are also updated. The updates through the moving window follow one of two defined scenarios: scenario 1 (increased length) and scenario 2 (fixed length).

Next, we focus on comparing the performance of the two real-time detection systems based on 1-SVM.FS and Hotelling's T^2 dynamic models using the two scenarios. The results are reported in Table III. For both scenarios, the Hotelling's T^2 system has shown poor performance in detecting abnormal wafers (DR=65.22%), while 1-SVM.FS system has been able to detect 95.65% of abnormal wafers. In addition, we have obtained lower false alarms rate using our detection system. For both detection systems, scenario 1 reduces false alarms compared to scenario 2.

TABLE III: PERFORMANCE OF 1-SVM.FS AND HOTELLING'S T^2 SYSTEMS ON THE DATASET 1

Moving window	Detection system	Detection Rate	False Alarms Rate
Scenario 1	1-SVM.FS	95.65	12.89
	Hotelling's T^2	65.22	13.43
Scenario 2	1-SVM.FS	95.65	19.25
	Hotelling's T^2	65.22	19.85

To confirm this hypothesis, FAR-DR curve is plotted in the Fig. 4 to study the behavior of our detection system regarding the two different scenarios, over the same range of ν defined above. It is clear that scenario 1 gives a significant reduction interm of false alarms compared to scenario 2. This is due to the increased size of its moving window where a new lot is added to the training database at each update. In fact one-class SVM requires many more positive training data to give an accurate decision boundary because its support vectors come only from the positive data. However scenario 2 tends to detect more quickly abnormal wafers (i.e. for any value of v, scenario 2 has higher or the same DR than scenario 1). The short fixed window in scenario 2 has a more efficient updating strategy and contains fewer abnormal wafers in the moving training dataset, which improves the performance of 1-SVM algorithm since the latter requires normal wafers to learn the classifier.

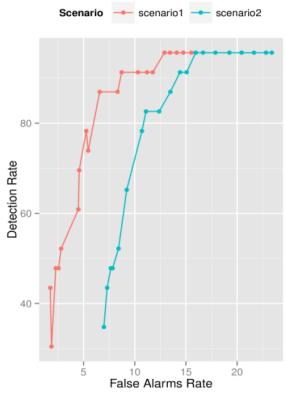

Fig. 4. FAR-DR curve comparing performances of 1-SVM.FS detection system using the two proposed scenarios.

Note that, in the first experiment considering only two months of production, we did not remove old data in the actual training set after the Δt period for the scenario 1, as has been recommended in Section V. This action takes place in the second experiment considering four months of production where we have a larger number of wafers.

A final comparison is realized between 1-SVM.FS and 1-SVM detection systems. The difference between these two systems is that the latter ignores the feature selection step used by the former. Another FAR-DR curve is plotted in the Fig. 5 illustrating this comparison. From this curve, a very important improvements achieved by applying our feature selection method MADe-FS. These improvements were

observed on each of the two performance measures (DR and FAR).

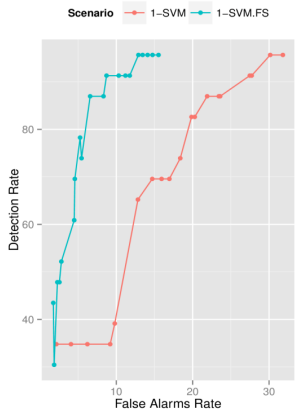

Fig. 5. FAR-DR curve showing the importance of our filter method MADe-FS to improve the performance of the 1-SVM classifier, according to scenario 1.

2) Dataset 2

Dataset 2 contains wafers from another category of products collected over four months of production. This dataset has higher dimensional space and lower percentage of abnormal wafers, compared to the first dataset. We set to 500 the size of the correct performance data. The validation set contains 2 abnormal wafers among of 100.

Following the same procedure used in dataset 1 for selecting optimal parameters, we have retained (q, ν)=(0.75, 0.04) for 1-SVM.FS model and (β, α)=(0.8, 0.01) for Hotelling's T^2 model. We set Δt to 2 months.

