

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE
APPLIQUÉES

PAR
MOUDACHE SALIM

PRÉDICTION DU RISQUE LOGICIEL, UNE APPROCHE BASÉE
SUR LA PROBABILITÉ ET L'IMPACT DES FAUTES:
ÉVALUATION EMPIRIQUE.

Août 2018

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

Résumé

De nos jours, la priorisation des tests logiciels est une tâche essentielle du volet assurance qualité du processus de développement des logiciels. En effet, tester l'intégralité d'un système logiciel reste la plupart du temps impossible. En outre, cette opération peut être très coûteuse et très longue. L'utilisation de techniques de forage de données, spécifiquement les algorithmes de Machine Learning, pour la prédition de fautes est une approche puissante avec un grand potentiel pour aiguiller l'équipe de test de façon plus ciblée et efficace. Cela permet de fournir des indications précieuses qui permettraient de construire des scénarios de test plus appropriés, et une meilleure allocation des ressources. De cette façon, les testeurs vont concentrer leurs efforts en particulier sur des classes à haut risque. Ceci permet de réduire considérablement le temps et les coûts de la phase d'assurance qualité.

Dans cette étude, nous explorons la piste de l'utilisation d'un nouveau modèle basé sur le risque pour la classification des classes logicielles selon leur risque de présenter des fautes, à l'aide de différents algorithmes d'apprentissage. Le risque est évalué en fonction de deux facteurs principaux: la probabilité qu'une classe soit sujette à contenir des fautes et l'impact de leur propagation sur le reste du système. Nous avons utilisé diverses métriques orientées objet pour représenter les deux facteurs de risque. La construction du modèle de risque utilisé est basée sur le principe de la distance euclidienne entre une paire de métriques et le point de risque maximum, une première corrélée à la probabilité d'occurrence de fautes et une deuxième qui représente l'impact de leur propagation.

Nous avons utilisé un ensemble de données provenant de différentes versions d'un système open source Java (ANT). Le jeu de données est utilisé pour générer les différentes variantes du modèle de risque, à l'aide de différentes combinaisons de paires de métriques orientées objet, en calculant la distance euclidienne entre elles. Ensuite, des modèles de classificateurs ont été construits par apprentissage supervisé pour la prédition de l'existence de fautes dans une première phase, puis leurs niveaux de sévérité dans une seconde étape. L'objectif était d'établir une distinction entre les classes contenant des fautes de sévérité normale et élevée. Nous avons utilisé différents algorithmes d'apprentissage : Naïve Bayes (NB), J48, Random Forest (RF), Support Vector Machines (SVM) et Multilayer Parceptron (ANN). Enfin, nous avons introduit la prédition du risque logiciel basée sur le nombre de défauts et leurs niveaux respectifs de gravité. Le modèle de risque évalué dans cette étude (plusieurs variantes) se révèle d'un grand potentiel et reste ouvert à amélioration.

Abstract

Nowadays, software tests prioritization is a crucial task of the software quality insurance process, since testing the whole software system is most of the time unfeasible. Furthermore, this operation can be very costly and heavily time consuming. The use of data mining techniques, specifically machine learning algorithms, to predict fault-prone components is described as a powerful tool with a great potential to help reducing effectively testing time and costs, and ensuring the delivery of a high quality product. This gives valuable insights that would allow building accurate software testing scenarios and better allocation of resources. In this way, testers will focus their efforts on relatively high risk classes.

In this study, we investigate the potential of a novel risk-based model for software faulty classes' classification using different machine learning algorithms. The risk is evaluated based on two factors: the probability of a fault-prone class and its impact on the rest of the system. We used various object-oriented metrics to represent the two risk factors. The risk of a class is modeled using the Euclidean distance between a pair of object-oriented metrics, one that better describes the probability of fault-prone occurrence and a second metric representing the impact of its propagation.

We used a dataset collected from different versions of one Java open source system (ANT). The dataset is used to build various variants of the risk model, using different combinations of object-oriented metrics pairs following the Euclidean distance formula. Then, supervised machine learning based models were constructed for software fault prediction and fault severity level prediction. The objective was to distinguish between classes containing normal and high severity faults. We used different machine learning algorithms: Naive Bayes (NB), J48, Random Forest (RF), Support Vector Machines (SVM) and Multilayer Perceptron (ANN). Finally, we investigated software risk prediction based on the number of defects and their respective levels of severity. The risk model evaluated in this study proves to be with a great potential and is still open to improvement.

Remerciements

Je tiens tout d'abord à exprimer mon profond respect et toute ma gratitude à mon directeur de recherche, Mourad BADRI, qui m'a fait confiance et m'a grand ouvert les portes du laboratoire de recherche en génie logiciel. Pour sa disponibilité, son suivi, ses remarques et ses orientations très constructives et précieuses, du début jusqu'à la fin.

Mes plus vifs remerciements vont aux jurys, pour l'honneur qu'ils me font en voulant bien accepter d'examiner mon modeste travail, pour leurs temps et l'attention qu'ils y portent en le faisant.

Sans oublier, bien sûr, mes très chers parents pour l'éducation qu'ils m'ont prodiguée, avec tous les moyens et au prix de tous les sacrifices qu'ils ont consentis à mon égard. Pour le sens du devoir qu'ils m'ont enseigné depuis mon enfance. Les anges gardiens qui éclairent mon chemin et qui me soutiennent toujours dans ce que j'entreprends de faire. À mes très chères sœurs Thiziri et Baya, mes très chers frères Rabah et Soufyane ainsi que sa femme Linda avec qui j'ai vécu mes années les plus tendres. Vous qui ne cessez de m'encourager, grâce à qui je me suis érigé au point où j'en suis aujourd'hui.

J'adresse mes sincères remerciements à mes proches ici au CANADA, qui m'ont accueilli à bras ouverts. À tous les membres de ma famille et mes amis avec lesquels j'ai partagé mes moments de joie et de bonheur. Malgré la distance, vous restez toujours présents quelque part.

Enfin, que toute personne m'ayant aidé de près ou de loin à l'accomplissement de ce modeste travail, ayant cru et continuant de croire en moi, trouve ici l'expression de ma reconnaissance.

Table Des Matières

Résumé	i
Abstract	ii
Remerciements	iii
Table Des Matières	iv
Liste des Tables	vii
Liste des Figures	ix
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Problématique	2
1.3 Organisation du Mémoire	3
2 ÉTAT DE L'ART : un Aperçu	4
2.1 Introduction	4
2.2 Métriques Orientées Objet	4
2.2.1 Métriques de complexité	5
2.2.2 Métriques de taille	5
2.2.3 Métriques de couplage	5
2.2.4 Response For Class	6
2.3 Solutions Existantes	6
2.3.1 Seuils des métriques	6
2.3.2 Classification binaire	7
2.3.3 Niveau de sévérité	8
2.4 Limitations	8
2.5 Piste Potentielle	9
2.6 Conclusion	9

3 MÉTHODOLOGIE DE LA RECHERCHE	10
3.1 Introduction	10
3.2 Questions de Recherches	10
3.3 Présentation des Données	11
3.4 Construction des Modèles de Risque	12
3.5 Prétraitements et Préparation des Données	13
3.6 Machine Learning	14
3.6.1 Naïve bayes	15
3.6.2 Arbres de décisions	15
3.6.3 Machines à vecteurs de support	15
3.6.4 Réseaux de neurones artificiels	15
3.7 Méthodes d'Évaluation	16
3.7.1 10-Fold validation	16
3.7.2 G-mean	16
3.7.3 Area under the curve	18
4 EXPÉRIMENTATIONS ET RÉSULTATS	20
4.1 Introduction	20
4.2 Classification Binaire	20
4.2.1 Description de l'expérimentation	20
4.2.2 Rappel des questions de recherche	21
4.2.3 Résultats	21
4.2.4 Discussion	23
4.3 Détection du Niveau de Sévérité	23
4.3.1 Introduction	23
4.3.2 Rappel des questions de recherche	24
4.3.3 Détection des fautes de sévérité normale	24
4.3.4 Détection des fautes de sévérité élevées	26
4.3.5 Détection du niveau de sévérité des fautes	28
4.3.6 Conclusion	29
4.4 Prédiction de Risque	30
4.4.1 Introduction	30
4.4.2 Rappel des questions de recherche	30
4.4.3 Définition du risque empirique	30
4.4.4 Modèle de risque multidimensionnel	31
4.4.5 Description de l'expérimentation	31
4.4.6 Résultats	32
4.4.7 Conclusion	33
5 DISCUSSIONS ET CONCLUSION	35
5.1 Menaces de Validité	35
5.1.1 Risques de validité externes	35
5.1.2 Risques de validité internes	35

5.1.3 Risques de validité conceptuelle :	36
5.2 Conclusion et Perspectives	36
BIBLIOGRAPHIE	38
ANNEXE A Classification Binaire	45
ANNEXE B Prédiction des Fautes de Sévérité Normale	53
ANNEXE C Prédiction des Fautes de Sévérité Élevée	58
ANNEXE D Prédiction du Niveau de Sévérité	63
ANNEXE E Prédiction de Risque	73

Liste des Tables

3.1	Statistiques des Versions de ANT	12
3.2	Matrice de Confusion	16
3.3	Matrice de Confusion Multi-Classes	17
4.1	Statistiques ANT Oversampling	21
4.2	Résultats RT (LOC, Fan-in) Classification Binaire	22
4.3	Statistiques sur les Fautes de Sévérité Normales pour ANT	24
4.4	Résultats de RT (LOC, Fan-in) pour la Prédiction de Bogues de Sévérité Normale	25
4.5	Statistiques sur les Défauts de Sévérité Élevée pour ANT	26
4.6	Résultats de RT (RFC, Ca) pour la Prédiction de Bogues de Sévérité Élevée	27
4.7	Statistiques sur le Niveau de Sévérité pour ANT	28
4.8	Résultats de RT (LOC, RFC) pour la Prédiction du Niveau de Sévérité	29
4.9	Résultats de RT (LOC, RFC) pour la Prédiction de Risque	33
A.1	Résultats RT (LOC, Ca) Classification Binaire	46
A.2	Résultats RT (LOC, CBO) Classification Binaire	47
A.3	Résultats RT (LOC, Ce) Classification Binaire	48
A.4	Résultats RT (WMC, Ca) Classification Binaire	49
A.5	Résultats RT (WMC, CBO) Classification Binaire	50
A.6	Résultats RT (WMC, Ce) Classification Binaire	51
A.7	Résultats RT (WMC, Fan-in) Classification Binaire	52
B.1	Résultats de RT (LOC, Ce) pour la Prédiction de Bogues de Sévérité Normale	54
B.2	Résultats de RT (WMC, CBO) pour la Prédiction de Bogues de Sévérité Normale	55
B.3	Résultats de RT (RFC, Ca) pour la Prédiction de Bogues de Sévérité Normale	56
B.4	Résultats de RT (LOC, RFC) pour la Prédiction de Bogues de Sévérité Normale	57
C.1	Résultats de RT (LOC, Ce) pour la Prédiction de Bogues de Sévérité Élevée	59

C.2	Résultats de RT (LOC, Fan-in) pour la Prédiction de Bogues de Sévérité Élevée	60
C.3	Résultats de RT (WMC, CBO) pour la Prédiction de Bogues de Sévérité Élevée	61
C.4	Résultats de RT (LOC, RFC) pour la Prédiction de Bogues de Sévérité Élevée	62
D.1	Résultats de RT (LOC, Ce) pour la Prédiction du Niveau de Sévérité	64
D.2	Résultats de RT (LOC, Fan-in) pour la Prédiction du Niveau de Sévérité	65
D.3	Résultats de RT (WMC, CBO) pour la Prédiction du Niveau de Sévérité	66
D.4	Résultats de RT (RFC, Ca) pour la Prédiction du Niveau de Sévérité	67
D.5	Résultats de RT* (RT (LOC, Ce), RT (LOC, Ca)) pour la Prédiction du Niveau de Sévérité	68
D.6	Résultats de RT* (RT (LOC, WMC), RT (LOC, Ca)) pour la Prédiction du Niveau de Sévérité	69
D.7	Résultats de RT* (RT (LOC, RFC), RT (LOC, Ca), RT (RFC, Ca)) pour la Prédiction du Niveau de Sévérité	70
D.8	Résultats de RT* (LOC, RT (RFC, Ca)) pour la Prédiction du Niveau de Sévérité	71
D.9	Résultats de RT* (RFC, RT (LOC, Ca)) pour la Prédiction du Niveau de Sévérité	72
E.1	Résultats de RT (LOC, Ce) pour la Prédiction de Risque	74
E.2	Résultats de RT (LOC, Fan-in) pour la Prédiction de Risque	75
E.3	Résultats de RT (WMC, CBO) pour la Prédiction de Risque	76
E.4	Résultats de RT (LOC, RFC) pour la Prédiction de Risque	77
E.5	Résultats de RT* (RT (LOC, Ce), RT (LOC, Ca)) pour la Prédiction de Risque	78
E.6	Résultats de RT* (RT (LOC, RFC), RT (LOC, Ca), RT (RFC, Ca)) pour la Prédiction de Risque	79
E.7	Résultats de RT* (RT (LOC, WMC), RT (LOC, Ca)) pour la Prédiction de Risque	80
E.8	Résultats de RT* (LOC, RT (RFC, Ca)) pour la Prédiction de Risque	81
E.9	Résultats de RT* (RFC, RT (LOC, Ca)) pour la Prédiction de Risque	82

Liste des Figures

3.1	Familles de Métriques de Code Source et Facteurs de Risque	12
3.2	Modèle de Risque 2D	13
3.3	Courbe ROC	18

Chapitre. 1

INTRODUCTION

1.1 Introduction

De nos jours, les besoins en génie logiciel sont de plus en plus importants relativement à la complexité croissante des applications développées. Avec cette complexité croissante, le risque d'occurrence de fautes connaît lui aussi le même phénomène. La présence de fautes dans un système logiciel dégrade non seulement sa qualité, mais augmente également le coût de son développement et de sa maintenance [1]. Pour rectifier ces erreurs, plus particulièrement les plus sévères, et veiller à avoir un logiciel de haute qualité, il doit être rigoureusement testé. Les activités d'assurance qualité, et en particulier les tests logiciels, sont très importantes dans ce contexte. Cependant, il n'est pas réaliste d'appliquer un effort de test égal à tous les composants d'un système logiciel volumineux et complexe. Concrètement, dans le monde de l'industrie, les tests logiciels doivent souvent être faits sous une pression énorme en raison de contraintes de délais serrés et de ressources limitées. En effet, tester l'intégralité d'un système complexe est une tâche à coûts élevés et prendrait trop de temps et de ressources. Effectuer des tests de manière exhaustive devient alors dans la plupart des cas impossible, sauf peut-être pour des cas vraiment triviaux [2]. Pour toutes ces raisons, l'orientation des tests, qui est une pratique où les efforts de l'équipe d'assurance qualité sont amenés à tester rigoureusement seulement les parties susceptibles d'être les plus risquées, apparaît comme la meilleure solution. Toutefois, ce processus peut être coûteux et peut prendre beaucoup de temps [3]. Cela reste également une des tâches les plus difficiles. Un défi majeur qui plane sur cet objectif consiste à identifier les fautes le plus tôt possible dans les étapes de développement [4].

Plusieurs études qui ont été menées dans le volet de l'assurance qualité logicielle ont abordé cette problématique. Une solution triviale proposée était de tester aléatoirement [5], où les cas de test sont choisis tout à fait au hasard par un algorithme pour construire un scénario de test. Cette méthode est qualifiée de naïve et n'est certainement pas assez précise pour couvrir toutes les fautes. Une autre proposition très active a été

la prédiction de défaillance logicielle [6]. Beaucoup d'études axées sur ce principe ont été menées pour rechercher la réponse à la simple question: cette classe contient-elle des fautes? C'est ainsi que des chercheurs ont proposé des modèles de classification binaire qui conviennent le mieux pour répondre à ce genre de questions. La prédiction de fautes logicielles vise à orienter la priorisation des tests. La faiblesse de l'utilisation de modèles binaires pour soutenir la construction de scénarios de test, est qu'ils fournissent des informations simples : une classe donnée est-elle sujette à contenir des fautes ou non? C'est ce que nous dénotons par P , la probabilité de présence ou d'absence de fautes. Il est fréquent que des classes logicielles défectueuses possèdent un plus grand nombre de défauts par rapport à d'autres classes. Elles peuvent alors nécessiter des efforts supplémentaires pour les fixer. La priorisation des tests uniquement basée sur l'information binaire: classe fautive ou non fautive, peut aboutir à une utilisation non efficiente des ressources [7].

Récemment, certaines études ont approché le défi de détecter les niveaux de sévérité des fautes. La sévérité des fautes mesure la gravité de l'impact d'une faute sur un système et ses utilisateurs [8], que nous dénotons comme I . Le fruit de ces études a permis la construction de modèles de prédiction basés sur la gravité des fautes [9], qui peuvent associer à une faute différents niveaux de gravité, par exemple : faible, élevée ou non fautive. Ceci est beaucoup plus utile que la simple classification binaire, où la logique veut que les fautes aient le même niveau de sévérité. Ainsi, cette approche permet une planification des tests de qualité supérieure et une allocation de ressources à faible coût en prédisant les fautes critiques dans le système. Le problème avec ce genre d'approches, c'est qu'ils ne considèrent pas le nombre de défauts, qui en lui-même peut affecter considérablement l'effort de test.

1.2 Problématique

Il y a deux aspects principaux qui définissent le risque d'une classe ou d'un module: le nombre de fautes qu'il peut contenir et l'impact de leur propagation sur le système selon leurs niveaux de sévérité. Nous avons étudié différentes variantes du modèle de risque proposé par F. Touré dans [10] qui repose sur la distance euclidienne. La logique derrière le choix des paramètres pour la construction de ce modèle à deux dimensions est la suivante: un axe sera représenté par une métrique qui corrèle le mieux à P (la probabilité d'occurrence de fautes). Dans la seconde dimension, une métrique qui représente I (l'impact des fautes). Par la suite, nous avons exploré leur capacité à prédire l'existence de fautes et le niveau de sévérité associé. Finalement, nous introduisons la prédiction du risque logiciel. Notre motivation et notre contribution avec cette recherche consistent à introduire un nouveau modèle basé sur le risque qui englobe des informations précieuses à la fois sur le nombre de défauts et leurs niveaux de sévérité respectifs. Notre intention est de pallier aux inconvénients des modèles de classification binaire et des modèles basés sur le niveau de sévérité et à orienter les tests plus efficacement.

Nous avons le but de recueillir des informations précieuses sur le niveau de la gravité des fautes sans perdre les informations sur le nombre de fautes contenues dans une classe donnée. Pour ce faire, nous introduisons le concept de prédiction de risque afin d'atteindre cet objectif.

1.3 Organisation du Mémoire

Ce mémoire se compose de 5 chapitres principaux. Le premier chapitre est une introduction générale à la priorisation des tests logiciels et à la problématique liée à ce domaine. Après la mise en avant du contexte de cette étude, nous dévoilons nos motivations et objectifs à travers ce travail de recherche.

Le deuxième chapitre présente une revue sommaire de l'état de l'art actuel dans le domaine de l'assurance qualité logicielle en lien avec nos objectifs. Après une brève introduction qui rappelle le contexte, nous introduisons quelques éléments clés du jargon de la priorisation des tests, et plus en détail, les métriques orientées objet. Par la suite, nous exposons des solutions existantes proposées et leurs limitations. Pour clôturer ce chapitre, nous présentons la piste que nous avons choisie de creuser qui est basée sur les modèles de risque et nous finissons par une conclusion.

Dans le troisième chapitre, nous donnons les détails de la méthodologie de recherche que nous avons suivie. Nous avons réalisé une étude empirique sur les modèles de risques (variantes). Nous commencerons par exposer les questions de recherche que nous avons voulu vérifier. Pour ce faire, nous présentons d'abord les données que nous avons utilisées. Ensuite, nous détaillons la façon dont sont construits nos modèles de risque et les prétraitements réalisés sur ces derniers. Enfin, nous donnons des informations sur la panoplie d'algorithmes de Machine Learning que nous avons utilisés et nos méthodes d'évaluation.

Au quatrième chapitre, nous présentons nos résultats et les expérimentations que nous avons menées pour les obtenir. Pour chaque objectif des questions de recherche, nous décrivons l'expérimentation associée, les résultats obtenus et une discussion des résultats. Il est à noter que davantage de résultats sont donnés en annexe à ce mémoire.

Le dernier chapitre mentionne les risques éventuels sur la validité de notre étude ainsi que les futurs travaux et pistes de réflexions intéressantes. Pour finir, nous concluons ce mémoire en résumant les principales contributions de cette étude.

Chapitre. 2

ÉTAT DE L'ART : un Aperçu

2.1 Introduction

La priorisation des tests dans le volet assurance qualité du cycle de développement logiciel est sujette à beaucoup de recherches depuis plusieurs années; ce qui explique le développement considérable de ce secteur. Plusieurs facteurs mis en place ont pu rendre cette évolution opportune. Notamment, nous pouvons citer: la mise à disposition de banques de données libres d'accès au grand public, le développement d'outils de calcul de métriques ainsi que la nécessité d'assurer la qualité et la maintenance de ses produits de la part des entreprises et l'engouement du monde de l'industrie logicielle à cet essor. Les données en question présentent des informations sur, entre autres, les fautes détectées, leurs niveaux de sévérité respectifs et des métriques de code logiciel. Elles sont utilisées comme entrées pour mener des expériences qui visent à estimer le potentiel de ces métriques pour la priorisation des tests en utilisant différents procédés d'analyse. Cependant, les techniques et méthodes utilisées aujourd'hui restent non maîtrisées complètement et ouvertes à des améliorations. Aussi, beaucoup de questions de recherche demeurent sans réponse.

2.2 Métriques Orientées Objet

Nous nous penchons sur le paradigme orienté objet, grandement utilisé de nos jours. Dans ce modèle de développement, un découpage en classes d'objets qui encapsulent des propriétés (attributs) et des responsabilités (méthodes) communes est effectué. En plus des différentes manières de faire interagir des instances de classes entre elles, des notions plus complexes telles que l'héritage et le polymorphisme viennent donner un surcroit de puissance à ce paradigme. Toutes ces particularités font apparaître alors de nouveaux attributs logiciels. Nous pouvons en citer: la cohésion, la complexité, l'héritage et le couplage entre objets. Une métrique logicielle permet de mesurer un attribut logiciel. L'intérêt de ces métriques est qu'elles procurent une information quantitative propre à

chaque projet, en plus d'être objective.

Une métrique simple peut être obtenue à l'aide d'une mesure simple telle que compter le nombre de lignes de code. Mais, elle peut également être plus édifiée en mettant en coopération plusieurs attributs de différentes complexités. Par conséquent, le nombre de métriques possibles est de suite démultiplié. Plusieurs études ont été menées dans l'objectif de définir des patterns et des liens entre ces métriques et la qualité logicielle. La qualité logicielle étant en elle-même un concept abstrait et complexe, cette tâche s'avère difficile. Pour J. Voas et W.W. Agresti [11], la qualité met en jeu plusieurs aspects: la sûreté, la fiabilité, la testabilité, la maintenabilité, la tolérance aux fautes, la sécurité et la disponibilité.

En parcourant la littérature, on constate que les métriques de code source orientées objet telles que définies par Chidamber & Kemerer [12] sont les plus utilisées et donnent de bons résultats. En plus de cela, elles ont déjà été validées empiriquement par le passé par de nombreuses études [9, 13–19]. C'est pour toutes ces raisons que notre choix s'est naturellement porté sur elles pour effectuer notre étude. Les métriques orientées objet (du moins celles que nous avons utilisées) peuvent être subdivisées comme suit :

2.2.1 Métriques de complexité

WMC (Weighted Methods per Class). Comme son nom l'indique, WMC (en français : méthodes pondérées par classe) est une somme pondérée par la complexité cyclomatique de chaque méthode d'une classe [20]. Si la valeur de cette métrique est élevée pour une classe donnée, la classe sera plus difficile à tester et même à être comprise [21].

2.2.2 Métriques de taille

LOC (Lines Of Code), traduit par le nombre de lignes de code. Cette métrique donne une idée de la taille d'une classe avec le nombre de lignes de code qui la composent. Plus une classe est grande de taille, plus communément appelée classe 'Big-Mac', plus elle est disposée à contenir des fautes [22]. Cela est également indicateur de mauvaises répartitions des responsabilités entre classes, ce qui peut nécessiter une restructuration. Si toutes les classes sont de petites tailles, cela signifie que le problème persiste également. Il existe alors des responsabilités de même nature qui peuvent être regroupées dans les mêmes classes. Le juste milieu est souvent difficile à trouver.

2.2.3 Métriques de couplage

- **CBO (Coupling Between Objects)** : Le couplage [23] entre objets met en évidence le nombre de classes auxquelles une classe donnée est couplée et inversement. En fait, c'est une somme de deux autres métriques de couplage existantes qui sont Fan-in et Fan-out [24]. Elle sert à évaluer l'interdépendance entre classes d'un même système. Une classe indépendante sera plus facile à tester, à réutiliser et à maintenir.

- ***Ca (Afferent Coupling)*** : Le couplage afférent nous dénombre les classes d'un paquetage donné qui dépendent de classes extérieures. Le couplage afférent donne une idée sur l'indépendance [25].
- ***Ce (Efferent Coupling)*** : Le couplage efférent, à l'inverse de Ca, représente le nombre de classes extérieures qui dépendent de classes dans un paquetage donné. Le couplage efférent mesure la responsabilité [25].
- ***Fan-in*** : Le nombre d'appels des modules pour une fonction/méthode pour une classe donnée [26]. Cette mesure est surtout utilisée pour identifier les modules qui requièrent une restructuration et un plus grand effort de test [27].

