

Chapitre X

Reconstruction de la masse d'une résonance grâce au *Machine Learning*

Sommaire

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 2 | Événements utilisés | 2 |
| 2.1 | Génération avec FASTSIM | 3 |
| 2.2 | Sélection des événements | 3 |
| 2.3 | Événements obtenus et pondération | 5 |
| 2.4 | Cible et variables d'entrée des modèles | 6 |
| 3 | Arbres de décision améliorés | 7 |
| 3.1 | Arbres de décision | 7 |
| 3.2 | <i>Gradient Boosting</i> | 8 |
| 3.3 | Fonction de coût et <i>Gradient Descent</i> | 8 |
| 3.4 | Sous-entraînement et surentraînement | 9 |
| 4 | Réseaux de neurones profonds | 9 |
| 4.1 | Neurones | 10 |
| 4.2 | Réseaux de neurones | 12 |
| 4.3 | Entraînement | 12 |
| 5 | Optimisation des hyper-paramètres et choix d'un modèle | 15 |
| 5.1 | Variables d'entrée | 17 |
| 5.2 | Type de modèle | 18 |
| 5.3 | Fonction de coût | 20 |
| 5.4 | Algorithme d'optimisation | 21 |
| 5.5 | Autres hyper-paramètres | 22 |
| 6 | Discussions | 27 |
| 6.1 | Effet de l'empilement | 29 |
| 6.2 | Effet de la reconstruction des particules | 30 |
| 6.3 | Effet des faux taus hadroniques | 31 |
| 6.4 | Effet de la séparation des canaux | 34 |
| 6.5 | Effet de la définition de E_T^{miss} | 37 |
| 6.6 | Effet de l'intervalle de masse | 37 |
| 6.7 | Modèle final | 38 |
| 7 | Utilisation du modèle dans les analyses CMS | 38 |
| 8 | Conclusion | 38 |

1 Introduction

L'utilisation de l'intelligence artificielle (IA) s'est grandement développée au cours des dernières années. L'IA est la capacité qu'ont des programmes à prendre des décisions, selon les informations qui leurs sont données par exemple sur leur environnement, de manière à maximiser leurs chances

de réussite. L'entreprise Google DeepMind a par exemple développé AlphaGo [1], un programme destiné à jouer au jeu de Go, qui a battu en 2016 le champion du monde de la discipline 4 à 1.

Le *Machine Learning* (ML) est une branche de l'IA dans laquelle un modèle (algorithme ou programme) s'améliore à réaliser une tâche par accumulation d'expérience sur des jeux de données d'entraînement, sans pour autant être programmé explicitement pour réaliser cette tâche. Pour y parvenir, les jeux de données d'entraînement comprennent les informations $\{\vec{x}_i\}$ à donner au modèle ainsi que les « bonnes réponses » $\{y_{\text{vraie},i}\}$ qu'il doit fournir en sortie. L'objectif du modèle est donc de donner une fonction F approximant celle reliant les entrées $\{\vec{x}_i\}$ aux cibles $\{y_{\text{vraie},i}\}$. Il peut alors donner une prédiction $y_{\text{préd}}$ sur une nouvelle entrée selon $y_{\text{préd}} = F(\vec{x})$. La tâche du modèle est :

une classification lorsque y est discrète, par exemple lorsqu'il s'agit de déterminer si une image représente un chat ou un chien [2];

une régression lorsque y est continue, par exemple estimer le prix d'un bien immobilier [3].

Les applications du ML à la physique des particules sont variées et proposent de nombreux sujets d'étude [4-7]. Dans les chapitres précédents, le ML est déjà activement utilisé pour diverses tâches :

- identification des jets issus de quarks b (b -tagging) avec DEEPCSV [8-10];
- identification des taus hadroniques avec DEEPTAU [11];
- catégorisation des événements comme exposé dans le chapitre 4 [12, 13].

Dans les événements $H \rightarrow \tau\tau$ présentés au chapitre 1, et plus généralement lors de tout processus physique $X \rightarrow \tau\tau$ où une particule X se désintègre en paire de leptons tau, des neutrinos sont émis lors des désintégrations des taus. Or, ils sont invisibles dans les détecteurs tels que CMS ou ATLAS. Il est donc impossible de déterminer la masse invariante totale du système $\tau\tau$ issu de X . Plusieurs méthodes ont été développées afin de reconstruire la masse du système $\tau\tau$ [14-16]. Dans le cadre des analyses $H \rightarrow \tau\tau$, la collaboration CMS utilise SVFIT [17].

La reconstruction la masse de la particule X , ou résonance, se désintégrant en paire de leptons tau grâce au *Machine Learning* a été étudiée par BÄRTSCHI & coll. [18] dans le cas où X est un boson de Higgs avec une masse entre 80 et 300 GeV. Ils ont obtenu une résolution de 8,4 % sur la masse du Higgs, contre 17 % avec SVFIT. Le temps de calcul nécessaire à l'obtention de la masse est de plus bien plus court avec le ML. L'utilisation du ML est donc très prometteuse. Cependant, ces travaux utilisent des événements générés avec une simulation grossière du détecteur CMS basée sur DELPHES [19, 20] et sans empilement, notion introduite dans le chapitre 2.

Les travaux présentés dans ce chapitre vont plus loin. La génération des événements, introduite dans la section 2, utilise FASTSIM [21-24] pour modéliser le détecteur CMS. Bien qu'il ne s'agisse pas de la simulation complète basée sur GEANT4 [25-27], FASTSIM est bien plus proche de la réalité que DELPHES. De plus, l'empilement est pris en compte. Les modèles obtenus sont ainsi directement utilisables dans de réelles analyses, telles que celle présentée dans le chapitre 4.

Deux types de modèle sont étudiés :

- des arbres de décision améliorés, introduits section 3;
- des réseaux de neurones profonds, introduits section 4.

La comparaison des modèles obtenus et la sélection de l'un d'entre eux est présentée section 5. Dans la section 6, divers effets sur les performances des modèles sont discutés, en particulier la prise en compte de l'empilement. Enfin, l'utilisation en conditions réelles du modèle issu de ces travaux dans des analyses de physique est présentée dans la section 7.

2 Événements utilisés

L'objectif des modèles à entraîner est de reconstruire la masse des particules se désintégrant en paire de leptons tau. Il s'agit d'une tâche de régression, il faut donc entraîner les modèles sur le plus de valeurs différentes possible. Dans l'optique d'une utilisation dans les analyses telles que celle présentée dans le chapitre 4, il a été choisi d'utiliser des événements $\mathcal{H} \rightarrow \tau\tau$ où \mathcal{H} est le boson de Higgs du modèle standard h dont la masse est modifiée, à l'instar de ce qu'ont fait BÄRTSCHI & coll. [18]. La cible du modèle est donc la masse $m_{\mathcal{H}}$.

2.1 Génération avec FASTSIM

Nous avons généré nos propres données simulées [28] afin d’obtenir des événements indépendants de ceux utilisés dans les analyses. Dans le contexte de la collaboration CMS, nous avons utilisé FASTSIM [21–24]. Cet outil permet de procéder à l’ensemble de la simulation des événements introduite chapitre 2, de la génération du processus physique à la reconstruction des objets physiques par le détecteur.

Les processus physiques sont générés par PYTHIA 8 [29] avec les réglages CUEP8M1 [30, 31]. L’énergie dans le centre de masse est de 13 TeV. Pour ne pas générer d’événements indésirables, seules les collisions créant un boson de Higgs par fusion de gluons, mode dominant pour le modèle standard, sont autorisées. De plus, le rapport de branchement $\mathcal{BR}(\mathcal{H} \rightarrow \tau\tau)$ est fixé à 1, c’est-à-dire que \mathcal{H} se désintègre forcément en paires de leptons taus.

La masse de \mathcal{H} varie de 50 à 800 GeV par pas de 1 GeV. Il est important d’utiliser l’intervalle le plus étendu possible, il correspond à la gamme utile des modèles obtenus. L’effet de l’étendue de cet intervalle est discuté dans la section 6. Lorsque $m_{\mathcal{H}}$ est supérieure à 800 GeV, les propriétés de \mathcal{H} , basées sur celles de h , ne permettent pas d’obtenir des valeurs de $m_{\mathcal{H}}$ cohérentes avec la méthode de génération utilisée. Nous ne considérerons pas de masse plus haute. Bien qu’il soit possible pour une particule de se désintégrer en deux taus dès que sa masse est plus élevée que $2m_{\tau} = 3,5$ GeV, la sélection des événements présentée dans la section 2.2 rejette plus de 99 % des événements lorsque $m_{\mathcal{H}} < 50$ GeV. Nous ne considérerons pas de masse plus basse. L’efficacité des sélections appliquées est représentée sur la figure X.1. S’il est possible d’appliquer des poids aux événements afin d’équilibrer l’entraînement sur l’ensemble des valeurs de la cible, plus d’événements sont générés à basse masse afin d’obtenir des topologies d’événements variées malgré la faible efficacité de sélection. Ainsi, la quantité d’événements générés pour chaque valeur de $m_{\mathcal{H}}$ est de :

- 60 000 pour $50 \leq m_{\mathcal{H}} < 300$;
- 20 000 pour $300 \leq m_{\mathcal{H}} < 500$;
- 10 000 pour $500 \leq m_{\mathcal{H}} \leq 800$.

L’empilement est modélisé par superposition du signal $\mathcal{H} \rightarrow \tau\tau$ à des événements dits de « biais minimum » [29]. Il s’agit d’événements pouvant contenir des interactions dures, mais n’activant pas de chemin de déclenchement. La quantité d’empilement ajoutée à l’événement $\mathcal{H} \rightarrow \tau\tau$ suit le profil de l’année 2017.

2.2 Sélection des événements

2.2.1 Canaux $\tau_h\tau_h$, $\mu\tau_h$, $e\tau_h$ et $e\mu$

La sélection des événements se fait comme exposé dans le chapitre 4 pour l’année 2017 et les canaux $\tau_h\tau_h$, $\mu\tau_h$, $e\tau_h$ et $e\mu$ y étant exploités, à l’exception des coupures servant à séparer la région de signal des régions de contrôle et de détermination, sur $m_T^{(\mu)}$ dans le canal $\mu\tau_h$, $m_T^{(e)}$ dans le canal $e\tau_h$, D_{ζ} dans le canal $e\mu$. La construction du *dilepton* est inchangée. La correspondance des objets du *dilepton* avec ceux ayant activé le chemin de déclenchement n’est pas vérifiée. Ce choix permet d’obtenir un modèle dont les prédictions auront non seulement un sens dans les régions de contrôle et de détermination, mais aussi plus facilement dans le contexte d’autres analyses dans lesquelles les sélections peuvent différer.

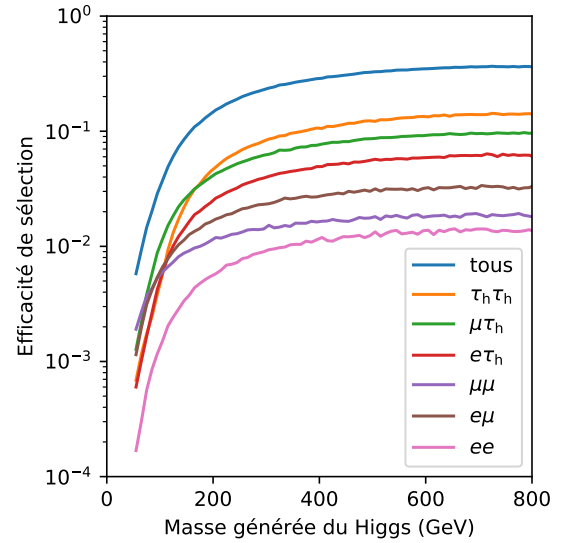


Figure X.1 – Efficacité de sélection des événements pour $m_{\mathcal{H}} \in [50, 800]$ GeV dans les différents canaux et pour tous les canaux.

En plus des canaux listés ci-dessus, nous avons également sélectionné des événements des canaux $\mu\mu$ et ee , selon les procédures présentées ci-après.

2.2.2 Canal $\mu\mu$

Sélection des muons Tout muon respectant les critères listés ci-après est retenu pour jouer le rôle de L_1 ou L_2 dans le *dilepton* :

- $p_T^\mu > 10 \text{ GeV}$;
- $|\eta^\mu| < 2.4$;
- paramètres d'impact $d_z < 0,2 \text{ cm}$ et $d_{xy} < 0,045 \text{ cm}$;
- $I^\mu < 0,15 p_T^\mu$;
- passer le point de fonctionnement *medium* du *muonID*.

Sélection du *dilepton* L'événement est retenu à condition qu'au moins une paire $L_1 L_2 = \mu\mu$ puisse être construite avec L_1 et L_2 de charges électriques opposées. Il est de plus requis que L_1 et L_2 soient séparés dans le plan (η, ϕ) tel que $\Delta R > 0,3$. Si plus d'une paire possible existe dans l'événement, une seule est retenue selon la logique exposée dans le chapitre 4.

Vétos de leptons supplémentaires Les vetos de leptons supplémentaires doivent être respectés, c'est-à-dire que l'événement ne contient pas :

- de second muon tel que $p_T^\mu > 10 \text{ GeV}$, $|\eta^\mu| < 2,4$, passant le point de fonctionnement *medium* du *muonID* et d'isolation $I^\mu < 0,3 p_T^\mu$;
- de second électron tel que $p_T^e > 10 \text{ GeV}$, $|\eta^e| < 2,5$, passant le point de fonctionnement à 90 % d'efficacité de l'*electron ID MVA* et d'isolation $I^e < 0,3 p_T^e$, l'électron devant passer le veto d'électron de conversion et présenter moins de deux points de passage manquants dans le trajectographe.

2.2.3 Canal ee

Sélection des électrons Tout électron respectant les critères listés ci-après est retenu pour jouer le rôle de L_1 ou L_2 dans le *dilepton* :

- $p_T^e > 20 \text{ GeV}$;
- $|\eta^e| < 2.4$;
- paramètres d'impact $d_z < 0,2 \text{ cm}$ et $d_{xy} < 0,045 \text{ cm}$;
- $I^e < 0,1 p_T^e$;
- passer le point de fonctionnement à 90 % d'efficacité de l'*electron ID MVA*.

Sélection du *dilepton* L'événement est retenu à condition qu'au moins une paire $L_1 L_2 = ee$ puisse être construite avec L_1 et L_2 de charges électriques opposées. Il est de plus requis que L_1 et L_2 soient séparés dans le plan (η, ϕ) tel que $\Delta R > 0,5$. Si plus d'une paire possible existe dans l'événement, une seule est retenue selon la logique exposée dans le chapitre 4.

Vétos de leptons supplémentaires Les vetos de leptons supplémentaires doivent être respectés, c'est-à-dire que l'événement ne contient pas :

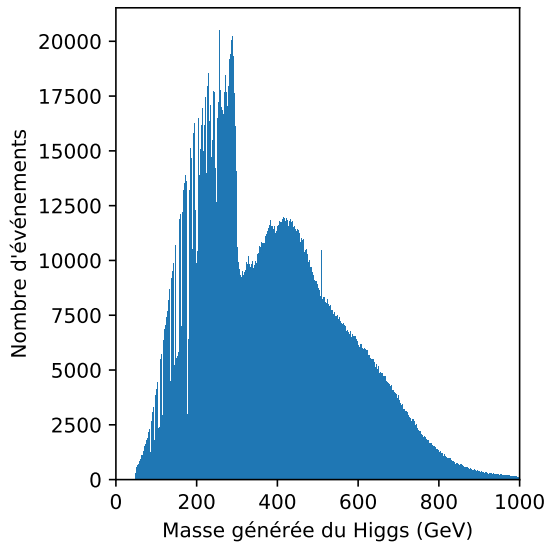
- de second muon tel que $p_T^\mu > 10 \text{ GeV}$, $|\eta^\mu| < 2,4$, passant le point de fonctionnement *medium* du *muonID* et d'isolation $I^\mu < 0,3 p_T^\mu$;
- de second électron tel que $p_T^e > 10 \text{ GeV}$, $|\eta^e| < 2,5$, passant le point de fonctionnement à 90 % d'efficacité de l'*electron ID MVA* et d'isolation $I^e < 0,3 p_T^e$, l'électron devant passer le veto d'électron de conversion et présenter moins de deux points de passage manquants dans le trajectographe.

2.3 Événements obtenus et pondération

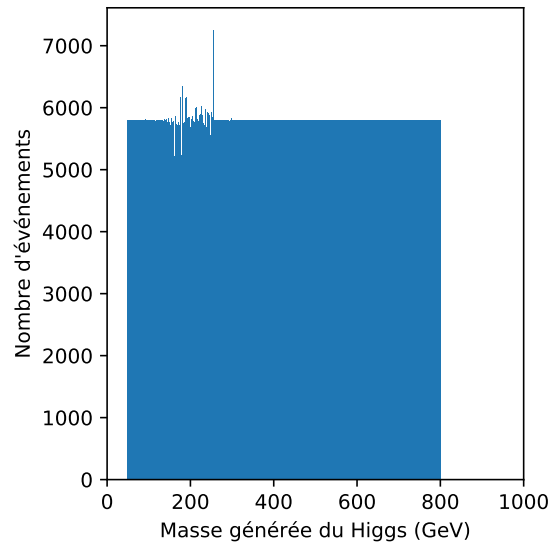
Plus de 22 millions d'événements ont été générés. Environ 3 millions sont sélectionnés selon les critères présentés précédemment. La distribution de $m_{\mathcal{H}}$ dans ces événements sélectionnés est représentée sur la figure X.2a. Quelques événements présentent des valeurs de $m_{\mathcal{H}}$ au-delà de 800 GeV, cet effet est dû à la largeur de cette particule, représentée sur la figure X.3 en fonction de sa masse. La largeur à 800 GeV est ainsi d'environ 300 GeV. Le réglage $m_{\mathcal{H}} = 800$ GeV donne donc des événements contenant un boson dont la masse effective se situe entre 500 et 1100 GeV, d'où la queue de la distribution observée à haute masse sur la figure X.2a. À basse masse en revanche, la largeur est inférieure à 100 MeV, cet effet n'est donc pas présent. La cible du modèle est la masse effective du boson. Les événements retenus dans la suite sont ceux où celle-ci se situe bien entre 50 et 800 GeV, d'où la disparition de la queue à haute masse sur la figure X.2b.

Ces événements sont de plus séparés en trois groupes selon les proportions suivantes :

- 70 % pour l'entraînement. Ce sont ces événements que les modèles pourront exploiter afin d'ap-



(a) Distribution brute sur tous les événements.



(b) Distribution pondérée pour les événements d'entraînement.

Figure X.2 – Distributions de la masse générée de \mathcal{H} .

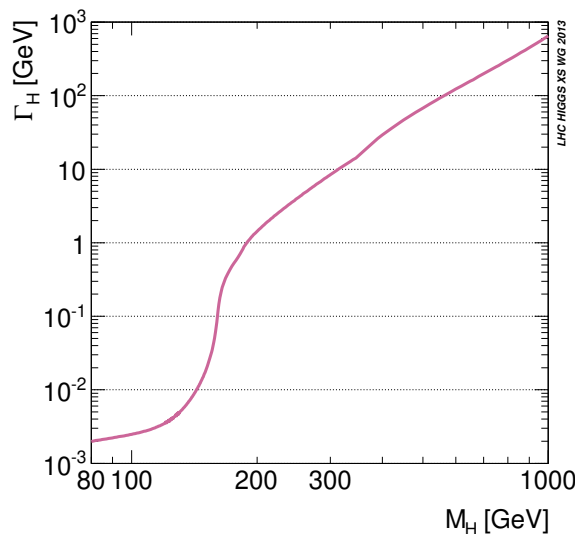


Figure X.3 – Largeur du boson de Higgs du modèle standard [32].

prendre à prédire correctement $m_{\mathcal{H}}$;

- 20 % pour la validation. Ces événements permettent de vérifier qu'il n'y a pas de surentraînement, c'est-à-dire que le modèle ne se spécialise pas vis-à-vis du jeu d'entraînement ;
- 10 % pour les tests. Ces événements ne sont pas utilisés lors des entraînements et permettent donc de tester les modèles sur des données inédites. Sauf contre-indication, les figures sont toutes obtenues avec ce groupe d'événements.

Afin de réaliser un entraînement équitable entre les différentes valeurs de $m_{\mathcal{H}}$, un poids est associé à chaque événement de manière à ce que la distribution pondérée de $m_{\mathcal{H}}$ soit plate. La distribution pondérée de $m_{\mathcal{H}}$ sur les événements utilisés pour l'entraînement des modèles est représentée sur la figure X.2b.

2.4 Cible et variables d'entrée des modèles

La cible des modèles est la masse de la particule générée \mathcal{H} se désintégrant en paire de leptons tau. Un tel événement est illustré sur la figure X.4. Les variables d'entrée doivent être des observables accessibles expérimentalement, c'est-à-dire issues de la reconstruction des événements présentée dans le chapitre 2.

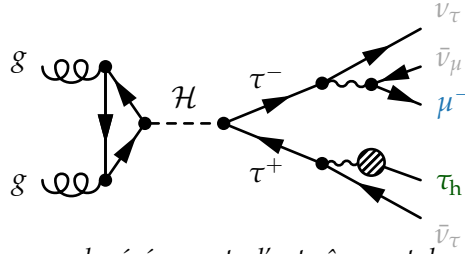


Figure X.4 – Diagramme de Feynman des événements d'entraînement des modèles dans le cas du canal $\mu\tau_h$.