Table IV summarizes the performances achieved by the two systems under the two different scenarios. The results reveal a degree of similarity between the performances of both systems. High performance was obtained using both systems. Observations resulting from the comparison of two scenarios in dataset 1 are confirmed in dataset 2. Scenario 1 has lower false alarms rate, while scenario 2 detect more effectively abnormal wafers.

TABLE IV: PERFORMANCE OF 1-SVM.FS AND HOTELLING'S T^2 SYSTEMS ON THE DATASET 2

Moving window	Detection system	Detection Rate	False Alarms Rate
Scenario 1	1-SVM.FS	83.33	5.89
	Hotelling's T^2	83.33	6.34
Scenario 2	1-SVM.FS	91.67	8.62
	Hotelling's T^2	91.67	9.12

VIII. CONCLUSION

In this paper, we proposed a new real-time fault detection system based on the machine learning 1-SVM algorithm and our filter method for feature selection. A dynamic detection was realized by updating the database following two proposed scenarios. The efficacy of our system has been demonstrated using two real-world industrial datasets. For both scenarios, results from the two datasets showed that our system could detect most of the abnormal wafers with an admissible percentage of false alarms. In addition, our system outperformed the detection system based on the Hotelling's T^2 test in the dataset 1, and similar performance was obtained in dataset 2 with slightly lower rate of false alarms.

ACKNOWLEDGMENT

This study has been done within the framework of a joint collaboration of STMicroelectronics in Crolles, France, and the TIMC laboratory of the Grenoble University in Grenoble, France. The authors would like to thank the ANRT (Association Nationale de la Recherche et de la Technologie) which has partially financed this study.

REFERENCES

[1] T. Kourti, J. Lee, and J. Macgregor, "Experiences with industrial applications of projection methods for multivariate statistical process control," *Computers and Chemical Engineering*, vol. 20, pp. S745–S750, 1996.
[2] A. Cinar and C. Undey, "Statistical process and controller performance monitoring: a tutorial on current methods and future directions," in *Proc. American Control Conference*, 1999, vol. 4, pp. 2625–2630.
[3] B. Wise, N. Gallagher, S. Butler *et al.*, "A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process," *Journal of Chemometrics*, vol. 13, pp. 389–422, 1999.
[4] D. Garcia-Alvarez, M. Fuente, P. Vega, and G. Sainz, "Fault detection and diagnosis using multivariate statistical techniques in a wastewater treatment plant," in *Proc. 7th IFAC International Symposium on Advanced Control of Chemical Processes*, 2009, pp. 952–957.
[5] F. Baggiani and S. Marsili-Libelli, "Real-time fault detection and isolation in biological wastewater treatment plants," *Water science and technology*, vol. 60, no. 11, pp. 2949–2961, 2009.
[6] J. Lee, C. Yoo, S. Choi, P. Vanrolleghem, and I. Lee, "Nonlinear process monitoring using kernel principal component analysis," *Chemical engineering Science*, vol. 59, no. 1, pp. 223–234, 2004.
[7] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.
[8] V. Vapnick, "Sv machines for pattern recognition," *Statistical Learning Theory*, John Wiley Sons, pp. 496–498, 1998.
[9] A. Rakotomamonjy, "Variable selection using svm based criteria," *J. Mach. Learn. Res.*, vol. 3, pp. 1357–1370, Mar. 2003.
[10] S. Mahadevan and S. Shah, "Fault detection and diagnosis in process data using one-class support vector machines," *Journal of Process Control*, vol. 19, pp. 1627–1639, 2009.
[11] A. H. Hassan, S. Lambert-Lacroix, and F. Pasqualini, "A new approach of one class support vector machines for detecting abnormal wafers in semi-conductor," in *Proc. Fourth Meeting on Statistics and Data Mining, ser. MSDM '13,* 2013, pp. 35–41.
[12] S. Burke, "Missing values, outliers, robust statistics and non-parametric methods," *Statistics and Data Analysis*, vol. 2002, 2001.
[13] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, Jun. 1998.
[14] F. R. Hampel, "The Influence Curve and Its Role in Robust Estimation," *Journal of the American Statistical Association*, vol. 69, no. 346, pp. 383–393, 1974.