2.2.4 Response For Class

Réponse pour une classe (RFC) est une métrique de complexité et de couplage à la fois. Elle est définie par le nombre de méthodes qui peuvent être appelées par une instance d'une classe [12]. La métrique RFC indique le degré de communication entre une classe et le reste du système à travers les invocations de méthodes externes qu'elle peut potentiellement effectuer [28]. RFC est étroitement liée à l'effort de test et à la complexité. Plus une classe fait appel à des services fournis par d'autres classes, peut-être même d'autres paquetages ou modules, plus elle sera difficile à tester.

2.3 Solutions Existantes

En vue de l'importante évolution continue que connaît le monde de l'industrie logicielle, les procédés de tests conventionnels tels que les techniques de couverture (des instructions, des scénarios d'exécution, etc.) ne sont plus possibles vu la taille et la complexité importante. Des méthodes triviales, telles que le choix arbitraire ou même basé sur l'expérience des testeurs, pour le choix des tests les plus importants, de manière complètement aléatoire, s'avèrent très peu concluantes et efficaces. Beaucoup de recherches dans ce domaine se sont focalisées sur la priorisation des tests [15, 29–38]. Dans ce qui suit, nous donnons un aperçu de solutions proposées dans la littérature:

2.3.1 Seuils des métriques

Shatnawi s'est engagé dans une étude [14] pour déterminer les valeurs de seuils pour les métriques CK au-delà desquelles une classe donnée serait à un plus grand risque de contenir des fautes. Il mène une analyse empirique de trois systèmes logiciels, inspirée par l'évaluation d'un modèle épidémiologique [39] de seuils de risque, pour calculer les seuils de valeurs acceptables pour chaque métrique. Une analyse de régression logistique lui permit de limiter la suite de métriques CK à un sous-ensemble représenté par les métriques de complexité et de couplage (WMC, CBO et RFC) comme étant les plus corrélées avec la probabilité de présence de défauts. Après avoir déterminé les seuils sur

la version 2.0 d'Eclipse, l'auteur effectue une validation sur la version 2.1 en construisant un modèle (un arbre de régression) et en croisant les résultats expérimentaux avec les résultats réels. Il démontre alors, la très bonne capacité à prédire les fautes avec les seuils déterminés plus tôt dans la version antérieure d'Eclipse. Par souci de comparaison, l'auteur détermine les seuils pour d'autres systèmes (il valida également leur capacité prédictive) et remarqua que les métriques importantes sont les mêmes pour tous les systèmes, mais les valeurs des seuils sont fortement tributaires du système en question et ne peuvent pas, par conséquent, être utilisées de manière inter-systèmes.

2.3.2 Classification binaire

Les modèles de prédiction semi-supervisée ont été utilisés. Ce genre de modèles nécessite moins de données sur les fautes que les modèles supervisés. Lu et al. [40, 41] ont étudié l'utilisation de l'apprentissage semi-supervisé pour la prédiction de la prédisposition aux fautes logicielles avec les algorithmes: Random Forest et réduction de dimension. Ils ont établi que réduire la dimensionnalité des métriques du code source améliore significativement le modèle d'apprentissage semi-supervisé.

De son côté, Catal a exploré différents algorithmes de classification semi-supervisée pour la prédiction de fautes [42]. Il a comparé quatre méthodes, à savoir : Support Vector Machine, Class Mass Normalization, Low-Density Separation et Expectation-Maximization. Il a conclu que la normalisation faible densité a donné les meilleurs résultats pour les grands ensembles de données, mais pourrait également être utilisée pour les plus petits. Bien que ces algorithmes soient supervisés, ils conviennent aux données non balancées, ce qui est souvent le cas quand on aborde les problèmes de prédiction de fautes, comme les logiciels présentent plus de classes non fautives que celles qui sont fautives. Cependant, des données sur les fautes sont nécessaires pour utiliser ce type d'approches, ce qui les rend plus difficiles à automatiser (complètement) que les approches non supervisées. Par conséquent, elles ne conviennent pas à l'utilisation dans les premiers stades du cycle de développement étant donné que ces données n'existent pas encore.

En revanche, les modèles d'apprentissage non supervisés dépassent cette lacune puisqu'ils ne nécessitent aucune donnée sur les fautes. Dans [43], Bishnu & Bhattacherjee ont utilisé une approche similaire à celle de Catal et al. dans [44, 45], en utilisant l'algorithme K-means et les mêmes valeurs de seuil pour prédire les modules défectueux. Cependant, ils ont utilisé l'algorithme Quad-Tree combiné à un algorithme génétique pour initialiser les clusters utilisés dans l'algorithme K-means. Selon les auteurs, la performance de classification de leur modèle est aussi bonne que les résultats obtenus avec les modèles supervisés qui sont construits à l'aide de données sur les fautes. A. Boucher et M. Badri ont étudié en [46] trois techniques de calcul de seuils qui peuvent être utilisées pour la prédiction de fautes: courbes ROC, VARL (valeur d'un niveau de risque acceptable) et Alves Ranking. Ils concluent que courbes ROC donne les meilleures performances, mais qu'Alves Ranking est un bon choix aussi. L'avantage de l'algorithme

Alves Ranking sur la technique des courbes ROC, c'est qu'il est complètement non supervisé; il ne requiert pas de données sur les fautes et peut être utilisé lorsqu'elles ne sont pas disponibles. Comme ils le mentionnent dans [47], ils ont adapté le modèle de HySOM, qui à l'origine marche à un niveau de granularité fonctionnelle, au niveau de granularité de classe logicielle du paradigme orienté objet.

2.3.3 Niveau de sévérité

Plus récemment, plusieurs études ont démontré que les modèles de prédiction de fautes basés sur le niveau de sévérité des bogues sont beaucoup plus utiles que les modèles de classification binaire qui eux déterminent seulement si un module est sujet à contenir des fautes ou non. E. Hong [48] a proposé un nouveau modèle de prédiction de sévérité à l'aide de deux modules mesurant le gravité MS (Module de gravité) et MS RT (Module de la densité de gravité). Les modèles proposés sont différents des précédents modèles [9, 49, 50] par l'assignation d'un coefficient de gravité au modèle, plutôt que de traiter chaque niveau de sévérité seul. Grâce à une étude empirique à l'aide des projets JM1 et PC4 de l'ensemble de données de la NASA, ils ont conclu que les modèles MS réalisent de meilleures performances que les précédents modèles de prédiction et que les modèles MSD montrent de bonnes performances avec JM1. En outre, les réseaux neuronaux artificiels multicouches ont montré les meilleures performances parmi les trois algorithmes de classification utilisés pour la construction des modèles de prédiction.

2.4 Limitations

Les méthodes connues aujourd'hui ne sont pas tout à fait bien maîtrisées et présentent quelques limitations qu'il faudra dépasser. Le plus grand challenge dans le domaine de l'assurance qualité est la détection de fautes dans les premières phases du cycle de vie du logiciel. Or, les techniques d'apprentissage supervisées et semi-supervisées nécessitent la présence de données sur les fautes, qui n'existent pas encore lorsqu'on n'utilise pas des modèles de croisement, pour être utilisées. Pour ces raisons, les chercheurs se sont orientés vers les techniques d'apprentissage non supervisées.

Un grand nombre de recherches se focalise sur la classification binaire des classes fautives et non fautives. Alors que cette information, bien qu'utile, n'est pas suffisante pour l'orientation des efforts de test dans le cas d'un grand projet. La priorisation de tests uniquement basée sur l'information binaire : classe défectueuse et non défectueuse, peut aboutir à une utilisation non efficiente des ressources [7]. Ce type de classification assume que tous les défauts ont le même degré de sévérité, bien que dans le monde réel ce n'est pas le cas. Une classe qui contient des fautes qui font crasher le système et une autre classe qui contient une faute triviale (d'affichage par exemple) ne requièrent pas le même effort de test et laps de temps pour être fixées.

Les modèles de prédiction basés sur le niveau de sévérité des fautes viennent surpasser cette lacune. Mais, ils sont encore loin de fournir l'information suffisante pour une

orientation des tests efficiente. Cette approche néglige une information essentielle qui est le nombre de fautes. En effet, une classe qui contient plusieurs défauts de sévérité élevée, nécessitera plus de temps et de ressources et aura certainement une priorité plus haute qu'une autre classe qui contient une faute de sévérité élevée et une deuxième faute de sévérité normale.

2.5 Piste Potentielle

F. Touré a présenté un nouveau modèle basé sur le risque dans sa thèse [10] construit à partir de la théorie du risque de Bernoulli [51], qui est largement utilisée en gestion de projets. L'intérêt de cette méthode est qu'elle procure une évaluation objective du risque associé aux différents éléments d'un système. Un risque, appelé événement redouté, est étudié sur deux dimensions : la probabilité de survenance et l'impact de cet événement redouté sur le reste du système. Réduire le risque quantifié d'un composant revient à diminuer alors l'un ou les deux axes qui composent ce risque. Le modèle de risque est basé sur le calcul de la distance euclidienne entre un couple de métriques orientées objet et le point de risque maximal, suivant ainsi deux axes : la probabilité d'occurrence des fautes et l'impact des fautes.

Il a démontré, en utilisant quatre différentes méthodes de validation croisée (le fitting, 10-Fold Validation, Leave-One-Out Validation et le $10 \times 70/30$), que ces modèles peuvent couvrir 70% des cas de test qui ont été proposés par des testeurs confirmés. Il a également commencé à explorer l'utilisation de ces modèles pour la priorisation des tests. Les résultats de cette étude sont exploratoires et ont besoin d'être plus approfondis. Dans ce contexte, nous avons voulu explorer le potentiel de ces modèles dans notre étude.

2.6 Conclusion

Après comparaison de plusieurs recherches, nous avons constaté que la suite de métriques orientées objet CK revient très fréquemment dans la littérature [52–54]. Elle a également été validée empiriquement par de nombreuses études [21, 55, 56]. Ces métriques se sont révélées être les plus significatives dans la prédiction des fautes [9, 52, 57]. La plupart des études se focalisent sur la classification binaire; or cette dernière ne donne pas une information suffisante. Les approches basées sur le niveau de sévérité apportent des indications encore plus utiles, mais négligent le nombre de défauts qui constitue une information très importante. C'est dans cette optique que nous nous sommes lancés dans cette étude avec la motivation de palier à certaines des limitations relevées dans l'état de l'art de la priorisation des tests.

Chapitre. 3

MÉTHODOLOGIE DE LA RECHERCHE

3.1 Introduction

Les méthodes utilisées dans la priorisation des tests aujourd’hui ne permettent pas de capturer les deux facteurs qui affectent le risque qui sont le nombre de fautes et leurs niveaux de sévérité respectifs. C’est dans ce contexte que nous utilisons une nouvelle définition du risque qui englobe ces deux entités. Nous avons utilisé un ensemble de données provenant du système ANT écrit en Java pour construire différents modèles de risque. Dans une première phase, on s'est intéressés à la classification binaire des classes fautives. Dans une seconde étape, nous avons investigué la prédiction de niveaux de sévérité des bogues. Nous avons utilisé l'échelle qualitative suivante pour la classification des fautes selon leur gravité : normale, élevée ou ne contient pas de fautes. Comme dernière phase, nous avons exploré le potentiel de la prédiction de risque. Dans ce qui suit, nous allons exposer nos questions de recherche et les détails de notre méthodologie de recherche.

3.2 Questions de Recherches

Dans cette étude, nous voulons répondre aux questions de recherche suivantes :

QR1 : Le modèle de risque mentionné précédemment est-il adapté pour prédire l'existence de fautes logicielles?

Étant donné que l'utilisation d'un tel modèle de risque n'a pas encore été largement investiguée pour la prédiction de fautes, nous avons décidé de vérifier s'il peut réaliser une bonne performance.

QR2 : Une imbrication de modèles de risque (une combinaison de variantes du modèle de base) peut-elle effectuer une meilleure prédiction de l'occurrence de fautes logicielles?

L'hypothèse que nos modèles de risque à deux dimensions ne contiennent pas suffisamment d'informations pour nous donner des informations précieuses sur les fautes n'est pas à écarter. Afin de contrer cela, ou de l'améliorer, nous investiguons également des modèles de risque multidimensionnels où nous imbriquons une combinaison de modèles de risque à deux dimensions.

QR3 : Les modèles de risque mentionnés peuvent-ils prédire les niveaux de sévérité des fautes?

Le modèle de risque considéré n'a jamais été utilisé séparément, ni pour la classification binaire des classes fautives et encore moins pour la prédiction du niveau de sévérité des fautes. Nous avons alors décidé d'explorer son potentiel pour la prédiction de la gravité des fautes avec deux niveaux de gravité : élevée et normale ou pas sujettes à contenir des fautes.

QR4 : Une imbrication des modèles de risque mentionnés précédemment peut-elle améliorer la prédiction de la gravité des fautes?

Dans un souci d'amélioration, nous avons creusé également l'utilisation de modèles de risque multidimensionnels pour la prédiction de la sévérité.

QR5 : Notre modèle de risque peut-il prédire le risque des classes logicielles et nous donner des informations à la fois sur le nombre de fautes, mais aussi leurs niveaux de sévérité?

3.3 Présentation des Données

Sous le format d'un fichier Excel, nous disposons d'un ensemble de données décrivant différentes versions du système ANT (<http://ant.apache.org/>), de la version 1.3 à la version 1.7, où chaque classe est décrite par le nombre de fautes qu'elle contient, leur niveau de gravité respectif qui peut être soit : bloquant, critique, majeur, normal, mineur ou trivial. Également, ce qui nous intéresse, les métriques orientées objet que nous avons exposées précédemment. Pour les nommer, il s'agit de : WMC, LOC, CBO, Ce, Ca, Fan-in et pour finir RFC.

Le tableau suivant décrit les différentes versions du système ANT utilisées et met en évidence le nombre de classes, classes défectueuses et le ratio de classes fautives avant et après la duplication de fautes [9]. La duplication de fautes est une méthodologie simple qui consiste à dupliquer les classes qui comprennent plus d'une faute autant de fois qu'elles présentent de fautes.

Système	#Classes	#Fautives	Ratio	#Classes-D	#Fautives-D	Ratio-D
ANT 1.3	126	60	47.62%	201	135	67.16%
ANT 1.4	178	38	21.35%	190	50	26.21%
ANT 1.5	293	106	36.62%	366	179	48.91%
ANT 1.6	352	45	12.78%	368	61	16.58%
ANT 1.7	675	70	10.37%	776	101	13.01%

Table 3.1: Statistiques des Versions de ANT

3.4 Construction des Modèles de Risque

À partir des données présentées ci-dessus, nous avons construit un ensemble de modèles de risque tel que défini dans la thèse de F. Touré [10]. Nous avons sélectionné différentes combinaisons de métriques en respectant la disposition suivante : en prenant m_1 une métrique qui décrit le mieux la probabilité de présence de faute, ces études [9, 15, 58, 59] attestent que ce sont les métriques de taille et de complexité qui jouent ce rôle, et m_2 une métrique qui représente l'impact des fautes. Les études [60] et [61] mentionnent que les métriques de couplage accomplissent cette tâche. On peut donc diviser les métriques orientées objet en deux grandes familles comme l'illustre la figure suivante [10] :

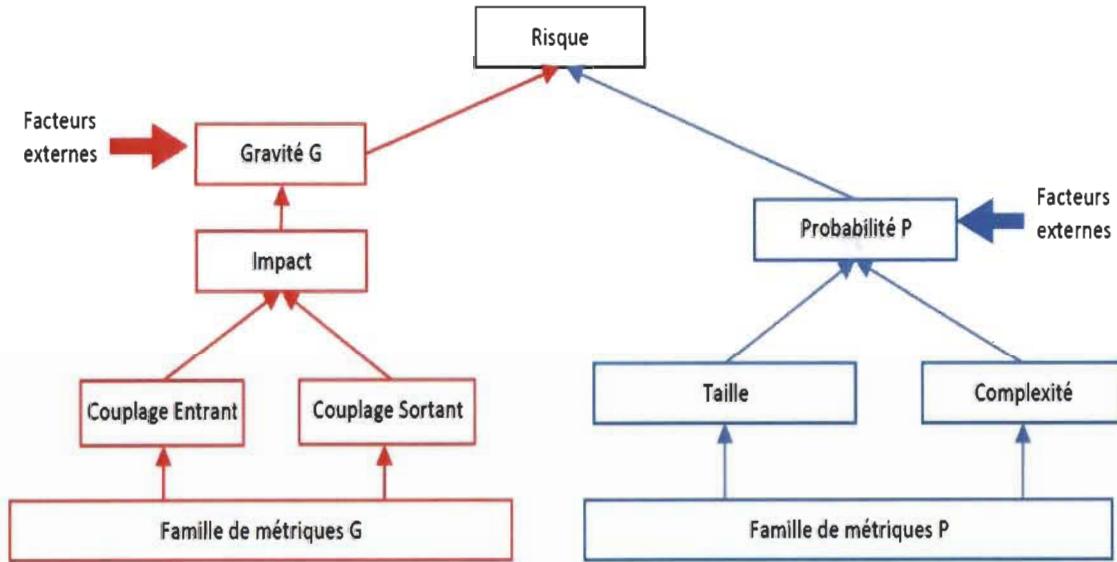


Figure 3.1: Familles de Métriques de Code Source et Facteurs de Risque

Le risque théorique d'une classe donnée C est calculé comme suit :

$$RT_{m_1, m_2}(C) = \sqrt{(m_1(C) - \max_i(m_1(C_i)))^2 + (m_2(C) - \max_i(m_2(C_i)))^2}$$

La figure suivante [10] illustre le modèle de risque à deux dimensions. L'axe des abscisses présente la métrique m1 et l'axe des ordonnées représente la métrique m2. Le point M quant à lui, représente le point de risque maximum des deux métriques et il est calculé comme suit : $M = (\max(m_1(C_i)), \max(m_2(C_i)))$.

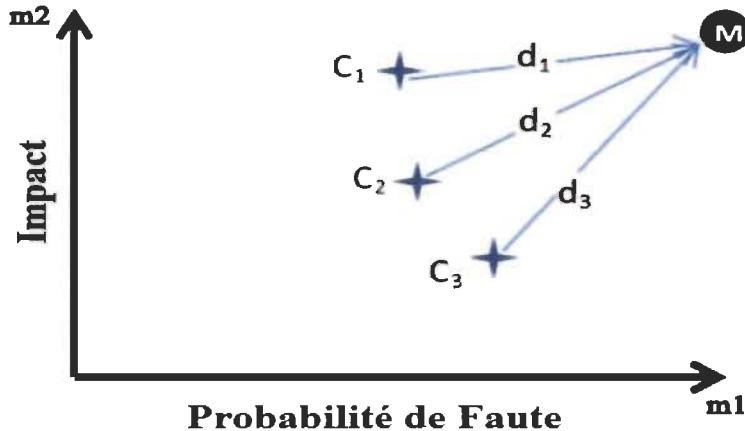


Figure 3.2: Modèle de Risque 2D

Afin d'effectuer une étude empirique sur le modèle de risque considéré pour la prédiction de fautes logicielles sur le système ANT, nous avons construit les modèles de risque suivants : RT (LOC, Fan-in), RT (WMC, fan-in), RT (WMC, CBO), RT (LOC, CBO), RT (LOC, Ce), RT (LOC, Ca), RT (WMC, Ce), RT (WMC, Ca), RT (RFC, Ca), RT (LOC, WMC), RT (LOC, RFC) et enfin RT (WMC, RFC). Nous donnons des explications supplémentaires sur la méthodologie que nous avons suivie pour mener à bien notre étude empirique dans le reste du présent document.

3.5 Prétraitements et Préparation des Données

La majorité, si ce n'est pas tous, des algorithmes de forage de données traitent les données sous forme d'une matrice unique appelée 'Mining View' dans le jargon du domaine des sciences de données. Toutefois, les données obtenues au début d'un projet ne sont pas nécessairement dans ce format. En outre, les ensembles de données contiennent fréquemment des valeurs manquantes, combinaisons de données impossibles et hors

intervalles. Il y a aussi certains algorithmes qui sont compatibles avec seulement quelques types d'attributs uniquement. Certaines données pourraient alors être incompatibles avec certains algorithmes. Le prétraitement des données [62] comprend le nettoyage, la normalisation, la transformation, l'extraction de caractéristiques, la sélection d'attributs et ainsi de suite. L'étape de prétraitement de données pourrait prendre jusqu'à 80% du temps et des ressources d'un projet de science des données. Heureusement, il existe des outils qui pourraient nous aider à éviter les situations de GIGO (Garbage In Garbage Out) sous la forme de filtres généralement.

En termes de prétraitement des données, les opérations suivantes ont été effectuées sur chaque version de ANT :

- Regroupement des bogues bloquants, critiques et majeurs pour former les fautes de sévérité élevée. D'un autre côté, les fautes de sévérité normale, mineure et triviale formeront les fautes de sévérité normale.
- Binarisation du nombre de fautes d'une classe donnée dans un nouvel attribut appelé « BugsBinary », c'est-à-dire 0 signifie que la classe est faiblement défectueuse et 1 fortement défectueuse.
- Dans un souci de compatibilité lors de l'utilisation des algorithmes d'apprentissage automatique, la transformation du filtre type nominal a été appliquée à la classe (attribut) à prédire.
- Afin d'améliorer les résultats de la prédiction, une normalisation linéaire a été réalisée sur les modèles pour obtenir des valeurs dans l'intervalle [-1, 1].
- Nous explorons également la contribution de techniques de stratification comme l'oversampling en appliquant le filtre Smote 35-65 % comme indiqué dans [63] et la duplication des classes fautives [9].

3.6 Machine Learning

Dans cette section, nous présentons une brève description des algorithmes de Machine Learning que nous avons utilisés en nous aidant de l'API de Weka [64] pour le langage Java. Les algorithmes d'apprentissage machine sont communément utilisés en classification des classes fautives afin d'appréhender les relations entre la métrique du code source (attributs) et les défauts (la classe à prédire ou le target). Ces algorithmes passent par une première phase d'entraînement en utilisant les ensembles de données des métriques et des fautes. Dans notre étude, nous avons utilisé les algorithmes d'apprentissage suivants: Naïve Bayes (NB), J48, Random Forest (RF), Support Vectors machines (SVM) et Multilayer Perceptron (ANN).

3.6.1 Naïve bayes

L'algorithme des réseaux bayésiens classifie les instances données (classes dans notre cas) par la construction d'un graphe orienté, où des modèles de risque seront représentés par les nœuds et leurs indépendances sous forme de liens, pour trier les classes défectueuses des non défectueuses [65]. Il peut être utilisé sous différentes variantes. L'une des plus populaires est la Naïve Bayes Network que nous avons choisie pour notre étude qui applique des assumptions d'indépendances moins fortes que le classificateur Naïve Bayes.

3.6.2 Arbres de décisions

Les algorithmes J48 et Random Forest sont utilisés pour la construction d'arbres de décision. Ils calculent l'efficacité avec laquelle chaque attribut répartit les données (dans notre cas comme classe sujette à contenir des fautes ou non dans une première expérience puis en niveaux de sévérité) en déterminant les seuils discriminants les catégories de la cible. L'arbre de décision qui en résulte est facile à comprendre, car il est suffisamment explicite [3]. Weka inclut un outil permettant la visualisation d'arbres de décision. Il y a deux principaux avantages quant à l'exploration de données visuelles par rapport au forage de données en plus de l'implication directe de l'utilisateur [66]: elle peut gérer facilement des données non homogènes et très bruyantes. De plus, la visualisation est intuitive et ne nécessite aucune compréhension des paramètres des algorithmes, mathématiques ou statistiques complexes. Ce type d'algorithmes a été également utilisé dans de nombreuses études traitant la prédiction de fautes [3, 15, 67].

3.6.3 Machines à vecteurs de support

L'algorithme Support Vector Machine est fondé sur la théorie de l'apprentissage statistique, ce qui le rend parfait pour à la fois la régression et la classification. Cet algorithme alloue moins de poids aux individus qui sont loin de la tendance. Il est aussi adapté aux données qui ne suivent pas une distribution linéaire [3]. Malhotra et al. [17] le décrivent comme le meilleur algorithme d'apprentissage machine à utiliser. Il a également été utilisé dans d'autres études traitant de la prédiction de la prédisposition aux fautes [3, 68].

3.6.4 Réseaux de neurones artificiels

Les réseaux neuronaux artificiels (ANN) sont divers et largement utilisés dans plusieurs domaines. Dans notre cas, un Perceptron multicouches (feedforward ANN avec un algorithme de rétropropagation) est utilisé, comme dans [68]. Cette topologie ANN spécifique consiste à avoir plusieurs couches de neurones, où chaque couche peut avoir un nombre différent de neurones. Chaque neurone de chaque couche est lié aux neurones des couches précédente et suivante. Le réseau est d'abord entraîné à l'aide

de données d'apprentissage, et l'algorithme de rétro-propagation met à jour les poids des neurones. Plusieurs études ont utilisé cet algorithme pour la prédition de fautes [15, 17, 65, 67].

3.7 Méthodes d'Évaluation

Nous avons utilisé les algorithmes d'apprentissage machine mentionnés précédemment durant nos expérimentations pour construire des modèles de prédition de fautes logicielles en veillant à appliquer une validation croisée 10-fold. Pour évaluer l'efficacité de la prédition de modèles de prédition, nous avons utilisé l'AUC (l'aire sous la courbe) et la métrique de la moyenne géométrique (g-mean), ce qui peut être facilement calculé à l'aide de la matrice de confusion résultant de la classification comme illustré dans la figure suivante:

Classifié	Actuel	
	Fautive	Non-Fautive
Fautive	Vrais positifs (TP)	Faux positifs (FP)
Non-Fautive	Faux négatifs (FN)	Vrais négatifs (TN)

Table 3.2: Matrice de Confusion

3.7.1 10-Fold validation

Le 10-Fold Validation est une technique de validation dans laquelle l'ensemble de données de départ est partitionné en 10 sous-ensembles de même taille. Neuf de ces échantillons serviront de données d'entrées pour la phase d'entraînement tandis que le dernier servira d'ensemble de test pour évaluer et valider les performances de l'algorithme d'apprentissage. Un processus itératif (10 itérations en tout) est mis en place, dans lequel, il y aura variation de sous-ensembles tests et des échantillons d'entraînement. Une matrice de confusion moyenne sera générée à la toute fin de ce processus de validation.