Les variables considérées sont :

- les impulsions de L_1 et L_2 les produits de désintégration visibles des taus c'est-à-dire le muon et le τ_h dans l'exemple de la figure X.4 :
 $p_T^{L_1}, \eta^{L_1}, \phi^{L_1}, p_T^{L_2}, \eta^{L_2}, \phi^{L_2}$;
- l'énergie transverse manquante pour rendre compte de la présence des neutrinos :
 $E_T^{\text{miss}}, \phi^{E_T^{\text{miss}}}$;
- la matrice M de covariance de E_T^{miss} , rendant compte de l'incertitude sur la mesure de E_T^{miss} :
 M_{xx}, M_{xy}, M_{yy} ;
- le nombre attendu de neutrinos lié à l'état final identifié N_v^{reco} , déterminé à partir du canal obtenu par la sélection des événements, c'est-à-dire sans utilisation des informations générées ;
- les masses transverses $m_T(L_1, E_T^{\text{miss}}), m_T(L_2, E_T^{\text{miss}})$ et $m_T(L_1, L_2)$ définies par

$$m_T(A, B) = \sqrt{2 p_T^A p_T^B (1 - \cos(\phi^A - \phi^B))} ; \quad (\text{X.1})$$

- la masse transverse totale m_T^{tot} définie par

$$m_T^{\text{tot}} = \sqrt{m_T^2(L_1, E_T^{\text{miss}}) + m_T^2(L_2, E_T^{\text{miss}}) + m_T^2(L_1, L_2)} ; \quad (\text{X.2})$$

- les impulsions des deux jets principaux (de plus haut p_T) présents dans l'événement :
 $p_T^{\text{jet } 1}, \eta^{\text{jet } 1}, \phi^{\text{jet } 1}, p_T^{\text{jet } 2}, \eta^{\text{jet } 2}, \phi^{\text{jet } 2}$;
- l'Activité Hadronique Additionnelle (AHA), définie par une somme des impulsions des jets autres que les deux principaux :
 $p_T^{\text{AHA}}, \eta^{\text{AHA}}, \phi^{\text{AHA}}$ avec

$$\vec{p}^{\text{AHA}} = \sum_{\text{jet } i, i > 2} \vec{p}^{\text{jet } i} ; \quad (\text{X.3})$$

- la quantité de jets utilisés pour déterminer $\vec{p}^{\text{AHA}}, N_{\text{jets}}^{\text{AHA}}$;
- le nombre de vertex principaux d'empilement, N_{PU} .

Des modèles sont entraînés sur l'ensemble de ces 27 variables ainsi que sur des sous-ensembles de cette liste.

3 Arbres de décision améliorés

La librairie XGBOOST [33] (*eXtreme Gradient Boosting*) permet de construire des arbres de décision améliorés. De nombreuses compétitions Kaggle [34] ont été remportées grâce eux. Ils présentent l'avantage d'être généralement plus à entraîner que des réseaux de neurones présentés section 4, et peuvent fournir des prédictions même si une des entrées est manquante, ce qui n'est pas le cas des réseaux de neurones.

3.1 Arbres de décision

Les arbres de décision sont une succession de questions, dont les réponses mènent à un résultat final, comme illustré sur la figure X.5. Chaque réponse à une question crée une « branche » (en bleu), les réponses finales sont les « feuilles » (en rouge et vert).

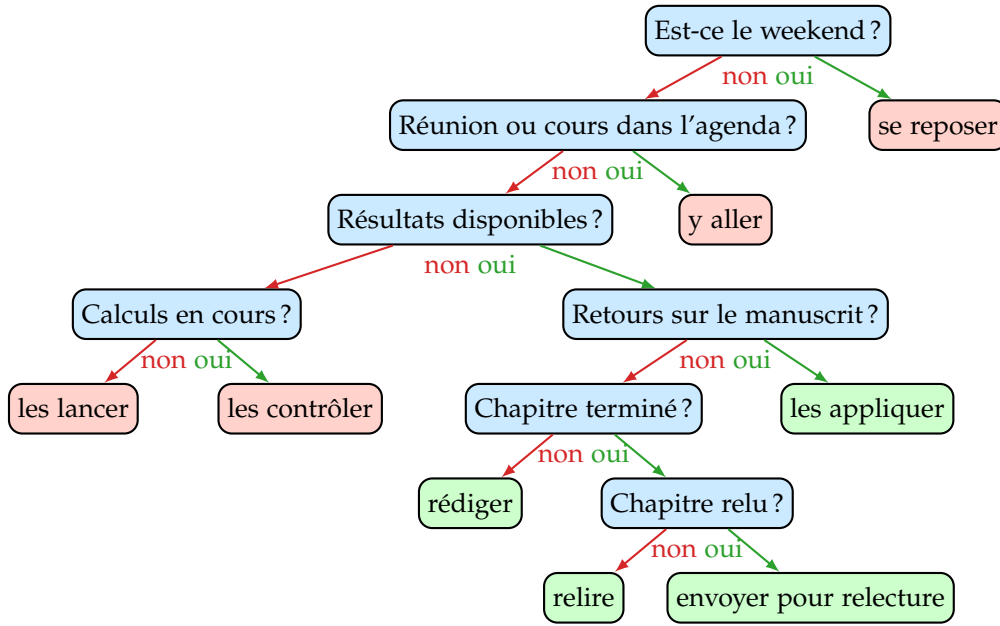


Figure X.5 – Exemple d'un arbre de décision utilisable par un doctorant.

De tels arbres peuvent être utilisés avec des variables numériques. Dans ce cas, les questions consistent en une condition sur l'une des variables, par exemple $p_T^\mu > 50 \text{ GeV}$. Le choix de la variable (p_T^μ) et de la coupure correspondante (50 GeV) à utiliser pour former deux nouvelles branches b_1 (condition fausse) et b_2 (condition vraie) se base sur la similarité S , définie comme

$$S = \frac{1}{Nn} \left(\sum_{i=1}^N \sum_{j=1}^n x_{j,i} \right)^2 \quad (\text{X.4})$$

où N est la taille du jeu de données d'entraînement (quantité d'échantillons), n le nombre de variables d'entrée différentes et $x_{j,i}$ la valeur de la variable x_j dans le i^{e} échantillon.

Le gain G obtenu par la création de deux nouvelles branches b_1 et b_2 s'exprime

$$G = S_{b_1} + S_{b_2} - S_{b_1+b_2} \quad (\text{X.5})$$

avec $S_{b_1+b_2}$ la similarité du jeu de donnée non séparé, S_{b_1} (S_{b_2}) la similarité du jeu de donnée se retrouvant dans la branche b_1 (b_2). La condition retenue pour former les deux branches est celle présentant le gain le plus élevé. Ce processus est alors itéré sur chacune des nouvelles branches, jusqu'à ce que :

- le gain soit inférieur à γ ;
- la profondeur de l'arbre (nombre de conditions successives) est supérieure à $N_{\text{max}}^{\text{prof.}}$.
- la quantité d'échantillons dans une branche est inférieure à $N_{\text{min}}^{\text{échant.}}$.

Les paramètres γ , $N_{\max}^{\text{prof.}}$ et $N_{\min}^{\text{échant.}}$, fixés par l'utilisateur, sont nommés « hyper-paramètres ». Ils ne doivent pas être confondus avec les paramètres propres à l'arbre, déterminés lors de la construction des branches.

3.2 Gradient Boosting

La technique du *Gradient Boosting* consiste en l'utilisation de modèles simples, ici des arbres de décision, pour obtenir un modèle global plus robuste. La construction se fait de manière itérative.

À chaque étape $k \geq 1$, un arbre de décision M_k nommé estimateur est construit avec pour objectif de prédire

$$y_{\text{vraie},i} - F_{k-1}(\vec{x}_i) \quad (\text{X.6})$$

pour une entrée \vec{x}_i , avec $y_{\text{vraie},i}$ la valeur que doit prédire le modèle global pour l'entrée \vec{x}_i et F_{k-1} la fonction du modèle global issu de l'étape $k-1$, F_0 étant égale à M_0 , l'arbre de décision obtenu sans *Gradient Boosting*. Le modèle M_k corrige donc l'écart résiduel des prédictions $\{y_{\text{préd},i}\}$ du modèle global à $\{y_{\text{vraie},i}\}$. Les prédictions F_k du modèle global s'expriment alors

$$y_{\text{préd},i} = F_k(\vec{x}_i) = F_{k-1}(\vec{x}_i) + \eta M_k(\vec{x}_i) \quad (\text{X.7})$$

avec η le taux d'apprentissage, inférieur à 1, permettant de corriger progressivement l'écart résiduel. L'itération s'arrête lorsque le nombre maximal d'estimateurs $N_{\max}^{\text{estim.}}$ est atteint. Les grandeurs η et $N_{\max}^{\text{estim.}}$ sont également des hyper-paramètres.

3.3 Fonction de coût et Gradient Descent

Une fonction de coût (*loss function*) compare les prédictions d'un modèle aux valeurs vraies. Elle doit être différentiable et est définie de manière à être minimale lorsque les prédictions sont égales aux valeurs vraies, c'est-à-dire lorsque le modèle est parfait. Les fonctions de coûts les plus répandues sont :

MSE *Mean Squared Error* ou erreur quadratique moyenne,

$$L_{\text{MSE}}(\{y_{\text{vraie},i}\}, \{y_{\text{préd},i}\}) = \frac{1}{2N} \sum_{i=1}^N (y_{\text{préd},i} - y_{\text{vraie},i})^2 ; \quad (\text{X.8})$$

MAE *Mean Absolute Error* ou erreur absolue moyenne,

$$L_{\text{MAE}}(\{y_{\text{vraie},i}\}, \{y_{\text{préd},i}\}) = \frac{1}{N} \sum_{i=1}^N |y_{\text{préd},i} - y_{\text{vraie},i}| ; \quad (\text{X.9})$$

MAPE *Mean Absolute Percentile Error* ou erreur absolue relative moyenne,

$$L_{\text{MAPE}}(\{y_{\text{vraie},i}\}, \{y_{\text{préd},i}\}) = \frac{100}{N} \sum_{i=1}^N \left| \frac{y_{\text{préd},i} - y_{\text{vraie},i}}{y_{\text{vraie},i}} \right|. \quad (\text{X.10})$$

L'objectif du modèle M_k défini dans la section précédente est de prédire, pour \vec{x}_i ,

$$y_{\text{vraie},i} - F_{k-1}(\vec{x}_i) = - \frac{\partial L_{\text{MSE}}(y_{\text{vraie},i}, F_{k-1}(\vec{x}_i))}{\partial F_{k-1}(\vec{x}_i)}. \quad (\text{X.11})$$

Il est ainsi possible de généraliser le *Gradient Boosting* en considérant que l'objectif de M_k est de prédire

$$- \frac{\partial L(y_{\text{vraie},i}, F_{k-1}(\vec{x}_i))}{\partial F_{k-1}(\vec{x}_i)} = - \vec{\nabla}_{F_{k-1}(\vec{x}_i)} (L(y_{\text{vraie},i}, F_{k-1}(\vec{x}_i))) \quad (\text{X.12})$$

avec L une fonction de coût quelconque. Il s'agit du *Gradient Descent*, où l'objectif est de minimiser L . La fonction de coût est un hyper-paramètre du modèle.

3.4 Sous-entraînement et surentraînement

La construction d'un modèle, aussi appelé « entraînement », est un processus itératif visant à minimiser la fonction de coût. Dans le cas des arbres de décision améliorés créés avec XGBOOST, l'entraînement cesse lorsque le nombre d'estimateur maximal est atteint, lorsqu'il est renseigné. Dans tous les cas, il est légitime de se demander si le modèle obtenu à la fin de l'entraînement est optimal.

Il faut que l'entraînement soit suffisamment long pour que le modèle propose les prédictions les plus précises possible. Autrement dit, il faut que le modèle ait le temps d'apprendre. S'il ne l'a pas, les prédictions ne sont pas aussi précises qu'elles pourraient l'être, c'est le sous-entraînement. La valeur de la fonction de coût appliquée au jeu de données d'entraînement diminuant lors de l'apprentissage, un critère pourrait être de l'utiliser afin de déterminer si le modèle apprend encore ou non. Arrivé à un plateau, le modèle ne s'améliore plus et l'entraînement s'arrête.

Cette approche masquerait toutefois une spécialisation du modèle. En effet, un modèle peut apprendre à prédire parfaitement $\{y_{\text{vraie},i}\}$ sur le jeu de données d'entraînement, ce qui correspond à une fonction de coût nulle, mais être moins bon qu'un modèle entraîné moins longtemps lorsqu'il est utilisé sur d'autres données. C'est le surentraînement. Cet effet peut être évité en utilisant un jeu de données dit de « validation », non utilisé pour régler les paramètres du modèle.

L'intérêt du jeu de validation est illustré sur la figure X.6. Un modèle sous-entraîné ou dont l'entraînement est optimal présente des erreurs similaires dans les deux jeux de données. Dans le cas d'un surentraînement, les erreurs continuent à diminuer sur le jeu d'entraînement, mais pas sur le jeu de validation. Une fonction d'évaluation E , éventuellement égale à la fonction de coût L , permet de quantifier ces erreurs et de mettre fin à l'entraînement avant de surentraîner le modèle. Il s'agit de l'arrêt prématuré (*early stopping*).

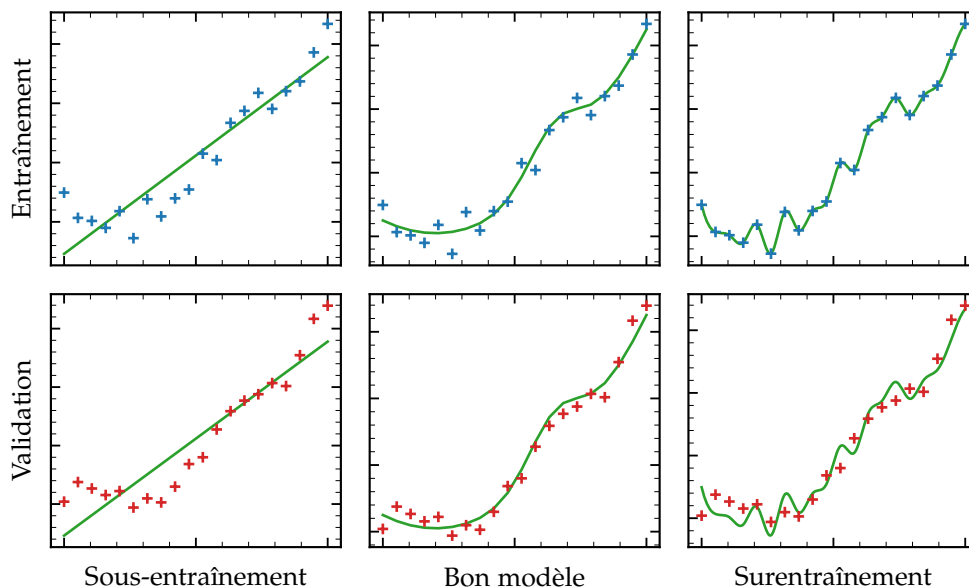


Figure X.6 – Illustrations du sous-entraînement et du surentraînement. Un même modèle est peu (gauche), suffisamment (milieu) ou trop entraîné (droite). Ses prédictions (ordonnées) en fonction de l'entrée (abscisses) sont tracées en vert. Le jeu de données d'entraînement (de validation) est représenté par des croix bleues (rouges) sur la ligne du haut (bas).

Dans le cas des arbres de décision améliorés, une itération de l'entraînement consiste en l'ajout d'un estimateur, comme exposé dans la section 3.2. Un arrêt prématuré est réalisé lorsque l'erreur quadratique moyenne ne diminue pas sur le jeu de validation pendant 5 itérations.

4 Réseaux de neurones profonds

Les réseaux de neurones (NN, *Neural Networks*) sont un autre type de modèle permettant d'approximer la fonction reliant les entrées $\{\vec{x}_i\}$ aux cibles $\{y_{\text{vraie},i}\}$ [35]. La section 4.1 introduit le

concept de neurone dans le cadre du ML. Puis, les réseaux de neurones sont présentés dans la section 4.2. L'entraînement de ce type de modèle est discuté section 4.3.

4.1 Neurones

4.1.1 Principe

Un neurone est une entité ayant un certain nombre d'entrées $x_j, j \in \{1, \dots, n\}$, auxquelles sont associées des poids w_j , un biais b et une fonction f dite d'« activation », discutée section 4.1.2. Les poids w_j et le biais b sont les paramètres du neurone, la fonction d'activation est un hyper-paramètre. La sortie s du neurone s'exprime

$$s = f \left(\sum_{j=1}^n w_j x_j + b \right). \quad (\text{X.13})$$

Le fonctionnement d'un neurone est résumé sur la figure X.7.

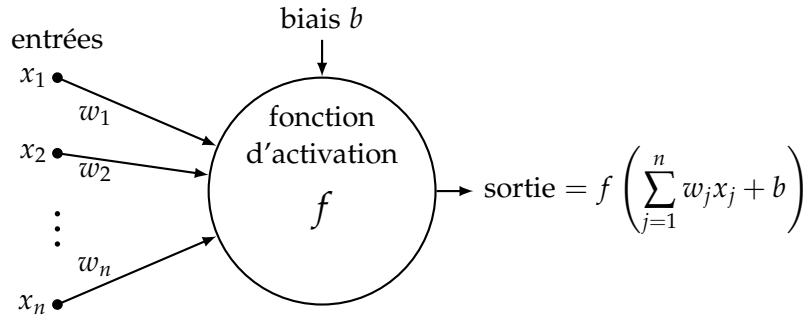


Figure X.7 – Structure d'un neurone. Une fonction f dite d'« activation » est appliquée à la somme des entrées x_j pondérées par les poids w_j et du biais b afin d'obtenir la valeur de sortie.

4.1.2 Fonctions d'activation

En principe, toute fonction définie sur l'ensemble d'existence des entrées x_j peut être utilisée comme fonction d'activation. Celles-ci étant généralement à valeurs réelles et unidimensionnelles, les fonctions sont définies sur \mathbb{R} . Les plus utilisées sont :

tangente hyperbolique notée \tanh , définie par

$$\tanh : x \mapsto \frac{e^x - e^{-x}}{e^x + e^{-x}} ; \quad (\text{X.14})$$

sigmoïde notée sig , définie par

$$\text{sig} : x \mapsto \frac{e^x}{1 + e^x} = \frac{1}{1 + e^{-x}} ; \quad (\text{X.15})$$

Softsign notée Ssg , définie par

$$\text{Ssg} : x \mapsto \frac{x}{1 + |x|} ; \quad (\text{X.16})$$

ReLU (*Rectified Linear Unit*), définie par

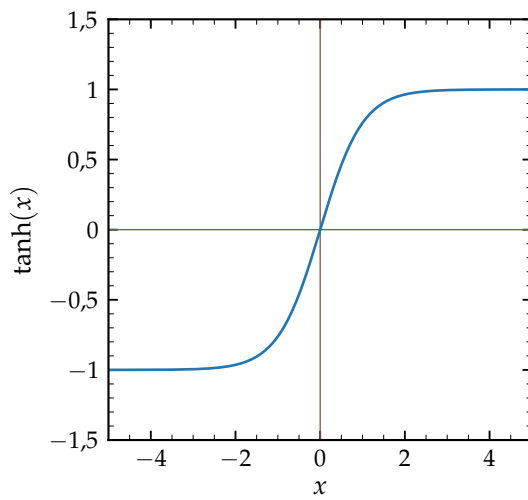
$$\text{ReLU} : x \mapsto \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} ; \quad (\text{X.17})$$

Softplus notée Spl , définie par

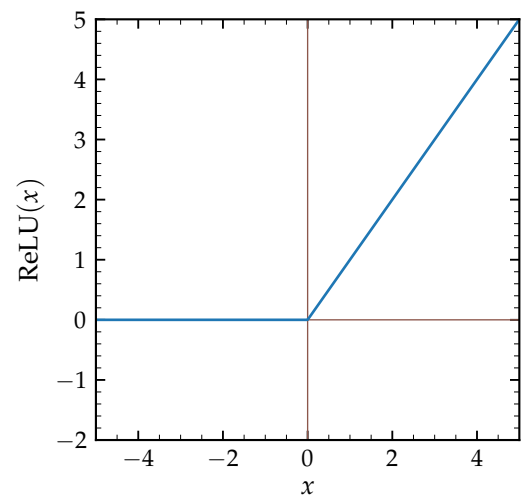
$$\text{Spl} : x \mapsto \ln(1 + e^x) ; \quad (\text{X.18})$$

ELU (*Exponential Linear Unit*), définie par

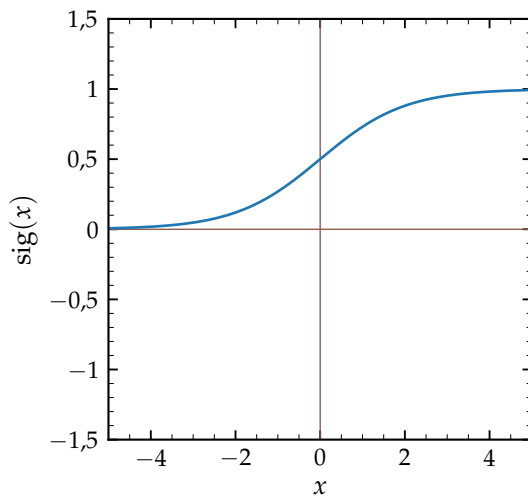
$$\text{ELU} : x \mapsto \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases}, \quad \alpha = 1 ; \quad (\text{X.19})$$



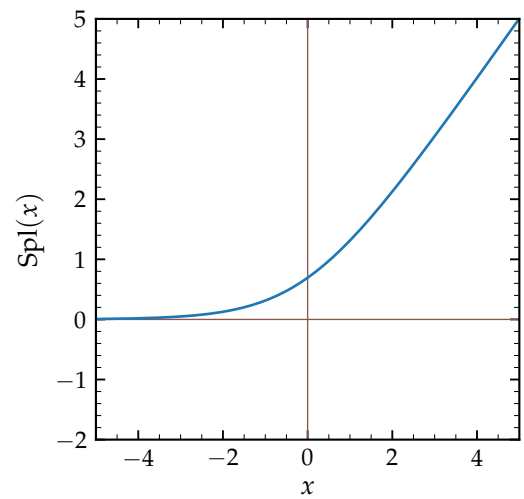
(a) Tangente hyperbolique.