Ali Hajj Hassan is a Ph.D. student at Grenoble University, France. He works as an engineer in the Process Control Department at STMicroelectronics Crolles. He received the applied mathematics and statistics degree from University of Joseph Fourier, Grenoble, France, in 2010. His Ph.D. subject is about multivariate statistical approaches for detecting abnormal wafers in semiconductor.

Table des figures

Liste des tableaux

Bibliographie

[1] Abe, N., Zadrozny, B., and Langford, J. Outlier detection by active learning. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2006), KDD '06, ACM, pp. 504–509. 23, 29

[2] Addison, J. F. D., Wermter, S., and MacIntyre, J. Effectiveness of feature extraction in neural network architectures for novelty detection. In *9th International Conference on Artificial Neural Networks* (January 1999), vol. 2, pp. 976–981. 22

[3] Aizerman, M. A., Braverman, E. A., and Rozonoer, L. Theoretical foundations of the potential function method in pattern recognition learning. In *Automation and Remote Control,* (1964), no. 25, pp. 821–837. 49

[4] Albrecht, S., Busch, J., Kloppenburg, M., Metze, F., and Tavan, P. Generalized radial basis function networks for classification and novelty detetion : self-organization of optional bayesian decision. *Neural Netw. 13*, 10 (Dec. 2000), 1075–1093. 20

[5] Aronszajn, N. Theory of reproducing kernels. *Transactions of the American Mathematical Society 68*, 3 (1950), 337–404. 49

[6] Axelberg, P. G. V., Gu, I. Y.-H., and Bollen, M. H. J. Support vector machine for classification of voltage disturbances. *IEEE Transactions on Power Delivery 22*, 3 (July 2007), 1297–1303. 31

[7] Baggiani, F., and Marsili-Libelli, S. Real-time fault detection and isolation in biological wastewater treatment plants. *Water science and technology 60*, 11 (2009), 2949–2961. 12

[8] Bakar, Z. A., Mohamed, R., Ahmad, A., and Deris, M. M. A comparative study for outlier detection techniques in data mining. In *2006 IEEE Conference on Cybernetics and Intelligent Systems* (June 2006), pp. 1–6. 19

[9] Bertsekas, D. P. *Nonlinear Programming*. Athena Scientific, MA, 1995. 61

[10] Blum, A. L., and Langley, P. Selection of relevant features and examples in machine learning. *Artif. Intell. 97*, 1-2 (Dec. 1997), 245–271. 70

[11] Bolton, R. J., and Hand, D. J. Statistical fraud detection : A review. *Statistical Science 17*, 3 (2002), 235–255. 20

[12] Boser, B. E., Guyon, I., and Vapnik, V. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* (New York, NY, USA, 1992), COLT '92, ACM, pp. 144–152. 49

[13] Bradley, P. S., and Mangasarian, O. L. Feature selection via concave minimization and support vector machines. In *Machine Learning Proceedings of the Fifteenth International Conference(ICML'98)* (1998), Morgan Kaufmann, pp. 82–90. 72

[14] Breunig, M. M., Kriegel, H. P., Ng, R. T., and Sander, J. Lof : Identifying density-based local outliers. *SIGMOD Rec. 29*, 2 (May 2000), 93–104. 32

[15] Burges, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov. 2*, 2 (June 1998), 121–167. 37

[16] Burke, S. Missing values, outliers, robust statistics & non-parametric methods. *LCGC Europe Online supplement* (2001), 19–24. 74

[17] Chandola, V., Banerjee, A., and Kumar, V. Anomaly detection : A survey. *ACM Comput. Surv. 41*, 3 (July 2009), 15 :1–15 :58. 22, 23, 24, 27, 35, 36, 133

[18] Chang, C. C., and Lin, C. J. Libsvm : A library for support vector machines. *ACM Trans. Intell. Syst. Technol. 2*, 3 (May 2011), 27 :1–27 :27. 31

[19] Chen, Y. W., and Lin, C. J. Combining SVMs with Various Feature Selection Strategies. In *Feature Extraction*, I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, Eds., vol. 207 of *Studies in Fuzziness and Soft Computing*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, ch. 13, pp. 315–324. 72