3.7.2 G-mean

Le g-mean a été défini spécifiquement pour la classification des données non balancées [17] et a également été utilisé dans d'autres études sur la prédition de la prédisposition aux fautes [14, 17]. Les deux métriques : FPR (taux de faux positifs) et FNR (taux de faux négatifs) sont calculées comme suit :

$$FPR = \frac{FP}{FP+TN}$$

$$FNR = \frac{FN}{FN+TP}$$

Le FPR donne le pourcentage des faux positifs parmi toutes les valeurs négatives réelles. En revanche, la métrique FNR donne le pourcentage des faux négatifs parmi toutes les valeurs positives réelles. Le plus faible est chacune de ces deux métriques, meilleure est la classification.

En ce qui concerne le calcul du g-mean, dans le cas d'une classification binaire, la précision des résultats positifs (TPR) et la précision des négatifs (TNR) [17] (qui sont les paramètres opposés de FNR et FPR respectivement) sont d'abord calculées en premier. Contrairement au FPR et FNR, où il vaut mieux avoir de faibles valeurs, plus élevés sont le TPR et le TNR, le g-mean sera alors plus élevé et meilleure sera la classification. La métrique du g-mean sera satisfaisante si le TPR et le TNR sont tous deux élevés, sinon ce ne sera pas le cas. Les calculs se font comme suit :

$$\begin{aligned} TPR &= 1 - FNR = \frac{TP}{TP+FN} \\ TNR &= 1 - FPR = \frac{TN}{TN+FP} \\ g - mean &= \sqrt{TPR \times TNR} \end{aligned}$$

Dans le cas d'une classification multilabels, il existe une autre formule pour le calcul de g-mean [69] avec l'utilisation du rappel pour chaque classe R_i , où k est le nombre de catégories distinctes. Le tableau suivant décrit une matrice de confusion d'une prédiction à k classes, où C_i représente la $i^{\text{ème}}$ classe :

		classe Prédite			
		C_1	C_2	...	C_k
Vraie Classe	C_1	n_{11}	n_{12}	...	n_{1k}
	C_2	n_{21}	n_{22}	...	n_{2k}
.
.
.
C_k	n_{k1}	n_{k2}	n_{kk}

Table 3.3: Matrice de Confusion Multi-Classes

Le rappel de la $i^{\text{ème}}$ classe, dénoté R_i est donné par:

$$R_i = \frac{n_{ii}}{\sum_{j=1}^k n_{ij}}$$

Par la suite, le g-mean peut-être calculé avec la formule ci-dessous:

$$g - mean = \left(\prod_{i=1}^k R_i \right)^{1/k}$$

Il est courant d'utiliser une correction ϵ , surtout quand le nombre de classes est conséquent. Comme nous avons trois classes distinctes, quand $R_i = 0$ dans notre étude, nous avons choisi de le remplacer par $\epsilon = 0.001$.

Nous avons suivi l'interprétation suivante de g-mean pour nos résultats :

- $g - mean < 0.5$ signifie mauvaise classification.
- $0.5 < g - mean < 0.6$ signifie faible classification.
- $0.6 < g - mean < 0.7$ signifie classification acceptable.
- $0.7 < g - mean < 0.8$ signifie bonne classification.
- $0.8 < g - mean < 0.9$ signifie très bonne classification.
- $0.9 < g - mean$ signifie excellente classification.

3.7.3 Area under the curve

L'aire sous la courbe (AUC) de la fonction d'efficacité du récepteur (ROC) est utilisée dans l'évaluation des classificateurs. ROC est une fonction paramétrée de la sensibilité et 1- la spécificité en fonction du seuil variant entre 0 et 1. La courbe ROC est tracée donc au moyen de deux variables: une variable binaire et une autre continue. La variable binaire est l'occurrence d'une classe sujette à contenir des fautes ou non. La variable continue sera l'un de nos modèles de risque. Le modèle est évalué en considérant toutes ses valeurs comme des seuils potentiels qui peuvent être utilisés pour décider si une classe est défectueuse (\geq seuil) ou non défectueuse ($<$ seuil). Pour chaque seuil potentiel, une matrice de confusion est créée comme indiqué dans la table précédente. Chaque matrice peut être utilisée pour calculer deux mesures importantes: la sensibilité et la spécificité, respectivement TPR et TNR comme présenté plus haut.

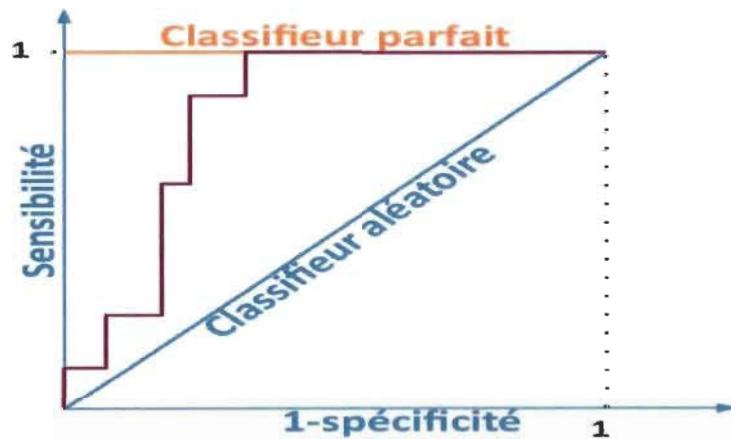


Figure 3.3: Courbe ROC

L'aire sous la courbe ROC (AUC) montre une analyse de compromis visuel entre le taux de classes correctement classées comme sujettes aux erreurs et le taux de classes incorrectement classées comme non sujettes aux erreurs. L'AUC est une valeur unique qui évalue le pouvoir discriminant dans la courbe entre les classes défectueuses et non défectueuses [70].

L'aire sous la ligne diagonale vaut 0,5. Dans ce cas précis, on obtient un classificateur complètement aléatoire. Par conséquent, la courbe qui discrimine bien entre les deux classes devrait avoir une aire supérieure à 0,5 et devrait s'approcher du coin supérieur gauche, c'est-à-dire 1. Hosmer et Lemeshow ont proposé l'utilisation des règles suivantes pour évaluer la performance des classificateurs [71]:

- $AUC = 0.5$ signifie mauvaise classification, assimilée à une classification aléatoire.
- $0.5 < AUC < 0.6$ signifie faible classification.
- $0.6 < AUC < 0.7$ signifie classification acceptable.
- $0.7 < AUC < 0.8$ signifie bonne classification.
- $0.8 < AUC < 0.9$ signifie très bonne classification.
- $0.9 < AUC$ signifie excellente classification.

Chapitre. 4

EXPÉRIMENTATIONS ET RÉSULTATS

4.1 Introduction

Comme nous l'avons mentionné plutôt, il n'y a aucun travail qui a investigué le potentiel du modèle de risque considéré pour la prédition des fautes de logiciel. Dans cette optique-là, nous avons effectué une étude empirique sur le modèle de risque considéré pour la prédition de la prédisposition aux fautes de logiciel sur le système ANT dans une première expérience. Notre seconde expérimentation vise à vérifier le potentiel de nos modèles (différentes variantes) à la prédition du niveau de sévérité. Comme expérience finale, on s'attaque à la prédition de risque. Dans ce qui suit, nous donnons un récapitulatif de toutes nos expérimentations.

4.2 Classification Binaire

4.2.1 Description de l'expérimentation

Tout d'abord, nous avons étudié le potentiel des modèles de risque (variantes évaluées) mentionnés précédemment dans la classification binaire des classes fautives. Le but de cette classification est de construire un modèle capable de prédire si une classe ou un module est défectueux ou non.

En outre, nous avons exploré la contribution des méthodes de stratification. Ici, nous donnons les statistiques après l'application de la duplication de fautes et du filtre Smote, des techniques de sur échantillonnage. Vous noterez que nous n'avons touché ni à ANT 1.3 ni à la version 1.5, car le ratio des classes défectueuses est déjà supérieur à 35%. Comme expliqué dans la sous-section sur les prétraitements, nous avons suivi une stratification de 35 à 65%.

Système	#Classes-D	#Fautes-D	Ratio-D	#Classes-B	#Fautes-B	Ratio-B
ANT 1.3	201	135	67.16%	-	-	-
ANT 1.4	178	38	21.35%	216	76	35.18%
ANT 1.5	366	179	48.91%	-	-	-
ANT 1.6	368	61	16.58%	473	166	35.09%
ANT 1.7	776	101	13.01%	1039	364	35.05%

Table 4.1: Statistiques ANT Oversampling

4.2.2 Rappel des questions de recherche

Pour rappel, l'objectif de ces expérimentations est de répondre aux questions de recherche 1 et 2. Est-ce que nos modèles de risque sont capables de prédire les fautes?

4.2.3 Résultats

Le tableau ci-dessous décrit les résultats de la meilleure performance du modèle de risque qui est RT (LOC, Fan-in). Le tableau présente les résultats d'avant, d'après duplication des fautes et d'après suréchantillonnage. Les meilleures performances ont été mises en gras pour mieux attirer l'attention dessus. Il est à noter que les résultats des autres modèles étudiés peuvent être trouvés dans l'annexe A de ce mémoire.

On peut constater que les meilleurs résultats ont été obtenus grâce aux variantes: RT (LOC, Ce), RT (LOC, Fan-in), RT (LOC, RFC), RT (RFC, Ca) et RT (WMC, CBO); et que les variantes incluant la métrique LOC sont un cran au-dessus niveau performance.

		TPR	TNR	g-m	AUC	TPR-D	TNR-D	g-m-D	AUC-D	TPR-B	TNR-B	g-m-B	AUC-B
ANT 1.3	NB	0.733	0.758	0.745	0.781	0.741	0.909	0.821	0.867	-	-	-	-
	J48	0.733	0.773	0.753	0.732	0.852	0.727	0.787	0.815	-	-	-	-
	RF	0.633	0.682	0.657	0.730	0.919	0.682	0.792	0.885	-	-	-	-
	RLog	0.667	0.818	0.739	0.836	0.837	0.788	0.812	0.892	-	-	-	-
	SVM	0.400	0.955	0.618	0.677	0.993	0.076	0.275	0.530	-	-	-	-
	ANN	0.733	0.818	0.774	0.830	0.875	0.732	0.817	0.888	-	-	-	-
ANT 1.4	NB	0.237	0.971	0.480	0.729	0.340	0.943	0.566	0.740	0.329	0.929	0.553	0.715
	J48	0.053	0.936	0.223	0.484	0.480	0.850	0.639	0.667	0.566	0.807	0.676	0.655
	RF	0.289	0.807	0.525	0.655	0.560	0.821	0.678	0.769	0.513	0.700	0.599	0.702
	RLog	0.184	0.971	0.423	0.743	0.260	0.957	0.499	0.769	0.355	0.900	0.565	0.744
	SVM	0.000	1.000	0.000	0.500	0.220	0.971	0.462	0.596	0.197	0.971	0.438	0.584
	ANN	0.263	0.921	0.494	0.689	0.440	0.886	0.624	0.758	0.472	0.847	0.635	0.735
ANT 1.5	NB	0.226	0.947	0.463	0.762	0.464	0.914	0.651	0.810	-	-	-	-
	J48	0.708	0.674	0.691	0.700	0.765	0.770	0.767	0.736	-	-	-	-
	RF	0.528	0.727	0.620	0.703	0.810	0.717	0.762	0.814	-	-	-	-
	RLog	0.292	0.925	0.520	0.780	0.615	0.861	0.728	0.836	-	-	-	-
	SVM	0.075	0.979	0.271	0.527	0.441	0.920	0.637	0.681	-	-	-	-
	ANN	0.585	0.807	0.687	0.790	0.740	0.797	0.770	0.824	-	-	-	-
ANT 1.6	NB	0.111	0.967	0.328	0.694	0.131	0.961	0.355	0.751	0.277	0.928	0.507	0.717
	J48	0.000	1.000	0.000	0.469	0.000	1.000	0.000	0.489	0.669	0.756	0.711	0.713
	RF	0.289	0.876	0.503	0.615	0.557	0.883	0.701	0.747	0.566	0.730	0.643	0.710
	Rlog	0.044	0.984	0.208	0.717	0.115	0.977	0.335	0.761	0.307	0.919	0.531	0.727
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500	0.108	0.964	0.323	0.536
	ANN	0.000	1.000	0.000	0.694	0.061	0.982	0.254	0.753	0.447	0.866	0.624	0.720
ANT 1.7	NB	0.214	0.969	0.455	0.787	0.366	0.964	0.594	0.805	0.349	0.930	0.570	0.790
	J48	0.000	0.999	0.000	0.552	0.248	0.993	0.496	0.754	0.673	0.824	0.745	0.800
	RF	0.243	0.929	0.475	0.653	0.545	0.910	0.704	0.801	0.585	0.778	0.675	0.791
	RLog	0.143	0.991	0.376	0.817	0.248	0.987	0.495	0.846	0.426	0.919	0.625	0.820
	SVM	0.000	1.000	0.000	0.500	0.079	0.994	0.280	0.537	0.319	0.941	0.548	0.630
	ANN	0.109	0.994	0.346	0.817	0.347	0.970	0.580	0.848	0.598	0.849	0.721	0.813

Table 4.2: Résultats RT (LOG₂ Fan-in) Classification Binaire

4.2.4 Discussion

On constate que les meilleures performances ont été obtenues grâce aux algorithmes ANN et Random Forest. Nous avons une très bonne classification pour la version ANT 1.3 (avec un g-mean de 0.821); sans doute vu l'abondance de fautes dans cette version précoce. Pour le reste des versions, nous avons quand même de bonnes classifications (un g-mean > 0.745). La duplication de fautes apporte un grand pont de performance surtout pour les versions 1.4, 1.6 et 1.7 où le ratio de défauts est moindre (avoisine les 20% pour la version 1.4 et seulement les 11% pour les deux autres versions). L'application du filtre Smote, quant à lui, apporte une nette amélioration par rapport à la duplication, mais pas assez significative. Cette petite chute de performances serait due au fonctionnement des algorithmes d'apprentissage, qui lors de la phase d'entraînement, ont besoin d'informations sur les classes fautives.

Les meilleurs résultats ont été obtenus avec la variante du modèle de risque RT (LOC, Fan-in) qui englobe une métrique de taille. On a déjà retrouvé dans la littérature que LOC (Nombre de lignes de code) est un bon indicateur de la probabilité d'occurrence de fautes. C'est également pour cette raison que parmi les cinq modèles les plus concluants, à savoir : RT (LOC, Ce), RT (LOC, Fan-in), RT (LOC, RFC), RT (RFC, Ca) et RT (WMC, CBO), les modèles qui comportent la métrique LOC présentent de meilleures performances. Il paraît logique que plus le code est volumineux, plus il est susceptible de contenir des fautes. De plus, nous avons vu que Fan-in est un bon indicateur des classes qui nécessitent une restructuration et un plus grand effort de test; ce qui voudrait dire qu'il indique la présence de fautes. On peut voir que la fusion de deux métriques: de taille et de couplage, plus particulièrement LOC et Fan-in, dans notre modèle de risque est satisfaisante pour la prédiction de présence de fautes.

4.3 Détection du Niveau de Sévérité

4.3.1 Introduction

Durant cette deuxième série d'expérimentations, nous nous sommes penchés sur la prédiction du niveau de sévérité des fautes. Nous avons restreint le nombre de nos modèles de risque à seulement cinq; les plus performants pour la classification binaire que sont: RT (LOC, Ce), RT (LOC, Fan-in), RT (LOC, RFC), RT (RFC, Ca) et RT (WMC, CBO). Comme le suréchantillonnage n'apportait pas une grande amélioration, nous nous contentons de la duplication de fautes, largement utilisée dans la littérature, comme technique de stratification. De plus, en utilisant le filtre Smote nous risquons fortement de biaiser nos données d'entrées. C'est pourquoi nous nous en abstiens.

Dans l'objectif de mieux comprendre le comportement du modèle de risque étudié, nous avons choisi de prédire les niveaux de sévérité séparément, puis dans un second temps rassemblés dans une classification multi-classes. Nous donnons plus de détails sur cette deuxième phase de notre étude dans ce qui suit.

4.3.2 Rappel des questions de recherche

L'objectif de ces expérimentations est de répondre aux questions de recherche 3 et 4. Est-ce que nos modèles de risque peuvent prédire le niveau de sévérité des fautes?

4.3.3 Détection des fautes de sévérité normale

Nous voulons savoir ce que donne notre modèle de risque dans le cas de la prédiction de fautes de sévérité normale telle que définie précédemment. Pour ce faire, on réalise une classification binaire où la cible est un attribut binaire « NormalBugs » qui vaut 1 lorsqu'une classe donnée contient au moins une faute de sévérité normale sinon il vaudra 0. La table suivante nous donne quelques statistiques sur les défauts de sévérité normale:

Système	#Classes	#Fautives	Ratio	#Classes-D	#Fautives-D	Ratio-D
ANT 1.3	126	55	43.64%	285	214	75.09%
ANT 1.4	178	29	16.29%	186	37	19.89%
ANT 1.5	293	64	21.84%	330	101	30.61%
ANT 1.6	352	32	9.09%	362	42	11.60%
ANT 1.7	675	56	7.51%	765	76	9.93%

Table 4.3: Statistiques sur les Fautes de Sévérité Normales pour ANT

Les meilleurs résultats sont obtenus avec la variante RT (LOC, Fan-in). Le tableau suivant présente ses résultats. Les résultats des modèles restants peuvent être retrouvés plus loin dans l'annexe B.

On peut voir que les meilleurs résultats ont été obtenus grâce aux variantes : RT (LOC, Ce), RT (LOC, Fan-in) et RT (LOC, RFC). Toutes ces variantes incluent la métrique LOC. Les deux premières variantes avec une métrique de couplage (Ce et Fan-in) sont un peu plus performantes.

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meanD	AUCD
ANT 1.3	NB	0.770	0.723	0.724	0.811	0.804	0.803	0.803	0.873
	J48	0.786	0.801	0.791	0.724	0.881	0.782	0.825	0.794
	RF	0.690	0.682	0.686	0.795	0.912	0.754	0.818	0.923
	RLog	0.778	0.746	0.751	0.854	0.874	0.770	0.815	0.892
	SVM	0.714	0.639	0.609	0.677	0.751	0.249	0.432	0.500
	ANN	0.760	0.739	0.758	0.843	0.853	0.735	0.787	0.882
ANT 1.4	NB	0.831	0.301	0.407	0.654	0.812	0.384	0.481	0.714
	J48	0.837	0.163	0.369	0.481	0.796	0.218	0.163	0.481
	RF	0.697	0.247	0.333	0.530	0.769	0.577	0.654	0.680
	RLog	0.826	0.189	0.184	0.682	0.812	0.364	0.456	0.733
	SVM	0.837	0.163	0.369	0.500	0.801	0.199	0.159	0.500
	ANN	0.821	0.160	0.047	0.679	0.817	0.386	0.483	0.722
ANT 1.5	NB	0.785	0.321	0.368	0.755	0.742	0.471	0.496	0.813
	J48	0.782	0.218	0.413	0.479	0.785	0.761	0.772	0.746
	RF	0.713	0.413	0.496	0.680	0.803	0.758	0.778	0.834
	RLog	0.782	0.297	0.326	0.771	0.758	0.533	0.577	0.830
	SVM	0.778	0.217	0.411	0.498	0.733	0.434	0.438	0.584
	ANN	0.783	0.432	0.760	0.784	0.780	0.691	0.729	0.822
ANT 1.6	NB	0.895	0.202	0.349	0.750	0.873	0.239	0.372	0.818
	J48	0.909	0.091	0.083	0.475	0.884	0.116	0.320	0.481
	RF	0.855	0.311	0.478	0.733	0.870	0.611	0.721	0.828
	RLog	0.901	0.118	0.176	0.779	0.870	0.156	0.216	0.824
	SVM	0.909	0.091	0.083	0.500	0.884	0.116	0.320	0.500
	ANN	0.909	0.091	0.083	0.688	0.879	0.115	0.101	0.801
ANT 1.7	NB	0.914	0.288	0.474	0.783	0.907	0.404	0.576	0.801
	J48	0.921	0.108	0.188	0.671	0.914	0.276	0.443	0.715
	RF	0.875	0.284	0.464	0.631	0.892	0.578	0.708	0.784
	RLog	0.930	0.223	0.399	0.824	0.918	0.324	0.498	0.842
	SVM	0.925	0.075	0.263	0.500	0.901	0.111	0.115	0.506
	ANN	0.927	0.185	0.363	0.808	0.911	0.250	0.498	0.761

Table 4.4: Résultats de RT (LOC, Fan-in) pour la Prédiction de Bogues de Sévérité Normale

On constate que les meilleures performances ont été obtenues grâce aux algorithmes d'arbres de décisions Random Forest et J48. Pareillement que la classification binaire, nous avons une très bonne classification pour la version ANT 1.3 (un g-mean de 0.825). Pour le reste des versions, nous avons également une bonne classification. Grâce à la duplication de fautes, la performance connaît un grand bond, surtout pour les versions 1.4, 1.6 et 1.7 où le ratio de défauts est moindre.

Les meilleurs résultats ont été obtenus avec la variante du modèle de risque RT (LOC, Fan-in) composée d'une métrique de couplage. Le couplage à une forte relation

avec la propagation des fautes et de ce fait sur l'impact de ces dernières. Plus les méthodes d'une classe sont sollicitées, plus le risque de propagation de ses fautes est élevé. On peut également constater que les variantes avec des métriques de complexité donnent de moins bons résultats pour la prédiction de fautes de sévérité normale. Les mesures de complexité ne sont pas capables de détecter ce niveau de sévérité. En effet, les métriques WMC et RFC sont étroitement liées à la complexité et au grand effort de test. Alors que, les fautes de sévérité normale requièrent moins d'efforts pour être fixées [72]; contrairement aux fautes de sévérité élevée auxquelles on assigne d'ailleurs une priorité plus importante.

4.3.4 Détection des fautes de sévérité élevées

De l'autre part, nous avons suivi le même procédé avec l'attribut binaire « HighBugs» qui vaut 1 lorsqu'une classe donnée contient au moins une faute de sévérité élevée sinon il vaudra 0. La table suivante nous donne quelques statistiques sur les fautes de sévérité élevée:

Système	#Classes	#Fautives	Ratio	#Classes-D	#Fautives-D	Ration-D
ANT 1.3	126	60	47.62%	279	147	52.69%
ANT 1.4	178	12	6.74%	179	13	7.26%
ANT 1.5	293	55	18.77%	298	60	20.13%
ANT 1.6	352	19	5.40%	-	-	-
ANT 1.7	745	22	2.95%	748	25	3.34%

Table 4.5: Statistiques sur les Défauts de Sévérité Élevée pour ANT

La duplication de fautes n'a pas pu être effectuée pour la version 1.6 parce qu'elle contient 19 fautes de sévérité élevée, présentes dans 19 classes distinctes. Nous ne présentons donc pas de résultats pour cette version après duplication pour ce cas de figure particulier.

Les meilleurs résultats sont cette fois-ci obtenus avec la variante RT (RFC, Ca) et RT (LOC, RFC). Nous avons pu obtenir une très bonne classification pour la version 1.3 ($g\text{-mean} = 0.814$), mais seulement des classifications acceptables pour les versions 1.4 et 1.5. Quant aux deux dernières versions, la classification est mauvaise, si on prend en considération le $g\text{-mean}$. Cela peut être expliqué par la rareté de fautes de sévérité élevée présentes dans ces deux versions qui même après duplication reste moindre à 5%. L'équipe d'assurance qualité a certainement concentré ses efforts sur la correction des fautes de sévérité élevée, ce qui explique leur quasi-disparition.

On constate que les meilleurs modèles présentent tous deux la métrique RFC dans leur composition. Du niveau performance, le modèle RT (RFC, Ca) l'emporte de peu. La métrique de taille n'est pas un bon indicateur des fautes de sévérité élevée. Par contre, les métriques de complexité et de couplage accomplissent ce rôle. Plus une classe est complexe et au couplage excessif, plus elle a un impact important. Le reste

des résultats obtenus grâce aux autres modèles de risque peuvent être retrouvés dans l'annexe C.