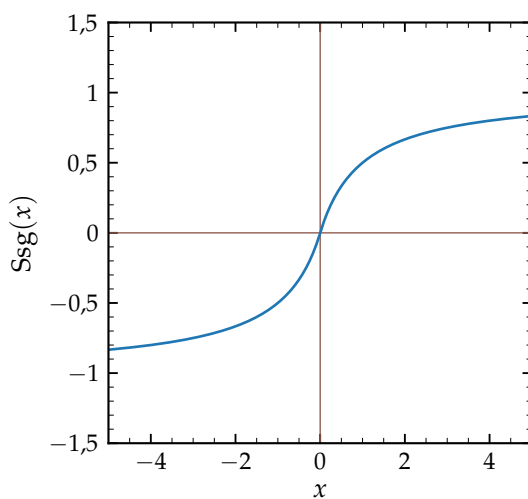
(b) ReLU.



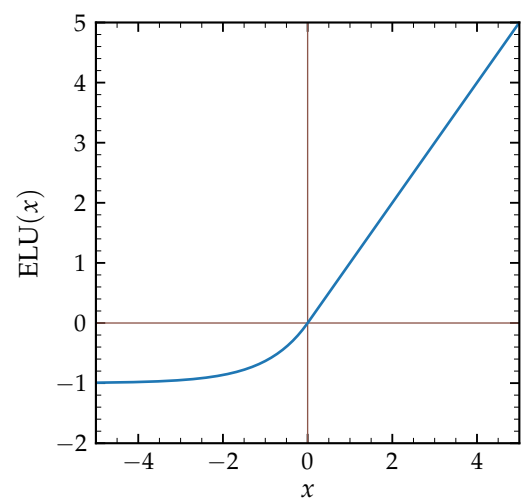
(c) Sigmoïde.



(d) Softplus.



(e) Softsign.



(f) ELU.

Figure X.8 – Exemples de fonctions d'activation. À gauche, des fonctions à valeurs bornées, généralement utilisées en classification. À droite, des fonctions à valeurs non bornées, utilisables pour des tâches de régression.

SELU (*Scaled Exponential Linear Unit*), définie par

$$\text{SELU} : x \mapsto \lambda \times \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases}, \quad \alpha \simeq 1,67, \quad \lambda \simeq 1,05; \quad (\text{X.20})$$

ou encore la fonction linéaire identité $\mathbb{1} : x \mapsto x$. Certaines d'entre elles sont représentées sur la figure X.8.

4.2 Réseaux de neurones

Un NN est obtenu par l'interconnexion de plusieurs neurones entre eux. Ces connexions peuvent se faire selon diverses architectures [35]. Nous utilisons ici, comme dans les travaux de BÄRTSCHI & coll. [18], une architecture normale profonde à propagation avant complètement connectée (*normal deep feedforward fully-connected*), c'est-à-dire avec :

- des neurones répartis en couches (normale);
- plusieurs couches « cachées », situées entre celles d'entrée et de sortie (profonde);
- les entrées des neurones de la couche k :
 - prises parmi les sorties de ceux de la couche $k - 1$ (à propagation avant),
 - étant toutes les sorties de ceux de la couche $k - 1$ (complètement connectée).

Le nombre de neurones par couche cachée N_N est pris constant. Le nombre de couches cachées (*hidden layers*) est noté N_L . Le NN ayant une structure profonde, il s'agit d'un DNN (*Deep Neural Network*).

La tâche du réseau est une régression sur une seule grandeur, $m_{\mathcal{H}}$, à partir de n variables d'entrée x_j , $j \in \{1, \dots, n\}$. La couche de sortie est donc composée d'un seul neurone dont la fonction d'activation est l'identité. La couche d'entrée comporte n neurones, chacun ayant une entrée correspondant à une variable avec un poids de 1, un biais de zéro et identité comme fonction d'activation. Il s'agit donc d'une couche d'adaptation entre le nombre d'entrées n_{in} et le nombre de neurones dans la couche suivante N_N . La fonction d'activation des neurones des couches cachées est identique, plusieurs possibilités sont essayées dans la section 5. La structure obtenue est représentée sur la figure X.9.

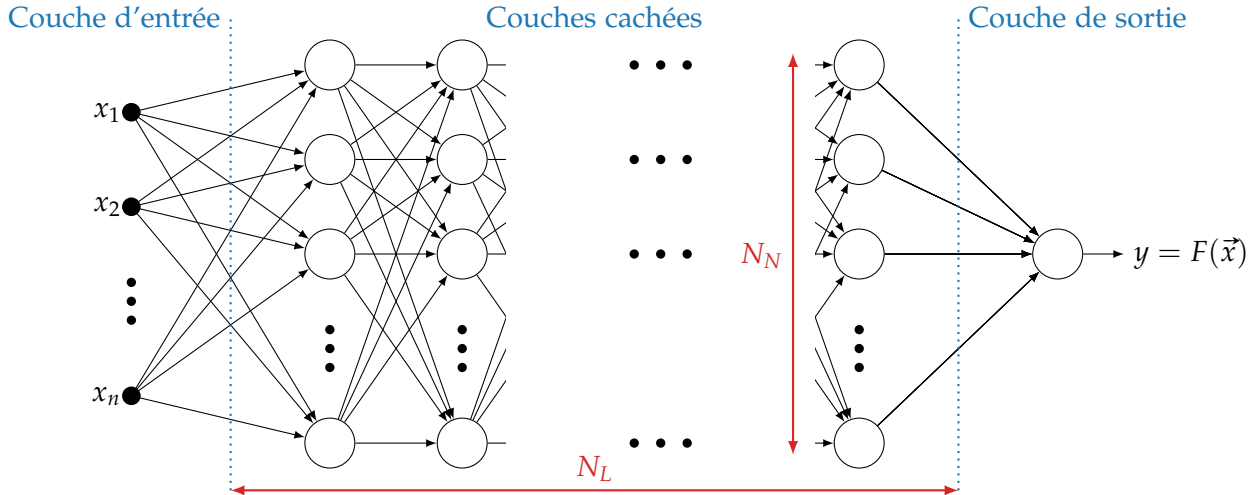


Figure X.9 – Structure normale profonde à propagation avant complètement connectée d'un réseau de neurones. Une couche d'entrée comporte autant de neurones que de variables x_i . La couche de sortie en comporte autant que de valeurs à donner, c'est-à-dire une. Les fonctions d'activation de ces deux couches sont linéaires. Entre elles se trouvent N_L couches cachées, chacune contenant N_N neurones. Diverses fonctions d'activation peuvent être utilisées dans les couches cachées.

4.3 Entraînement

L'entraînement d'un NN est le réglage des paramètres des neurones du réseau situés sur les couches cachées et la couche de sortie. Il s'agit des poids w_i et du biais b . Pour un DNN avec $n_{\text{in}} = 27$

variables d'entrée, $N_L = 3$ couches cachées de $N_N = 1000$ neurones, le nombre de paramètres est ainsi de

$$N_{\text{params.}} = \underbrace{N_N \times (n_{\text{in}} + 1)}_{\text{couche cachée 1}} + \underbrace{(N_L - 1) \times N_N \times (N_N + 1)}_{\text{autres couches cachées}} + \underbrace{N_N + 1}_{\text{couche de sortie}} \\ = 28\,000 + 2 \times 1\,001\,000 + 1001 = 2\,031\,001, \quad (\text{X.21})$$

soit près de deux millions. Les termes « +1 » correspondent aux biais b à ajouter au nombre d'entrées des neurones.

4.3.1 Initiation des paramètres

Les biais b_i sont initialement fixés à 0, les poids w_i à une valeur constante donnée ou aléatoirement selon une loi de probabilité. Le mode d'initiation est un hyper-paramètre du modèle. Lors de ces travaux, nous avons testé les lois normale et uniforme. Dans le cas des DNNs, ces modes d'initiation peuvent être améliorés par la méthode de GLOROT & BENGIO [36] afin de faciliter l'entraînement. Il s'agit alors des lois « Glorot uniforme » et « Glorot normale », également testées.

4.3.2 Fonction de coût et optimisation des paramètres

Les modifications apportées aux paramètres ont pour objectif l'amélioration des prédictions du modèle. La qualité de ces prédictions est quantifiée par une fonction de coût L à minimiser, comme exposé section 3.3. Il s'agit donc de trouver le minimum de L dans l'espace à D dimensions formé par les $D = N_{\text{params.}}$ paramètres à régler. Cela peut être fait de manière itérative par *Gradient Descent*.

Le *Gradient Descent* [37] détermine le gradient de L , $\vec{\nabla}(L)$, autour de la « position » du modèle dans l'espace à D dimensions. Chaque paramètre p (w_i et b de chaque neurone) est alors modifié selon

$$p \rightarrow p - \eta \vec{\nabla}(L) \cdot \vec{e}_p = p - \eta \frac{\partial L}{\partial p} \quad (\text{X.22})$$

avec η le taux d'apprentissage, c'est-à-dire que la position du modèle est déplacée en suivant la pente du gradient vers un point plus bas. Le taux d'apprentissage est généralement pris entre 0 et 1.

La modification des paramètres du NN pourrait être réalisée ainsi pour chaque événement du jeu de données d'entraînement. Or, la nature des données à analyser peut mener à une stagnation, si deux événements donnent modifications opposées. Afin d'éviter ce phénomène, la mise à jour des paramètres se fait à partir de « mini-lots », introduits section 4.3.3. Des algorithmes d'optimisation, adaptés aux mini-lots et dérivés du *Gradient Descent*, sont présentés section 4.3.4.

4.3.3 Mini-lots et époques

Un mini-lot est un sous-ensemble du jeu de données. L'entraînement se base alors sur la moyenne du gradient de la fonction de coût sur le mini-lot, au lieu de la valeur de ce gradient pour chaque événement.

Une « époque » de l'entraînement correspond à une utilisation de tous les mini-lots, c'est-à-dire de tous les événements du jeu de données, pour modifier les paramètres du NN. Le nombre maximal d'époques autorisé est de 500, avec un arrêt prématuré au bout de 20 époques sans diminution de l'erreur absolue moyenne sur les données de validation.

Pour ne pas biaiser l'entraînement à cause de l'ordre du jeu de données, il est mélangé aléatoirement à chaque nouvelle époque. La composition des mini-lots est donc également aléatoire. La taille des mini-lots est fixée à $2^{11} = 2048$ événements. Une taille de la forme 2^n permet d'optimiser l'utilisation des GPUs (*Graphics Processing Unit*) sur lesquels l'entraînement se fait [35]. Les points de masse générés étant les entiers entre 50 et 800 GeV, soit 750 points de masse, 2048 événements pris au hasard est un compromis entre un petit mini-lot et une bonne probabilité de couvrir une large gamme de masse au sein d'un mini-lot.

4.3.4 Algorithmes d'optimisation

Plusieurs algorithmes d'optimisation existent [35], présentés ci-après. Le premier, SGD, est l'adaptation directe du *Gradient Descent* aux mini-lots. Cependant, fixer le taux d'apprentissage η est ardu et les modèles y sont très sensibles [35]. C'est pourquoi d'autres algorithmes d'optimisation ont été développés.

Stochastic Gradient Descent (SGD) [38] L'algorithme SGD applique le principe du *Gradient Descent* en estimant le gradient de la fonction de coût par une moyenne sur le mini-lot. Cette moyenne introduit un bruit dû à la composition aléatoire des mini-lots qui reste non nul même une fois le minimum de L atteint. Pour palier cet effet, le taux d'apprentissage η peut être diminué à chaque époque [35]. La condition sur les taux d'apprentissage η_k avec k l'époque afin de s'assurer de la convergence du modèle optimisé par SGD est

$$\sum_{k=1}^{\infty} \eta_k = \infty, \quad \sum_{k=1}^{\infty} \eta_k^2 < \infty. \quad (\text{X.23})$$

La mise à jour des paramètres à la fin d'un mini-lot pendant l'époque k est alors réalisée selon

$$p \rightarrow p - \eta_k \langle \vec{\nabla}(L) \rangle_{\text{mini-lot}} \cdot \vec{e}_p = p - \eta_k \left\langle \frac{\partial L}{\partial p} \right\rangle_{\text{mini-lot}}. \quad (\text{X.24})$$

SGD avec moments [35] Les moments sont une « mémoire » des valeurs du gradient de la fonction de coût des époques précédentes. Ce peut être vu comme une inertie du mouvement du modèle dans l'espace des paramètres, prise en compte à travers une vitesse \vec{v} définie initialement par l'utilisateur et mise à jour à chaque mini-lot selon

$$\vec{v}[t-1] \rightarrow \vec{v}[t] = \alpha \vec{v}[t-1] - \eta_k \langle \vec{\nabla}(L)[t] \rangle_{\text{mini-lot}} \quad (\text{X.25})$$

$$\Rightarrow \vec{v}[t] \cdot \vec{e}_p = v_p[t] = \alpha v_p[t-1] - \eta_k \left\langle \frac{\partial L}{\partial p} [t] \right\rangle_{\text{mini-lot}} \quad (\text{X.26})$$

avec t le numéro de l'itération de l'entraînement et $0 \leq \alpha < 1$ le paramètre des moments. La mise à jour des paramètres lors de l'itération t se fait alors selon

$$p[t-1] \rightarrow p[t] = p[t-1] + v_p[t] = p[t-1] + \alpha v_p[t-1] - \eta_k \left\langle \frac{\partial L}{\partial p} [t] \right\rangle_{\text{mini-lot}}. \quad (\text{X.27})$$

Adaptive Gradient (Adagrad) [39] L'algorithme Adagrad adapte le taux d'apprentissage individuellement pour chaque paramètre p à l'aide d'une variable de mémoire \vec{r} . Elle est initialement définie à $\vec{0}$ et est modifiée à chaque mini-lot selon

$$\vec{r} \cdot \vec{e}_p = r_p \rightarrow r_p + \left\langle \frac{\partial L}{\partial p} \right\rangle_{\text{mini-lot}}^2. \quad (\text{X.28})$$

La mise à jour des paramètres se fait alors selon

$$p \rightarrow p - \eta \frac{1}{\sqrt{r_p} + \delta} \left\langle \frac{\partial L}{\partial p} \right\rangle_{\text{mini-lot}} \quad (\text{X.29})$$

où δ est une variable de régularisation évitant les divisions par zéro. Le taux d'apprentissage effectif pour le paramètre p est ainsi η divisé par la somme quadratique des gradients précédents $\sqrt{r_p}$.

Plus un paramètre modifie la valeur de la fonction de coût, plus sa modification est progressive. Dans l'optique de la recherche d'un minimum, cela revient à descendre une pente lentement et à se mouvoir rapidement dans une direction plane. Cependant, l'accumulation depuis le début de l'entraînement des gradients au carré dans r_p peut mener à une diminution excessive du taux d'apprentissage effectif d'un paramètre.

RMSProp [40] L'algorithme RMSProp consiste en une légère modification de AdaGrad. Une décroissance exponentielle de la mémoire des gradients passés est mise en place en remplaçant (X.28) par

$$r_p \rightarrow \rho r_p + (1 - \rho) \left\langle \frac{\partial L}{\partial p} \right\rangle_{\text{mini-lot}}^2 \quad (\text{X.30})$$

où $0 < \rho < 1$ est le tau de diminution de la mémoire. RMSProp est ainsi une version de AdaGrad dont la mémoire est plus adaptée à la situation locale.

Adaptive Delta (Adadelata) À l'instar de RMSProp, Adadelata est une modification de AdaGrad visant à modifier l'effet de mémoire. La variable r_p est mise à jour par (X.28). Cependant, la valeur précédente de r_p est également utilisée lors de la mise à jour de p . Ainsi, lors de l'itération t ,

$$p[t-1] \rightarrow p[t] = p[t-1] - \frac{\sqrt{r_p[t-1]} + \delta}{\sqrt{r_p[t]} + \delta} \left\langle \frac{\partial L}{\partial p} \right\rangle_{\text{mini-lot}}[t]. \quad (\text{X.31})$$

Il n'y a donc pas besoin de définir un taux d'apprentissage initial avec Adadelata.

Adaptive Moments (Adam) [35, 41] L'algorithme Adam combine la méthode des moments et RMSProp. il adapte donc le taux d'apprentissage pour chaque paramètre à chaque mini-lot. Pour cela sont définis initialement :

- le pas $\epsilon = 0,001$;
- les moments d'ordres 1 et 2, $\vec{v} = \vec{0}$ et $\vec{r} = \vec{0}$;
- les taux de diminution de moments d'ordre 1 et 2, $\rho_1 = 0,9$ et $\rho_2 = 0,999$;
- le paramètre temporel $t = 0$.

Puis, à chaque mini-lot, les moments sont redéfinis selon

$$\vec{v} \cdot \vec{e}_p = v_p \rightarrow \rho_1 v_p + (1 - \rho_1) \left\langle \frac{\partial L}{\partial p} \right\rangle_{\text{mini-lot}}, \quad \vec{r} \cdot \vec{e}_p = r_p \rightarrow \rho_2 r_p + (1 - \rho_2) \left\langle \frac{\partial L}{\partial p} \right\rangle_{\text{mini-lot}}^2. \quad (\text{X.32})$$

Le biais d'initiation des moments est corrigé en appliquant

$$t \rightarrow t + 1, \quad v_p \rightarrow \frac{v_p}{1 - \rho_1^t}, \quad r_p \rightarrow \frac{r_p}{1 - \rho_2^t}. \quad (\text{X.33})$$

Les paramètres du modèle sont alors mis à jour selon

$$p \rightarrow p - \epsilon \frac{v_p}{\sqrt{r_p} + \delta} \quad (\text{X.34})$$

où $\delta = 10^{-8}$ permet de stabiliser les calculs en évitant une division par zéro.

5 Optimisation des hyper-paramètres et choix d'un modèle

Le choix d'un modèle et de ses hyper-paramètres est l'objet de cette section. Deux types de modèle sont étudiés :

- des arbres de décision améliorés, introduits section 3, notés XGB;
- des réseaux de neurones profonds, introduits section 4, notés DNN.

Les hyper-paramètres des XGBs sont :

- la profondeur maximale des arbres $N_{\text{max}}^{\text{prof.}}$;
- la quantité d'échantillons minimale dans une branche $N_{\text{min}}^{\text{échant.}}$;
- le nombre d'arbres $N_{\text{max}}^{\text{estim.}}$;
- le gain minimal γ ;
- le taux d'apprentissage η ;
- la fonction de coût L ;
- la liste des variables d'entrée.

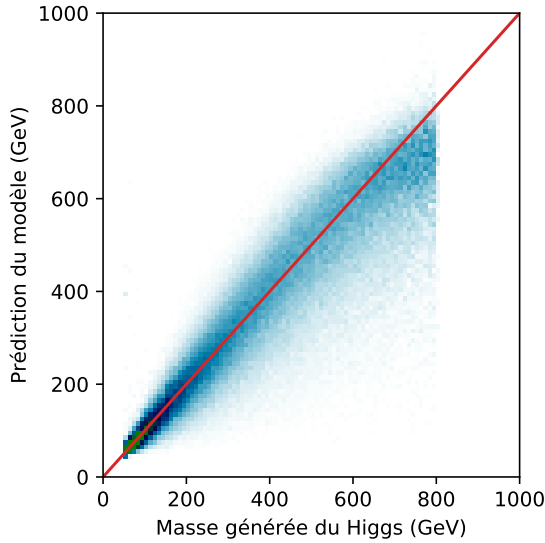
Les hyper-paramètres des DNNs sont :

- le nombre de couches cachées N_L ;
- le nombre de neurones par couche cachée N_N ;
- la fonction d'activation des neurones des couches cachées ;
- l'algorithme d'optimisation ;
- la fonction de coût L ;
- le mode d'initiation des poids ;
- la liste des variables d'entrée.

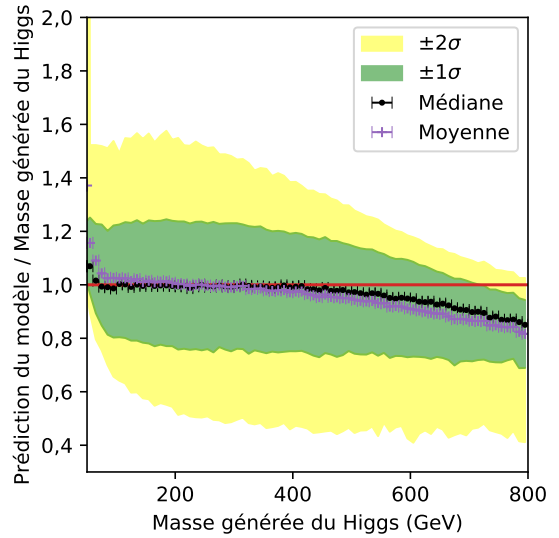
Les modèles entraînés ont pour but de prédire la masse générée du boson de Higgs m_H . Une représentation graphique possible afin de montrer les performances d'un modèle serait de représenter ses prédictions $y_{\text{préd}}$ en fonction de y_{vraie} , dans un histogramme à deux dimensions. La figure X.10a montre un tel histogramme pour un de nos modèles. L'objectif des modèles est alors de se rapprocher autant que possible de la première bissectrice, tracée en rouge. Toutefois la large gamme explorée, de 50 à 800 GeV, rend difficile la visualisation des performances à basse masse. Or, cette région est importante car elle contient les bosons Z et h du modèle standard. La réponse r du modèle, définie comme

$$r = \frac{y_{\text{préd}}}{y_{\text{vraie}}} = \frac{F(\vec{x})}{m_H}, \quad (\text{X.35})$$

permet de ramener l'objectif des modèles à 1 sur toute la gamme de masse. La réponse du même modèle est ainsi représentée sur la figure X.10b. Pour chaque intervalle de 10 GeV sur m_H , la distribution de r est déterminée. La valeur moyenne et la médiane de cette distribution sont données, et ainsi que le premier ($\pm 1\sigma$) et le second quantile ($\pm 2\sigma$). Une évaluation quantifiée des modèles reste nécessaire.