[20] Cinar, A., and Undey, C. Statistical process and controller performance monitoring : a tutorial on current methods and future directions. In *Proceedings of American Control Conference* (1999), vol. 4, pp. 2625–2630. 12

[21] Coheni, G., Hilario, M., Sax, H., Hugonnet, S., Pellegrini, C., and Geissbuhler, A. An application of one-class support vector machine to nosocomial infection detection. *Studies in health technology and informatics 107* (1996), 716–720. 13, 72

[22] Cortes, C., and Vapnik, V. Support-vector networks. *Mach. Learn. 20*, 3 (Sept. 1995), 273–297. 29, 37, 38, 47

[23] Courant, R., and Hilbert, D. *Methods of Mathematical Physics*, vol. 1. Interscience Publishers, Inc., New York, NY, 1953. 50

[24] Dechang, C., Yufeng, K., and Chang-Tien, L. Spatial weighted outlier detection. In *Proceedings of SIAM Conference on Data Mining* (2006), pp. 614–618. 26

[25] Desforges, M. J., Jacob, P. J., and Cooper, J. E. Applications of probability density estimation to the detection of abnormal conditions in engineering. In *Proceedings of Institute of Mechanical Engineers* (1998), vol. 212, pp. 687–703. 34

[26] Ding, C., He, X., Zha, H., and Simon, H. D. Adaptive dimension reduction for clustering high dimensional data. In *Proceedings of the 2002 IEEE International Conference on Data Mining* (Washington, DC, USA, 2002), ICDM '02, IEEE Computer Society, pp. 147–154. 63

[27] Donoho, D. L., and Huber, P. J. *A Festschrift for Eric Lehman. The notion of breakdown point.* Wadsworth, Belmont, CA, 1983. 74

[28] Ertoz, L., Eilertson, E., Lazarevic, A., Tan, P. N., Kumar, V., Srivastava, J., and Dokas, P. Minds - minnesota intrusion detection system. In *Data Mining - Next Generation Challenges and Future Directions*. MIT Press, 2004. 22

[29] Eskin, E. Anomaly detection over noisy data using learned probability distributions. In *Proceedings of the Seventeenth International Conference on Machine Learning* (San Francisco, CA, USA, 2000), ICML '00, Morgan Kaufmann Publishers Inc., pp. 255–262. 34

[30] Fagarasan, I., and Iliescu, S. S. T. Applications of fault detection methods to industrial processes. *WTOS 7*, 6 (June 2008), 812–821. 20

[31] Fan, J., and Li, R. Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association 96*, 456 (2001), 1348–1360. 71

[32] Fan, J., and Li, R. Statistical challenges with high dimensionality : Feature selection in knowledge discovery. In *International Congress of Mathematicians* (2006), European Mathematical Society, pp. 595–622. 71

[33] Ferré, L. Selection of components in principal component analysis : A comparison of methods. *Computational Statistics & Data Analysis 19*, 6 (1995), 669 – 682. 67

[34] Fletcher, R. *Practical Methods of Optimization ; (2Nd Ed.)*. Wiley-Interscience, New York, NY, USA, 1987. 46

[35] Garcia-Alvarez, D., Fuente, M., Vega, P., and Sainz, G. Fault detection and diagnosis using multivariate statistical techniques in a wastewater treatment plant. In *7th IFAC International Symposium on Advanced Control of Chemical Processes* (2009), pp. 952–957. 12

[36] Guller, I., and Ubeyli, E. D. Multiclass support vector machines for eeg-signal classification. *IEEE Transactions on Information Technology in Biomedicine 11*, 2 (March 2007), 117–126. 31

[37] Guyon, I., and Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res. 3* (Mar. 2003), 1157–1182. 70, 71

[38] Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. Gene selection for cancer classification using support vector machines. *Mach. Learn. 46*, 1-3 (Mar. 2002), 389–422. 71, 73, 77, 78

[39] Hampel, F. R. The Influence Curve and Its Role in Robust Estimation. *Journal of the American Statistical Association 69*, 346 (1974), 383–393. 74

[40] He, Z., Xu, X., and Deng, S. Discovering cluster-based local outliers. *Pattern Recogn. Lett. 24*, 9-10 (June 2003), 1641–1650. 32