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meanD	AUCD
ANT 1.3	NB	0.517	0.894	0.680	0.798	0.741	0.818	0.779	0.873
	J48	0.850	0.727	0.786	0.717	0.884	0.727	0.802	0.769
	RF	0.650	0.697	0.673	0.769	0.912	0.697	0.797	0.876
	RLog	0.667	0.818	0.739	0.817	0.912	0.727	0.814	0.882
	SVM	0.400	0.924	0.608	0.662	1.000	0.000	0.000	0.500
	ANN	0.573	0.809	0.770	0.762	0.904	0.656	0.781	0.872
ANT 1.4	NB	0.167	0.994	0.407	0.767	0.077	0.988	0.276	0.759
	J48	0.000	0.976	0.000	0.408	0.000	0.994	0.000	0.446
	RF	0.333	0.958	0.565	0.726	0.462	0.952	0.663	0.830
	RLog	0.000	0.994	0.000	0.791	0.000	0.994	0.000	0.775
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.000	0.997	0.000	0.645	0.000	0.997	0.000	0.609
ANT 1.5	NB	0.182	0.950	0.416	0.699	0.217	0.945	0.453	0.722
	J48	0.055	0.958	0.229	0.492	0.267	0.924	0.496	0.590
	RF	0.400	0.887	0.596	0.695	0.467	0.887	0.643	0.736
	RLog	0.036	0.966	0.187	0.724	0.133	0.962	0.358	0.740
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.016	0.967	0.133	0.707	0.146	0.959	0.454	0.688
ANT 1.6	NB	0.000	0.991	0.000	0.587	-	-	-	-
	J48	0.000	1.000	0.000	0.473	-	-	-	-
	RF	0.053	0.952	0.224	0.447	-	-	-	-
	RLog	0.000	0.997	0.000	0.618	-	-	-	-
	SVM	0.000	1.000	0.000	0.500	-	-	-	-
	ANN	0.000	0.999	0.000	0.577	-	-	-	-
ANT 1.7	NB	0.136	0.988	0.367	0.671	0.200	0.989	0.445	0.671
	J48	0.000	1.000	0.000	0.462	0.000	0.997	0.000	0.511
	RF	0.136	0.976	0.365	0.646	0.200	0.975	0.442	0.731
	RLog	0.045	0.999	0.213	0.785	0.080	0.999	0.283	0.794
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.000	0.999	0.000	0.680	0.041	0.999	0.346	0.693

Table 4.6: Résultats de RT (RFC, Ca) pour la Prédition de Bogues de Sévérité Élevée

4.3.5 Détection du niveau de sévérité des fautes

Dans ce chapitre, nous avons essayé de construire des modèles capables de prédire le niveau de gravité d'une classe donnée dans un seul modèle de classification multi-classes. C'est-à-dire, un seul modèle qui peut prédire si une classe n'est pas défectueuse ou bien défectueuse avec un niveau de gravité élevé ou normal. Pour ce faire, nous avons construit une nouvelle variable cible que nous avons nommée « Severity » comme suit:

$$\text{Severity} = \begin{cases} 0 & \text{Non Fautive} \\ 1 & \text{Bogues de Sévérité Normale} \\ 2 & \text{Bogues de Sévérité Élevée} \end{cases}$$

La table suivante nous donne quelques statistiques sur les niveaux de sévérité pour le système ANT :

Système	#C	#S=0	#S=1	#S=2	#C-D	#S-D=0	#S-D=1	#S-D=2
ANT 1.3	126	66	0	60	213	66	0	147
ANT 1.4	178	140	26	12	185	140	32	13
ANT 1.5	293	193	45	55	321	193	68	60
ANT 1.6	352	307	26	19	358	307	32	19
ANT 1.7	745	675	48	22	765	675	65	25

Table 4.7: Statistiques sur le Niveau de Sévérité pour ANT

Nous avons obtenu une très bonne classification pour la version ANT 1.3 (avec un g-mean égal à 0.827), mais que des résultats acceptables pour les autres versions, mise à part la version 1.6 où nous avons une mauvaise classification. Il ne faut pas oublier que c'est pour cette version que la duplication de faute apporte le moins, vu qu'il n'y a que six classes d'ajoutées. Plus de résultats peuvent être trouvés en consultant l'annexe D.

Pour cette expérimentation, il s'agit du modèle RT (LOC, RFC) qui donne les meilleurs résultats en concurrence avec RT (RFC, Ca). La force de ce modèle est qu'il se compose d'une métrique de taille 'LOC' et d'une métrique de Complexité/Couplage 'RFC', ce qui lui confère un grand potentiel informatif. Comme nous l'avons vu dans les deux sections précédentes: les modèles comportant LOC sont de bons prédicteurs de fautes de sévérité normale alors que ceux qui ne la comportent pas n'arrivent pas à bien détecter ce niveau de sévérité. De l'autre côté, concernant le niveau de sévérité élevée, les modèles de risque qui encapsulent la métrique RFC sont de bons prédicteurs alors que la métrique de taille LOC n'arrive pas à détecter les fautes de sévérité élevée. De plus, la deuxième composante était dans les deux cas une métrique de couplage. C'est là que RFC, étant une métrique de couplage et de complexité à la fois, arrive parfaitement à créer la conjoncture entre prédiction des deux niveaux de sévérité en s'alliant à la métrique LOC.

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meanD	AUCD
ANT 1.3	NB	0.762	0.747	0.739	0.815	0.808	0.813	0.810	0.879
	J48	0.730	0.726	0.729	0.729	0.779	0.726	0.749	0.797
	RF	0.675	0.675	0.675	0.766	0.84	0.736	0.776	0.879
	RLog	0.762	0.756	0.757	0.834	0.826	0.797	0.811	0.892
	SVM	0.754	0.738	0.728	0.746	0.831	0.824	0.827	0.827
	ANN	0.683	0.667	0.764	0.763	0.83	0.787	0.812	0.885
ANT 1.4	NB	0.781	0.371	0.325	0.728	0.768	0.434	0.393	0.755
	J48	0.787	0.213	0.316	0.471	0.751	0.273	0.236	0.575
	RF	0.657	0.434	0.398	0.665	0.719	0.604	0.600	0.738
	RLog	0.781	0.331	0.325	0.737	0.773	0.403	0.382	0.759
	SVM	0.787	0.213	0.316	0.500	0.757	0.243	0.316	0.500
	ANN	0.784	0.259	0.330	0.733	0.773	0.403	0.382	0.757
ANT 1.5	NB	0.672	0.455	0.000	0.736	0.614	0.538	0.442	0.775
	J48	0.655	0.427	0.000	0.635	0.620	0.700	0.534	0.715
	RF	0.584	0.607	0.362	0.677	0.660	0.740	0.680	0.750
	RLog	0.666	0.436	0.000	0.760	0.626	0.560	0.513	0.779
	SVM	0.659	0.341	0.000	0.500	0.611	0.448	0.339	0.528
	ANN	0.674	0.563	0.000	0.756	0.626	0.674	0.691	0.768
ANT 1.6	NB	0.849	0.222	0.294	0.678	0.832	0.223	0.279	0.713
	J48	0.872	0.128	0.316	0.468	0.858	0.142	0.316	0.486
	RF	0.787	0.256	0.342	0.602	0.802	0.420	0.436	0.687
	RLog	0.866	0.147	0.248	0.703	0.846	0.191	0.280	0.726
	SVM	0.872	0.128	0.316	0.500	0.858	0.142	0.316	0.500
	ANN	0.872	0.128	0.316	0.679	0.858	0.142	0.316	0.722
ANT 1.7	NB	0.887	0.298	0.385	0.800	0.877	0.408	0.425	0.812
	J48	0.906	0.120	0.254	0.591	0.894	0.264	0.341	0.750
	RF	0.839	0.296	0.332	0.684	0.856	0.553	0.600	0.776
	RLog	0.907	0.223	0.334	0.817	0.890	0.302	0.375	0.834
	SVM	0.906	0.094	0.316	0.500	0.882	0.118	0.316	0.500
	ANN	0.906	0.111	0.316	0.817	0.891	0.287	0.368	0.831

Table 4.8: Résultats de RT (LOC, RFC) pour la Prédiction du Niveau de Sévérité

4.3.6 Conclusion

Dans l'ensemble, nous constatons que le modèle de risque présente de bons résultats pour la classification du niveau de sévérité des fautes. Il présente de très bonnes performances pour la détection de fautes de sévérité normale, mais il est moins précis pour la détection de fautes de sévérité élevée. Malgré cela, les résultats restent bons pour la prédiction des fautes de sévérité élevée. Ce qui est tout à fait normal, vu le petit nombre de fautes de sévérité élevée présents dans le système ANT.

Les meilleurs résultats ont été obtenus avec la variante du modèle de risque RT

(LOC, RFC) dont la force réside dans sa composition d'une métrique de taille et d'une métrique de complexité/couplage. Nos résultats coïncident avec la littérature quant aux choix des métriques, ce qui est assez bon signe. On constate que cette fois-ci également, que les meilleures performances ont été obtenues grâce aux algorithmes ANN et Random Forest.

4.4 Prédiction de Risque

4.4.1 Introduction

Dans la dernière partie de cette étude, nous introduisons la notion de prédiction du risque. Nous commençons par définir le risque empirique d'une classe [10], que nous dénotons RE, qui prend en considération à la fois la sévérité des fautes et leur nombre. Comme vous allez le constatez, il s'agit d'un attribut numérique continu. Nous présentons donc d'abord les algorithmes de régression que nous avons utilisés pour réaliser nos expérimentations. L'objectif est de prédire le risque empirique d'une classe à l'aide de nos modèles de risque théoriques comme données d'entrées. Nous avons également décidé d'explorer la piste d'imbrication de modèle de risque à deux dimensions pour aboutir à des modèles de risque multidimensionnels. Nous apportons plus d'éclaircissement à ce concept dans ce qui suit.

4.4.2 Rappel des questions de recherche

Avec cette dernière série d'expérimentations, nous tentons de répondre à la question de recherche 5. Nos modèles de risque peuvent-ils prédire le risque des classes logicielles?

4.4.3 Définition du risque empirique

Nous définissons le risque empirique (RE) d'une classe C donnée avec la formule suivante qui prend en compte à la fois le nombre de fautes et leurs différents niveaux de gravité respectifs.

$$RE(C) = \sum_{i=1}^6 S_i \times N_i$$

Où S_i est le poids donné au niveau de sévérité: 6 représente le plus haut niveau de gravité qui est bloquant et 1 correspond au plus bas, bogues triviaux. N_i sera le nombre de défauts observés pour le niveau de gravité i pour la classe C.

Le RE d'une classe est un entier positif qui englobe des informations sur le nombre de fautes contenues et leurs niveaux de sévérité respectifs. Une classe qui contient plusieurs fautes de sévérité élevée, possédera un risque empirique élevé. Un RE nul pour une classe signifie qu'elle ne contient pas de faute (ou du moins aucune faute détectée par les testeurs).

4.4.4 Modèle de risque multidimensionnel

L'idée derrière les modèles de risque multidimensionnels que nous dénotons RT^* est l'imbrication de modèles de risque à deux dimensions que nous avons présentés précédemment en suivant la même logique déjà décrite. Donc, au lieu de calculer la distance euclidienne entre deux métriques orientées objet et le point de risque maximum, nous allons calculer cette distance entre deux instances du modèle de risque à deux dimensions. Nous allons prendre RT_1 , un modèle de risque RT (m_1, m_2), sur l'axe des abscisses et RT_2 , un modèle de risque RT (m_3, m_4) sur l'axe des ordonnées.

Le modèle de risque multidimensionnel d'une classe donnée C est calculé comme suit:

$$RT_{RT_1, RT_2}^*(C) = \sqrt{(RT_1(C) - \max_i(RT_1(C_i)))^2 + (RT_2(C) - \max_i(RT_2(C_i)))^2}$$

Nous nous sommes inspirés de nos résultats obtenus précédemment pour choisir les combinaisons des modèles de risque à deux dimensions qui vont nous servir de support pour la construction des modèles multidimensionnels. La logique suivie est de prendre un des meilleurs modèles de risque qui prédit l'existence de faute (classification binaire) pour représenter l'axe P de probabilité de fautes d'un côté. De l'autre côté, un des meilleurs modèles de risque pour la prédiction du niveau de sévérité qui lui, va décrire l'axe I de l'impact des fautes. Nous avons construit les modèles de risque multidimensionnels suivants: RT^* (RT (LOC, Ce), RT (RFC, Ca)), RT^* (RT (LOC, WMC), RT (RFC, Ca)). Nous avons également exploré les variantes suivantes: RT^* (RT (LOC, Ca), RFC), RT^* (LOC, RT (RFC, Ca)) ainsi que RT^* (RT (LOC, RFC), RT (LOC, Ca), RT (LOC, Ca)) en utilisant la formule de la distance euclidienne à trois dimensions.

Ensuite, dans un objectif totalement exploratoire, nous avons investigué le potentiel de ces nouveaux modèles de risque multidimensionnels pour les expérimentations menées précédemment. Les résultats des modèles de risque multidimensionnels pour la classification binaire, la prédiction du niveau de sévérité des fautes ainsi que la prédiction du risque des classes logicielles peuvent être retrouvés en consultant leurs annexes respectives qui accompagnent ce mémoire.

4.4.5 Description de l'expérimentation

Dans cette partie, nous avons utilisé les algorithmes d'apprentissage automatique avec le schéma de régression et essayé de prédire le risque empirique d'une classe donnée en effectuant une validation croisée 10-fold à partir de nos modèles de risque théoriques.

De façon générale, la régression est une méthode statistique [73] modélisant le lien entre une variable dépendante Y (qui sera le risque empirique dans notre cas) et des variables explicatives indépendantes X_i (nos modèles de risque théorique) grâce à l'équation suivante:

$$Y = b_0 + b_1X_1 + b_2X_2 + \cdots + b_nX_n + \epsilon$$

On parle de régression simple, lorsque $n = 1$ et de régression multiple pour $n > 1$. Il est à noter que n représente le nombre de variables indépendantes. Nous avons effectué des régressions simples en variant le modèle de risque choisi. ϵ indique l'écart entre la valeur réelle de la variable dépendante Y et l'estimation du modèle de prédiction (i.e. l'erreur). Le facteur b_0 représente l'ordonnée à l'origine. Le coefficient b_i symbolise le poids de la variable X_i dans l'interprétation de la variable Y . Pour chaque coefficient, la p-value sera calculée et comparée au seuil $\alpha = 5\%$. Pour l'évaluation des modèles, nous donnons le coefficient de corrélation (r) dont la valeur appartient à l'intervalle $[0, 1]$. Plus le coefficient de corrélation est proche de 1, meilleur est le modèle de prédiction.

Nous avons utilisé: la régression linéaire (LR), le processus gaussien (GP), Random Forest (RF), M5P un algorithme d'arbres de décision, SmoReg et le réseau de neurones artificiels (ANN).

4.4.6 Résultats

Le tableau ci-dessous présente les résultats de la prédiction du risque logiciel. Nous donnons le coefficient de corrélation (r), Erreur Moyenne Absolue (MAE), erreur quadratique moyenne (RMSE), erreur absolue relative (RAE) et erreur quadratique relative (RSE) de la meilleure prédiction de modèle de risque d (LOC, RFC) après avoir expérimenté tous les modèles de risque nommés précédemment.

Plus de résultats peuvent être retrouvés en consultant l'annexe E. On peut y remarquer que les modèles les plus performants sont : RT*(RT (LOC, RFC), RT (LOC, Ca), RT (RFC, Ca)), RT (LOC, Fan-in) et RT (RFC, Ca). Le meilleur coefficient de corrélation obtenu est de 0.659 pour la version ANT 1.3. Pour la version ANT 1.5, il est de 0.538. Pour les deux versions ANT 1.4 et 1.7, il dépasse les 0.4 malgré le peu de fautes présentes dans ces versions. Les performances sont moindres pour la version 1.6 qui contient très peu de fautes. Pour rappel, les résultats donnés passent le test de significativité de la p-value comparée au seuil $\alpha = 5\%$. Nous pouvons également constater que l'erreur moyenne absolue est globalement très faible.

		r	MAE	RMSE	RAE	RSE
ANT 1.3	LR	0.651	5.004	7.211	66.902	75.478
	Gaussian	0.629	5.117	7.381	68.424	77.260
	RF	0.439	6.091	9.282	81.437	97.162
	M5P	0.651	5.004	7.211	66.902	75.478
	SmoReg	0.659	4.769	7.358	63.763	77.022
	ANN	0.557	5.659	8.059	75.669	84.354
ANT 1.4	RL	0.391	1.257	2.001	81.595	91.507
	Gaussian	0.355	1.287	2.033	83.597	92.987
	RF	0.203	1.335	2.419	86.661	110.591
	M5P	0.391	1.257	2.001	81.595	91.507
	SmoReg	0.351	0.972	2.378	63.118	108.742
	ANN	0.264	1.371	2.196	88.996	100.391
ANT 1.5	RL	0.468	2.494	3.496	81.300	88.229
	Gaussian	0.504	2.337	3.414	76.189	86.168
	RF	0.347	2.587	4.007	84.311	101.126
	M5P	0.431	2.461	3.594	80.201	90.703
	SmoReg	0.472	2.096	3.607	68.333	91.047
	ANN	0.482	2.644	3.571	86.194	90.104
ANT 1.6	RL	0.232	0.858	1.584	92.253	96.965
	Gaussian	0.243	0.845	1.580	90.890	96.733
	RF	0.070	0.878	1.930	94.385	118.11
	M5P	0.208	0.853	1.596	91.653	97.679
	SmoReg	0.087	0.533	1.710	57.277	104.679
	ANN	0.121	1.004	1.698	107.894	103.903
ANT 1.7	RL	0.452	0.678	1.384	85.464	89.128
	Gaussian	0.459	0.640	1.377	80.728	88.711
	RF	0.280	0.681	1.669	85.763	107.51
	M5P	0.452	0.678	1.384	85.464	89.128
	SmoReg	0.341	0.440	1.501	55.435	96.686
	ANN	0.343	0.870	1.499	109.589	96.517

Table 4.9: Résultats de RT (LOC, RFC) pour la Prédition de Risque

4.4.7 Conclusion

Les résultats montrent un grand coefficient de corrélation pour les versions 1.3 (0.659) et 1.5 (0.504). Également pour 1.4 et 1.7 même si ces deux versions contiennent moins de fautes. Par contre, le manque de performances persiste avec la version 1.6. Comme vu précédemment sur les premières phases de cette étude, probablement en raison de l'absence de bogues dans cette version, les algorithmes ne disposent pas de suffisamment de données d'entrée pour la phase d'entraînement.

Les modèles de risque multidimensionnels présentent de bons résultats comparés aux modèles de risque à deux dimensions, mis à part RT (LOC, RFC) qui les surpasse de très peu. La performance de ce modèle de risque pour la prédiction du risque théorique est en lien avec sa capacité à prédire le niveau de sévérité des fautes comme vu dans l’expérimentation précédente. En effet, grâce à la composante LOC, ce modèle est capable de prédire la probabilité de présence de faute. Sa deuxième composante RFC lui permet de prédire la gravité des fautes. Le modèle de risque théorique arrive à estimer le risque empirique avec un bon coefficient de corrélation lorsque les données sont plus ou moins balancées et reste ouvert à des améliorations.

Chapitre. 5

DISCUSSIONS ET CONCLUSION

5.1 Menaces de Validité

Les principales menaces à la validité qui pourraient biaiser les résultats de notre étude sont :

5.1.1 Risques de validité externes

Tout d'abord, nous nous sommes appuyés exclusivement sur différentes versions du système ANT pour mener nos expériences. Ce système étant écrit en JAVA, les métriques orientées objet extraites s'imprègnent des caractéristiques de ce langage de programmation, ce qui peut avoir un impact sur les possibilités de généralisation. Nos conclusions ne peuvent être généralisées pour d'autres langages utilisant le paradigme orienté objet. Le modèle de risque considéré, et les variantes étudiées, doivent être validés avec d'autres systèmes pour confirmer nos résultats, qui sont globalement bons pour la classification binaire et la prédiction du niveau de sévérité.

5.1.2 Risques de validité internes

Une autre menace réside dans notre formulation du risque empirique. En effet, le choix des poids pour les niveaux de gravité reste encore subjectif. Même si en donnant le poids le plus élevé de 6 pour le plus haut niveau de gravité, nous avons suivi une certaine logique. Le problème reste dans la catégorisation de la sévérité en elle-même qui n'est pas formelle. Le choix de la catégorie d'appartenance des bogues est basé uniquement sur l'expérience des testeurs (développeurs) qui les reportent. Cette assignation pourrait être changée vers une autre mieux élaborée et de manière objective qui reflète mieux l'impact des bogues sur le système étudié.

5.1.3 Risques de validité conceptuelle :

En plus de cela, notre le modèle de risque est basé sur la distance euclidienne. Il ne doit pas être écarté qu'il existe une autre formule de calcul de distance comme la distance de Manhattan ou une autre plus complexe qui décrit mieux la relation entre P (la probabilité de faute) et I (l'impact des fautes). Toutefois, ce choix n'est pas arbitraire, il est basé sur la théorie des risques de Bernoulli largement utilisée en gestion de risque.

Pour finir, la logique derrière le choix des modèles de risque à deux dimensions à imbriquer pour former les modèles de risque multidimensionnels est à relativiser. En effet, notre choix est basé uniquement sur les résultats obtenus, en prenant un modèle de risque à deux dimensions qui est bon prédicteur pour la probabilité d'occurrence de fautes et un second qui est bon prédicteur de l'impact des fautes. Sans réfléchir, on pourrait imbriquer des modèles de cette façon jusqu'à l'infini; il faudrait le faire de manière plus objective. Par exemple : Est-ce que c'est sensé d'imbriquer deux modèles de risque à deux dimensions qui ont une composante (métrique) commune ? Cela est une question qui reste ouverte.

5.2 Conclusion et Perspectives

Durant cette recherche, nous avons mené une étude empirique sur le potentiel d'un nouveau modèle de risque pour la prédiction de fautes logicielles. Dans une première phase, après une étape de prétraitement, nous avons d'abord investigué la classification binaire. Puis, nous nous sommes penchés sur la prédiction du niveau de sévérité sur les versions considérées du système ANT. Nous avons calculé le g-mean et l'AUC pour l'évaluation de nos modèles construits avec une validation croisée 10-fold.

En ce qui concerne la classification binaire, nous avons obtenu de très bons modèles sur toutes les versions du système ANT que nous avons étudiées. Les techniques de stratification que nous avons utilisées, à savoir: la duplication des erreurs; ont permis de surmonter le problème de non-balancement des données. Notamment sur les versions ANT 1.6 et 1.7 qui présentent un faible nombre d'erreurs.

Concernant la prédiction des défauts de sévérité normale, nous avons obtenu de très bons modèles pour ANT 1.3 et de bons modèles pour les autres versions. Cela est dû à la présence importante de bogues de gravité normale sur cette version. En ce qui concerne les bogues de sévérité élevée, nous avons pu construire de très bons modèles pour ANT 1.3, des modèles acceptables pour 1.4 et 1.5 mais des modèles faibles pour les autres versions. Une explication possible de ce déficit est le très faible pourcentage de ces fautes parmi ces versions (moins de 5,50% même après la duplication des fautes).

En ce qui concerne la prédiction du niveau de sévérité, les résultats montrent de bons modèles de classification avec la version ANT 1.5 et des modèles acceptables avec les autres versions, sauf la version 1.6. Comme nous l'avons vu au cours de cette étude, l'absence de défauts, due évidemment aux corrections, sur les versions récentes rend

plus difficile la construction de modèles de prédiction efficaces. Le modèle de risque présenté dans cette étude s'avère être doté d'un grand potentiel non seulement sur la prédiction de fautes logicielles, mais également sur le niveau de prédiction de la gravité et il est certain qu'il peut être amélioré.

Comme perspectives, entreprendre une étude de validation de notre travail actuel en y considérant un ensemble d'autres systèmes serait plus que bénéfique. Nous souhaitons également explorer la validation entre-versions de nos modèles. Il faudra pour cela, construire un modèle à partir d'une version antérieure d'un système et le tester sur ses versions nouvelles.

BIBLIOGRAPHIE

- [1] Tim Menzies, Zach Milton, Burak Turhan, Bojan Cukic, Yue Jiang, and Ayşe Bener. Defect prediction from static code features: current results, limitations, new approaches. *Automated Software Engineering*, 17(4):375–407, 2010.
- [2] Antonia Bertolino. Software testing research: Achievements, challenges, dreams. In *2007 Future of Software Engineering*, pages 85–103. IEEE Computer Society, 2007.
- [3] Julie Moeyersoms, Enric Junqué de Fortuny, Karel Dejaeger, Bart Baesens, and David Martens. Comprehensible software fault and effort prediction: A data mining approach. *Journal of Systems and Software*, 100:80–90, 2015.
- [4] Karel Dejaeger, Thomas Verbraken, and Bart Baesens. Toward comprehensible software fault prediction models using bayesian network classifiers. *IEEE Transactions on Software Engineering*, 39(2):237–257, 2013.
- [5] Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, and W Eric Wong. Code coverage of adaptive random testing. *IEEE Transactions on Reliability*, 62(1):226–237, 2013.
- [6] Ruchika Malhotra. A systematic review of machine learning techniques for software fault prediction. *Applied Soft Computing*, 27:504–518, 2015.
- [7] Santosh S Rathore and Sandeep Kumar. An empirical study of some software fault prediction techniques for the number of faults prediction. *Soft Computing*, 21(24):7417–7434, 2017.
- [8] Donald E Harter, Chris F Kemerer, and Sandra A Slaughter. Does software process improvement reduce the severity of defects? a longitudinal field study. *IEEE Transactions on Software Engineering*, 38(4):810–827, 2012.
- [9] Yuming Zhou and Hareton Leung. Empirical analysis of object-oriented design metrics for predicting high and low severity faults. *IEEE Transactions on software engineering*, 32(10):771–789, 2006.

- [10] Fadel Toure. *Orientation de l'effort des tests unitaires dans les systèmes orientés objet: une approche basée sur les métriques logicielles*. PhD thesis, Université Laval, 2016.
- [11] Jeffrey Voas and William W Agresti. Software quality from a behavioral perspective. *IT professional*, 6(4):46–50, 2004.
- [12] Shyam R Chidamber and Chris F Kemerer. A metrics suite for object oriented design. *IEEE Transactions on software engineering*, 20(6):476–493, 1994.
- [13] Raed Shatnawi, Wei Li, James Swain, and Tim Newman. Finding software metrics threshold values using roc curves. *Journal of software maintenance and evolution: Research and practice*, 22(1):1–16, 2010.
- [14] Raed Shatnawi. A quantitative investigation of the acceptable risk levels of object-oriented metrics in open-source systems. *IEEE Transactions on software engineering*, 36(2):216–225, 2010.
- [15] Tibor Gyimothy, Rudolf Ferenc, and Istvan Siket. Empirical validation of object-oriented metrics on open source software for fault prediction. *IEEE Transactions on Software engineering*, 31(10):897–910, 2005.
- [16] Bassey Isong and Ekabua Obeten. A systematic review of the empirical validation of object-oriented metrics towards fault-proneness prediction. *International Journal of Software Engineering and Knowledge Engineering*, 23(10):1513–1540, 2013.
- [17] Ruchika Malhotra and Ankita Jain Bansal. Fault prediction considering threshold effects of object-oriented metrics. *Expert Systems*, 32(2):203–219, 2015.
- [18] Marian Jureczko. Significance of different software metrics in defect prediction. *Software Engineering: An International Journal*, 1(1):86–95, 2011.
- [19] Liguo Yu. Using negative binomial regression analysis to predict software faults: A study of apache ant. 2012.
- [20] Thomas J McCabe. A complexity measure. *IEEE Transactions on software Engineering*, (4):308–320, 1976.
- [21] Victor R Basili, Lionel C. Briand, and Walcélion L Melo. A validation of object-oriented design metrics as quality indicators. *IEEE Transactions on software engineering*, 22(10):751–761, 1996.
- [22] Marian Jureczko and Diomidis Spinellis. Using object-oriented design metrics to predict software defects. *Models and Methods of System Dependability. Oficyna Wydawnicza Politechniki Wrocławskiej*, pages 69–81, 2010.