(a) Histogramme à deux dimensions de $y_{\text{préd}}$ en fonction de y_{vraie} .



(b) Réponse du modèle $y_{\text{préd}}/y_{\text{vraie}}$ en fonction de y_{vraie} .

Figure X.10 – Exemples de graphiques rendant compte des performances des modèles.

Il est difficile de définir un seul score quantifiant la qualité d'un modèle. Plusieurs métriques sont considérées afin d'évaluer les modèles :

- les valeurs de L_{MSE} , L_{MAE} , L_{MAPE} ;
- la résolution relative des modèles, ou « largeur à 1σ », notée $\Delta_{1\sigma}$, estimée par la moyenne de la largeur du premier quantile de la distribution de la réponse r des modèles sur des intervalles de 10 GeV sur $y_{\text{vraie}} = m_H$, c'est-à-dire

$$\Delta_{1\sigma} = \left\langle \sigma \left(\frac{y_{\text{préd}}}{y_{\text{vraie}}} \right) \right\rangle_{y_{\text{vraie}} \in [n, n+1] \times 10 \text{ GeV}}. \quad (\text{X.36})$$

Il s'agit donc de la moyenne de la largeur verticale des bandes vertes ($\pm 1\sigma$) sur les graphiques des réponses des modèles comme celui de la figure X.10b.

Pour toutes ces métriques, l'objectif est la plus petite valeur possible. De plus, quatre domaines de masse sont définis :

- basse masse : $m_{\mathcal{H}} < 150 \text{ GeV}$, incluant en particulier les bosons Z et h ;
- masse moyenne : $150 \text{ GeV} \geq m_{\mathcal{H}} < 500 \text{ GeV}$;
- haute masse : $m_{\mathcal{H}} \geq 500 \text{ GeV}$;
- toute masse : aucune restriction sur $m_{\mathcal{H}}$.

Ils permettent de comparer les performances des modèles sur certaines gammes de masse uniquement. Sauf contre-indication, toute la gamme de masse est considérée.

5.1 Variables d'entrée

Les différentes variables d'entrée considérées sont listées dans la section 2.4. La plupart de celles-ci sont généralement déjà exploitées dans les analyses en cours. Ce n'est toutefois pas garanti, en particulier pour les variables relatives à l'activité hadronique additionnelle. L'utilisation de variables supplémentaires demande, en plus de la mise en place de leur obtention, de reprendre potentiellement de longues étapes de calculs. Se restreindre à un sous-ensemble des variables d'entrée, si cela ne dégrade pas la qualité des modèles, pourrait donc faciliter l'intégration de nos modèles.

Les sous-ensembles des variables d'entrée sont définis par les restrictions suivantes :

- sans N_{PU} : la variable N_{PU} n'est pas utilisée ;
- sans N_{ν}^{reco} : la variable N_{ν}^{reco} n'est pas utilisée ;
- sans AHA : les variables d'activité hadronique additionnelle ne sont pas utilisées ;
- sans jets : les variables relatives aux jets (dont AHA) ne sont pas utilisées ;
- sans m_{T} : les masses transverses ne sont pas utilisées ;
- sans METcov : la matrice de covariance de $E_{\text{T}}^{\text{miss}}$ n'est pas utilisée.

L'application de plusieurs de ces restrictions est également testée.

Les performances modèles entraînés avec les différents ensembles de variables d'entrée sont données figure X.11 pour les XGBs et figure X.12 pour les DNNs. Les modèles concernés par plusieurs restrictions sont comptés de manière pondérée dans chaque groupe correspondant à une restriction unique. Par exemple, un modèle soumis à la restriction « sans N_{PU} » et « sans N_{ν}^{reco} » a un poids de $\frac{1}{2}$ dans chacun de ces deux groupes. Une pondération par la quantité de modèles dans chaque groupe est de plus appliquée pour supprimer le biais lié à la quantité accrue de modèles dans le groupe sans restrictions. Les histogrammes ainsi créés sont superposés. Il est alors possible de voir les contributions de chacune des restrictions aux valeurs obtenues sur la métrique d'évaluation illustrée.

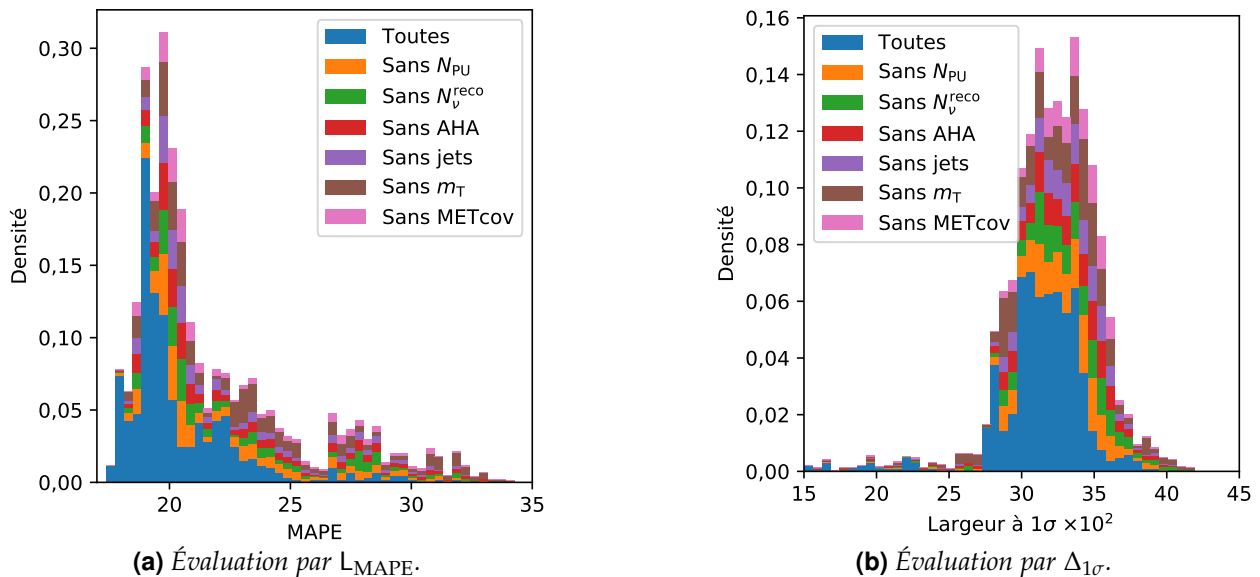


Figure X.11 – Évaluations des XGBs regroupés selon les variables d'entrée.

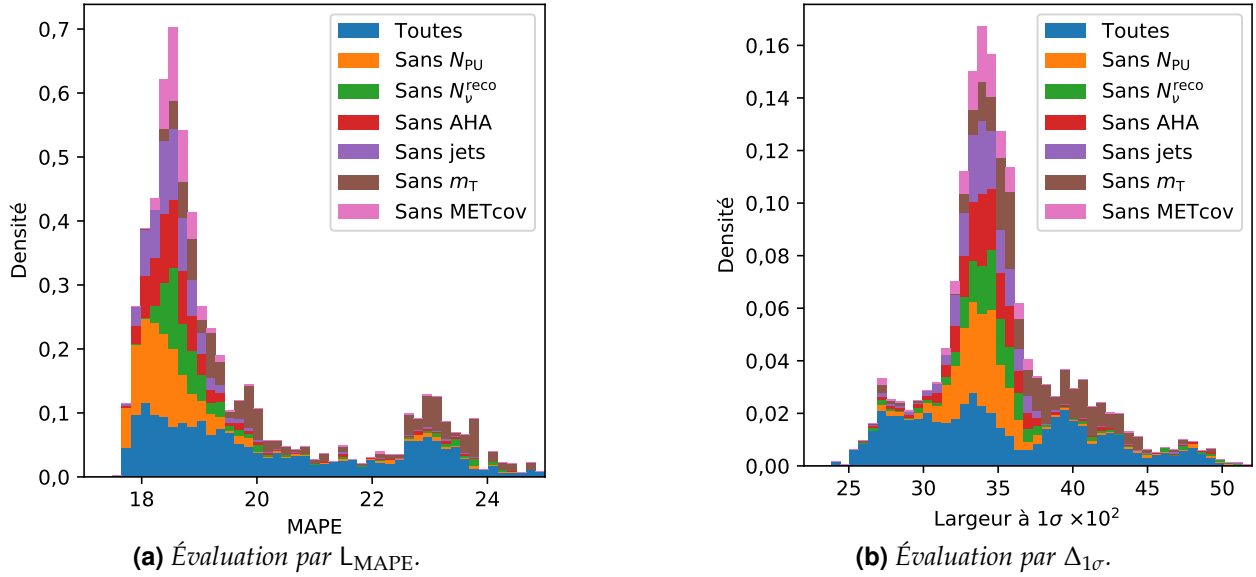


Figure X.12 – Évaluations des DNNs regroupés selon les variables d'entrée.

Dans le cas des XGBs, l'évaluation des modèles par L_{MAPE} , en figure X.11a, donne des valeurs situées entre 17 et 35. Le cœur de la distribution, à $L_{MAPE} = 19 \pm 2$, est plutôt constitué de modèles utilisant toutes les entrées dans sa partie gauche ($L_{MAPE} < 19$) et de modèles utilisant un sous-ensemble d'entrées dans sa partie droite ($19 < L_{MAPE} < 22$). De plus, les basses valeurs de L_{MAPE} , en-dessous de 18,5, sont presque exclusivement obtenues avec des modèles utilisant toutes les entrées. À l'inverse, la queue à hautes valeurs de la distribution obtenue ($L_{MAPE} > 23$) est largement dominée par les contributions des modèles avec un sous-ensemble d'entrées.

La plupart des XGBs ont une largeur $\Delta_{1\sigma}$, en figure X.11b, située entre 27 et 38. Cependant, les XGBs utilisant toutes les variables d'entrée exhibent une distribution de $\Delta_{1\sigma}$ légèrement décalée vers de plus faibles valeurs.

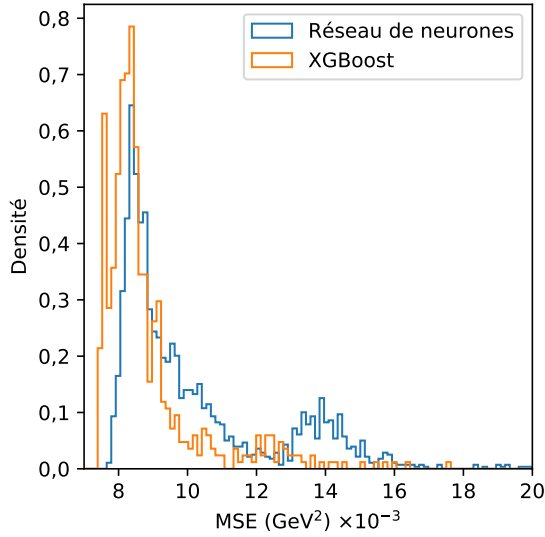
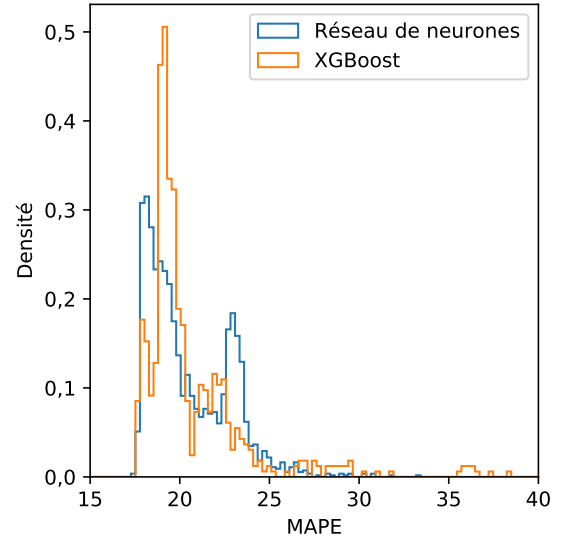
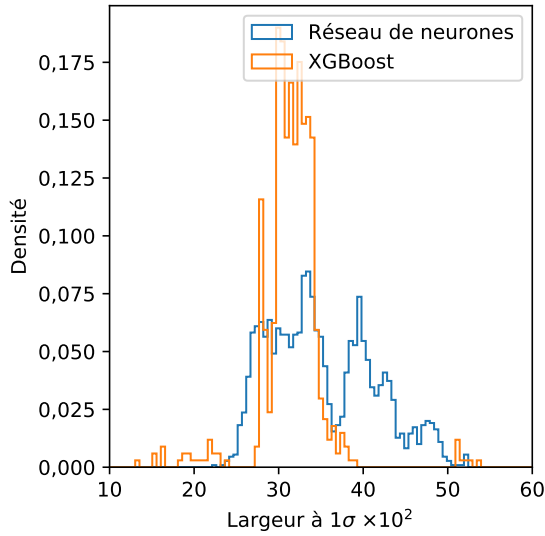
Dans le cas des DNNs, la distribution de la métrique L_{MAPE} , en figure X.12a, contient des valeurs situées entre 17,5 et 25. Les DNNs n'utilisant pas N_{ν}^{reco} se situent à $L_{MAPE} > 18$. Cette variable permet aux modèles de différencier les canaux hadroniques, semi-leptoniques et leptoniques, dont la séparation est discutée dans la section 6. Ceux n'utilisant pas m_T présentent également des valeurs de L_{MAPE} uniquement au-delà de 18. L'utilisation de ces variables permet donc d'obtenir de meilleurs modèles. Elles sont de plus facilement obtenues à partir du *dilepton*, défini chapitre 4, et de E_T^{miss} . Les analyses avec deux leptons tau dans l'état final exploitent déjà ces observables, leur utilisation par nos modèles est donc à la fois pertinente, car les scores de L_{MAPE} obtenus sont meilleurs, et sans incidence sur la facilité d'intégration du modèle à l'analyse. Les DNNs avec $L_{MAPE} \lesssim 18$ exploitent presque tous les variables relatives aux jets, à l'AHA et à la matrice de covariance de E_T^{miss} . Ces entrées sont donc vraisemblablement utiles aux DNNs afin de réaliser la régression. Enfin, la restriction sur N_{PU} ne semble pas dégrader les performances des DNNs selon L_{MAPE} .

La distribution de $\Delta_{1\sigma}$, en figure X.12b, montre que les modèles utilisant toutes les entrées peuvent se répartir en plusieurs groupes, aux alentours des valeurs 0,275, 0,335, 0,395, 0,425 et 0,475. À 0,395 apparaît également un groupe de modèles entraînés sans m_T . À 0,335 se trouvent la majorité des modèles entraînés avec une restriction des entrées. Pour $\Delta_{1\sigma} < 0,3$, les modèles sont très majoritairement ceux utilisant l'ensemble des variables proposées.

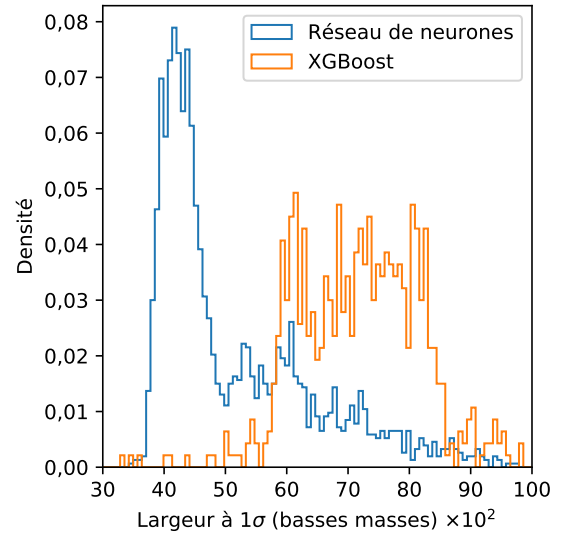
L'utilisation de toutes les variables listées dans la section 2.4 est donc corrélée avec de meilleures performances selon les métriques L_{MAPE} et $\Delta_{1\sigma}$. Par la suite, seuls les modèles utilisant toutes les variables, au nombre de 27, sont considérés.

5.2 Type de modèle

Les figures X.13 et X.14 présentent les distributions des scores de L_{MSE} , L_{MAPE} et $\Delta_{1\sigma}$ pour l'ensemble des DNNs et des XGBs utilisant toutes les variables d'entrée.

(a) Évaluation par L_{MSE} .(b) Évaluation par L_{MAPE} .**Figure X.13** – Évaluations des XGBs et des DNNs par L_{MSE} et L_{MAPE} .

(a) Sur toute la gamme de masse.



(b) À basse masse.

Figure X.14 – Évaluations des XGBs et des DNNs par $\Delta_{1\sigma}$.

L'évaluation par L_{MSE} , en figure X.13a, favorise les XGBs. Le cœur de la distribution de L_{MSE} pour ces modèles est en effet à $8,1 \times 10^3 \text{ GeV}^2$ contre $8,5 \times 10^3 \text{ GeV}^2$ pour les DNNs. En revanche, l'évaluation par L_{MAPE} , en figure X.13b, favorise les DNNs avec un groupe de DNNs à $L_{MAPE} = 18$ contre 19 pour les XGBs. Un second groupe de DNNs est présent à $L_{MAPE} = 23$. L'existence de ces deux groupes est due à l'utilisation de plusieurs algorithmes d'optimisation, comme discuté dans la section 5.4.

La résolution des modèles est évaluée par $\Delta_{1\sigma}$ en figure X.14a pour toute la gamme de masse et en figure X.14b pour les basses masses. Sur l'ensemble de la gamme de masse, les XGBs ont un score de $0,32 \pm 0,04$ et les DNNs se répartissent en plusieurs groupes à environ 0,28, 0,33, 0,40, 0,42 et 0,48. Les XGBs sont ainsi compétitifs d'après cette évaluation. Cependant, les performances des modèles à basse masse, c'est-à-dire pour $m_H < 150 \text{ GeV}$, sont importantes car c'est dans cette gamme de masse que se trouvent les bosons Z et h du modèle standard. Sur la figure X.14b, l'évaluation à basse masse par $\Delta_{1\sigma}$ les XGBs ont un score de $0,70 \pm 0,15$ alors que les DNNs se répartissent en deux groupes, le premier à $0,42 \pm 0,05$ et le second entre 0,50 et 1,0. Le premier ensemble de DNNs propose les meilleures résolutions sur les masses des particules du modèle standard.

La réévaluation des modèles par L_{MSE} et L_{MAPE} à basse masse, en figures X.15a et X.15b, confirme

l'obtention de meilleures performances avec les DNNs. En effet, les DNNs sont les seuls modèles avec $L_{MSE} < 1,5 \times 10^3 \text{ GeV}^2$ et $L_{MAPE} < 28$ à basse masse. Les XGBs ont des scores de L_{MSE} et L_{MAPE} généralement entre $1,5 \times 10^3 \text{ GeV}^2$ et $4,0 \times 10^3 \text{ GeV}^2$ et entre 28 et 43, respectivement. La suite de la sélection d'un modèle est donc faite parmi les DNNs.

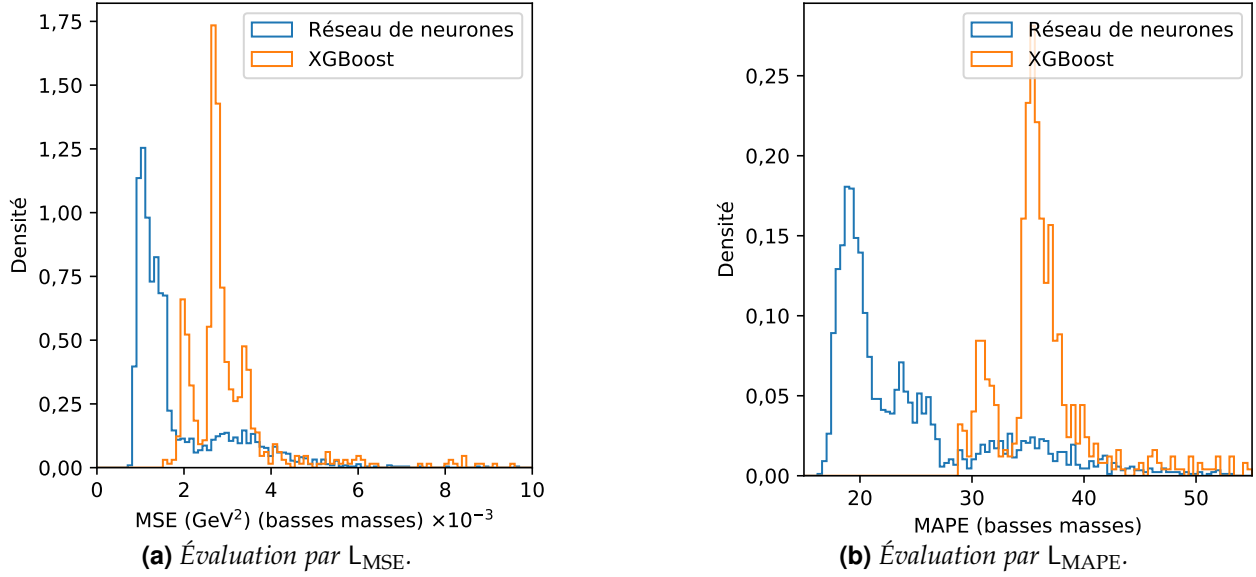


Figure X.15 – Évaluations des XGBs et des DNNs par L_{MSE} et L_{MAPE} à basse masse.

5.3 Fonction de coût

Les évaluations des DNNs, regroupés d'après la fonction de coût utilisée lors de leurs entraînements, selon les métriques L_{MSE} , L_{MAPE} , L_{MAE} sur toute la gamme de masse et $\Delta_{1\sigma}$ à basse masse sont représentées sur la figure X.16.