[41] Heller, K. A., Svore, K. M., Keromytis, A. D., and Stolfo, S. J. One class support vector machines for detecting anomalous windows registry accesses. In *Proc. of the workshop on Data Mining for Computer Security* (2003). 13, 72

[42] Hotelling, H. *Techniques of Statistical Analysis*. McGraw-Hili, New York, 1947. 34

[43] Hsu, C. W., Chang, C. C., and Lin, C. J. A practical guide to support vector classification. Tech. rep., Departement of Computer Science, National Taiwan University, 2010. 53

[44] Huang, J., Lu, J., and Ling, L. C. X. Comparing naive bayes, decision trees, and svm with auc and accuracy. In *Third IEEE International Conference on Data Mining, ICDM 2003* (2003), IEEE Computer Society, pp. 553–556. 31

[45] Iglewicz, B., and Hoaglin, D. *How to detect and handle outliers*. ASQC Quality Press, 1993. 74

[46] John, G. H., Kohavi, R., and Pfleger, K. Irrelevant features and the subset selection problem. In *Machine Learning : proceedings of the Eleventh international* (1994), Morgan Kaufmann, pp. 121–129. 71

[47] Keerthi, S. S., and Lin, C. J. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural Comput. 15*, 7 (July 2003), 1667–1689. 51

[48] Knorr, E. M., and Ng, R. T. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24rd International Conference on Very Large Data Bases* (San Francisco, CA, USA, 1998), VLDB '98, Morgan Kaufmann Publishers Inc., pp. 392–403. 32

[49] Kohavi, R., and John, G. H. Wrappers for feature subset selection. *Artif. Intell. 97*, 1-2 (Dec. 1997), 273–324. 70

[50] Kourti, T., Lee, J., and MacGregor, J. Experiences with industrial applications of projection methods for multivariate statistical process control. *Computers and Chemical Engineering 20* (1996), S745–S750. 12

[51] Kresta, J. V., MacGregor, J. F., and Marlin, T. E. Multivariate statistical monitoring of process operating performance. *The Canadian Journal of Chemical Engineering 69*, 1 (1991), 35–47. 12

[52] Kubat, M., and Matwin, S. Addressing the curse of imbalanced training sets : One-sided selection. In *Proceedings of the Fourteenth International Conference on Machine Learning* (1997), Morgan Kaufmann, pp. 179–186. 84

[53] Kuhn, H. W., and Tucker, A. W. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability* (Berkeley, Calif., 1951), University of California Press, pp. 481–492. 46

[54] Lazarevic, A., Kumar, V., and Srivastava, J. Intrusion detection : A survey. In *Managing Cyber Threats*. Springer US, 2005, pp. 19–78. 20

[55] Lebart, L., Morineau, A., and Piron, M. *Statistique Exploratoire Multidimensionnelle*. Dunod, Paris, France, 2004. 64

[56] Lee, J., Yoo, C., Choi, S., Vanrolleghem, P., and Lee, I. Nonlinear process monitoring using kernel principal component analysis. *Chemical engineering Science 59*, 1 (2004), 223–234. 12

[57] Lin, H. T., and Lin, C. J. A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Tech. rep., Department of Computer Science, National Taiwan University, 2003. 51

[58] Lin, J., Keogh, E., Fu, A., and Van Herle, H. Approximations to magic : Finding unusual medical time series. In *Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems* (Washington, DC, USA, 2005), CBMS '05, IEEE Computer Society, pp. 329–334. 20

[59] Ma, J., and Perkins, S. Time-series novelty detection using one-class support vector machines. In *International Joint Conference on Neural Networks* (July 2003), vol. 3, pp. 1741–1745. 19

[60] MacGregor, J. F., Kourti, T., and Nomikos, P. Analysis, monitoring and fault diagnosis of industrial processes using multivariate statistical projection methods. IFAC, 13th Triennial Word Congress, pp. 145-150, San Francisco, CA. 68

[61] Mahadevan, S., and Shah, S. L. Fault detection and diagnosis in process data using one-class support vector machines. *Journal of Process Control 19*, 10 (2009), 1627 – 1639. 12, 13, 20, 72