- [23] Edward Yourdon and Larry L. Constantine. *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1st edition, 1979. ISBN 0138544719.
- [24] Lionel C Briand, Jurgen Wust, and Hakim Lounis. Using coupling measurement for impact analysis in object-oriented systems. In *Software Maintenance, 1999.(ICSM'99) Proceedings. IEEE International Conference on*, pages 475–482. IEEE, 1999.
- [25] Robert Martin. Oo design quality metrics. *An analysis of dependencies*, 12:151–170, 1994.
- [26] Sallie Henry and Dennis Kafura. Software structure metrics based on information flow. *IEEE transactions on Software Engineering*, (5):510–518, 1981.
- [27] LH Rosenberg, R Stanko, and A Gallo. Risk-based object oriented testing. 24th swe. *NASA, Greenbelt, MD, USA*, 1999.
- [28] Grady Booch. Object-oriented design with applications,(1991). *Redwood City, CA, enj armn/Cummings*, 1991.
- [29] Shivkumar Shivaji, E James Whitehead, Ram Akella, and Sunghun Kim. Reducing features to improve code change-based bug prediction. *IEEE Transactions on Software Engineering*, 39(4):552–569, 2013.
- [30] Iker Gondra. Applying machine learning to software fault-proneness prediction. *Journal of Systems and Software*, 81(2):186–195, 2008.
- [31] Karim O Elish and Mahmoud O Elish. Predicting defect-prone software modules using support vector machines. *Journal of Systems and Software*, 81(5):649–660, 2008.
- [32] Erik Arisholm, Lionel C Briand, and Eivind B Johannessen. A systematic and comprehensive investigation of methods to build and evaluate fault prediction models. *Journal of Systems and Software*, 83(1):2–17, 2010.
- [33] Sunghun Kim, E James Whitehead Jr, and Yi Zhang. Classifying software changes: Clean or buggy? *IEEE Transactions on Software Engineering*, 34(2):181–196, 2008.
- [34] S Kanmani, V Rhymend Uthariaraj, V Sankaranarayanan, and P Thambidurai. Object-oriented software fault prediction using neural networks. *Information and software technology*, 49(5):483–492, 2007.
- [35] Huanjing Wang, Taghi M Khoshgoftaar, and Amri Napolitano. A comparative study of ensemble feature selection techniques for software defect prediction. In *Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on*, pages 135–140. IEEE, 2010.

- [36] Tim Menzies, Jeremy Greenwald, and Art Frank. Data mining static code attributes to learn defect predictors. *IEEE transactions on software engineering*, (1):2–13, 2007.
- [37] Issam H Laradji, Mohammad Alshayeb, and Lahouari Ghouti. Software defect prediction using ensemble learning on selected features. *Information and Software Technology*, 58:388–402, 2015.
- [38] Weiwei Li, Zhiqiu Huang, and Qing Li. Three-way decisions based software defect prediction. *Knowledge-Based Systems*, 91:263–274, 2016.
- [39] Ralf Bender. Quantitative risk assessment in epidemiological studies investigating threshold effects. *Biometrical Journal*, 41(3):305–319, 1999.
- [40] Huihua Lu, Bojan Cukic, and Mark Culp. Software defect prediction using semi-supervised learning with dimension reduction. In *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering*, pages 314–317. ACM, 2012.
- [41] Huihua Lu, Bojan Cukic, and Mark Culp. A semi-supervised approach to software defect prediction. In *Computer Software and Applications Conference (COMPSAC), 2014 IEEE 38th Annual*, pages 416–425. IEEE, 2014.
- [42] Cagatay Catal. A comparison of semi-supervised classification approaches for software defect prediction. *Journal of Intelligent Systems*, 23(1):75–82, 2014.
- [43] Partha S Bishnu and Vandana Bhattacherjee. Software fault prediction using quad tree-based k-means clustering algorithm. *IEEE Transactions on knowledge and data engineering*, 24(6):1146–1150, 2012.
- [44] Cagatay Catal, Ugur Sevim, and Banu Diri. Metrics-driven software quality prediction without prior fault data. In *Electronic Engineering and Computing Technology*, pages 189–199. Springer, 2010.
- [45] Cagatay Catal, Ugur Sevim, and Banu Diri. Clustering and metrics thresholds based software fault prediction of unlabeled program modules. In *Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on*, pages 199–204. IEEE, 2009.
- [46] Alexandre Boucher and Mourad Badri. Software metrics thresholds calculation techniques to predict fault-proneness: An empirical comparison. *Information and Software Technology*, 96:38–67, 2018.
- [47] Alexandre Boucher and Mourad Badri. Predicting fault-prone classes in object-oriented software: an adaptation of an unsupervised hybrid som algorithm. In *Software Quality, Reliability and Security (QRS), 2017 IEEE International Conference on*, pages 306–317. IEEE, 2017.

- [48] Euyseok Hong. Software fault-proneness prediction using module severity metrics. *International Journal of Applied Engineering Research*, 12(9):2038–2043, 2017.
- [49] Yogesh Singh, Arvinder Kaur, and Ruchika Malhotra. Empirical validation of object-oriented metrics for predicting fault proneness models. *Software quality journal*, 18(1):3, 2010.
- [50] Raed Shatnawi and Wei Li. The effectiveness of software metrics in identifying error-prone classes in post-release software evolution process. *Journal of systems and software*, 81(11):1868–1882, 2008.
- [51] OF RISK and DANIEL BERNOULLI. Exposition of a new theory on the measurement. *Econometrica*, 22(1):23–36, 1954.
- [52] Danijel Radjenović, Marjan Heričko, Richard Torkar, and Aleš Živković. Software fault prediction metrics: A systematic literature review. *Information and Software Technology*, 55(8):1397–1418, 2013.
- [53] Akhilendra Singh Chauhan and Sanjay Kumar Dubey. Analytical review of fault-proneness for object oriented systems.
- [54] Ruchika Malhotra. *Empirical research in software engineering: concepts, analysis, and applications*. Chapman and Hall/CRC, 2016.
- [55] Mahmoud O Elish, Ali H Al-Yafei, and Muhammed Al-Mulhem. Empirical comparison of three metrics suites for fault prediction in packages of object-oriented systems: A case study of eclipse. *Advances in Engineering Software*, 42(10):852–859, 2011.
- [56] Segla Kpodjedo, Filippo Ricca, Philippe Galinier, Yann-Gaël Guéhéneuc, and Giuliano Antoniol. Design evolution metrics for defect prediction in object oriented systems. *Empirical Software Engineering*, 16(1):141–175, 2011.
- [57] Ping Yu, Tarja Systa, and Hausi Muller. Predicting fault-proneness using oo metrics. an industrial case study. In *Software Maintenance and Reengineering, 2002. Proceedings. Sixth European Conference on*, pages 99–107. IEEE, 2002.
- [58] Lionel C Briand, Jürgen Wüst, John W Daly, and D Victor Porter. Exploring the relationships between design measures and software quality in object-oriented systems. *Journal of systems and software*, 51(3):245–273, 2000.
- [59] Martin Iliev, Bilal Karasneh, Michel RV Chaudron, and Edwin Essenijs. Automated prediction of defect severity based on codifying design knowledge using ontologies. In *Realizing Artificial Intelligence Synergies in Software Engineering (RAISE), 2012 First International Workshop on*, pages 7–11. IEEE, 2012.

- [60] William O Bearden and Richard G Netemeyer. *Handbook of marketing scales: Multi-item measures for marketing and consumer behavior research*. Sage, 1999.
- [61] Cagatay Catal and Banu Diri. Software fault prediction with object-oriented metrics based artificial immune recognition system. In *International Conference on Product Focused Software Process Improvement*, pages 300–314. Springer, 2007.
- [62] Mamdouh Refaat. *Data preparation for data mining using SAS*. Elsevier, 2010.
- [63] Taghi M Khoshgoftaar, Chris Seiffert, Jason Van Hulse, Amri Napolitano, and Andres Folleco. Learning with limited minority class data. In *Machine Learning and Applications, 2007. ICMLA 2007. Sixth International Conference on*, pages 348–353. IEEE, 2007.
- [64] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.
- [65] Raed Shatnawi. Improving software fault-prediction for imbalanced data. In *Innovations in Information Technology (IIT), 2012 International Conference on*, pages 54–59. IEEE, 2012.
- [66] Daniel A Keim. Information visualization and visual data mining. *IEEE transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.
- [67] Arvinder Kaur and Kamaldeep Kaur. Performance analysis of ensemble learning for predicting defects in open source software. In *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on*, pages 219–225. IEEE, 2014.
- [68] Ruchika Malhotra and Ankita Jain. Fault prediction using statistical and machine learning methods for improving software quality. *Journal of Information Processing Systems*, 8(2):241–262, 2012.
- [69] Yanmin Sun, Mohamed S Kamel, and Yang Wang. Boosting for learning multiple classes with imbalanced class distribution. In *Data Mining, 2006. ICDM’06. Sixth International Conference on*, pages 592–602. IEEE, 2006.
- [70] Raed Shatnawi. The application of roc analysis in threshold identification, data imbalance and metrics selection for software fault prediction. *Innovations in Systems and Software Engineering*, 13(2-3):201–217, 2017.
- [71] David W Hosmer and Stanley Lemeshow. Special topics. *Applied Logistic Regression, Second Edition*, pages 260–351, 2000.
- [72] Lionel Marks, Ying Zou, and Ahmed E Hassan. Studying the fix-time for bugs in large open source projects. In *Proceedings of the 7th International Conference on Predictive Models in Software Engineering*, page 11. ACM, 2011.

- [73] Franz Faul, Edgar Erdfelder, Axel Buchner, and Albert-Georg Lang. Statistical power analyses using g* power 3.1: Tests for correlation and regression analyses. *Behavior research methods*, 41(4):1149–1160, 2009.

ANNEXE. A

Classification Binaire

		TPR	TNR	g-m	ACC	TPR-D	TNR-D	g-m-D	ACC-D	TPR-B	TNR-B	g-m-B	ACC-B
ANT 1.3	NB	0.567	0.924	0.754	0.731	0.741	0.909	0.796	0.821	-	-	-	-
	J48	0.733	0.773	0.754	0.753	0.852	0.727	0.811	0.787	-	-	-	-
	RF	0.633	0.667	0.651	0.650	0.919	0.667	0.836	0.783	-	-	-	-
	RL	0.667	0.818	0.746	0.739	0.837	0.788	0.821	0.812	-	-	-	-
	SVM	0.400	0.955	0.690	0.618	1.000	0.045	0.687	0.212	-	-	-	-
	ANN	0.684	0.827	0.772	0.730	0.873	0.752	0.818	0.876	-	-	-	-
ANT 1.4	NB	0.211	0.971	0.809	0.453	0.340	0.943	0.784	0.566	0.342	0.929	0.722	0.564
	J48	0.079	0.921	0.742	0.27	0.400	0.857	0.737	0.585	0.566	0.814	0.593	0.679
	RF	0.316	0.814	0.708	0.507	0.560	0.836	0.763	0.684	0.526	0.743	0.667	0.625
	RL	0.184	0.971	0.803	0.423	0.260	0.957	0.774	0.499	0.368	0.9	0.713	0.576
	SVM	0.000	1.000	0.787	0.000	0.200	0.971	0.768	0.441	0.184	0.971	0.694	0.423
	ANN	0.159	0.962	0.476	0.245	0.408	0.904	0.620	0.487	0.491	0.846	0.721	0.65
ANT 1.5	NB	0.226	0.947	0.686	0.463	0.464	0.920	0.697	0.653	-	-	-	-
	J48	0.708	0.695	0.700	0.701	0.765	0.770	0.768	0.767	-	-	-	-
	RF	0.519	0.754	0.669	0.626	0.799	0.738	0.768	0.768	-	-	-	-
	RL	0.292	0.925	0.696	0.52	0.615	0.861	0.740	0.728	-	-	-	-
	SVM	0.075	0.979	0.652	0.271	0.436	0.920	0.683	0.633	-	-	-	-
	ANN	0.532	0.844	0.683	0.589	0.740	0.797	0.770	0.760	-	-	-	-
ANT 1.6	NB	0.111	0.967	0.858	0.328	0.131	0.961	0.823	0.355	0.289	0.928	0.704	0.518
	J48	0.000	1.000	0.872	0.000	0.000	1.000	0.834	0.000	0.608	0.792	0.61	0.694
	RF	0.267	0.876	0.798	0.484	0.557	0.873	0.821	0.697	0.554	0.726	0.666	0.634
	RL	0.044	0.984	0.864	0.208	0.115	0.977	0.834	0.335	0.307	0.919	0.704	0.531
	SVM	0.000	1.000	0.872	0.000	0.000	1.000	0.834	0.000	0.102	0.964	0.662	0.314
	ANN	0.000	0.996	0.000	0.000	0.061	0.982	0.258	0.106	0.455	0.869	0.724	0.63
ANT 1.7	NB	0.229	0.969	0.899	0.471	0.366	0.964	0.887	0.594	0.36	0.93	0.731	0.579
	J48	0.000	0.999	0.905	0.000	0.248	0.993	0.896	0.496	0.599	0.836	0.629	0.707
	RF	0.200	0.921	0.854	0.429	0.465	0.910	0.852	0.650	0.563	0.776	0.702	0.661
	RL	0.143	0.991	0.911	0.376	0.248	0.985	0.889	0.494	0.42	0.919	0.744	0.621
	SVM	0.000	1.000	0.906	0.000	0.089	0.994	0.876	0.297	0.343	0.936	0.729	0.567
	ANN	0.109	0.994	0.346	0.185	0.310	0.976	0.555	0.422	0.592	0.847	0.757	0.714

Table A.1: Résultats RT (LOC, Ca) Classification Binaire

		TPR	TNR	g-m	ACC	TPR-D	TNR-D	g-m-D	ACC-D	TPR-B	TNR-B	g-m-B	ACC-B
ANT 1.3	NB	0.567	0.924	0.754	0.724	0.807	0.758	0.791	0.782	-	-	-	-
	J48	0.733	0.773	0.754	0.753	0.852	0.727	0.811	0.787	-	-	-	-
	RF	0.633	0.682	0.659	0.657	0.919	0.682	0.841	0.792	-	-	-	-
	RL	0.667	0.818	0.746	0.739	0.837	0.788	0.821	0.812	-	-	-	-
	SVM	0.400	0.955	0.690	0.618	1.000	0.045	0.687	0.212	-	-	-	-
	ANN	0.684	0.827	0.772	0.730	0.873	0.752	0.818	0.876	-	-	-	-
ANT 1.4	NB	0.237	0.971	0.815	0.480	0.340	0.943	0.784	0.566	0.316	0.929	0.713	0.542
	J48	0.053	0.936	0.747	0.223	0.480	0.850	0.753	0.639	0.553	0.814	0.583	0.671
	RF	0.316	0.800	0.697	0.503	0.560	0.807	0.742	0.672	0.526	0.693	0.634	0.604
	RL	0.184	0.971	0.803	0.423	0.260	0.957	0.774	0.499	0.355	0.9	0.708	0.565
	SVM	0.000	1.000	0.787	0.000	0.200	0.971	0.768	0.441	0.197	0.971	0.699	0.438
	ANN	0.532	0.844	0.683	0.589	0.740	0.797	0.770	0.760	0.481	0.849	0.719	0.643
ANT 1.5	NB	0.226	0.947	0.686	0.463	0.464	0.920	0.697	0.653	-	-	-	-
	J48	0.708	0.695	0.700	0.701	0.765	0.770	0.768	0.767	-	-	-	-
	RF	0.509	0.733	0.652	0.611	0.804	0.727	0.765	0.765	-	-	-	-
	RL	0.292	0.925	0.696	0.520	0.615	0.861	0.740	0.728	-	-	-	-
	SVM	0.075	0.979	0.652	0.271	0.441	0.920	0.686	0.637	-	-	-	-
	ANN	0.585	0.807	0.727	0.687	0.754	0.775	0.765	0.764	-	-	-	-
ANT 1.6	NB	0.111	0.967	0.858	0.328	0.131	0.961	0.823	0.355	0.271	0.928	0.698	0.502
	J48	0.000	1.000	0.872	0.000	0.000	1.000	0.834	0.000	0.56	0.798	0.579	0.669
	RF	0.267	0.876	0.798	0.484	0.557	0.876	0.823	0.699	0.524	0.723	0.653	0.616
	RL	0.044	0.984	0.864	0.208	0.115	0.977	0.834	0.335	0.307	0.919	0.704	0.531
	SVM	0.000	1.000	0.872	0.000	0.000	1.000	0.834	0.000	0.078	0.967	0.655	0.275
	ANN	0.000	0.996	0.000	0.000	0.061	0.982	0.258	0.106	0.45	0.867	0.721	0.626
ANT 1.7	NB	0.229	0.969	0.899	0.471	0.366	0.964	0.887	0.594	0.36	0.93	0.731	0.579
	J48	0.000	0.999	0.905	0.000	0.248	0.993	0.896	0.496	0.569	0.837	0.608	0.69
	RF	0.214	0.911	0.846	0.442	0.535	0.904	0.856	0.695	0.536	0.748	0.674	0.633
	RL	0.143	0.991	0.911	0.376	0.248	0.985	0.889	0.494	0.42	0.919	0.744	0.621
	SVM	0.000	1.000	0.906	0.000	0.089	0.993	0.875	0.297	0.332	0.936	0.725	0.558
	ANN	0.110	0.994	0.347	0.187	0.311	0.975	0.555	0.422	0.604	0.845	0.76	0.721

Table A.2: Résultats RT (LOC, CBO) Classification Binaire

ANNEXE A. CLASSIFICATION BINAIRE

		TPR	TNR	g-m	ACC	TPR-D	TNR-D	g-m-D	ACC-D	TPR-B	TNR-B	g-m-B	ACC-B
ANT 1.3	NB	0.567	0.924	0.754	0.724	0.741	0.909	0.796	0.821	-	-	-	-
	J48	0.733	0.758	0.746	0.745	0.852	0.727	0.811	0.787	-	-	-	-
	RF	0.617	0.606	0.611	0.611	0.919	0.636	0.826	0.765	-	-	-	-
	RL	0.667	0.818	0.746	0.739	0.837	0.788	0.821	0.812	-	-	-	-
	SVM	0.400	0.955	0.690	0.618	1.000	0.061	0.692	0.247	-	-	-	-
ANT 1.4	ANN	0.686	0.826	0.772	0.731	0.873	0.752	0.818	0.876	-	-	-	-
	NB	0.237	0.971	0.815	0.480	0.340	0.943	0.784	0.566	0.329	0.921	0.713	0.551
	J48	0.053	0.936	0.747	0.223	0.480	0.850	0.753	0.639	0.592	0.814	0.612	0.694
	RF	0.342	0.807	0.708	0.525	0.580	0.814	0.753	0.687	0.605	0.779	0.718	0.686
	RL	0.184	0.971	0.803	0.423	0.260	0.964	0.779	0.501	0.368	0.9	0.713	0.576
ANT 1.5	SVM	0.000	1.000	0.787	0.000	0.180	0.971	0.763	0.418	0.197	0.971	0.699	0.438
	ANN	0.159	0.962	0.476	0.245	0.408	0.904	0.620	0.487	0.487	0.846	0.719	0.643
	NB	0.226	0.947	0.686	0.463	0.469	0.914	0.697	0.655	-	-	-	-
	J48	0.708	0.674	0.686	0.701	0.771	0.770	0.770	0.770	-	-	-	-
	RF	0.500	0.717	0.638	0.611	0.799	0.717	0.757	0.757	-	-	-	-
ANT 1.6	RL	0.283	0.925	0.693	0.520	0.615	0.861	0.740	0.728	-	-	-	-
	SVM	0.066	0.979	0.648	0.271	0.419	0.925	0.678	0.623	-	-	-	-
	ANN	0.522	0.850	0.685	0.584	0.740	0.797	0.770	0.760	-	-	-	-
	NB	0.111	0.971	0.861	0.328	0.131	0.964	0.826	0.355	0.295	0.928	0.706	0.523
	J48	0.000	1.000	0.872	0.000	0.000	1.000	0.834	0.000	0.639	0.756	0.611	0.695
ANT 1.7	RF	0.244	0.886	0.804	0.465	0.541	0.886	0.829	0.692	0.524	0.739	0.664	0.623
	RL	0.022	0.984	0.861	0.147	0.098	0.977	0.832	0.309	0.307	0.919	0.704	0.531
	SVM	0.000	1.000	0.872	0.000	0.000	1.000	0.834	0.000	0.078	0.977	0.662	0.277
	ANN	0.000	0.996	0.000	0.000	0.067	0.982	0.266	0.116	0.459	0.869	0.726	0.635
	NB	0.214	0.969	0.898	0.455	0.356	0.967	0.888	0.587	0.36	0.93	0.731	0.579
ANT 1.8	J48	0.000	0.995	0.905	0.000	0.248	0.993	0.896	0.496	0.569	0.837	0.608	0.69
	RF	0.200	0.919	0.851	0.429	0.505	0.908	0.856	0.677	0.536	0.748	0.674	0.633
	RL	0.143	0.991	0.911	0.376	0.248	0.987	0.890	0.495	0.42	0.919	0.744	0.621
	SVM	0.000	1.000	0.906	0.000	0.069	0.996	0.875	0.262	0.332	0.936	0.725	0.558
	ANN	0.112	0.993	0.350	0.190	0.311	0.975	0.555	0.422	0.604	0.845	0.76	0.721

Table A.3: Résultats RT (LOC, Ce) Classification Binaire

		TPR	TNR	g-m	ACC	TPR-D	TNR-D	g-m-D	ACC-D	TPR-B	TNR-B	g-m-B	ACC-B
ANT 1.3	NB	0.150	0.848	0.516	0.357	0.941	0.076	0.657	0.267	-	-	-	-
	J48	0.617	0.530	0.571	0.572	0.807	0.409	0.677	0.575	-	-	-	-
	RF	0.500	0.652	0.579	0.571	0.822	0.485	0.711	0.631	-	-	-	-
	RL	0.217	0.909	0.579	0.444	0.985	0.015	0.667	0.122	-	-	-	-
	SVM	0.017	0.985	0.524	0.129	1.000	0.000	0.672	0.000	-	-	-	-
	ANN	0.210	0.867	0.532	0.309	0.947	0.100	0.436	0.794	-	-	-	-
ANT 1.4	NB	0.079	0.964	0.775	0.276	0.120	0.964	0.742	0.340	0.105	0.957	0.657	0.317
	J48	0.000	1.000	0.787	0.000	0.140	0.921	0.716	0.359	0.605	0.729	0.575	0.664
	RF	0.263	0.886	0.753	0.483	0.520	0.879	0.784	0.676	0.513	0.793	0.694	0.638
	RL	0.053	0.979	0.781	0.228	0.080	0.971	0.737	0.279	0.132	0.957	0.667	0.355
	SVM	0.000	1.000	0.787	0.000	0.000	0.993	0.732	0.000	0.013	0.986	0.644	0.114
	ANN	0.048	0.971	0.315	0.083	0.287	0.931	0.545	0.387	0.37	0.87	0.694	0.631
ANT 1.5	NB	0.057	0.968	0.638	0.235	0.123	0.963	0.552	0.344	-	-	-	-
	J48	0.406	0.765	0.635	0.557	0.642	0.759	0.702	0.698	-	-	-	-
	RF	0.311	0.727	0.577	0.475	0.687	0.663	0.675	0.675	-	-	-	-
	RL	0.038	0.973	0.635	0.192	0.385	0.840	0.617	0.569	-	-	-	-
	SVM	0.000	0.995	0.635	0.000	0.039	0.989	0.525	0.196	-	-	-	-
	ANN	0.048	0.961	0.347	0.086	0.493	0.811	0.665	0.583	-	-	-	-
ANT 1.6	NB	0.044	0.984	0.864	0.208	0.000	0.977	0.815	0.000	0.054	0.971	0.649	0.229
	J48	0.000	1.000	0.872	0.000	0.033	0.980	0.823	0.180	0.187	0.906	0.274	0.411
	RF	0.067	0.909	0.801	0.247	0.344	0.909	0.815	0.559	0.44	0.818	0.685	0.6
	RL	0.022	0.993	0.869	0.148	0.016	0.990	0.829	0.126	0.024	0.987	0.649	0.154
	SVM	0.000	1.000	0.872	0.000	0.000	1.000	0.834	0.000	0	1	0.649	0
	ANN	0.000	1.000	0.000	0.000	0.000	0.995	0.000	0.000	0.025	0.98	0.645	0.252
ANT 1.7	NB	0.014	0.987	0.895	0.118	0.079	0.984	0.866	0.279	0.049	0.981	0.654	0.22
	J48	0.000	1.000	0.906	0.000	0.000	1.000	0.870	0.000	0.448	0.738	0.463	0.575
	RF	0.071	0.948	0.866	0.259	0.426	0.948	0.880	0.635	0.505	0.861	0.736	0.66
	RL	0.000	0.999	0.905	0.000	0.040	0.997	0.872	0.200	0.025	0.985	0.649	0.156
	SVM	0.000	1.000	0.906	0.000	0.000	0.999	0.869	0.000	0	0.999	0.649	0
	ANN	0.000	0.999	0.000	0.000	0.048	0.995	0.222	0.089	0.054	0.977	0.653	0.372