L'évaluation par L_{MSE} est représentée figure X.16a. Les DNNs entraînés avec $L = L_{MSE}$ y présentent un score compris entre $7,8 \times 10^3 \text{ GeV}^2$ et $15 \times 10^3 \text{ GeV}^2$, la majorité d'entre eux se trouvant en-dessous de $11 \times 10^3 \text{ GeV}^2$ avec un pic de leur distribution à $8,7 \times 10^3 \text{ GeV}^2$. Les DNNs entraînés avec $L = L_{MAE}$ se situent majoritairement entre $7,7 \times 10^3 \text{ GeV}^2$ et $10 \times 10^3 \text{ GeV}^2$, avec un pic de leur distribution à $8,3 \times 10^3 \text{ GeV}^2$. Les DNNs entraînés avec $L = L_{MAPE}$ se répartissent en deux groupes, le premier entre $7,9 \times 10^3 \text{ GeV}^2$ et $10 \times 10^3 \text{ GeV}^2$, le second entre $13 \times 10^3 \text{ GeV}^2$ et $16 \times 10^3 \text{ GeV}^2$. La fonction de coût L_{MAE} semble ainsi préférable à L_{MSE} lorsque la comparaison de fait sur L_{MSE} elle-même. Il est en revanche plus difficile de conclure quant à L_{MAPE} .

L'évaluation par L_{MAPE} , figure X.16b, montre également un avantage de L_{MAE} sur L_{MSE} . En effet, les modèles entraînés avec $L = L_{MAE}$ se situent majoritairement à $L_{MAPE} < 21$ alors que ceux entraînés avec $L = L_{MSE}$ sont plutôt dans la région $L_{MAPE} > 20$. Enfin, les valeurs de L_{MAPE} les plus basses sont atteintes par les modèles entraînés avec cette même fonction de coût. Il n'est donc pas pertinent d'affirmer que L_{MAPE} est préférable d'après les résultats de la figure X.16b.

Afin de compléter ces comparaisons, la figure X.16c représente l'évaluation des DNNs par L_{MAE} . La distribution obtenue avec les DNNs entraînés avec $L = L_{MSE}$ s'étend de 65 GeV à près de 10 GeV avec un pic à 71 GeV. En revanche, de nombreux modèles entraînés avec L_{MAE} ou L_{MAPE} se situent à $67 \pm 4 \text{ GeV}$.

Enfin, sur la figure X.16d se trouvent les distributions de $\Delta_{1\sigma}$ à basse masse pour ces trois groupes de DNNs. Les modèles utilisant L_{MSE} ont tous un score supérieur à 0,5. Ceux entraînés avec L_{MAE} se situent entre 0,4 et 0,6. Les modèles basés sur L_{MAPE} forment encore deux groupes, le premier entre 0,34 et 0,5, le second entre 0,5 et 0,64. Les fonctions de coût L_{MAPE} et L_{MAE} permettent donc d'obtenir des modèles avec une meilleure résolution à basse masse que L_{MSE} .

Les modèles entraînés avec $L = L_{MAE}$ ou $L = L_{MAPE}$ proposent de meilleurs scores que ceux obtenus avec $L = L_{MSE}$, quelle que soit la métrique d'évaluation utilisée. Lors des évaluations avec L_{MSE}

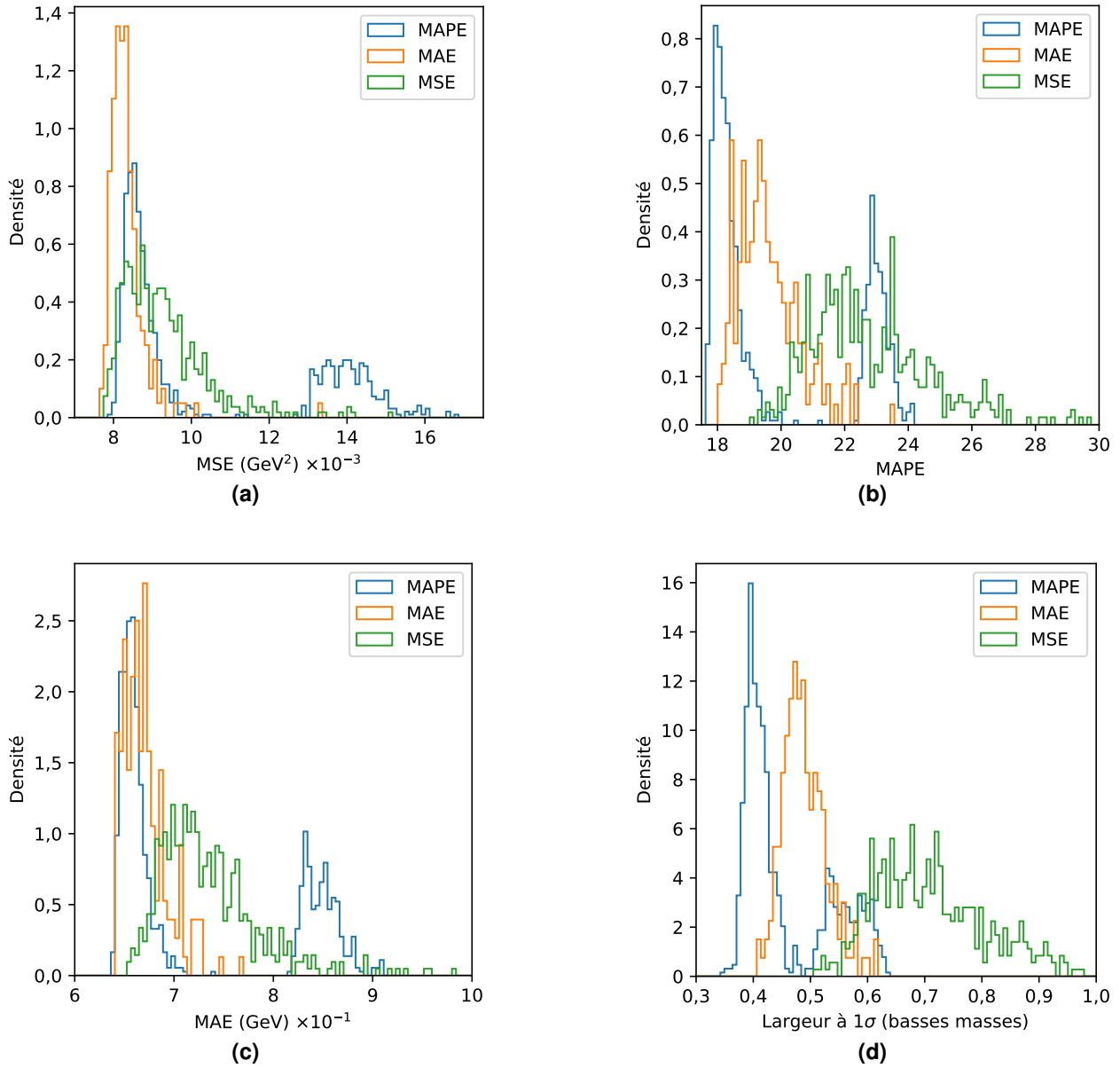


Figure X.16 – Évaluations des DNNs regroupés selon la fonction de coût par L_{MSE} , L_{MAPE} , L_{MAE} et $\Delta_{1\sigma}$.

ou L_{MAE} , aucun avantage net n'est visible entre $L = L_{MAE}$ et $L = L_{MAPE}$. En revanche, les métriques L_{MAPE} et $\Delta_{1\sigma}$ montrent que certains modèles entraînés avec $L = L_{MAPE}$ donnent de meilleurs résultats. La sélection d'un modèle est donc réalisée parmi ceux ayant comme fonction de coût L_{MAPE} .

5.4 Algorithme d'optimisation

Les algorithmes d'optimisation sont présentés dans la section 4.3.4. L'algorithme SGD ne permet pas aux modèles de converger, il est donc exclu de nos investigations. Deux algorithmes sont comparés, Adadelata et Adam.

Les évaluations des DNNs précédemment sélectionnés, regroupés d'après l'algorithme d'optimisation utilisé lors de leurs entraînements, selon les métriques L_{MAPE} sur toute la gamme de masse et $\Delta_{1\sigma}$ à basse masse sont représentées sur la figure X.17. Les deux groupes observés dans les sections précédentes sont identifiés comme étant les modèles entraînés respectivement par Adam et Adadelata. Dans le cadre de nos travaux, nous avons initialement utilisé Adam jusqu'à sélectionner le jeu de variables d'entrée (section 5.1) et la fonction de coût (section 5.3) à utiliser. C'est pourquoi ces deux groupes liés à Adam et Adadelata n'apparaissent que dans certains ensembles de modèles. Lors de la sélection finale, nous avons choisi de reprendre tous les modèles dès la début de la procédure.

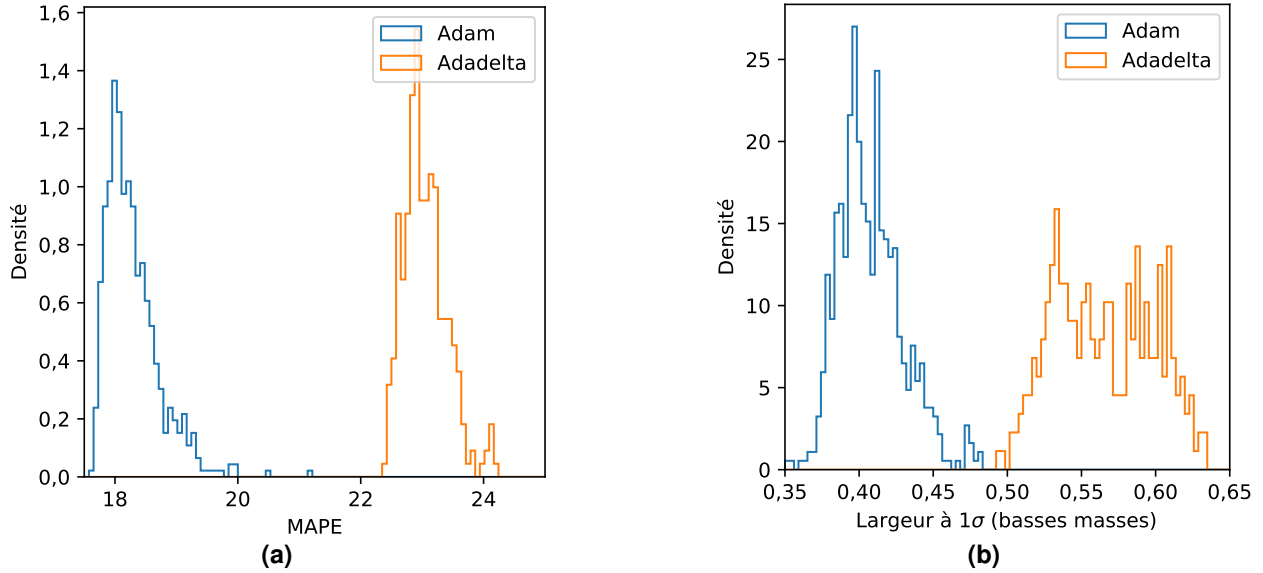


Figure X.17 – Évaluations des DNNs regroupés selon l'algorithme d'optimisation par L_{MAPE} et $\Delta_{1\sigma}$.

Sur la figure X.17a, les modèles optimisés par Adam présentent un score de L_{MAPE} entre 17,5 et 20 alors que ceux optimisés par Adadelta se situent entre 22,2 et 24,3. L'optimisation par Adam semble donc meilleure que celle par Adadelta. L'évaluation à basse masse par $\Delta_{1\sigma}$ sur la figure X.17b confirme cette observation. Les modèles optimisés par Adam se situent en effet entre 0,35 et 0,48, ceux optimisés par Adadelta entre 0,49 et 0,64.

L'algorithme d'optimisation Adam donne de meilleurs modèles qu'Adadelta.

5.5 Autres hyper-paramètres

Les hyper-paramètres restant à être fixés ainsi que les valeurs explorées sont :

- le nombre de couches cachées N_L , 2 à 5;
- le nombre de neurones par couche cachée N_N , 200 à 2000 par pas de 100;
- le mode d'initiation des poids (WI), uniforme (u), normale (n), Glorot uniforme (gu), Glorot normale (gn);
- la fonction d'activation (FA) des neurones des couches cachées, ReLU, SELU, ELU, Softplus.

Les évaluations à basse, moyenne et haute masse des DNNs utilisant les 27 variables d'entrée et entraînés par Adam avec $L = L_{MAPE}$, regroupés respectivement par N_L , N_N , mode d'initiation des poids et fonction d'activation sont données sur les figures X.18, X.19, X.20 et X.21 respectivement. La position du modèle final sélectionné est également indiquée.

Les regroupements définis par une valeur fixée d'un seul hyper-paramètre ne montrent aucune corrélation avec les valeurs des métriques d'évaluation utilisées. La méthode employée jusqu'ici pour sélectionner les hyper-paramètres ne permet donc pas de conclure. Nous avons alors choisi d'utiliser la procédure suivante :

1. Déterminer la valeur maximale $x_{\max}^{\text{métrique } m}$, sur l'ensemble des modèles sélectionnés, de chacune des métriques d'évaluation m utilisées. La valeur maximale autorisée $x_{\text{OK}}^{\text{métrique } m}$ pour la métrique m est initialement fixée à $x_{\max}^{\text{métrique } m}$;
2. Fixer la valeur maximale autorisée à 99 % de sa valeur actuelle pour chacune des métriques m ;
3. Rejeter tout modèle dont une des métriques donne un score supérieur à $x_{\text{OK}}^{\text{métrique } m}$;
4. Reprendre à l'étape 2 si plus de 10 modèles sont encore sélectionnés.

Les modèles ainsi sélectionnés, au nombre de 7, sont listés dans le tableau X.1 sans ordre particulier. Leurs réponses sont données sur les figures X.22 et X.23.

Aucun modèle avec $N_L \in \{2, 5\}$ n'est sélectionné. Pour 4 modèles, $N_L = 3$. Le nombre de neurones par couche cachée est de 1000 pour 4 modèles sur 7, dont 3 sur les 4 avec $N_L = 3$. Le WI le plus représenté est Glorot uniforme (5/7). Les FA sont disparates, chacune apparaissant une ou deux fois

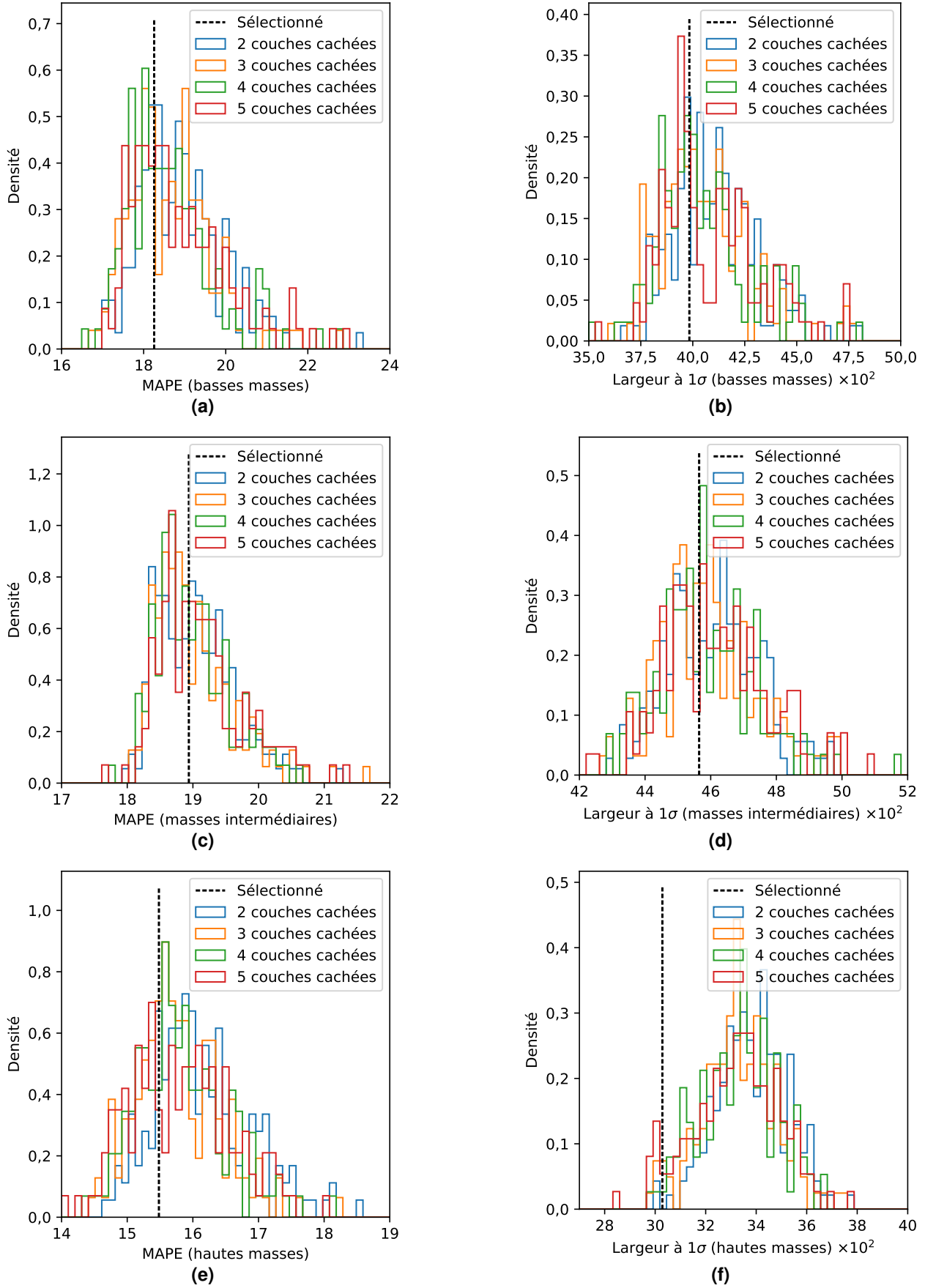


Figure X.18 – Évaluations des DNNs regroupés selon N_L par L_{MAPE} et $\Delta_{1\sigma}$.

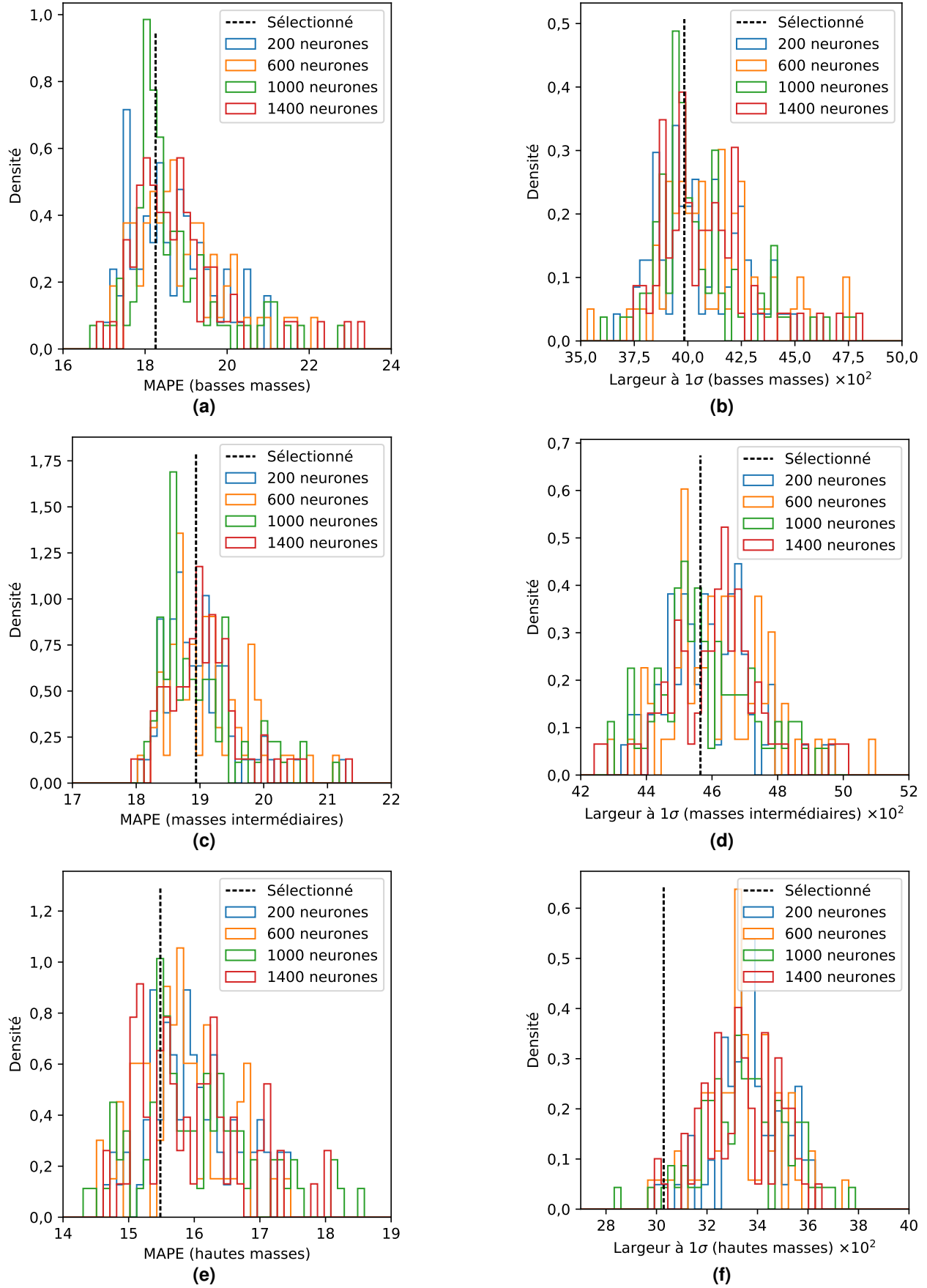


Figure X.19 – Évaluations des DNNs regroupés selon N_N par L_{MAPE} et $\Delta_{1\sigma}$.