[62] Maldonado, S., Weber, R., and Basak, J. Simultaneous feature selection and classification using kernel-penalized support vector machines. *Inf. Sci. 181*, 1 (Jan. 2011), 115–128. 51

[63] McCallum, A., Freitag, D., and Pereira, F. C. N. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning* (San Francisco, CA, USA, 2000), ICML '00, Morgan Kaufmann Publishers Inc., pp. 591–598. 34

[64] Mercer, J. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London* (1909). 50

[65] Mnassri, B., Ananou, B., El Adel, E., Ouladsine, M., and Gasnier, F. Detection et localisation de defauts des wafers par des approches statistiques multivariees et calcul des contributions. Conference Internationale Francophone d'Automatique (CIFA). 12

[66] Morris, A. J., and Martin, E. B. Process performance monitoring and fault detection through multivariate statistical process control. IFAC Conference SAFEPROCESS'97, Hull UK, pp. 1-14. 68

[67] Munoz-Mari, J., Gomez-Chova, L., Camps-Valls, G., and Calpe-Maravilla, J. Image classification with semi-supervised one-class support vector machine. In *Proc. SPIE* (2008), vol. 7109. 13

[68] Patidar, R., and Sharma, L. Credit card fraud detection using neural network. In *IJSCE ISSN : 2231-2307, Volume-1, Issue-NCAI2011* (2011). 20

[69] Pudil, P., and Novovicova, J. Novel methods for subset selection with respect to problem knowledge. *IEEE Intelligent Systems 13*, 2 (Mar. 1998), 66–74. 63

[70] Rajasegarar, S., Leckie, C., Bezdek, J. C., and Palaniswami, M. Centered hyperspherical and hyperellipsoidal one-class support vector machines for anomaly detection in sensor networks. *IEEE Transactions on Information Forensics and Security 5*, 3 (2010), 518–533. 19

[71] Rakotomamonjy, A. Variable selection using svm based criteria. *J. Mach. Learn. Res. 3* (Mar. 2003), 1357–1370. 78, 79, 80

[72] Rätsch, G., Mika, S., Schölkopf, B., and Müller, K. R. Constructing boosting algorithms from svms : An application to one-class classification. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 9 (Sept. 2002), 1184–1199. 30

[73] Roweis, S. T., and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE 290* (2000), 2323–2326. 63

[74] Salvador, S., and Chan, P. Learning states and rules for detecting anomalies in time series. *Applied Intelligence 23*, 3 (Dec. 2005), 241–255. 26

[75] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. Estimating the support of a high-dimensional distribution. Tech. Rep. 99-87, Microsoft Research, 1999. 30, 58, 59

[76] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. Estimating the support of a high-dimensional distribution. *Neural Comput. 13*, 7 (July 2001), 1443–1471. 12, 56, 60, 61, 62

[77] Schölkopf, B., Smola, A. J., and Müller, K. R. Advances in kernel methods. MIT Press, Cambridge, MA, USA, 1999, ch. Kernel Principal Component Analysis, pp. 327–352. 63

[78] Schölkopf, B., Smola, A. J., Williamson, R. C., and Bartlett, P. L. New support vector algorithms. *Neural Comput. 12*, 5 (May 2000), 1207–1245. 53, 55, 62

[79] Shawe-Taylor, J., and Cristianini, N. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004. 51

[80] Shin, H. J., Eom, D. H., and Kim, S. S. One-class support vector machines-an application in machine fault detection and classification. *Comput. Ind. Eng. 48*, 2 (Mar. 2005), 395–408. 20

[81] Simoglou, A., Martin, E. B., and Morris, A. J. Multivariate statistical process control in chemicals manufacturing. IFAC Conference SAFEPROCESS'97, Hull UK. 68

[82] Smola, A. J., and Schölkopf, B. A tutorial on support vector regression. *Statistics and Computing 14*, 3 (Aug. 2004), 199–222. 37

[83] Song, X., Wu, M., Jermaine, C., and Ranka, S. Conditional anomaly detection. *IEEE Trans. on Knowl. and Data Eng. 19*, 5 (May 2007), 631–645. 25