Table A.4: Résultats RT (WMC, Ca) Classification Binaire

ANNEXE A. CLASSIFICATION BINAIRE

		TPR	TNR	g-m	ACC	TPR-D	TNR-D	g-m-D	ACC-D	TPR-B	TNR-B	g-m-B	ACC-B
ANT 1.3	NB	0.250	0.879	0.579	0.469	0.933	0.076	0.652	0.266	-	-	-	-
	J48	0.633	0.697	0.667	0.664	0.889	0.485	0.756	0.657	-	-	-	-
	RF	0.667	0.576	0.619	0.620	0.896	0.561	0.786	0.709	-	-	-	-
	RL	0.433	0.833	0.643	0.601	0.948	0.318	0.741	0.549	-	-	-	-
	SVM	0.033	0.970	0.524	0.179	1.000	0.000	0.672	0.000	-	-	-	-
	ANN	0.393	0.798	0.698	0.486	0.920	0.429	0.675	0.837	-	-	-	-
ANT 1.4	NB	0.105	0.957	0.775	0.317	0.200	0.957	0.758	0.437	0.224	0.95	0.694	0.461
	J48	0.026	0.971	0.770	0.159	0.400	0.900	0.768	0.600	0.671	0.721	0.614	0.696
	RF	0.263	0.871	0.742	0.479	0.600	0.850	0.784	0.714	0.487	0.793	0.685	0.621
	RL	0.132	0.957	0.781	0.355	0.200	0.950	0.753	0.436	0.342	0.943	0.731	0.568
	SVM	0.000	1.000	0.787	0.000	0.000	0.986	0.726	0.000	0.013	0.986	0.644	0.114
	ANN	0.142	0.956	0.564	0.218	0.466	0.915	0.662	0.547	0.516	0.856	0.737	0.669
ANT 1.5	NB	0.132	0.963	0.662	0.357	0.223	0.968	0.604	0.465	-	-	-	-
	J48	0.575	0.754	0.689	0.658	0.715	0.733	0.724	0.724	-	-	-	-
	RF	0.443	0.749	0.638	0.576	0.726	0.722	0.724	0.724	-	-	-	-
	RL	0.160	0.947	0.662	0.389	0.615	0.840	0.730	0.719	-	-	-	-
	SVM	0.009	0.995	0.638	0.095	0.190	0.968	0.587	0.429	-	-	-	-
	ANN	0.423	0.880	0.644	0.518	0.639	0.815	0.733	0.697	-	-	-	-
ANT 1.6	NB	0.044	0.977	0.858	0.207	0.016	0.974	0.815	0.125	0.084	0.958	0.651	0.284
	J48	0.000	1.000	0.872	0.000	0.997	0.016	0.834	0.126	0.452	0.847	0.521	0.619
	RF	0.178	0.906	0.813	0.402	0.475	0.909	0.837	0.657	0.56	0.811	0.723	0.674
	RL	0.022	0.990	0.866	0.148	0.016	0.990	0.829	0.126	0.084	0.961	0.653	0.285
	SVM	0.000	1.000	0.872	0.000	0.000	1.000	0.834	0.000	0	0.993	0.645	0
	ANN	0.008	0.998	0.125	0.015	0.011	0.980	0.142	0.020	0.24	0.916	0.679	0.492
ANT 1.7	NB	0.043	0.987	0.898	0.206	0.079	0.987	0.869	0.279	0.124	0.978	0.679	0.348
	J48	0.000	1.000	0.906	0.000	0.168	0.969	0.865	0.403	0.522	0.822	0.564	0.655
	RF	0.229	0.941	0.874	0.464	0.535	0.935	0.883	0.707	0.519	0.804	0.705	0.646
	RL	0.014	0.997	0.905	0.118	0.079	0.994	0.875	0.280	0.209	0.953	0.692	0.446
	SVM	0.000	1.000	0.906	0.000	0.000	0.999	0.869	0.000	0.016	0.996	0.653	0.128
	ANN	0.000	0.998	0.000	0.000	0.102	0.979	0.341	0.164	0.468	0.849	0.715	0.637

Table A.5: Résultats RT (WMC, CBO) Classification Binaire

		TPR	TNR	g-m	ACC	TPR-D	TNR-D	g-m-D	ACC-D	TPR-B	TNR-B	g-m-B	ACC-B
ANT 1.3	NB	0.450	0.894	0.683	0.634	0.630	0.833	0.697	0.724	-	-	-	-
	J48	0.667	0.682	0.675	0.674	0.830	0.636	0.766	0.727	-	-	-	-
	RF	0.600	0.682	0.643	0.64	0.919	0.636	0.826	0.765	-	-	-	-
	RL	0.583	0.803	0.698	0.684	0.837	0.667	0.781	0.747	-	-	-	-
	SVM	0.267	0.970	0.635	0.509	1.000	0.000	0.672	0.000	-	-	-	-
	ANN	0.532	0.797	0.674	0.606	0.870	0.540	0.691	0.831	-	-	-	-
ANT 1.4	NB	0.263	0.957	0.809	0.502	0.360	0.957	0.800	0.587	0.342	0.929	0.722	0.564
	J48	0.132	0.929	0.758	0.35	0.400	0.929	0.789	0.610	0.395	0.843	0.469	0.577
	RF	0.263	0.829	0.708	0.467	0.560	0.821	0.753	0.678	0.408	0.786	0.653	0.566
	RL	0.211	0.971	0.809	0.453	0.360	0.957	0.800	0.587	0.421	0.879	0.718	0.608
	SVM	0.000	1.000	0.787	0.000	0.060	1.000	0.753	0.245	0.092	0.993	0.676	0.302
	ANN	0.210	0.959	0.450	0.308	0.366	0.938	0.600	0.475	0.499	0.854	0.729	0.655
ANT 1.5	NB	0.321	0.914	0.700	0.542	0.497	0.904	0.705	0.670	-	-	-	-
	J48	0.604	0.818	0.741	0.703	0.726	0.818	0.773	0.771	-	-	-	-
	RF	0.434	0.775	0.652	0.580	0.737	0.802	0.770	0.769	-	-	-	-
	RL	0.396	0.904	0.720	0.598	0.642	0.872	0.760	0.748	-	-	-	-
	SVM	0.179	0.963	0.679	0.415	0.547	0.904	0.730	0.703	-	-	-	-
	ANN	0.520	0.860	0.676	0.588	0.724	0.817	0.775	0.756	-	-	-	-
ANT 1.6	NB	0.133	0.967	0.861	0.359	0.197	0.961	0.834	0.435	0.259	0.909	0.681	0.485
	J48	0.000	1.000	0.872	0.000	0.000	1.000	0.834	0.000	0.512	0.831	0.561	0.652
	RF	0.089	0.919	0.813	0.286	0.393	0.909	0.823	0.598	0.536	0.834	0.729	0.669
	RL	0.044	0.997	0.875	0.209	0.082	0.977	0.829	0.283	0.259	0.906	0.679	0.484
	SVM	0.000	1.000	0.872	0.000	0.000	1.000	0.834	0.000	0.042	0.984	0.653	0.204
	ANN	0.019	0.997	0.188	0.037	0.031	0.981	0.213	0.055	0.364	0.87	0.692	0.582
ANT 1.7	NB	0.229	0.966	0.897	0.470	0.327	0.959	0.876	0.560	0.335	0.919	0.714	0.555
	J48	0.129	0.990	0.909	0.357	0.317	0.988	0.901	0.560	0.558	0.837	0.6	0.683
	RF	0.200	0.956	0.885	0.437	0.455	0.939	0.876	0.654	0.602	0.864	0.772	0.721
	RL	0.143	0.994	0.914	0.377	0.317	0.988	0.901	0.560	0.39	0.896	0.719	0.591
	SVM	0.000	1.000	0.906	0.000	0.010	1.000	0.871	0.100	0.261	0.95	0.708	0.498
	ANN	0.086	0.996	0.360	0.153	0.282	0.990	0.531	0.417	0.471	0.869	0.73	0.671

Table A.6: Résultats RT (WMC, Ce) Classification Binaire

		TPR	TNR	g-m	ACC	TPR-D	TNR-D	g-m-D	ACC-D	TPR-B	TNR-B	g-m-B	ACC-B
ANT 1.3	NB	0.283	0.848	0.579	0.490	0.911	0.076	0.637	0.263	-	-	-	-
	J48	0.817	0.485	0.643	0.629	0.881	0.439	0.736	0.622	-	-	-	-
	RF	0.533	0.606	0.571	0.568	0.852	0.515	0.741	0.662	-	-	-	-
	RL	0.133	0.894	0.532	0.345	0.985	0.030	0.672	0.172	-	-	-	-
	SVM	0.000	0.970	0.508	0.000	1.000	0.000	0.672	0.000	-	-	-	-
	ANN	0.193	0.829	0.510	0.279	0.965	0.179	0.560	0.816	-	-	-	-
ANT 1.4	NB	0.053	0.979	0.781	0.228	0.080	0.979	0.742	0.280	0.079	0.979	0.662	0.278
	J48	0.184	0.936	0.775	0.415	0.660	0.821	0.779	0.736	0.684	0.779	0.654	0.73
	RF	0.421	0.893	0.792	0.613	0.600	0.879	0.805	0.726	0.553	0.764	0.69	0.65
	RL	0.079	0.986	0.792	0.279	0.080	0.979	0.742	0.280	0.092	0.971	0.662	0.299
	SVM	0.000	1.000	0.787	0.000	0.000	0.986	0.726	0.000	0.066	0.986	0.662	0.255
	ANN	0.092	0.966	0.440	0.151	0.358	0.917	0.662	0.450	0.596	0.854	0.763	0.722
ANT 1.5	NB	0.000	0.979	0.625	0.000	0.140	0.973	0.566	0.369	-	-	-	-
	J48	0.679	0.679	0.679	0.679	0.765	0.690	0.727	0.727	-	-	-	-
	RF	0.434	0.791	0.662	0.586	0.737	0.706	0.721	0.721	-	-	-	-
	RL	0.019	0.968	0.625	0.136	0.313	0.904	0.615	0.532	-	-	-	-
	SVM	0.000	0.995	0.635	0.000	0.034	0.984	0.519	0.183	-	-	-	-
	ANN	0.055	0.970	0.400	0.099	0.457	0.830	0.697	0.559	-	-	-	-
ANT 1.6	NB	0.000	0.997	0.869	0.000	0.000	0.990	0.826	0.000	0.012	0.987	0.645	0.109
	J48	0.000	1.000	0.872	0.000	0.016	0.997	0.834	0.126	0.313	0.866	0.402	0.521
	RF	0.111	0.909	0.807	0.318	0.426	0.893	0.815	0.617	0.452	0.801	0.679	0.602
	RL	0.000	0.993	0.866	0.000	0.000	0.990	0.826	0.000	0.018	0.987	0.647	0.134
	SVM	0.000	1.000	0.872	0.000	0.000	1.000	0.834	0.000	0	0.997	0.647	0
	ANN	0.000	1.000	0.000	0.000	0.000	0.995	0.000	0.000	0.069	0.98	0.66	0.401
ANT 1.7	NB	0.014	0.990	0.898	0.118	0.069	0.982	0.863	0.260	0	0.994	0.646	0
	J48	0.000	1.000	0.906	0.000	0.000	1.000	0.870	0.000	0.376	0.821	0.441	0.556
	RF	0.071	0.950	0.867	0.260	0.406	0.951	0.880	0.621	0.536	0.833	0.729	0.668
	RL	0.000	0.997	0.903	0.000	0.040	0.996	0.871	0.200	0.016	0.982	0.644	0.127
	SVM	0.000	1.000	0.906	0.000	0.000	1.000	0.870	0.000	0	0.997	0.648	0
	ANN	0.000	0.999	0.000	0.000	0.029	0.996	0.180	0.054	0.13	0.948	0.661	0.479

Table A.7: Résultats RT (WMC, Fan-in) Classification Binaire

ANNEXE. B

Prédiction des Fautes de Sévérité Normale

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meanD	AUCD
ANT 1.3	NB	0.770	0.723	0.724	0.811	0.796	0.801	0.799	0.871
	J48	0.786	0.801	0.791	0.724	0.87	0.759	0.807	0.785
	RF	0.659	0.650	0.653	0.786	0.898	0.712	0.784	0.913
	RLog	0.778	0.746	0.751	0.854	0.874	0.770	0.815	0.892
	SVM	0.714	0.639	0.609	0.677	0.751	0.249	0.432	0.5
	ANN	0.760	0.738	0.777	0.843	0.853	0.735	0.787	0.883
ANT 1.4	NB	0.831	0.301	0.407	0.657	0.817	0.386	0.483	0.715
	J48	0.837	0.163	0.369	0.481	0.796	0.218	0.163	0.481
	RF	0.685	0.244	0.331	0.529	0.758	0.554	0.634	0.694
	RLog	0.826	0.189	0.184	0.682	0.806	0.342	0.426	0.734
	SVM	0.837	0.163	0.369	0.5	0.801	0.199	0.399	0.500
	ANN	0.821	0.160	0.047	0.678	0.814	0.382	0.483	0.719
ANT 1.5	NB	0.785	0.321	0.368	0.754	0.739	0.464	0.487	0.816
	J48	0.782	0.218	0.413	0.479	0.785	0.761	0.772	0.746
	RF	0.71	0.401	0.483	0.667	0.779	0.725	0.749	0.830
	RLog	0.782	0.286	0.302	0.769	0.758	0.528	0.570	0.830
	SVM	0.778	0.217	0.411	0.498	0.727	0.42	0.416	0.574
	ANN	0.784	0.423	0.535	0.782	0.78	0.689	0.729	0.822
ANT 1.6	NB	0.895	0.202	0.349	0.75	0.876	0.239	0.373	0.817
	J48	0.909	0.091	0.288	0.475	0.884	0.116	0.102	0.481
	RF	0.855	0.254	0.416	0.727	0.870	0.569	0.693	0.826
	RLog	0.901	0.118	0.176	0.778	0.870	0.156	0.216	0.824
	SVM	0.909	0.091	0.288	0.500	0.884	0.116	0.102	0.500
	ANN	0.909	0.091	0.288	0.681	0.881	0.124	0.119	0.750
ANT 1.7	NB	0.917	0.288	0.475	0.785	0.906	0.393	0.565	0.804
	J48	0.922	0.124	0.231	0.691	0.914	0.276	0.443	0.726
	RF	0.877	0.268	0.446	0.644	0.889	0.578	0.707	0.795
	RLog	0.930	0.223	0.399	0.824	0.918	0.324	0.498	0.842
	SVM	0.925	0.075	0.263	0.500	0.901	0.111	0.115	0.506
	ANN	0.928	0.189	0.366	0.808	0.916	0.330	0.510	0.828

Table B.1: Résultats de RT (LOC, Ce) pour la Prédiction de Bogues de Sévérité Normale

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meanD	AUCD
ANT 1.3	NB	0.236	0.901	0.462	0.679	1.000	0.000	0.000	0.744
	J48	0.691	0.690	0.691	0.656	0.944	0.437	0.642	0.701
	RF	0.673	0.676	0.674	0.726	0.981	0.606	0.771	0.834
	RLog	0.309	0.901	0.528	0.762	0.991	0.085	0.289	0.795
	SVM	0.018	0.972	0.133	0.495	1.000	0.000	0.000	0.500
	ANN	0.352	0.873	0.732	0.682	0.926	0.429	0.647	0.786
ANT 1.4	NB	0.000	0.960	0.000	0.588	0.054	0.960	0.228	0.72
	J48	0.000	1.000	0.000	0.481	0.000	1.000	0.000	0.468
	RF	0.207	0.879	0.427	0.574	0.541	0.886	0.692	0.762
	RLog	0.034	0.973	0.183	0.696	0.054	0.960	0.228	0.752
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.003	0.978	0.073	0.68	0.043	0.957	0.422	0.742
ANT 1.5	NB	0.785	0.309	0.348	0.71	0.73	0.427	0.427	0.732
	J48	0.758	0.336	0.398	0.564	0.764	0.763	0.763	0.719
	RF	0.73	0.429	0.515	0.653	0.748	0.684	0.712	0.785
	RLog	0.778	0.274	0.276	0.755	0.752	0.486	0.516	0.802
	SVM	0.782	0.218	0.413	0.5	0.706	0.339	0.222	0.523
	ANN	0.793	0.366	0.471	0.747	0.758	0.613	0.684	0.793
ANT 1.6	NB	0.000	0.975	0.000	0.577	0.000	0.975	0.000	0.613
	J48	0.000	1.000	0.000	0.475	0.000	0.994	0.000	0.482
	RF	0.156	0.941	0.383	0.660	0.500	0.925	0.680	0.764
	RLog	0.031	0.994	0.176	0.715	0.024	0.991	0.154	0.754
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.000	0.999	0.000	0.595	0.000	0.997	0.000	0.645
ANT 1.7	NB	0.036	0.990	0.188	0.650	0.053	0.985	0.228	0.666
	J48	0.000	1.000	0.000	0.478	0.000	1.000	0.000	0.484
	RF	0.286	0.955	0.522	0.711	0.526	0.945	0.705	0.819
	RLog	0.018	0.999	0.134	0.77	0.039	0.996	0.198	0.798
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.000	0.999	0.000	0.633	0.034	0.993	0.198	0.769

Table B.2: Résultats de RT (WMC, CBO) pour la Prédiction de Bogues de Sévérité Normale

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meanD	AUCD
ANT 1.3	NB	0.491	0.901	0.665	0.825	0.911	0.732	0.817	0.873
	J48	0.836	0.718	0.775	0.723	0.935	0.718	0.819	0.785
	RF	0.618	0.732	0.673	0.770	0.963	0.69	0.815	0.887
	RLog	0.618	0.845	0.723	0.837	0.935	0.676	0.795	0.878
	SVM	0.218	0.958	0.457	0.588	1.000	0.000	0.000	0.500
	ANN	0.711	0.776	0.771	0.821	0.937	0.655	0.787	0.869
ANT 1.4	NB	0.207	0.953	0.444	0.677	0.297	0.953	0.532	0.737
	J48	0.000	0.987	0.000	0.492	0.297	0.980	0.540	0.606
	RF	0.172	0.879	0.389	0.606	0.405	0.852	0.588	0.687
	RLog	0.069	0.993	0.262	0.703	0.270	0.966	0.511	0.751
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.032	0.993	0.180	0.702	0.251	0.962	0.534	0.744
ANT 1.5	NB	0.816	0.487	0.582	0.777	0.779	0.581	0.631	0.818
	J48	0.785	0.715	0.747	0.723	0.803	0.786	0.794	0.781
	RF	0.751	0.48	0.567	0.727	0.818	0.776	0.795	0.853
	RLog	0.819	0.488	0.584	0.813	0.8	0.652	0.700	0.854
	SVM	0.785	0.231	0.125	0.508	0.752	0.47	0.491	0.611
	ANN	0.811	0.541	0.643	0.808	0.666	0.749	0.281	0.799
ANT 1.6	NB	0.188	0.969	0.426	0.683	0.214	0.969	0.456	0.768
	J48	0.000	1.000	0.000	0.475	0.000	1.000	0.000	0.481
	RF	0.125	0.925	0.340	0.678	0.500	0.931	0.682	0.798
	RLog	0.031	0.991	0.176	0.785	0.071	0.978	0.264	0.825
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.000	0.997	0.000	0.694	0.020	0.984	0.145	0.824
ANT 1.7	NB	0.107	0.990	0.326	0.719	0.145	0.987	0.378	0.75
	J48	0.000	1.000	0.000	0.478	0.000	1.000	0.000	0.484
	RF	0.232	0.954	0.470	0.726	0.539	0.951	0.716	0.823
	RLog	0.036	0.994	0.188	0.811	0.079	0.990	0.280	0.826
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.018	0.997	0.174	0.763	0.115	0.988	0.342	0.819

Table B.3: Résultats de RT (RFC, Ca) pour la Prédiction de Bogues de Sévérité Normale

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meanD	AUCD
ANT 1.3	NB	0.582	0.901	0.724	0.832	0.850	0.803	0.826	0.889
	J48	0.855	0.732	0.791	0.724	0.930	0.704	0.809	0.790
	RF	0.636	0.690	0.663	0.788	0.991	0.620	0.784	0.915
	RLog	0.655	0.887	0.762	0.854	0.925	0.69	0.799	0.892
	SVM	0.582	0.915	0.730	0.749	0.958	0.451	0.657	0.704
	ANN	0.720	0.815	0.769	0.847	0.918	0.62	0.782	0.869
ANT 1.4	NB	0.172	0.953	0.405	0.662	0.243	0.953	0.481	0.720
	J48	0.000	1.000	0.000	0.481	0.027	0.987	0.163	0.481
	RF	0.138	0.799	0.332	0.532	0.486	0.826	0.634	0.689
	RLog	0.034	0.980	0.184	0.680	0.243	0.960	0.483	0.730
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.000	0.985	0.000	0.669	0.240	0.960	0.483	0.718
ANT 1.5	NB	0.778	0.353	0.421	0.758	0.758	0.539	0.584	0.819
	J48	0.782	0.218	0.413	0.479	0.785	0.761	0.772	0.746
	RF	0.724	0.394	0.476	0.671	0.794	0.732	0.759	0.830
	RLog	0.782	0.32	0.368	0.779	0.764	0.569	0.618	0.831
	SVM	0.778	0.217	0.411	0.498	0.739	0.47	0.495	0.605
	ANN	0.772	0.409	0.519	0.781	0.777	0.706	0.738	0.821
ANT 1.6	NB	0.125	0.963	0.347	0.767	0.143	0.953	0.369	0.822
	J48	0.000	1.000	0.000	0.475	0.000	1.000	0.000	0.481
	RF	0.188	0.922	0.416	0.726	0.524	0.919	0.694	0.825
	RLog	0.031	0.984	0.175	0.786	0.095	0.978	0.305	0.825
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.000	1.000	0.000	0.742	0.000	1.000	0.000	0.817
ANT 1.7	NB	0.304	0.962	0.540	0.802	0.408	0.961	0.626	0.816
	J48	0.054	0.993	0.231	0.691	0.197	0.993	0.443	0.726
	RF	0.196	0.933	0.428	0.638	0.526	0.929	0.699	0.788
	RLog	0.161	0.993	0.399	0.827	0.250	0.987	0.497	0.843
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.092	0.993	0.333	0.813	0.180	0.992	0.497	0.780

Table B.4: Résultats de RT (LOC, RFC) pour la Prédiction de Bogues de Sévérité Normale

ANNEXE. C

Prédiction des Fautes de Sévérité Élevée

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meanD	AUCD
ANT 1.3	NB	0.550	0.909	0.707	0.794	0.748	0.894	0.818	0.860
	J48	0.700	0.773	0.735	0.738	0.898	0.576	0.719	0.817
	RF	0.650	0.606	0.628	0.729	0.905	0.636	0.759	0.869
	RLog	0.683	0.818	0.748	0.837	0.850	0.788	0.819	0.892
	SVM	0.383	0.955	0.605	0.669	1.000	0.000	0.000	0.500
	ANN	0.687	0.828	0.768	0.825	0.867	0.707	0.789	0.884
ANT 1.4	NB	0.167	0.952	0.398	0.732	0.154	0.952	0.383	0.732
	J48	0.000	1.000	0.000	0.427	0.000	1.000	0.000	0.412
	RF	0.083	0.952	0.282	0.574	0.308	0.934	0.536	0.652
	RLog	0.083	0.994	0.288	0.744	0.077	0.994	0.277	0.752
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.000	1.000	0.000	0.684	0.000	1.000	0.000	0.678
ANT 1.5	NB	0.073	0.962	0.265	0.700	0.083	0.958	0.283	0.730
	J48	0.018	0.983	0.134	0.485	0.267	0.903	0.491	0.582
	RF	0.382	0.836	0.565	0.691	0.500	0.824	0.642	0.734
	RLog	0.036	0.975	0.188	0.721	0.067	0.975	0.255	0.740
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.036	0.970	0.194	0.714	0.088	0.953	0.294	0.730
ANT 1.6	NB	0.000	0.982	0.000	0.588	-	-	-	-
	J48	0.000	1.000	0.000	0.473	-	-	-	-
	RF	0.000	0.952	0.000	0.441	-	-	-	-
	RLog	0.000	1.000	0.000	0.605	-	-	-	-
	SVM	0.000	1.000	0.000	0.500	-	-	-	-
	ANN	0.000	1.000	0.000	0.566	-	-	-	-
ANT 1.7	NB	0.182	0.982	0.423	0.781	0.240	0.983	0.486	0.779
	J48	0.000	1.000	0.000	0.462	0.000	1.000	0.000	0.580
	RF	0.091	0.979	0.298	0.618	0.320	0.972	0.558	0.648
	RLog	0.000	0.999	0.000	0.798	0.000	0.997	0.000	0.819
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.000	1.000	0.000	0.694	0.000	1.000	0.000	0.693

Table C.1: Résultats de RT (LOC, Ce) pour la Prédition de Bogues de Sévérité Élevée