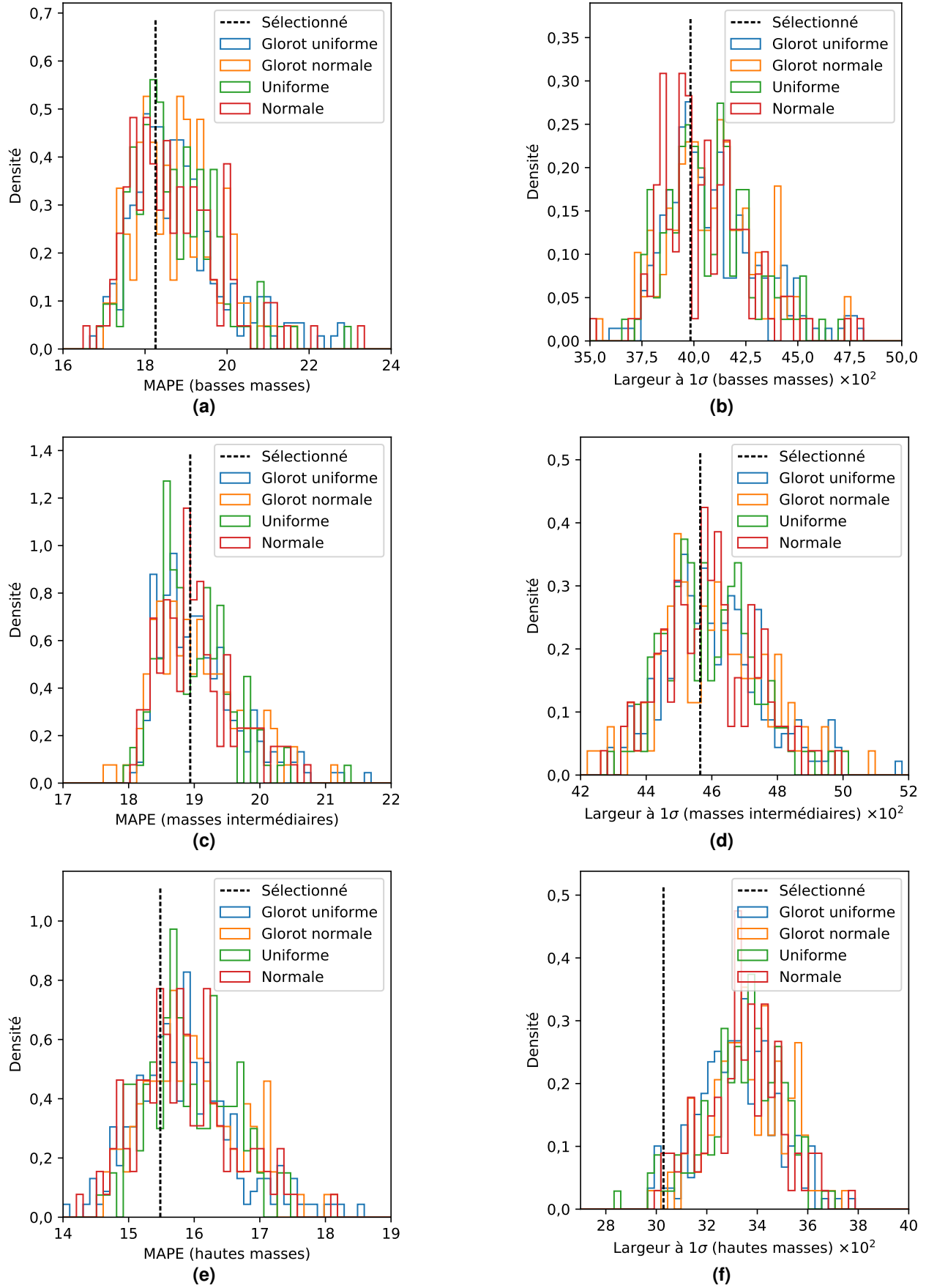


Figure X.20 – Évaluations des DNNs regroupés selon le mode d'initiation des poids par L_{MAPE} et $\Delta_{1\sigma}$.

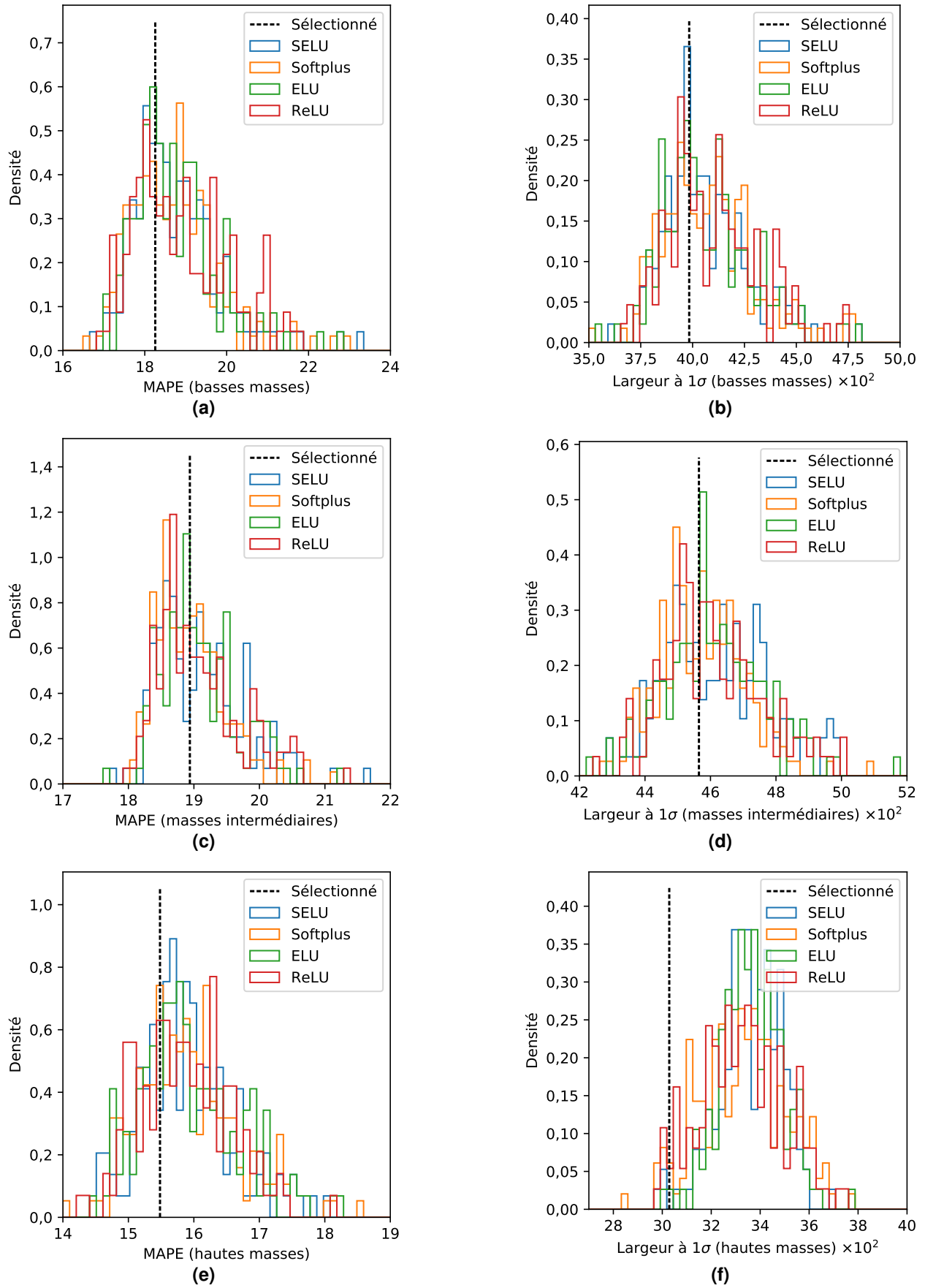


Figure X.21 – Évaluations des DNNs regroupés selon la fonction d'activation par L_{MAPE} et $\Delta_{1\sigma}$.

| Modèle | N_L | N_N | WI | FA |
|--------|-------|-------|----|----------|
| A | 3 | 1000 | gu | ELU |
| B | 3 | 1000 | gu | Softplus |
| C | 3 | 1000 | n | SELU |
| D | 3 | 1400 | gu | ReLU |
| E | 4 | 200 | gn | ReLU |
| F | 4 | 1000 | gu | ELU |
| G | 4 | 1400 | gu | Softplus |

Tableau X.1 – Liste des 7 modèles sélectionnés.

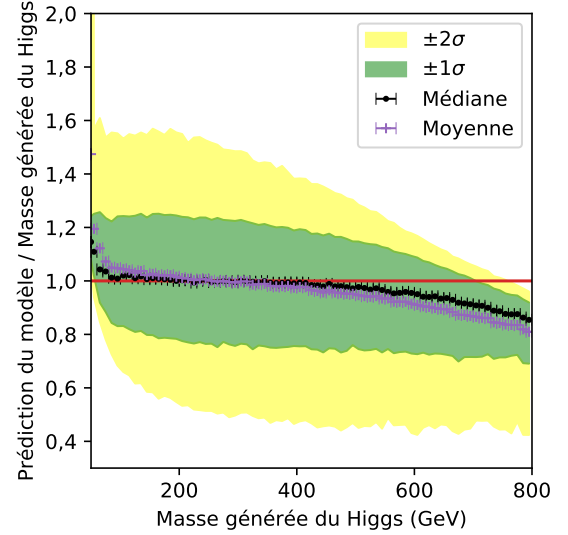


Figure X.22 – Réponse du modèle A.

dans la sélection.

Chacun de ces modèles présente une réponse proche de 1 entre 70 et 400 GeV avec une résolution à $\pm 1\sigma$ de l'ordre de 20 % à basse masse et 10 % à haute masse. Le modèle F conserve une réponse proche de 1 jusqu'à environ 500 GeV, cependant sa résolution à basse masse est légèrement dégradée par rapport aux autres modèles. Le modèle B a des hyper-paramètres « consensus », c'est-à-dire que chacune des valeurs de ses hyper-paramètres correspond à la valeur la plus représentée dans la sélection. C'est à partir de ce modèle que nous avons choisi de continuer notre étude. Les hyper-paramètres sont donc :

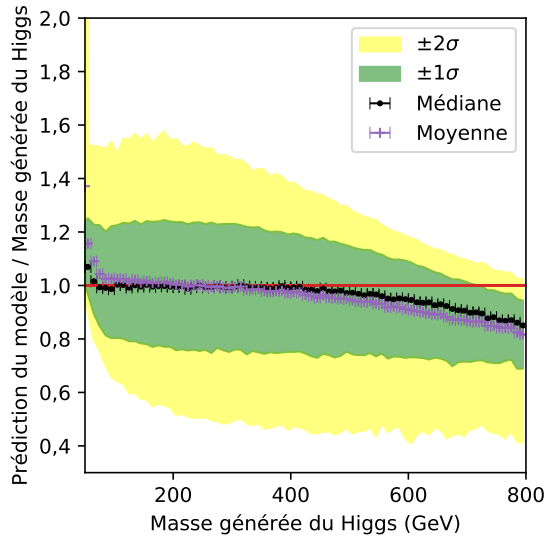
- 3 couches cachées;
- 1000 neurones par couche cachée;
- fonction d'activation Softplus, $x \mapsto \ln(1 + e^x)$;
- algorithme d'optimisation Adam, présenté en section 4.3.4;
- fonction de coût MAPE, définie équation (X.10);
- initiation des poids selon le mode « Glorot Uniforme » [36];
- la liste des 27 variables d'entrée égale à celle donnée en section 2.4.

En comparaison, le DNN de BÄRTSCHI & coll. [18] a pour hyper-paramètres :

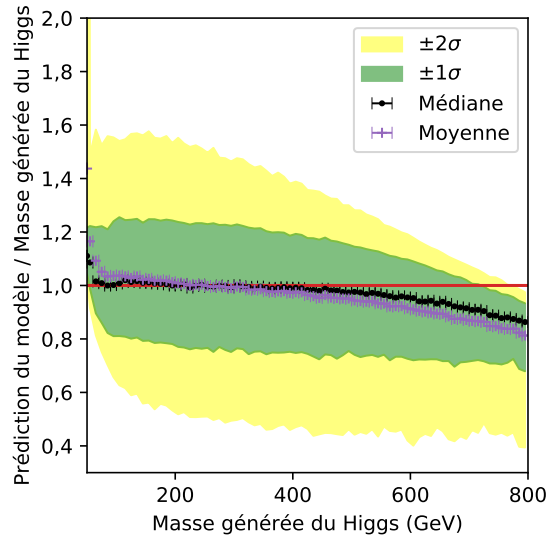
- 4 couches cachées;
- 200 neurones par couche cachée;
- fonction d'activation ReLU y compris pour la couche de sortie;
- algorithme d'optimisation Adam;
- fonction de coût MSE, définie équation (X.8);
- 17 variables d'entrée :
 - les impulsions, énergies et masses invariantes de L_1 et L_2 :
 $p_T^{L_1}, \eta^{L_1}, \phi^{L_1}, E^{L_1}, m_{\text{inv}}^{L_1}, p_T^{L_2}, \eta^{L_2}, \phi^{L_2}, E^{L_2}, m_{\text{inv}}^{L_2}$;
 - l'énergie transverse manquante :
 $E_T^{\text{miss}}, \phi^{E_T^{\text{miss}}}$;
 - la masse colinéaire du système di- τ , m_{coll} , définie dans [18];
 - 4 booléens donnant le type de canal (hadronique, semi-leptonique ou leptonique).

6 Discussions

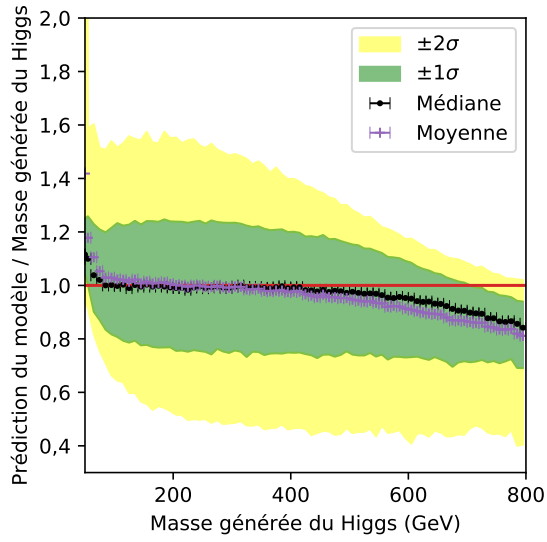
Les effets de l'empilement, de la reconstruction des particules, des faux taus hadroniques, de la séparation des canaux et de l'intervalle de masse sur les prédictions de masse sont discutées ci-après.



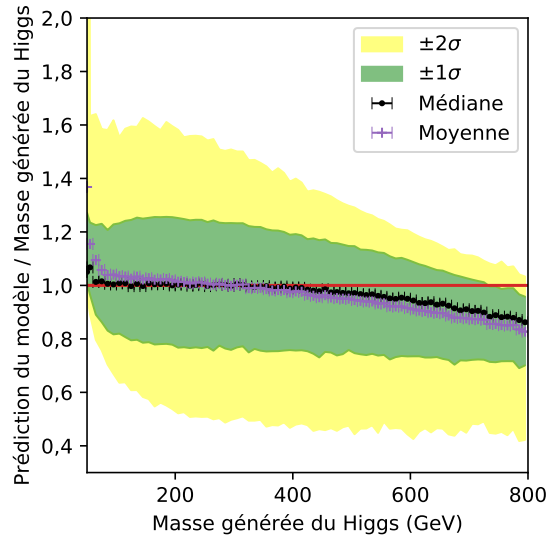
(a) Réponse du modèle B.



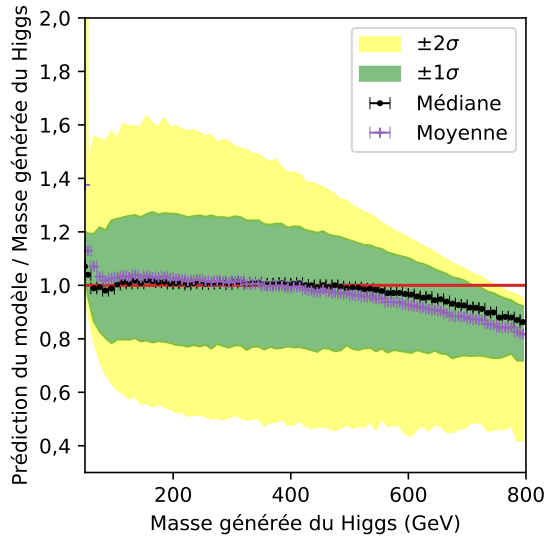
(b) Réponse du modèle C.



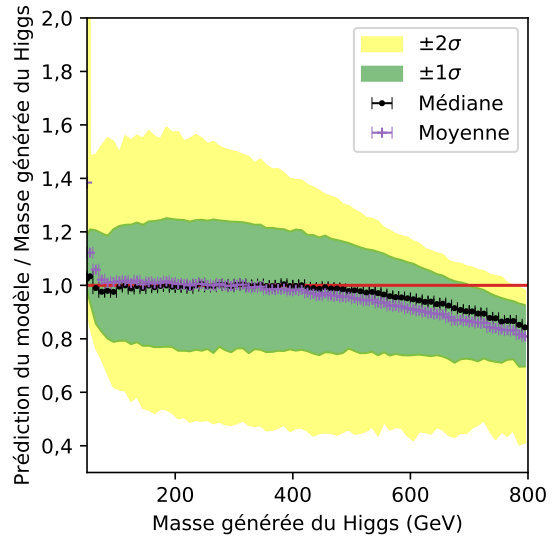
(c) Réponse du modèle D.



(d) Réponse du modèle E.



(e) Réponse du modèle F.



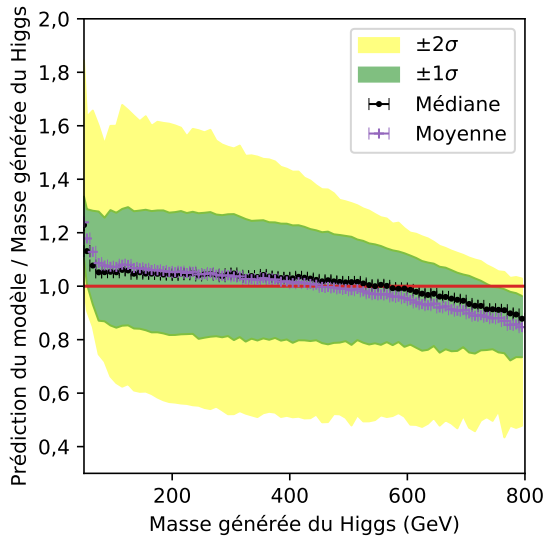
(f) Réponse du modèle G.

Figure X.23 – Réponse des modèles B à G.

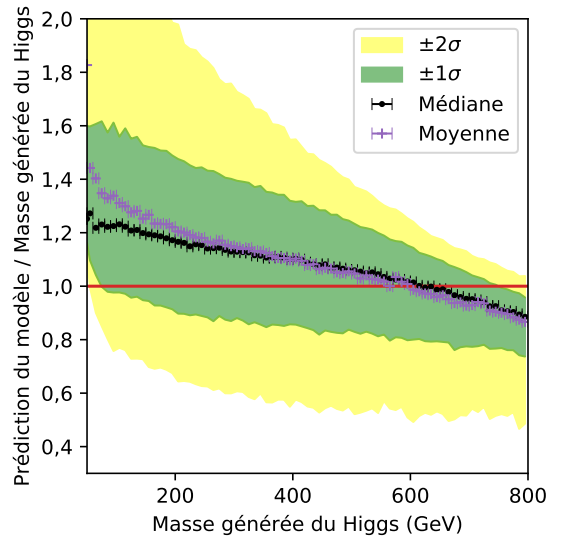
Lors de notre étude, ces effets ont été observés en parallèle de l’exploration des hyper-paramètres présentée en section 5 avec des modèles divers. À des fins de cohérence dans la comparaison des effets, nous utilisons ici le modèle B sélectionné précédemment comme référence.

6.1 Effet de l’empilement

Dans les travaux de BÄRTSCHI & coll. [18], l’empilement (PU, *Pile-Up*) n’est pas considéré. Nous avons donc souhaité déterminer l’effet du PU sur les prédictions de notre modèle. Pour cela, les mêmes événements que ceux décrits en section 2 ont été générés sans PU. Un DNN, noté B^{0PU} , est entraîné sur ces événements sans PU. Les hyper-paramètres de B^{0PU} sont ceux de B, à l’exception des variables d’entrées auxquelles N_{PU} est retiré, car $N_{PU} = 0$ pour tous les événements sans PU. La réponse de B^{0PU} sur les événements de test sans PU est représentée sur la figure X.24a. La réponse du modèle est de l’ordre de 1 pour m_H entre 80 GeV et 600 GeV avec une résolution à $\pm 1\sigma$ de l’ordre de 20 % à basse masse et 10 % à haute masse.



(a) Réponse de B^{0PU} sur les événements sans PU.

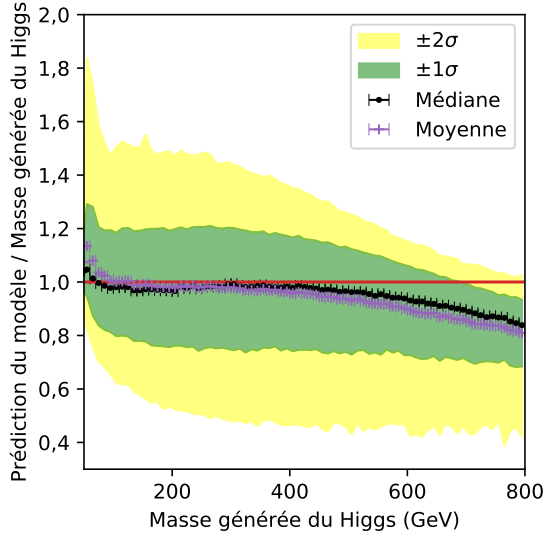


(b) Réponse de B^{0PU} sur les événements avec PU.