[84] Sukchotrat, T., Kim, S. B., and Tsung, F. One-class classification-based control charts for multivariate process monitoring. *IIE Transactions 42*, 2 (Nov. 2009), 107–120. 20

[85] Suzuki, E., Watanabe, T., Yokoi, H., and Takabayashi, K. Detecting interesting exceptions from medical test data with visual summarization. In *Proceedings of the Third IEEE International Conference on Data Mining* (Washington, DC, USA, 2003), ICDM '03, IEEE Computer Society, pp. 315–. 20

[86] Tang, J., Chen, Z., Fu, A. W. C., and Cheung, D. A robust outlier detection scheme for large data sets. In *In 6th Pacific-Asia Conf. on Knowledge Discovery and Data Mining* (2001), pp. 6–8. 32

[87] Tax, D. M. J., and Duin, R. P. W. Combining one-class classifiers. In *Proceedings of the Second International Workshop on Multiple Classifier Systems* (London, UK, UK, 2001), MCS '01, Springer-Verlag, pp. 299–308. 30

[88] Theiler, J., and Michael Cai, D. Resampling approach for anomaly detection in multispectral images. In *IN PROC. SPIE* (2003), pp. 230–240. 29

[89] Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B) 58* (1996), 267–288. 72

[90] Tong, H., and Bluck, D. An industrial application of principal component test to fault detection and identification. 3rd IFACWorkshop on On-line Fault Detection and Supervision in the Chemical Process Industries, Lyon, France. 68

[91] Valle, S., Li, W., and Qin, S. J. Selection of the number of principal components : The variance of the reconstruction error criterion with a comparison to other methods. *Industrial Engineering & Chemistry Research 38*, 11 (1999), 4389–4401. 67

[92] Vapnick, V. Sv machines for pattern recognition. In *Statistical learning theory*. john wiley sons, 1998, ch. 12, pp. 496–498. 12, 52, 89

[93] Vapnik, V. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. 23, 37, 41, 52, 62

[94] Viaene, S., Derrig, R. A., Baesens, B., and Dedene, G. A comparison of state-of-the-art classification techniques for expert automobile insurance claim fraud detection. *The Journal of Risk and Insurance 69*, 3 (2002), 373–421. 20

[95] Wang, D., and Romagnoli, J. Robust multi-scale principal components analysis with applications to process monitoring. *Process Control 15*, 8 (2005), 869–882. 72

[96] Wang, Y., Wong, J., and Miner, A. Anomaly intrusion detection using one class svm. In *Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC* (2004), IEEE, pp. 358 – 364. 13

[97] Wise, B., Gallagher, N., Butler, S., Jr, D., and Barna, G. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. *Journal of Chemometrics 13* (1999), 389–422. 12

[98] Yang, H., King, I., and Lyu, M. R. Multi-task learning for one-class classification. In *IEEEWorld Congress on Computational Intelligence* (July 2010), KDD '06, pp. 1–8. 19

[99] Young-Seon, J., In-Ho, K., Myong-Kee, J., and Dongjoon, K. A new feature felection method for one-class classification problems. *Systems, Man, and Cybermetics 42*, 6 (Nov 2012), 1500–1509. 23

[100] Yu, D., Sheikholeslami, G., and Zhang, A. Findout : Finding outliers in very large datasets. *Knowl. Inf. Syst. 4*, 4 (Oct. 2002), 387–412. 32

[101] Zhang, J., Ma, K. K., Er, M. H., and Chong, V. Tumor segmentation from magnetic resonance imaging by learning via one-class support vector machine. In *Proceedings from 7th International Workshop on Advanced Image Technology (IWAIT04)* (2004), pp. 207 – 211. 13, 72

[102] Zhifeng, L., and Xiaoou, T. Using support vector machines to enhance the performance of bayesian face recognition. *IEEE Transactions on Information Forensics and Security 2*, 2 (Juin 2007), 174–180. 31

[103] Zhu, J., Rosset, S., Hastie, T., and Tibshirani, R. 1-norm support vector machines. In *Neural Information Processing Systems* (2003), vol. 16. 72