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meand	AUCD
ANT 1.3	NB	0.550	0.924	0.713	0.793	0.755	0.879	0.815	0.863
	J48	0.700	0.788	0.743	0.750	0.898	0.576	0.719	0.817
	RF	0.683	0.682	0.683	0.753	0.912	0.697	0.797	0.875
	RLog	0.683	0.818	0.748	0.837	0.850	0.788	0.819	0.891
	SVM	0.400	0.955	0.618	0.677	1.000	0.000	0.000	0.500
	ANN	0.687	0.828	0.768	0.824	0.867	0.706	0.790	0.884
ANT 1.4	NB	0.167	0.952	0.398	0.732	0.154	0.952	0.383	0.735
	J48	0.000	1.000	0.000	0.427	0.000	1.000	0.000	0.412
	RF	0.083	0.952	0.282	0.574	0.231	0.940	0.466	0.645
	RLog	0.083	0.994	0.288	0.744	0.077	0.994	0.277	0.752
	SVM	0.000	1.000	0.000	0.5	0.000	1.000	0.000	0.500
	ANN	0.000	1.000	0.000	0.684	0.000	1.000	0.000	0.682
ANT 1.5	NB	0.073	0.962	0.265	0.700	0.083	0.958	0.283	0.733
	J48	0.018	0.983	0.134	0.485	0.267	0.903	0.491	0.582
	RF	0.382	0.836	0.565	0.691	0.500	0.824	0.642	0.721
	RLog	0.036	0.975	0.188	0.721	0.067	0.975	0.255	0.740
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.036	0.970	0.194	0.714	0.088	0.953	0.290	0.731
ANT 1.6	NB	0.000	0.982	0.000	0.588	-	-	-	-
	J48	0.000	1.000	0.000	0.473	-	-	-	-
	RF	0.000	0.952	0.000	0.441	-	-	-	-
	RLog	0.000	1.000	0.000	0.605	-	-	-	-
	SVM	0.000	1.000	0.000	0.500	-	-	-	-
	ANN	0.000	1.000	0.000	0.566	-	-	-	-
ANT 1.7	NB	0.182	0.982	0.423	0.781	0.240	0.983	0.486	0.775
	J48	0.000	1.000	0.000	0.462	0.000	1.000	0.000	0.580
	RF	0.091	0.979	0.298	0.618	0.320	0.979	0.560	0.671
	RLog	0.000	0.999	0.000	0.798	0.000	0.997	0.000	0.818
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.000	1.000	0.000	0.694	0.000	1.000	0.000	0.697

Table C.2: Résultats de RT (LOC, Fan-in) pour la Prédiction de Bogues de Sévérité Élevée

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meanD	AUCD
ANT 1.3	NB	0.250	0.909	0.477	0.667	0.993	0.045	0.212	0.750
	J48	0.617	0.727	0.670	0.626	0.891	0.530	0.687	0.743
	RF	0.533	0.576	0.554	0.585	0.864	0.591	0.715	0.793
	RLog	0.417	0.833	0.589	0.734	0.966	0.273	0.513	0.797
	SVM	0.033	0.955	0.178	0.494	1.000	0.000	0.000	0.500
	ANN	0.418	0.784	0.707	0.669	0.92	0.485	0.703	0.787
ANT 1.4	NB	0.000	0.988	0.000	0.754	0.000	0.994	0.000	0.751
	J48	0.000	1.000	0.000	0.427	0.000	1.000	0.000	0.412
	RF	0.083	0.928	0.278	0.591	0.231	0.934	0.464	0.669
	RLog	0.000	0.988	0.000	0.756	0.000	0.994	0.000	0.749
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.000	0.998	0.000	0.606	0.000	1.000	0.000	0.580
ANT 1.5	NB	0.018	0.971	0.133	0.559	0.017	0.975	0.127	0.598
	J48	0.000	1.000	0.000	0.474	0.000	1.000	0.000	0.497
	RF	0.273	0.878	0.489	0.570	0.367	0.887	0.570	0.594
	RLog	0.018	0.992	0.134	0.665	0.017	0.992	0.129	0.683
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.000	0.998	0.000	0.608	0.007	0.993	0.105	0.653
ANT 1.6	NB	0.000	1.000	0.000	0.427	-	-	-	-
	J48	0.000	0.997	0.000	0.511	-	-	-	-
	RF	0.158	0.964	0.390	0.496	-	-	-	-
	RLog	0.000	1.000	0.000	0.550	-	-	-	-
	SVM	0.000	1.000	0.000	0.500	-	-	-	-
	ANN	0.000	1.000	0.000	0.543	-	-	-	-
ANT 1.7	NB	0.136	0.989	0.367	0.570	0.160	0.992	0.398	0.626
	J48	0.000	1.000	0.000	0.462	0.000	0.999	0.000	0.466
	RF	0.000	0.975	0.000	0.468	0.160	0.979	0.396	0.569
	RLog	0.000	0.999	0.000	0.707	0.080	0.999	0.283	0.731
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.000	0.999	0.000	0.602	0.040	0.999	0.283	0.658

Table C.3: Résultats de RT (WMC, CBO) pour la Prédiction de Bogues de Sévérité Élevée

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meanD	AUCD
ANT 1.3	NB	0.583	0.909	0.728	0.822	0.810	0.818	0.814	0.882
	J48	0.700	0.788	0.743	0.750	0.898	0.576	0.719	0.817
	RF	0.667	0.621	0.644	0.737	0.912	0.636	0.762	0.870
	RLog	0.683	0.818	0.748	0.837	0.850	0.773	0.811	0.892
	SVM	0.567	0.909	0.718	0.738	0.837	0.803	0.820	0.820
	ANN	0.711	0.821	0.775	0.822	0.865	0.714	0.798	0.885
ANT 1.4	NB	0.083	0.952	0.282	0.742	0.077	0.952	0.271	0.739
	J48	0.000	1.000	0.000	0.427	0.000	1.000	0.000	0.412
	RF	0.167	0.946	0.397	0.578	0.308	0.928	0.534	0.650
	RLog	0.000	1.000	0.000	0.756	0.000	1.000	0.000	0.756
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.000	1.000	0.000	0.712	0.000	1.000	0.000	0.714
ANT 1.5	NB	0.091	0.954	0.294	0.708	0.100	0.954	0.309	0.733
	J48	0.018	0.983	0.134	0.485	0.267	0.903	0.491	0.582
	RF	0.382	0.849	0.569	0.712	0.500	0.828	0.643	0.739
	RLog	0.055	0.966	0.230	0.727	0.067	0.958	0.253	0.742
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.015	0.985	0.13	0.718	0.038	0.971	0.234	0.671
ANT 1.6	NB	0.000	0.979	0.000	0.585	-	-	-	-
	J48	0.000	1.000	0.000	0.473	-	-	-	-
	RF	0.000	0.964	0.000	0.450	-	-	-	-
	RLog	0.000	1.000	0.000	0.607	-	-	-	-
	SVM	0.000	1.000	0.000	0.500	-	-	-	-
	ANN	0.000	1.000	0.000	0.569	-	-	-	-
ANT 1.7	NB	0.182	0.981	0.422	0.796	0.240	0.979	0.485	0.788
	J48	0.000	1.000	0.000	0.462	0.000	1.000	0.000	0.580
	RF	0.091	0.972	0.297	0.591	0.240	0.971	0.483	0.648
	RLog	0.000	0.999	0.000	0.805	0.000	0.996	0.000	0.818
	SVM	0.000	1.000	0.000	0.500	0.000	1.000	0.000	0.500
	ANN	0.000	1.000	0.000	0.731	0.000	1.000	0.000	0.732

Table C.4: Résultats de RT (LOC, RFC) pour la Prédiction de Bogues de Sévérité Élevée

ANNEXE. D

Prédiction du Niveau de Sévérité

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meand	AUCD
ANT 1.3	NB	0.746	0.730	0.718	0.789	0.798	0.859	0.825	0.860
	J48	0.730	0.726	0.727	0.729	0.779	0.726	0.749	0.797
	RF	0.667	0.667	0.667	0.769	0.836	0.734	0.774	0.875
	RLog	0.762	0.756	0.757	0.835	0.822	0.795	0.807	0.892
	SVM	0.675	0.647	0.592	0.661	0.690	0.310	0.462	0.500
	ANN	0.743	0.734	0.766	0.807	0.829	0.789	0.812	0.882
ANT 1.4	NB	0.775	0.371	0.294	0.723	0.768	0.419	0.367	0.746
	J48	0.787	0.213	0.316	0.471	0.751	0.273	0.236	0.575
	RF	0.657	0.433	0.398	0.654	0.714	0.603	0.597	0.741
	RLog	0.792	0.333	0.334	0.737	0.773	0.403	0.382	0.759
	SVM	0.787	0.213	0.316	0.500	0.757	0.243	0.316	0.500
	ANN	0.787	0.306	0.343	0.734	0.773	0.403	0.382	0.758
ANT 1.5	NB	0.652	0.398	0.000	0.713	0.607	0.474	0.357	0.766
	J48	0.655	0.427	0.000	0.635	0.614	0.671	0.491	0.706
	RF	0.577	0.600	0.381	0.668	0.648	0.740	0.670	0.744
	RLog	0.652	0.398	0.000	0.756	0.626	0.517	0.488	0.780
	SVM	0.659	0.341	0.000	0.500	0.601	0.410	0.121	0.505
	ANN	0.678	0.547	0.000	0.759	0.632	0.656	0.673	0.767
ANT 1.6	NB	0.858	0.223	0.294	0.678	0.844	0.224	0.280	0.703
	J48	0.872	0.128	0.316	0.468	0.858	0.142	0.316	0.486
	RF	0.787	0.256	0.342	0.599	0.793	0.420	0.435	0.684
	RLog	0.872	0.147	0.249	0.704	0.846	0.158	0.235	0.723
	SVM	0.872	0.128	0.316	0.500	0.858	0.142	0.316	0.500
	ANN	0.871	0.128	0.316	0.656	0.857	0.148	0.316	0.691
ANT 1.7	NB	0.894	0.273	0.368	0.792	0.880	0.350	0.399	0.782
	J48	0.906	0.120	0.254	0.591	0.894	0.264	0.341	0.750
	RF	0.835	0.296	0.331	0.682	0.854	0.543	0.620	0.775
	RLog	0.909	0.210	0.334	0.818	0.893	0.303	0.375	0.833
	SVM	0.906	0.094	0.316	0.500	0.882	0.118	0.316	0.500
	ANN	0.907	0.125	0.316	0.805	0.891	0.292	0.369	0.830

Table D.1: Résultats de RT (LOC, Ce) pour la Prédiction du Niveau de Sévérité

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meanD	AUCD
ANT 1.3	NB	0.746	0.728	0.713	0.795	0.798	0.859	0.825	0.859
	J48	0.730	0.726	0.727	0.729	0.779	0.726	0.749	0.797
	RF	0.706	0.706	0.706	0.776	0.854	0.768	0.803	0.886
	RLog	0.762	0.756	0.757	0.834	0.822	0.795	0.807	0.892
	SVM	0.675	0.647	0.592	0.661	0.690	0.310	0.000	0.500
	ANN	0.745	0.736	0.766	0.808	0.828	0.789	0.812	0.882
ANT 1.4	NB	0.775	0.371	0.294	0.726	0.773	0.419	0.382	0.745
	J48	0.787	0.213	0.316	0.471	0.751	0.273	0.236	0.575
	RF	0.663	0.414	0.399	0.659	0.724	0.589	0.578	0.728
	RLog	0.792	0.333	0.334	0.737	0.773	0.403	0.382	0.759
	SVM	0.787	0.213	0.316	0.500	0.757	0.243	0.316	0.500
	ANN	0.787	0.301	0.343	0.734	0.773	0.403	0.382	0.758
ANT 1.5	NB	0.652	0.398	0.000	0.713	0.607	0.474	0.357	0.768
	J48	0.659	0.457	0.000	0.660	0.611	0.694	0.529	0.708
	RF	0.601	0.629	0.388	0.679	0.657	0.754	0.683	0.754
	RLog	0.652	0.398	0.000	0.756	0.626	0.517	0.488	0.779
	SVM	0.659	0.341	0.000	0.500	0.601	0.410	0.121	0.505
	ANN	0.677	0.550	0.000	0.759	0.633	0.659	0.702	0.767
ANT 1.6	NB	0.858	0.223	0.294	0.675	0.844	0.224	0.280	0.704
	J48	0.872	0.128	0.316	0.468	0.858	0.142	0.316	0.486
	RF	0.784	0.294	0.342	0.603	0.788	0.469	0.434	0.690
	RLog	0.869	0.147	0.249	0.704	0.846	0.158	0.235	0.724
	SVM	0.872	0.128	0.316	0.500	0.858	0.142	0.316	0.500
	ANN	0.871	0.128	0.316	0.660	0.857	0.148	0.316	0.700
ANT 1.7	NB	0.893	0.273	0.367	0.786	0.878	0.350	0.399	0.783
	J48	0.906	0.120	0.254	0.585	0.894	0.264	0.341	0.749
	RF	0.846	0.322	0.354	0.689	0.863	0.582	0.630	0.771
	RLog	0.909	0.210	0.334	0.817	0.892	0.302	0.375	0.833
	SVM	0.906	0.094	0.316	0.500	0.882	0.118	0.316	0.500
	ANN	0.907	0.125	0.316	0.811	0.891	0.290	0.368	0.829

Table D.2: Résultats de RT (LOC, Fan-in) pour la Prédiction du Niveau de Sévérité

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meanD	AUCD
ANT 1.3	NB	0.563	0.532	0.436	0.627	0.700	0.339	0.212	0.766
	J48	0.690	0.685	0.686	0.666	0.779	0.634	0.680	0.708
	RF	0.595	0.593	0.593	0.634	0.812	0.732	0.765	0.823
	RLog	0.651	0.633	0.612	0.744	0.756	0.515	0.550	0.796
	SVM	0.532	0.488	0.220	0.510	0.690	0.310	0.462	0.500
	ANN	0.624	0.601	0.714	0.677	0.775	0.607	0.705	0.773
ANT 1.4	NB	0.775	0.271	0.294	0.714	0.751	0.336	0.309	0.761
	J48	0.787	0.233	0.249	0.493	0.730	0.379	0.329	0.590
	RF	0.730	0.467	0.361	0.669	0.724	0.605	0.547	0.725
	RLog	0.764	0.230	0.314	0.754	0.751	0.321	0.309	0.772
	SVM	0.787	0.213	0.316	0.500	0.757	0.243	0.316	0.500
	ANN	0.774	0.245	0.284	0.743	0.743	0.353	0.379	0.761
ANT 1.5	NB	0.648	0.392	0.000	0.660	0.595	0.471	0.266	0.702
	J48	0.659	0.341	0.000	0.473	0.617	0.686	0.662	0.686
	RF	0.604	0.571	0.436	0.592	0.648	0.692	0.667	0.693
	RLog	0.645	0.373	0.000	0.718	0.626	0.494	0.461	0.739
	SVM	0.659	0.341	0.000	0.498	0.601	0.399	0.000	0.500
	ANN	0.652	0.428	0.219	0.714	0.625	0.585	0.589	0.732
ANT 1.6	NB	0.861	0.146	0.315	0.584	0.844	0.158	0.315	0.644
	J48	0.872	0.128	0.316	0.468	0.852	0.142	0.316	0.492
	RF	0.810	0.258	0.348	0.601	0.832	0.456	0.501	0.664
	RLog	0.866	0.147	0.316	0.611	0.849	0.158	0.315	0.678
	SVM	0.872	0.128	0.316	0.500	0.858	0.142	0.316	0.500
	ANN	0.871	0.128	0.316	0.577	0.856	0.142	0.316	0.622
ANT 1.7	NB	0.899	0.094	0.316	0.651	0.878	0.157	0.355	0.756
	J48	0.906	0.094	0.316	0.496	0.884	0.127	0.198	0.492
	RF	0.863	0.285	0.374	0.620	0.877	0.506	0.522	0.747
	RLog	0.905	0.107	0.316	0.742	0.881	0.147	0.299	0.775
	SVM	0.906	0.094	0.316	0.500	0.882	0.118	0.316	0.500
	ANN	0.905	0.094	0.316	0.640	0.882	0.136	0.324	0.748

Table D.3: Résultats de RT (WMC, CBO) pour la Prédiction du Niveau de Sévérité

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meand	AUCD
ANT 1.3	NB	0.714	0.696	0.680	0.803	0.789	0.813	0.800	0.878
	J48	0.770	0.774	0.771	0.710	0.831	0.774	0.799	0.796
	RF	0.690	0.688	0.689	0.799	0.850	0.749	0.788	0.872
	RLog	0.730	0.723	0.722	0.818	0.854	0.784	0.814	0.883
	SVM	0.683	0.655	0.605	0.669	0.690	0.310	0.462	0.500
	ANN	0.741	0.736	0.757	0.803	0.840	0.772	0.807	0.876
ANT 1.4	NB	0.792	0.471	0.368	0.741	0.757	0.463	0.380	0.752
	J48	0.815	0.398	0.441	0.544	0.762	0.420	0.449	0.583
	RF	0.713	0.441	0.504	0.603	0.724	0.557	0.600	0.694
	RLog	0.803	0.395	0.313	0.750	0.768	0.465	0.381	0.771
	SVM	0.787	0.213	0.316	0.500	0.757	0.243	0.316	0.500
	ANN	0.796	0.315	0.264	0.701	0.778	0.384	0.384	0.710
ANT 1.5	NB	0.672	0.544	0.205	0.754	0.623	0.614	0.486	0.782
	J48	0.638	0.623	0.238	0.683	0.704	0.734	0.667	0.731
	RF	0.645	0.686	0.431	0.721	0.713	0.779	0.718	0.787
	RLog	0.689	0.540	0.000	0.778	0.654	0.633	0.623	0.794
	SVM	0.662	0.348	0.000	0.505	0.620	0.442	0.341	0.531
	ANN	0.700	0.607	0.000	0.770	0.677	0.692	0.739	0.788
ANT 1.6	NB	0.855	0.241	0.325	0.626	0.844	0.274	0.332	0.710
	J48	0.872	0.128	0.316	0.468	0.858	0.142	0.316	0.486
	RF	0.767	0.217	0.244	0.564	0.804	0.387	0.365	0.657
	RLog	0.869	0.166	0.249	0.718	0.858	0.192	0.281	0.743
	SVM	0.872	0.128	0.316	0.500	0.858	0.142	0.316	0.500
	ANN	0.872	0.133	0.316	0.671	0.855	0.150	0.316	0.709
ANT 1.7	NB	0.897	0.145	0.315	0.783	0.884	0.206	0.309	0.810
	J48	0.906	0.094	0.316	0.496	0.890	0.254	0.348	0.734
	RF	0.862	0.285	0.334	0.693	0.882	0.525	0.578	0.830
	RLog	0.901	0.119	0.316	0.798	0.876	0.166	0.298	0.819
	SVM	0.906	0.094	0.316	0.500	0.882	0.118	0.316	0.500
	ANN	0.904	0.103	0.316	0.725	0.879	0.172	0.316	0.804

Table D.4: Résultats de RT (RFC, Ca) pour la Prédiction du Niveau de Sévérité

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meand	AUCD
ANT 1.3	NB	0.550	0.909	0.707	0.790	0.748	0.909	0.825	0.861
	J48	0.700	0.758	0.728	0.727	0.823	0.667	0.741	0.790
	RF	0.717	0.742	0.729	0.779	0.932	0.727	0.823	0.894
	RLog	0.700	0.818	0.757	0.834	0.844	0.788	0.815	0.892
	SVM	0.367	0.955	0.592	0.661	1.000	0.000	0.000	0.500
	ANN	0.653	0.839	0.766	0.811	0.861	0.758	0.812	0.883
ANT 1.4	NB	0.775	0.371	0.294	0.723	0.768	0.419	0.367	0.748
	J48	0.787	0.213	0.316	0.471	0.746	0.272	0.235	0.575
	RF	0.669	0.454	0.418	0.654	0.730	0.621	0.611	0.737
	RLog	0.792	0.333	0.334	0.736	0.773	0.403	0.382	0.759
	SVM	0.787	0.213	0.316	0.500	0.757	0.243	0.316	0.500
	ANN	0.788	0.307	0.343	0.734	0.773	0.403	0.382	0.759
ANT 1.5	NB	0.000	0.964	0.000	0.756	0.176	0.964	0.412	0.757
	J48	0.000	0.943	0.000	0.634	0.412	0.834	0.586	0.701
	RF	0.222	0.756	0.410	0.673	0.662	0.720	0.690	0.743
	RLog	0.000	0.959	0.000	0.761	0.235	0.953	0.474	0.775
	SVM	0.000	1.000	0.000	0.500	0.029	0.995	0.171	0.507
	ANN	0.000	0.917	0.000	0.752	0.462	0.887	0.666	0.771
ANT 1.6	NB	0.858	0.223	0.294	0.676	0.844	0.224	0.280	0.701
	J48	0.872	0.128	0.316	0.468	0.858	0.142	0.316	0.486
	RF	0.770	0.293	0.317	0.606	0.788	0.436	0.434	0.686
	RLog	0.872	0.147	0.249	0.704	0.846	0.158	0.235	0.723
	SVM	0.872	0.128	0.316	0.500	0.858	0.142	0.316	0.500
	ANN	0.871	0.128	0.316	0.653	0.857	0.148	0.316	0.685
ANT 1.7	NB	0.894	0.273	0.368	0.791	0.880	0.350	0.399	0.781
	J48	0.907	0.133	0.281	0.605	0.893	0.254	0.286	0.738
	RF	0.909	0.210	0.334	0.817	0.893	0.303	0.375	0.832
	RLog	0.906	0.094	0.316	0.500	0.882	0.118	0.316	0.500
	SVM	0.906	0.094	0.316	0.500	0.882	0.118	0.316	0.500
	ANN	0.907	0.126	0.316	0.809	0.891	0.291	0.368	0.828

Table D.5: Résultats de RT* (RT (LOC, Ce), RT (LOC, Ca)) pour la Prédiction du Niveau de Sévérité

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meanD	AUCD
ANT 1.3	NB	0.550	0.924	0.713	0.791	0.748	0.909	0.825	0.862
	J48	0.700	0.758	0.728	0.727	0.823	0.667	0.741	0.790
	RF	0.700	0.742	0.721	0.776	0.932	0.727	0.823	0.893
	RLog	0.700	0.818	0.757	0.834	0.844	0.788	0.815	0.893
	SVM	0.367	0.955	0.592	0.661	1.000	0.000	0.000	0.500
	ANN	0.650	0.840	0.766	0.810	0.861	0.758	0.812	0.883
ANT 1.4	NB	0.775	0.371	0.294	0.723	0.768	0.419	0.367	0.745
	J48	0.787	0.213	0.316	0.471	0.746	0.272	0.235	0.575
	RF	0.674	0.435	0.419	0.658	0.735	0.622	0.622	0.735
	RLog	0.792	0.333	0.334	0.736	0.773	0.403	0.382	0.759
	SVM	0.787	0.213	0.316	0.500	0.757	0.243	0.316	0.500
	ANN	0.788	0.307	0.343	0.734	0.773	0.403	0.382	0.759
ANT 1.5	NB	0.000	0.964	0.000	0.755	0.176	0.964	0.412	0.755
	J48	0.000	0.943	0.000	0.634	0.412	0.834	0.586	0.701
	RF	0.222	0.756	0.410	0.672	0.662	0.731	0.695	0.747
	RLog	0.000	0.959	0.000	0.761	0.235	0.953	0.474	0.775
	SVM	0.000	1.000	0.000	0.500	0.029	0.995	0.171	0.507
	ANN	0.000	0.916	0.000	0.752	0.462	0.886	0.664	0.771
ANT 1.6	NB	0.858	0.223	0.294	0.676	0.844	0.224	0.280	0.704
	J48	0.872	0.128	0.316	0.468	0.858	0.142	0.316	0.486
	RF	0.770	0.293	0.317	0.606	0.788	0.436	0.434	0.685
	RLog	0.872	0.147	0.249	0.704	0.846	0.158	0.235	0.723
	SVM	0.872	0.128	0.316	0.500	0.858	0.142	0.316	0.500
	ANN	0.871	0.128	0.316	0.653	0.857	0.148	0.316	0.686
ANT 1.7	NB	0.894	0.273	0.368	0.790	0.880	0.350	0.399	0.780
	J48	0.907	0.133	0.281	0.605	0.893	0.254	0.286	0.738
	RF	0.909	0.210	0.334	0.817	0.893	0.303	0.375	0.832
	RLog	0.906	0.094	0.316	0.500	0.882	0.118	0.316	0.500
	SVM	0.906	0.094	0.316	0.500	0.882	0.118	0.316	0.500
	ANN	0.907	0.125	0.316	0.810	0.891	0.291	0.368	0.829

Table D.6: Résultats de RT* (RT (LOC, WMC), RT (LOC, Ca)) pour la Prédiction du Niveau de Sévérité

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meand	AUCD
ANT 1.3	NB	0.600	0.909	0.739	0.815	0.803	0.818	0.810	0.880
	J48	0.700	0.758	0.728	0.727	0.823	0.667	0.741	0.790
	RF	0.683	0.667	0.675	0.766	0.925	0.652	0.776	0.879
	RLog	0.700	0.818	0.757	0.834	0.850	0.773	0.811	0.892
	SVM	0.583	0.909	0.728	0.746	0.837	0.818	0.827	0.827
	ANN	0.700	0.815	0.764	0.823	0.864	0.751	0.811	0.886
ANT 1.4	NB	0.781	0.371	0.325	0.728	0.768	0.434	0.393	0.755
	J48	0.787	0.213	0.316	0.471	0.746	0.272	0.235	0.575
	RF	0.663	0.435	0.399	0.666	0.724	0.605	0.600	0.739
	RLog	0.781	0.331	0.325	0.737	0.773	0.403	0.382	0.759
	SVM	0.787	0.213	0.316	0.500	0.757	0.243	0.316	0.500
	ANN	0.785	0.261	0.331	0.733	0.773	0.403	0.382	0.757
ANT 1.5	NB	0.000	0.953	0.000	0.755	0.235	0.938	0.470	0.762
	J48	0.000	0.948	0.000	0.609	0.412	0.834	0.586	0.699
	RF	0.178	0.751	0.365	0.690	0.647	0.756	0.700	0.753
	RLog	0.000	0.953	0.000	0.760	0.250	0.938	0.484	0.774
	SVM	0.000	1.000	0.000	0.500	0.118	0.974	0.339	0.530
	ANN	0.000	0.907	0.000	0.751	0.504	0.872	0.680	0.771
ANT 1.6	NB	0.849	0.222	0.294	0.677	0.832	0.223	0.279	0.713
	J48	0.872	0.128	0.316	0.468	0.858	0.142	0.316	0.486
	RF	0.781	0.275	0.341	0.603	0.799	0.436	0.436	0.686
	RLog	0.866	0.147	0.248	0.703	0.846	0.191	0.280	0.726
	SVM	0.872	0.128	0.316	0.500	0.858	0.142	0.316	0.500
	ANN	0.872	0.128	0.316	0.678	0.858	0.142	0.316	0.719
ANT 1.7	NB	0.889	0.298	0.386	0.800	0.877	0.408	0.425	0.813
	J48	0.907	0.133	0.281	0.605	0.893	0.254	0.286	0.739
	RF	0.842	0.296	0.332	0.684	0.858	0.553	0.600	0.775
	RLog	0.907	0.223	0.334	0.817	0.890	0.302	0.375	0.834
	SVM	0.906	0.094	0.316	0.500	0.882	0.118	0.316	0.500
	ANN	0.907	0.110	0.316	0.817	0.891	0.286	0.368	0.831