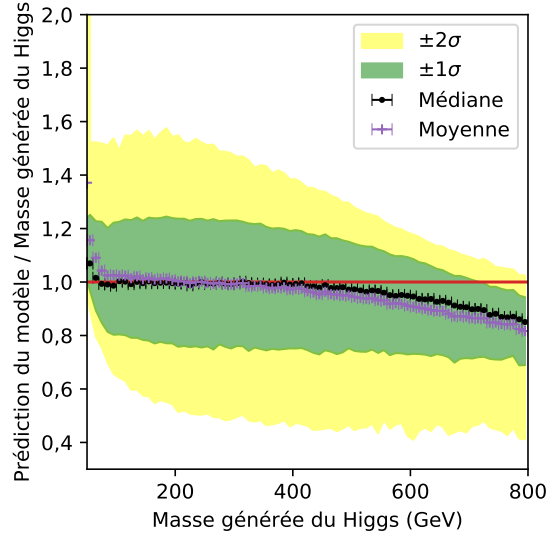
Figure X.24 – Réponses du modèle B^{0PU} sur les événements sans et avec PU.

Cependant, la réponse de B^{0PU} est dégradée sur des événements contenant du PU, figure X.24b. La réponse médiane se situe en effet à 1,2 à $m_H = 100$ GeV et diminue à 0,9 à $m_H = 800$ GeV contre 1,05 et 0,9 sans PU respectivement. La résolution relative à basse masse est de l’ordre de 30 %. Plus m_H est faible, plus basse est l’énergie portée par L_1 et L_2 , issue de la masse de \mathcal{H} . Alors, les particules du PU sont compétitives, en termes de propriétés cinématiques, avec L_1 et L_2 . Lors de la sélection des événements et de la construction du *dilepton* présentée au chapitre 4, il est ainsi possible que les particules utilisées en tant que L_1 ou L_2 soient en réalité issues du PU et non de la désintégration de \mathcal{H} . Plus m_H augmente, moins le PU est compétitif, d’où l’atténuation de l’effet sur la réponse observée du modèle. Il est donc primordial d’inclure le PU dans l’entraînement dans l’optique d’une utilisation de nos modèles dans les analyses de CMS.

La réponse du modèle B, entraîné avec PU, peut être comparée dans le cas d’événements sans PU, figure X.25a, au cas d’événements avec PU, figure X.25b (identique à X.23a). Le profil de PU utilisé pour générer les événements d’entraînement est celui de l’année 2017. Or, il apparaît que le modèle B est peu sensible au retrait du PU, les réponses étant similaires sur les figure X.25a et X.25b. L’utilisation de B sur des événements dont le profil de PU est légèrement différent de celui de l’année 2017, comme c’est le cas pour les autres années du Run II (2016, 2018) est ainsi directement envisageable.



(a) Réponse de B sur les événements sans PU.



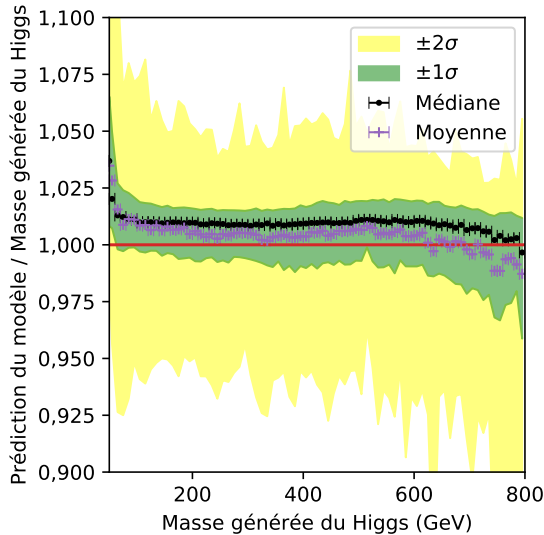
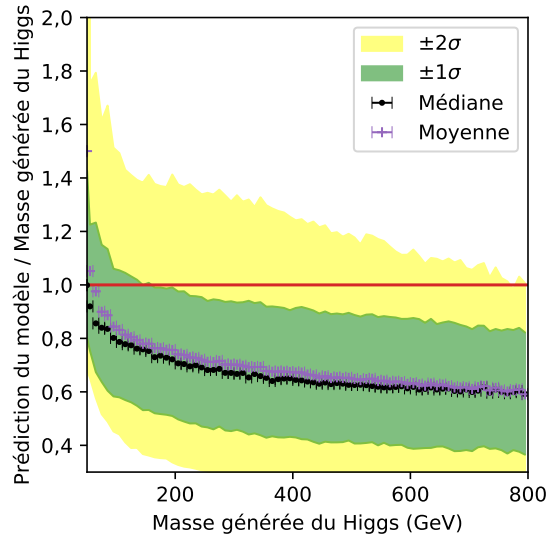
(b) Réponse de B sur les événements avec PU.

Figure X.25 – Réponses du modèle B sur les événements sans et avec PU.

6.2 Effet de la reconstruction des particules

La reconstruction des particules est présentée dans le chapitre 2. Son effet peut être caractérisé par l'étude du modèle B^{gen} , ayant les mêmes hyper-paramètres que B mais entraîné en utilisant les objets générés au lieu de ceux reconstruits pour L_1 , L_2 et E_T^{miss} , c'est-à-dire pour les trois objets physiques issus de la désintégration des leptons tau (deux parties visibles L_1 et L_2 et E_T^{miss} pour les neutrinos). En particulier, les valeurs de $\vec{p}_T^{L_1}$, $\vec{p}_T^{L_2}$ et \vec{E}_T^{miss} correspondent exactement à la réalité. Il s'agit donc du cas où les objets physiques directement liés à la désintégration de \mathcal{H} sont parfaitement reconstruits.

La figure X.26 montre les réponses du modèle B^{gen} sur les événements avec reconstruction parfaite et réelle. Dans le cas d'une reconstruction parfaite, la réponse de B^{gen} est de l'ordre de $1,01^{+0,01}_{-0,02}$. Il s'agit donc d'une estimation de $m_{\mathcal{H}}$ avec une précision de 2 %.

(a) Réponse de B^{gen} dans le cas d'une reconstruction parfaite.(b) Réponse de B^{gen} dans le cas d'une reconstruction réelle.**Figure X.26** – Réponses du modèle B^{gen} dans le cas d'une reconstruction des particules parfaite et réelle.

Les DNNs sont donc en mesure de comprendre la physique des événements $\mathcal{H} \rightarrow \tau\tau$ afin d'estimer $m_{\mathcal{H}}$ à partir des objets physiques générés correspondant aux objets effectivement reconstruits

par le détecteur. Cependant, comme le montre la figure X.26b, l'utilisation de B^{gen} sur les variables reconstruites, effectivement accessibles expérimentalement, ne permet pas d'obtenir $m_{\mathcal{H}}$. En effet, la réponse de B^{gen} avec ces variables est inférieure à 1 et de l'ordre de 0,7 à haute masse. De plus, la résolution relative est de l'ordre de 40 %. Une des tâches des DNNs est donc de corriger cet effet de reconstruction.

6.3 Effet des faux taus hadroniques

La phénoménologie des événements contenant une paire de leptons tau est décrite dans le chapitre 1. Ces leptons peuvent se désintégrer hadroniquement en tau hadronique (τ_h) ou leptoniquement en électron (e) ou en muon (μ). Il existe ainsi six canaux différents dans les événements avec une paire de leptons tau, pouvant être répartis en trois groupes :

- complètement hadronique : $\tau_h \tau_h$, avec deux τ_h ;
- semi-leptoniques : $\mu \tau_h$ et $e \tau_h$, ou simplement $\ell \tau_h$, avec un τ_h ;
- leptoniques : $\mu \mu$, $e \mu$ et ee , ou simplement $\ell \ell$, sans τ_h .

Les faux taus hadroniques (*fakes* τ_h) sont des objets physiques tels que des électrons, des muons et surtout des jets identifiés à tort comme des τ_h . Ils représentent près de 70 % des événements dans le canal $\tau_h \tau_h$, 38 % dans le canal $\mu \tau_h$ et 68 % dans le canal $e \tau_h$. Les *fakes* τ_h sont particulièrement difficiles à modéliser dans les simulations [42, 43].

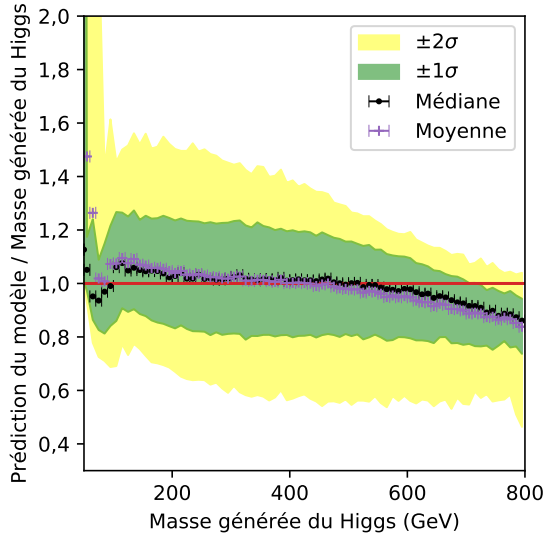
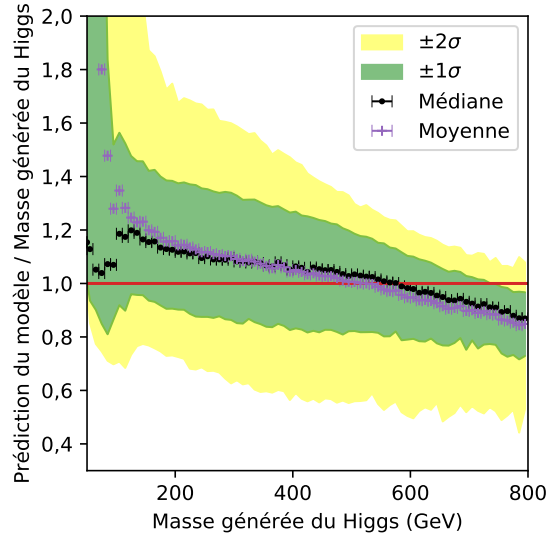
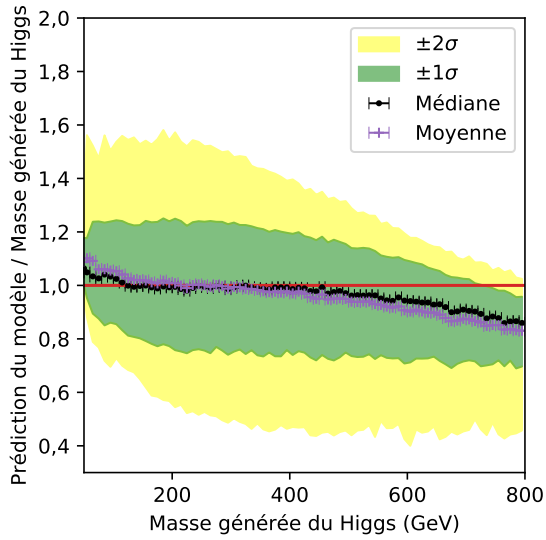
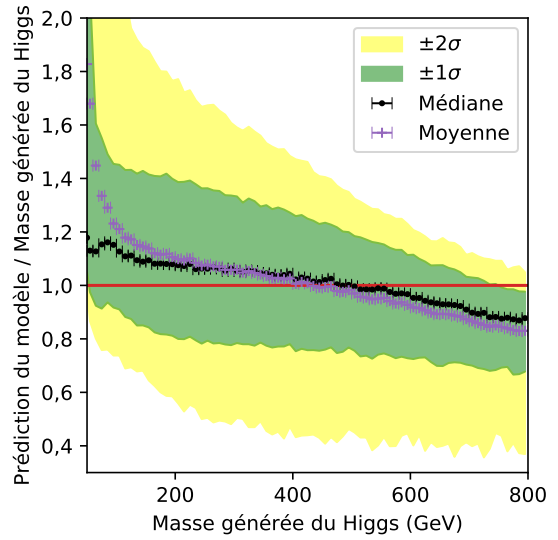
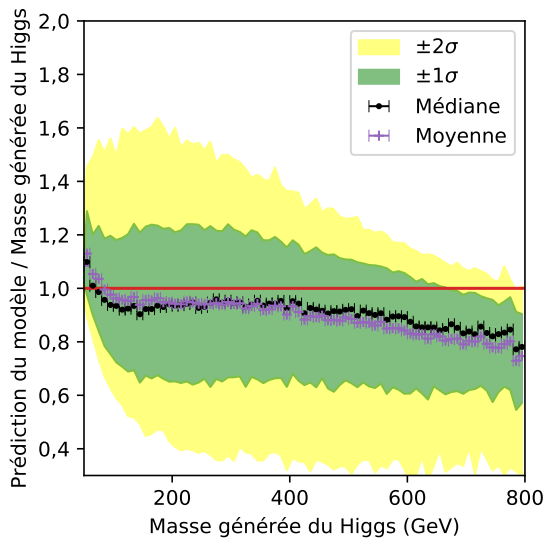
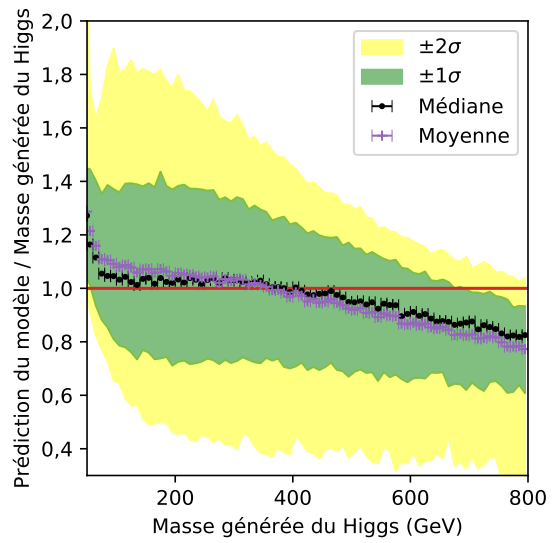
L'identification des τ_h est réalisée dans nos travaux à l'aide de l'algorithme DEEPTAU [11], qui présente un faible taux de mauvaise identification des τ_h , inférieur à 1 %. Cependant, une autre méthode d'identification des τ_h , basée sur un arbre de décision (BDT), peut être utilisée et présente un taux de mauvaise identification de jets en tant que τ_h pouvant atteindre 4 % [44]. Une sélection plus riche en *fakes* τ_h est ainsi obtenue.

Les réponses du modèle B sur chacun des trois groupes de canaux (hadronique, semi-leptoniques et leptoniques) sont représentées figure X.27 pour les deux ensembles de sélection des τ_h . Quel que soit le groupe d'état final, les réponses pour $m_{\mathcal{H}} > 600$ GeV ne sont pas affectées par la sélection des τ_h . En effet, pour de hautes valeurs de $m_{\mathcal{H}}$, les τ_h ont des impulsions suffisamment élevées pour être correctement sélectionnés par la séquence d'analyse. À basse masse en revanche, les *fakes* τ_h sont compétitifs et leur présence modifie la réponse du modèle qui augmente jusqu'à 20 % pour des masses entre 100 GeV et 600 GeV. L'effet le plus important se situe à très basse masse où la résolution est fortement dégradée. La figure X.28 montre la différence $y_{\text{préd}} - y_{\text{vraie}}$ entre les prédictions du modèle B et la valeur vraie de $m_{\mathcal{H}}$ pour des valeurs de $m_{\mathcal{H}}$ entre 50 et 200 GeV sur chacun des trois groupes de canaux et pour les deux ensembles de sélection des τ_h .

Les différences observée pour les canaux leptoniques ($\ell \ell$), figures X.28e et X.28f, sont bien moins importantes que dans les autres canaux. Les canaux leptoniques ne comportent aucun τ_h , seule la sélection des événements est modifiée. Un objet physique identifié comme un τ_h par le BDT et non par DEEPTAU peut en effet basculer d'un canal à l'autre, si le τ_h identifié par le BDT permet de construire un *dilepton*.

Dans le cas des canaux semi-leptoniques ($\ell \tau_h$), la différence entre $y_{\text{préd}}$ de B et y_{vraie} à basse masse est en moyenne inférieure à 10 GeV pour une sélection des τ_h par DEEPTAU, figure X.28c. La résolution relative est quant à elle inférieure à 25 %. Lors les τ_h sont identifiés par le BDT, figure X.28d, le modèle surestime $m_{\mathcal{H}}$ de 25 GeV en moyenne pour $70 \text{ GeV} < m_{\mathcal{H}} < 200 \text{ GeV}$ et de près de 40 GeV à $m_{\mathcal{H}} = 50 \text{ GeV}$. La résolution relative est de l'ordre de 25 % au-dessus de 70 GeV, moins bonne qu'avec DEEPTAU, et augmente drastiquement pour des masses plus basses, ce qui n'est pas le cas avec DEEPTAU. Il s'agit donc de la contribution des *fakes* τ_h .

Dans le canal $\tau_h \tau_h$, figures X.28a et X.28b, un effet similaire existe. La résolution relative est toujours de l'ordre de 20 %, mais la présence des *fakes* τ_h mène à une surestimation moyenne de 30 GeV pour $m_{\mathcal{H}} > 110 \text{ GeV}$ et pouvant aller jusqu'à 100 GeV pour $m_{\mathcal{H}} \simeq 50 \text{ GeV}$, soit une erreur de 200 %. La dégradation de la résolution à très basse masse commence dès 100 GeV, au lieu de 70 GeV pour les canaux $\ell \tau_h$. L'effet des *fakes* τ_h est donc plus important que dans les canaux $\ell \tau_h$, ce qui s'explique facilement par la présence de deux τ_h au lieu d'un seul. Pour $m_{\mathcal{H}} = 50 \text{ GeV}$, la résolution de B sur les événements avec DEEPTAU est également mauvaise. La sélection des τ_h se fait avec $p_T > 40 \text{ GeV}$, ce

(a) Canal $\tau_h \tau_h$, DEEPTAU.(b) Canal $\tau_h \tau_h$, BDT.(c) Canaux $\ell \tau_h$, DEEPTAU.(d) Canaux $\ell \tau_h$, BDT.(e) Canaux $\ell \ell$, DEEPTAU.(f) Canaux $\ell \ell$, BDT.Figure X.27 – Réponses du modèle B sur les différents types de canaux avec une quantité variable de fakes τ_h .

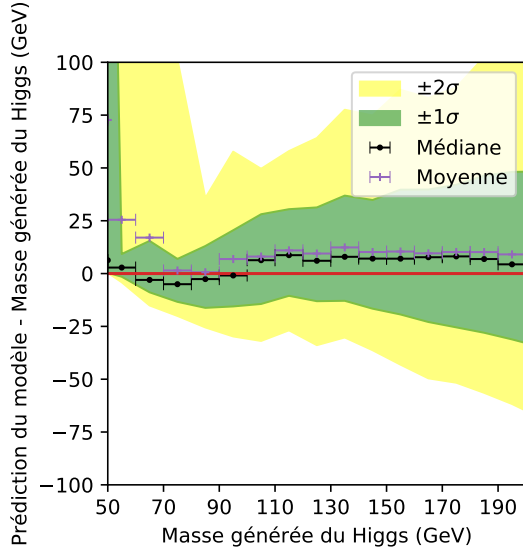
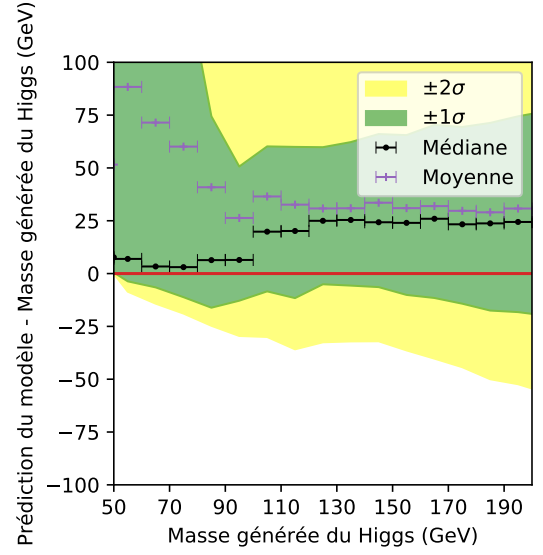
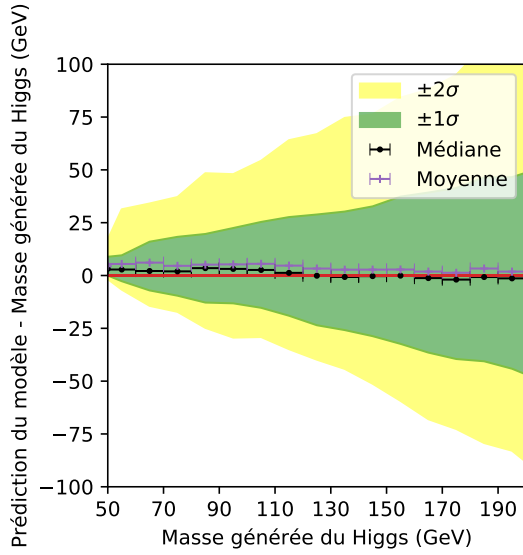
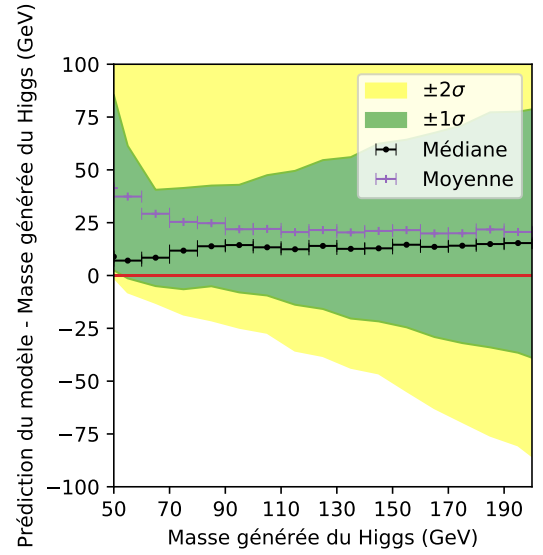
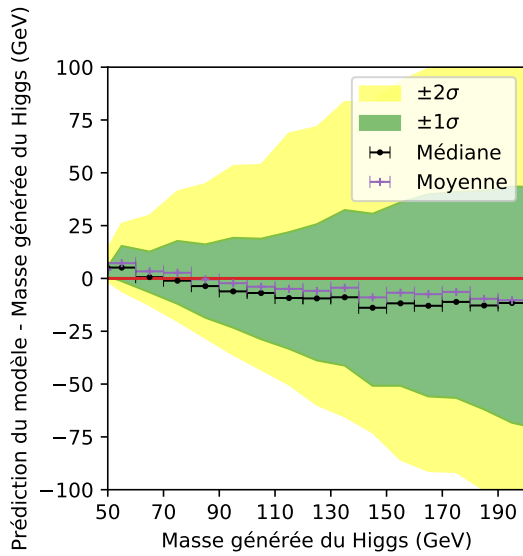
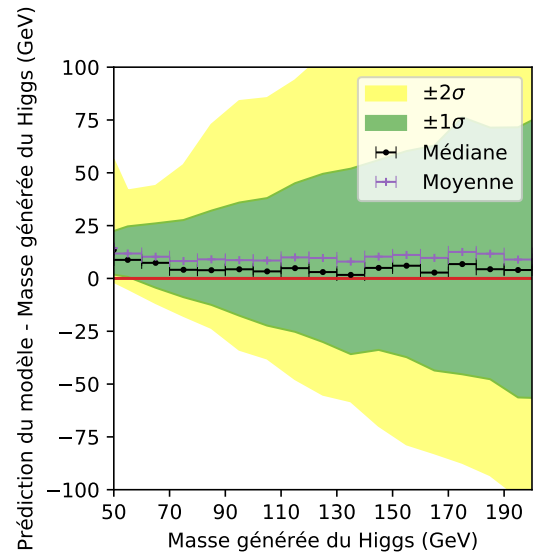
(a) Canal $\tau_h \tau_h$, DEEPTAU.(b) Canal $\tau_h \tau_h$, BDT.(c) Canaux $\ell \tau_h$, DEEPTAU.(d) Canaux $\ell \tau_h$, BDT.(e) Canaux $\ell \ell$, DEEPTAU.(f) Canaux $\ell \ell$, BDT.