Table D.7: Résultats de RT* (RT (LOC, RFC), RT (LOC, Ca), RT (RFC, Ca)) pour la Prédiction du Niveau de Sévérité

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meanD	AUCD
ANT 1.3	NB	0.567	0.909	0.718	0.798	0.748	0.894	0.818	0.864
	J48	0.717	0.788	0.751	0.732	0.884	0.682	0.777	0.782
	RF	0.667	0.682	0.674	0.772	0.912	0.652	0.771	0.881
	RLog	0.433	0.939	0.638	0.686	1.000	0.045	0.213	0.523
	SVM	0.654	0.831	0.766	0.810	0.862	0.758	0.812	0.883
	ANN	0.700	0.815	0.764	0.823	0.864	0.751	0.811	0.886
ANT 1.4	NB	0.775	0.372	0.294	0.725	0.773	0.419	0.382	0.748
	J48	0.787	0.213	0.316	0.471	0.751	0.273	0.236	0.575
	RF	0.663	0.435	0.399	0.666	0.724	0.605	0.600	0.739
	RLog	0.787	0.332	0.311	0.737	0.773	0.403	0.382	0.759
	SVM	0.787	0.213	0.316	0.500	0.757	0.243	0.316	0.500
	ANN	0.787	0.301	0.343	0.734	0.773	0.403	0.382	0.758
ANT 1.5	NB	0.000	0.953	0.000	0.755	0.235	0.938	0.470	0.762
	J48	0.000	0.948	0.000	0.609	0.412	0.834	0.586	0.699
	RF	0.178	0.751	0.365	0.690	0.647	0.756	0.700	0.753
	RLog	0.000	0.953	0.000	0.760	0.250	0.938	0.484	0.774
	SVM	0.000	1.000	0.000	0.500	0.118	0.974	0.339	0.530
	ANN	0.000	0.907	0.000	0.751	0.504	0.872	0.680	0.771
ANT 1.6	NB	0.858	0.223	0.294	0.676	0.844	0.224	0.280	0.703
	J48	0.872	0.128	0.316	0.468	0.858	0.142	0.316	0.486
	RF	0.872	0.147	0.249	0.704	0.846	0.158	0.235	0.723
	RLog	0.872	0.128	0.316	0.500	0.858	0.142	0.316	0.500
	SVM	0.872	0.128	0.316	0.500	0.858	0.142	0.316	0.500
	ANN	0.872	0.128	0.316	0.678	0.858	0.142	0.316	0.719
ANT 1.7	NB	0.894	0.273	0.368	0.793	0.880	0.350	0.399	0.783
	J48	0.906	0.120	0.254	0.591	0.894	0.264	0.341	0.750
	RF	0.827	0.308	0.342	0.672	0.855	0.591	0.637	0.774
	RLog	0.909	0.210	0.334	0.818	0.893	0.303	0.375	0.833
	SVM	0.906	0.094	0.316	0.500	0.882	0.118	0.316	0.500
	ANN	0.907	0.125	0.316	0.805	0.891	0.292	0.369	0.830

Table D.8: Résultats de RT* (LOC, RT (RFC, Ca)) pour la Prédiction du Niveau de Sévérité

		TPR	TNR	g-mean	AUC	TPRD	TNRD	g-meanD	AUCD
ANT 1.3	NB	0.517	0.939	0.697	0.768	0.707	0.909	0.802	0.846
	J48	0.667	0.788	0.725	0.723	0.837	0.652	0.738	0.801
	RF	0.667	0.712	0.689	0.769	0.925	0.652	0.776	0.876
	RLog	0.633	0.879	0.746	0.827	0.837	0.818	0.827	0.886
	SVM	0.267	0.985	0.512	0.626	1.000	0.000	0.000	0.500
	ANN	0.573	0.863	0.766	0.791	0.855	0.762	0.813	0.872
ANT 1.4	NB	0.775	0.371	0.294	0.703	0.773	0.419	0.382	0.723
	J48	0.787	0.213	0.316	0.471	0.746	0.257	0.315	0.541
	RF	0.702	0.461	0.453	0.674	0.730	0.623	0.637	0.738
	RLog	0.787	0.332	0.350	0.730	0.773	0.403	0.382	0.759
	SVM	0.787	0.213	0.316	0.500	0.757	0.243	0.316	0.500
	ANN	0.786	0.306	0.341	0.729	0.773	0.410	0.382	0.756
ANT 1.5	NB	0.000	0.964	0.000	0.723	0.162	0.964	0.395	0.725
	J48	0.000	0.948	0.000	0.616	0.412	0.798	0.573	0.706
	RF	0.133	0.720	0.310	0.663	0.647	0.731	0.688	0.739
	RLog	0.000	0.964	0.000	0.745	0.191	0.959	0.428	0.757
	SVM	0.000	1.000	0.000	0.500	0.029	0.995	0.171	0.509
	ANN	0.000	0.931	0.000	0.731	0.397	0.898	0.639	0.754
ANT 1.6	NB	0.858	0.223	0.294	0.636	0.844	0.224	0.280	0.686
	J48	0.872	0.128	0.316	0.468	0.858	0.142	0.316	0.486
	RF	0.770	0.236	0.287	0.593	0.774	0.401	0.415	0.675
	RLog	0.872	0.147	0.249	0.682	0.849	0.158	0.236	0.702
	SVM	0.872	0.128	0.316	0.500	0.858	0.142	0.316	0.500
	ANN	0.871	0.128	0.316	0.647	0.856	0.149	0.316	0.674
ANT 1.7	NB	0.893	0.273	0.367	0.755	0.878	0.340	0.393	0.745
	J48	0.906	0.094	0.316	0.539	0.895	0.274	0.348	0.778
	RF	0.848	0.271	0.354	0.651	0.854	0.572	0.624	0.765
	RLog	0.907	0.197	0.319	0.803	0.889	0.273	0.352	0.82
	SVM	0.906	0.094	0.316	0.500	0.882	0.118	0.316	0.500
	ANN	0.907	0.128	0.316	0.785	0.89	0.293	0.374	0.808

Table D.9: Résultats de RT* (RFC, RT (LOC, Ca)) pour la Prédiction du Niveau de Sévérité

ANNEXE. E

Prédiction de Risque

		r	MAE	RMSE	RAE	RSE
ANT 1.3	LR	0.586	5.421	7.742	72.49	81.04
	Gaussian	0.635	5.164	7.340	69.05	76.83
	RF	0.443	6.091	9.279	81.44	97.12
	M5P	0.577	5.449	7.826	72.85	81.91
	SmoReg	0.605	4.841	7.874	64.72	82.42
	ANN	0.556	5.660	8.055	75.67	84.32
ANT 1.4	RL	0.391	1.271	2.003	82.5	91.573
	Gaussian	0.075	1.422	2.177	92.295	99.554
	RF	0.209	1.31	2.392	85.002	109.372
	M5P	0.391	1.271	2.003	82.5	91.573
	SmoReg	0.385	0.972	2.379	63.106	108.764
	ANN	0.278	1.343	2.179	87.177	99.642
ANT 1.5	RL	0.361	2.652	3.710	86.435	93.629
	Gaussian	0.031	3.003	3.958	97.872	99.878
	RF	0.325	2.653	4.061	86.471	102.474
	M5P	0.404	2.448	3.681	79.776	92.894
	SmoReg	0.408	2.217	3.820	72.265	96.402
	ANN	0.477	2.696	3.743	87.889	94.46
ANT 1.6	RL	0.180	0.877	1.606	94.225	98.302
	Gaussian	-0.128	0.893	1.632	96.017	99.884
	RF	0.062	0.883	1.940	94.942	118.722
	M5P	0.262	0.842	1.570	90.533	96.122
	SmoReg	0.033	0.534	1.710	57.390	104.667
	ANN	0.083	0.948	1.719	101.886	105.23
ANT 1.7	RL	0.428	0.656	1.403	82.600	90.375
	Gaussian	0.124	0.734	1.550	92.425	99.842
	RF	0.286	0.672	1.658	84.691	106.811
	M5P	0.428	0.656	1.403	82.600	90.375
	SmoReg	0.218	0.438	1.610	55.216	103.691
	ANN	0.355	0.852	1.488	107.367	95.859

Table E.1: Résultats de RT (LOC, Ce) pour la Prédiction de Risque

		r	MAE	RMSE	RAE	RSE
ANT 1.3	LR	0.588	5.412	7.725	72.365	80.855
	Gaussian	0.222	7.096	9.480	94.883	99.231
	RF	0.442	6.029	9.280	80.610	97.132
	M5P	0.579	5.440	7.809	72.738	81.735
	SmoReg	0.602	4.954	7.830	66.241	81.960
	ANN	0.556	5.662	8.056	75.707	84.329
ANT 1.4	RL	0.392	1.266	2.001	82.161	91.501
	Gaussian	0.101	1.413	2.177	91.687	99.525
	RF	0.180	1.341	2.426	87.073	110.93
	M5P	0.323	0.972	2.378	63.109	108.73
	SmoReg	0.385	0.972	2.379	63.106	108.764
	ANN	0.279	1.343	2.179	87.195	99.622
ANT 1.5	RL	0.378	2.636	3.677	85.930	92.799
	Gaussian	0.048	2.998	3.957	97.732	99.860
	RF	0.345	2.594	4.017	84.555	101.364
	M5P	0.413	2.442	3.655	79.599	92.228
	SmoReg	0.416	2.217	3.800	72.272	95.897
	ANN	0.467	2.739	3.786	89.272	95.534
ANT 1.6	RL	0.190	0.874	1.602	93.913	98.036
	Gaussian	-0.123	0.891	1.632	95.710	99.881
	RF	0.055	0.907	1.962	97.509	120.067
	M5P	0.266	0.840	1.569	90.292	96.006
	SmoReg	0.088	0.534	1.710	57.357	104.652
	ANN	0.085	0.949	1.719	101.985	105.211
ANT 1.7	RL	0.438	0.656	1.395	82.590	89.879
	Gaussian	0.154	0.729	1.550	91.827	99.828
	RF	0.300	0.657	1.638	82.763	105.490
	M5P	0.431	0.656	1.401	82.643	90.263
	SmoReg	0.453	0.438	1.610	55.184	103.687
	ANN	0.356	0.852	1.488	107.365	95.850

Table E.2: Résultats de RT (LOC, Fan-in) pour la Prédiction de Risque

		r	MAE	RMSE	RAE	RSE
ANT 1.3	LR	0.233	7.018	9.311	93.835	97.462
	Gaussian	-0.134	7.316	9.540	97.826	99.854
	RF	0.342	7.441	10.527	99.487	110.186
	M5P	0.300	6.825	9.109	91.252	95.348
	SmoReg	0.201	6.691	10.67	89.466	111.77
	ANN	0.336	7.291	9.295	97.482	97.288
ANT 1.4	RL	0.286	1.376	2.088	89.313	95.462
	Gaussian	0.147	1.485	2.185	96.411	99.884
	RF	0.265	1.272	2.279	82.565	104.211
	M5P	0.361	1.250	2.032	81.125	92.908
	SmoReg	0.276	0.973	2.377	63.150	108.685
	ANN	0.178	1.324	2.291	85.907	104.762
ANT 1.5	RL	0.295	2.844	3.781	92.701	95.409
	Gaussian	-0.046	3.025	3.960	98.594	99.924
	RF	0.358	2.777	4.007	90.50	101.123
	M5P	0.380	2.606	3.662	84.948	92.403
	SmoReg	0.201	2.405	4.339	78.377	109.491
	ANN	0.296	3.159	4.093	102.959	103.282
ANT 1.6	RL	0.109	0.91	1.619	97.825	99.066
	Gaussian	-0.181	0.913	1.634	98.164	99.98
	RF	0.12	0.879	1.875	94.465	114.773
	M5P	0.152	0.869	1.614	93.36	98.781
	SmoReg	0.035	0.536	1.709	57.584	104.618
	ANN	-0.032	0.97	1.759	104.256	107.659
ANT 1.7	RL	0.226	0.748	1.531	94.195	98.616
	Gaussian	-0.069	0.768	1.552	96.683	99.968
	RF	0.168	0.751	1.569	94.578	101.065
	M5P	0.431	0.656	1.401	82.643	90.263
	SmoReg	0.073	0.442	1.609	55.634	103.653
	ANN	0.093	0.882	1.727	111.066	111.218

Table E.3: Résultats de RT (WMC, CBO) pour la Prédiction de Risque

		r	MAE	RMSE	RAE	RSE
ANT 1.3	LR	0.651	5.004	7.211	66.902	75.478
	Gaussian	0.629	5.117	7.381	68.424	77.260
	RF	0.439	6.091	9.282	81.437	97.162
	M5P	0.651	5.004	7.211	66.902	75.478
	SmoReg	0.659	4.769	7.358	63.763	77.022
	ANN	0.557	5.659	8.059	75.669	84.354
ANT 1.4	RL	0.391	1.257	2.001	81.595	91.507
	Gaussian	0.355	1.287	2.033	83.597	92.987
	RF	0.203	1.335	2.419	86.661	110.591
	M5P	0.391	1.257	2.001	81.595	91.507
	SmoReg	0.351	0.972	2.378	63.118	108.742
	ANN	0.264	1.371	2.196	88.996	100.391
ANT 1.5	RL	0.468	2.494	3.496	81.300	88.229
	Gaussian	0.504	2.337	3.414	76.189	86.168
	RF	0.347	2.587	4.007	84.311	101.126
	M5P	0.431	2.461	3.594	80.201	90.703
	SmoReg	0.472	2.096	3.607	68.333	91.047
	ANN	0.482	2.644	3.571	86.194	90.104
ANT 1.6	RL	0.232	0.858	1.584	92.253	96.965
	Gaussian	0.243	0.845	1.580	90.890	96.733
	RF	0.070	0.878	1.930	94.385	118.11
	M5P	0.208	0.853	1.596	91.653	97.679
	SmoReg	0.087	0.533	1.710	57.277	104.679
	ANN	0.121	1.004	1.698	107.894	103.903
ANT 1.7	RL	0.452	0.678	1.384	85.464	89.128
	Gaussian	0.459	0.640	1.377	80.728	88.711
	RF	0.280	0.681	1.669	85.763	107.51
	M5P	0.452	0.678	1.384	85.464	89.128
	SmoReg	0.341	0.440	1.501	55.435	96.686
	ANN	0.343	0.870	1.499	109.589	96.517

Table E.4: Résultats de RT (LOC, RFC) pour la Prédiction de Risque

		r	MAE	RMSE	RAE	RSE
ANT 1.3	LR	0.586	5.423	7.745	72.514	81.07
	Gaussian	0.581	7.129	8.525	95.316	89.236
	RF	0.436	5.984	9.485	80.007	99.281
	M5P	0.577	5.451	7.829	72.887	81.951
	SmoReg	0.601	4.954	7.838	66.235	82.047
	ANN	0.552	5.686	8.084	76.026	84.618
ANT 1.4	LR	0.391	1.271	2.003	82.468	91.558
	Gaussian	0.378	1.617	2.081	104.945	95.13
	RF	0.228	1.317	2.375	85.487	108.582
	M5P	0.391	1.271	2.003	82.468	91.558
	SmoReg	0.415	0.972	2.378	63.105	108.729
	ANN	0.278	1.344	2.179	87.215	99.636
ANT 1.5	LR	0.362	2.651	3.71	86.42	93.611
	Gaussian	0.344	3.024	3.78	98.564	95.379
	RF	0.345	2.603	4.019	84.834	101.418
	M5P	0.404	2.449	3.681	79.807	92.899
	SmoReg	0.408	2.216	3.821	72.24	96.413
	ANN	0.477	2.698	3.744	87.94	94.47
ANT 1.6	LR	0.18	0.877	1.606	94.214	98.296
	Gaussian	0.149	1.017	1.617	109.262	98.952
	RF	0.061	0.902	2.004	96.965	122.643
	M5P	0.266	0.84	1.569	90.319	96.016
	SmoReg	0.013	0.535	1.71	57.452	104.658
	ANN	0.083	0.948	1.72	101.929	105.25
ANT 1.7	LR	0.429	0.655	1.403	82.549	90.344
	Gaussian	0.421	0.944	1.451	118.844	93.478
	RF	0.281	0.681	1.681	85.721	108.291
	M5P	0.429	0.655	1.403	82.549	90.344
	SmoReg	0.46	0.438	1.61	55.19	103.67
	ANN	0.357	0.849	1.487	106.973	95.793

Table E.5: Résultats de RT* (RT (LOC, Ce), RT (LOC, Ca)) pour la Prédiction de Risque

		r	MAE	RMSE	RAE	RSE
ANT 1.3	LR	0.651	5.004	7.211	66.902	75.478
	Gaussian	0.643	7.237	8.558	96.761	89.579
	RF	0.439	6.091	9.282	81.437	97.162
	M5P	0.651	5.004	7.211	66.902	75.478
	SmoReg	0.659	4.766	7.353	63.719	76.968
	ANN	0.557	5.659	8.059	75.669	84.354
ANT 1.4	LR	0.391	1.257	2.001	81.595	91.507
	Gaussian	0.372	1.649	2.099	107.017	95.961
	RF	0.205	1.328	2.414	86.201	110.391
	M5P	0.391	1.257	2.001	81.595	91.507
	SmoReg	0.335	0.973	2.377	63.158	108.68
	ANN	0.264	1.371	2.196	88.996	100.391
ANT 1.5	LR	0.468	2.494	3.496	81.3	88.229
	Gaussian	0.46	3.018	3.688	98.37	93.055
	RF	0.345	2.593	4.01	84.526	101.196
	M5P	0.431	2.461	3.594	80.201	90.703
	SmoReg	0.454	2.223	3.665	72.459	92.479
	ANN	0.482	2.644	3.571	86.194	90.104
ANT 1.6	LR	0.232	0.858	1.584	92.253	96.965
	Gaussian	0.206	1.039	1.607	111.699	98.348
	RF	0.07	0.88	1.924	94.55	117.767
	M5P	0.208	0.853	1.596	91.653	97.679
	SmoReg	0.235	0.532	1.711	57.151	104.701
	ANN	0.121	1.003	1.698	107.806	103.955
ANT 1.7	LR	0.452	0.678	1.384	85.464	89.128
	Gaussian	0.446	0.972	1.452	122.385	93.546
	RF	0.287	0.675	1.656	84.97	106.658
	M5P	0.452	0.678	1.384	85.464	89.128
	SmoReg	0.312	0.439	1.61	55.238	103.667
	ANN	0.343	0.87	1.499	109.589	96.517

Table E.6: Résultats de RT* (RT (LOC, RFC), RT (LOC, Ca), RT (RFC, Ca)) pour la Prédiction de Risque

		r	MAE	RMSE	RAE	RSE
ANT 1.3	LR	0.585	5.43	7.757	72.599	81.191
	Gaussian	0.58	7.129	8.528	95.321	89.263
	RF	0.422	6.091	9.571	81.445	100.185
	M5P	0.576	5.458	7.841	72.971	82.072
	SmoReg	0.599	4.962	7.857	66.341	82.241
	ANN	0.552	5.687	8.084	76.043	84.621
ANT 1.4	LR	0.391	1.27	2.002	82.459	91.553
	Gaussian	0.378	1.617	2.081	104.946	95.129
	RF	0.229	1.309	2.36	84.98	107.924
	M5P	0.391	1.27	2.002	82.459	91.553
	SmoReg	0.316	0.972	2.377	63.104	108.7
	ANN	0.278	1.344	2.179	87.216	99.634
ANT 1.5	LR	0.364	2.65	3.706	86.364	93.516
	Gaussian	0.346	3.024	3.778	98.55	95.332
	RF	0.34	2.621	4.034	85.444	101.787
	M5P	0.405	2.448	3.678	79.788	92.824
	SmoReg	0.41	2.215	3.814	72.192	96.247
	ANN	0.477	2.698	3.744	87.948	94.474
ANT 1.6	LR	0.181	0.876	1.606	94.18	98.266
	Gaussian	0.15	1.017	1.616	109.292	98.935
	RF	0.058	0.909	2.01	97.653	123.039
	M5P	0.266	0.84	1.569	90.315	96.014
	SmoReg	0.019	0.535	1.71	57.452	104.655
	ANN	0.083	0.949	1.72	101.94	105.248
ANT 1.7	LR	0.43	0.655	1.402	82.541	90.278
	Gaussian	0.423	0.944	1.451	118.911	93.446
	RF	0.287	0.679	1.675	85.565	107.872
	M5P	0.43	0.655	1.402	82.541	90.278
	SmoReg	0.422	0.438	1.609	55.23	103.653
	ANN	0.357	0.849	1.487	106.973	95.791

Table E.7: Résultats de RT* (RT (LOC, WMC), RT (LOC, Ca)) pour la Prédiction de Risque

		r	MAE	RMSE	RAE	RSE
ANT 1.3	LR	0.603	5.342	7.603	71.428	79.582
	Gaussian	0.274	7.076	9.472	94.612	99.143
	RF	0.45	6.067	9.221	81.124	96.514
	M5P	0.603	5.342	7.603	71.428	79.582
	SmoReg	0.617	4.897	7.691	65.475	80.509
	ANN	0.556	5.662	8.06	75.7	84.365
ANT 1.4	LR	0.39	1.27	2.003	82.446	91.591
	Gaussian	0.096	1.416	2.177	91.934	99.523
	RF	0.19	1.319	2.411	85.607	110.242
	M5P	0.39	1.27	2.003	82.446	91.591
	SmoReg	0.411	0.972	2.379	63.107	108.764
	ANN	0.278	1.344	2.18	87.206	99.661
ANT 1.5	LR	0.361	2.652	3.711	86.45	93.651
	Gaussian	0.03	3.003	3.958	97.873	99.879
	RF	0.329	2.634	4.057	85.843	102.386
	M5P	0.403	2.447	3.685	79.768	93.005
	SmoReg	0.406	2.217	3.825	72.267	96.525
	ANN	0.477	2.696	3.743	87.888	94.46
ANT 1.6	LR	0.18	0.877	1.606	94.238	98.308
	Gaussian	-0.128	0.893	1.632	96.022	99.884
	RF	0.064	0.882	1.931	94.838	118.168
	M5P	0.262	0.842	1.57	90.536	96.123
	SmoReg	0.027	0.533	1.711	57.238	104.704
	ANN	0.083	0.948	1.719	101.883	105.225
ANT 1.7	LR	0.427	0.656	1.404	82.621	90.429
	Gaussian	0.123	0.734	1.55	92.457	99.843
	RF	0.28	0.695	1.693	87.596	109.05
	M5P	0.422	0.657	1.409	82.698	90.759
	SmoReg	0.279	0.439	1.61	55.242	103.664
	ANN	0.355	0.853	1.489	107.399	95.879

Table E.8: Résultats de RT* (LOC, RT (RFC, Ca)) pour la Prédiction de Risque

		r	MAE	RMSE	RAE	RSE
ANT 1.3	LR	0.546	5.706	8.052	76.291	84.287
	Gaussian	0.538	6.937	8.424	92.749	88.171
	RF	0.469	5.906	9.069	78.964	94.923
	M5P	0.507	5.855	8.432	78.288	88.255
	SmoReg	0.584	4.854	8.271	64.905	86.577
	ANN	0.549	5.617	8.08	75.104	84.571
ANT 1.4	LR	0.383	1.284	2.011	83.371	91.935
	Gaussian	0.375	1.571	2.062	102.0	94.257
	RF	0.159	1.379	2.438	89.524	111.462
	M5P	0.383	1.284	2.011	83.371	91.935
	SmoReg	0.31	0.972	2.375	63.105	108.589
	ANN	0.296	1.283	2.164	83.288	98.923
ANT 1.5	LR	0.32	2.747	3.776	89.527	95.285
	Gaussian	0.299	3.012	3.807	98.173	96.069
	RF	0.355	2.665	3.984	86.851	100.549
	M5P	0.432	2.452	3.591	79.911	90.618
	SmoReg	0.343	2.229	4.077	72.652	102.884
	ANN	0.46	2.774	3.811	90.431	96.178
ANT 1.6	LR	0.162	0.884	1.612	95.008	98.653
	Gaussian	0.135	0.994	1.618	106.864	99.05
	RF	0.028	0.927	1.927	99.597	117.956
	M5P	0.204	0.855	1.597	91.876	97.76
	SmoReg	-0.019	0.535	1.71	57.461	104.669
	ANN	0.071	0.949	1.726	101.98	105.671
ANT 1.7	LR	0.428	0.651	1.404	81.974	90.447
	Gaussian	0.421	0.891	1.433	112.17	92.286
	RF	0.298	0.682	1.671	85.909	107.62
	M5P	0.429	0.649	1.404	81.795	90.436
	SmoReg	0.432	0.438	1.609	55.171	103.636
	ANN	0.387	0.796	1.463	100.274	94.232

Table E.9: Résultats de RT* (RFC, RT (LOC, Ca)) pour la Prédition de Risque