Figure X.28 – Écarts à basse masse du modèle B sur les différents types de canaux avec une quantité variable de fakes τ_h .

qui est difficile à obtenir pour $m_{\mathcal{H}} = 50 \text{ GeV}$. Ces événements sont donc peu nombreux et vraisemblablement très contaminés par les *fakes* τ_h .

Les *fakes* τ_h introduisent donc un biais important sur une large gamme de masse et en particulier dans la région des bosons Z ($m_Z = 91,2 \text{ GeV}$) et h ($m_h = 125,1 \text{ GeV}$). L'inclusion des *fakes* τ_h dans l'entraînement est non trivial, car la masse à prédire n'est pas définie, les *fakes* τ_h n'étant pas des objets physiques provenant de \mathcal{H} .

6.4 Effet de la séparation des canaux

Les modèles construits sont entraînés et testés sur l'ensemble des événements, sans sélection sur le canal. Or, il est possible d'entraîner un DNN par canal afin de le spécialiser à la phénoménologie associée et obtenir, potentiellement, de meilleures estimations de $m_{\mathcal{H}}$.

Les modèles notés B^x possèdent les mêmes hyper-paramètres que B mais sont entraînés uniquement sur les événements du canal x .

6.4.1 Séparation en six canaux

Les figures X.29 et X.30 donnent les réponses des modèles $B^{\tau_h \tau_h}$, $B^{\mu \tau_h}$, $B^{e \tau_h}$ et $B^{\mu \mu}$, $B^{e \mu}$, B^{ee} testés sur leurs canaux respectifs, comparées à celles de B sur les mêmes canaux.

Dans le canal $\tau_h \tau_h$, la réponse est 10 % plus basse pour $B^{\tau_h \tau_h}$ (figure X.29a) que pour B (figure X.29b) pour $m_{\mathcal{H}} > 100 \text{ GeV}$. À basse masse, le comportement des deux modèles est similaire : une baisse locale de la réponse est observable pour $m_{\mathcal{H}} \simeq 80 \text{ GeV}$. La coupure sur l'impulsion transverse des τ_h étant de 40 GeV pour chacun des deux τ_h , il s'agit probablement d'une transition entre les événements avec une majorité de vrais τ_h ($m_{\mathcal{H}} > 80 \text{ GeV}$) et ceux avec une contamination importante par les *fakes* τ_h ($m_{\mathcal{H}} < 80 \text{ GeV}$). Pour $B^{\tau_h \tau_h}$ et B , la résolution relative sur le canal $\tau_h \tau_h$ est de 20 %.

Le modèle $B^{\mu \tau_h}$, figure X.29c, possède une réponse équivalente à celle de B sur les mêmes événements, figure X.29d, pour des masses inférieures à 400 GeV. À haute masse, la réponse du modèle B est toutefois plus proche de 1. Le même constat peut être fait dans le cas du canal $e \tau_h$, figures X.29e et X.29f. La réponse de B est toutefois plus proche de 1 que celle de $B^{e \tau_h}$ sur toute la gamme de masse.

Dans le cas des canaux leptoniques, figure X.30, l'utilisation de B plutôt que $B^{\mu \mu}$, $B^{e \mu}$ ou B^{ee} selon le canal permet d'améliorer la résolution relative sur $m_{\mathcal{H}}$ dont les valeurs sont données dans le tableau X.2. Les valeurs des réponses moyennes sont peu modifiées par rapport aux valeurs des résolutions.

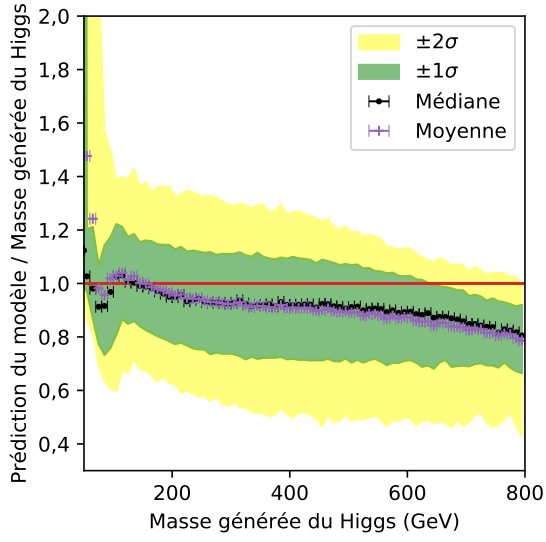
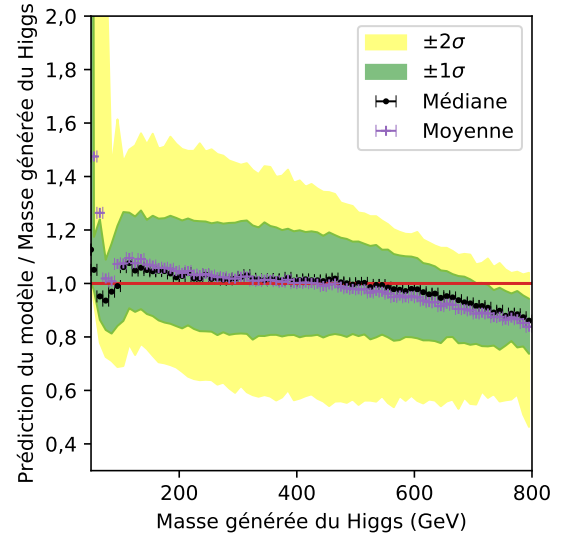
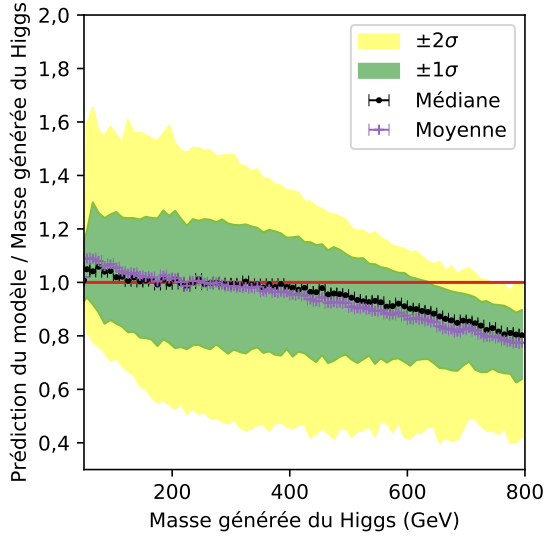
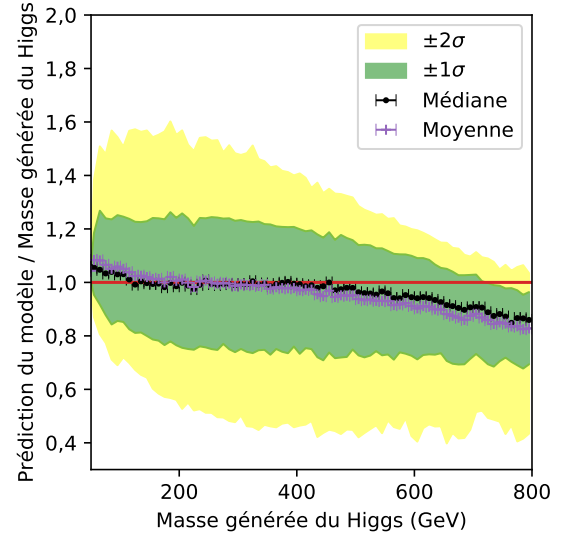
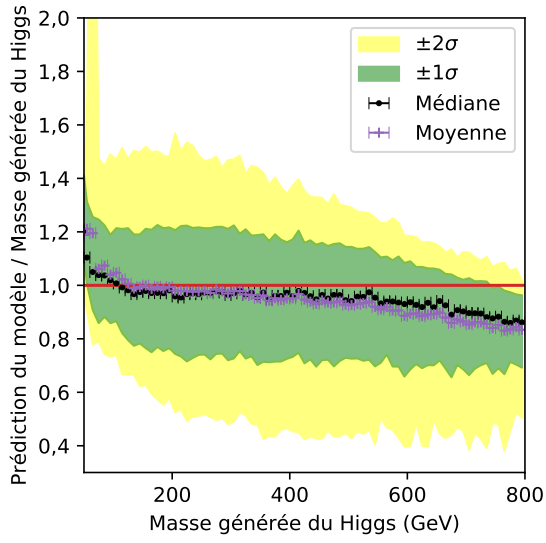
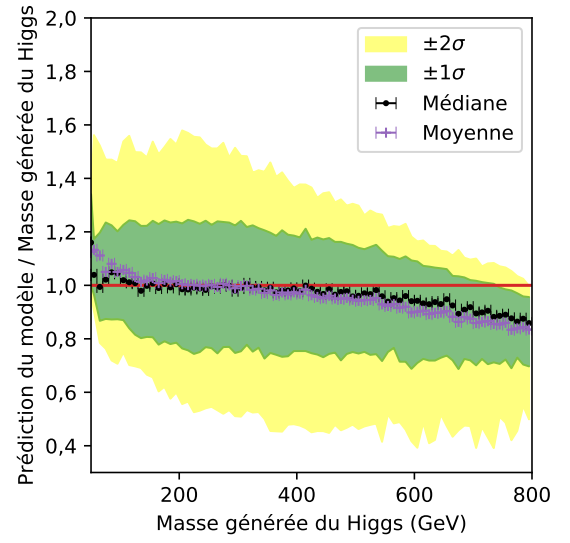
| Canal x | Modèle B^x | | Modèle B | |
|-----------|--------------|-----|------------|-----|
| | min | max | min | max |
| $\mu \mu$ | 20 | 50 | 10 | 40 |
| $e \mu$ | 20 | 40 | 20 | 30 |
| ee | 20 | 50 | 10 | 30 |

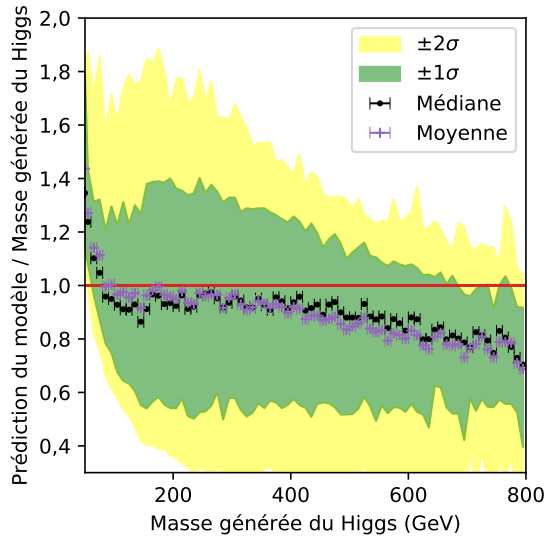
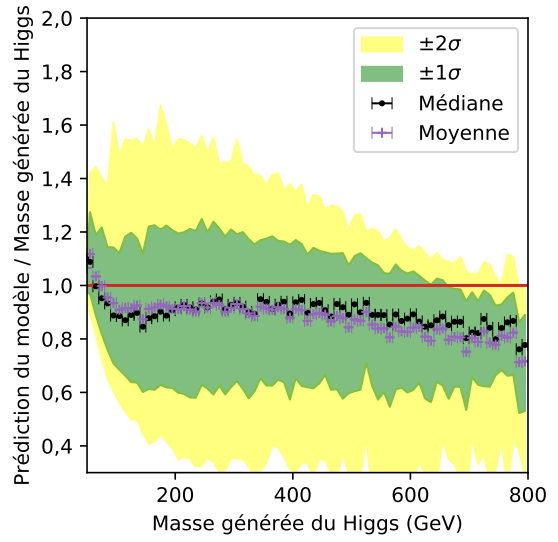
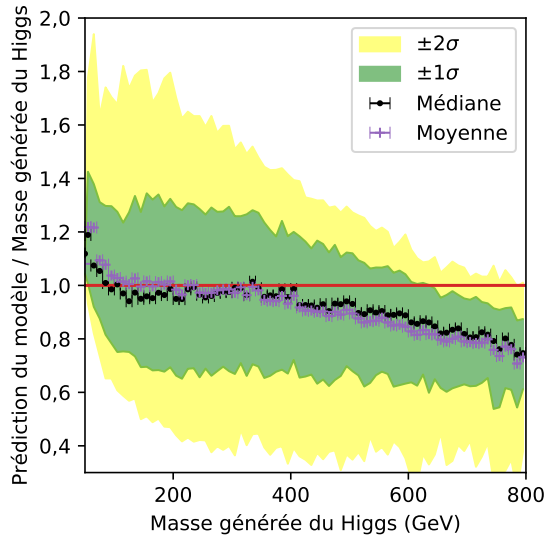
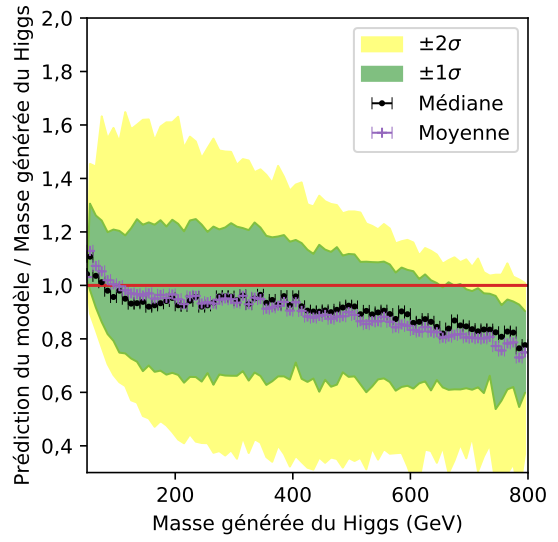
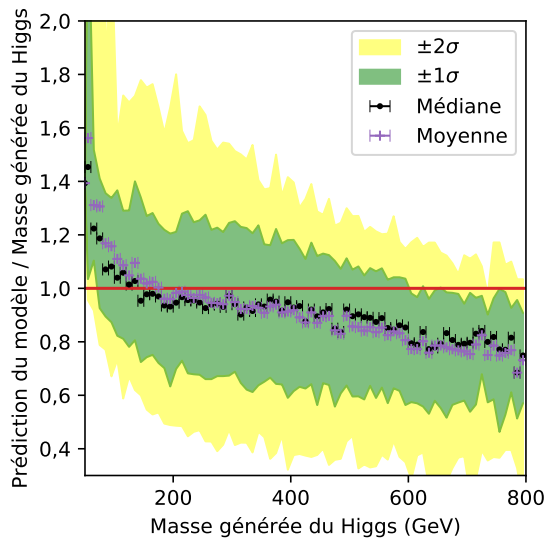
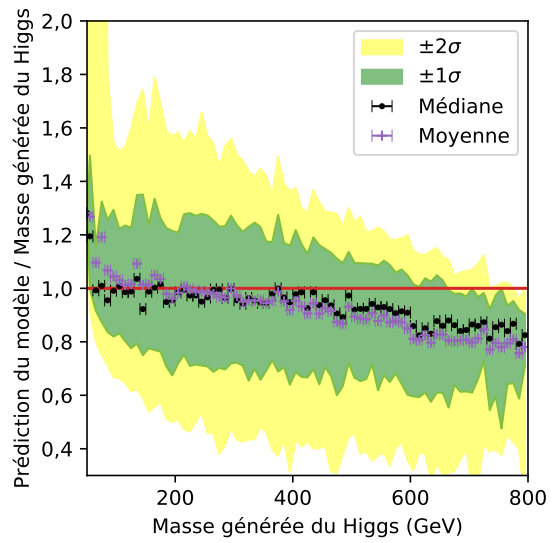
Tableau X.2 – Résolutions relatives minimale et maximale sur des intervalles de 10 GeV pour $B^{\mu \mu}$, $B^{e \mu}$ ou B^{ee} et B .

Il semble ainsi préférable d'utiliser un seul modèle global plutôt qu'un modèle par canal. Cet effet peut être dû à la statistique plus faible à disposition lors de l'entraînement des DNNs séparément pour chaque canal.

6.4.2 Séparation en trois groupes

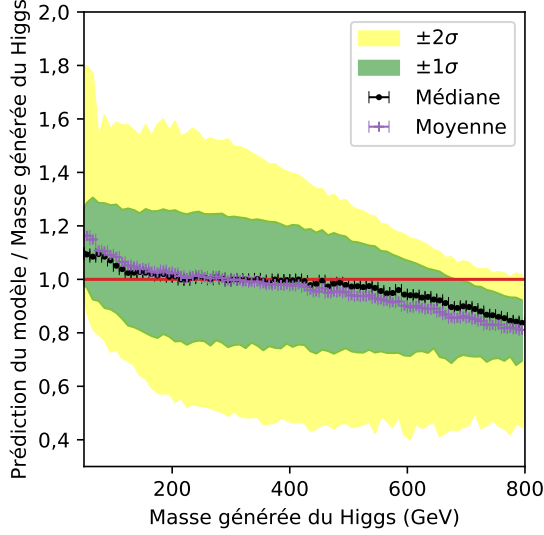
En dehors de toute considération de reconstruction des particules, la phénoménologie des canaux d'un même groupe est sensiblement la même. Au lieu de séparer les six canaux ($\tau_h \tau_h$, $\mu \tau_h$, $e \tau_h$, $\mu \mu$, $e \mu$, ee), il est possible de former trois groupes ($\tau_h \tau_h$, $\ell \tau_h$, $\ell \ell$), dans lesquels les quantités de τ_h et de neutrinos issus des désintégrations des leptons tau sont constantes. Cette nouvelle séparation permet ainsi d'avoir accès à de plus grandes quantités d'événements lors des entraînements, +100 % pour les canaux semi-leptoniques et +100 à +300 % pour les canaux leptoniques.

(a) Modèle $B^{T_h T_h}$ testé sur $\tau_h \tau_h$.(b) Modèle B testé sur $\tau_h \tau_h$.(c) Modèle $B^{\mu T_h}$ testé sur $\mu \tau_h$.(d) Modèle B testé sur $\mu \tau_h$.(e) Modèle $B^{e T_h}$ testé sur $e \tau_h$.(f) Modèle B testé sur $e \tau_h$.Figure X.29 – Comparaison des modèles entraînés par canal ($\tau_h \tau_h$, $\mu \tau_h$, $e \tau_h$) au modèle B .

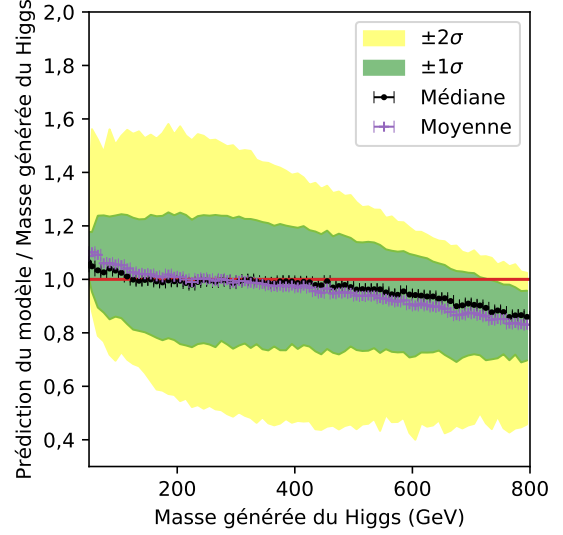
(a) Modèle $B^{\mu\mu}$ testé sur $\mu\mu$.(b) Modèle B testé sur $\mu\mu$.(c) Modèle $B^{e\mu}$ testé sur $e\mu$.(d) Modèle B testé sur $e\mu$.(e) Modèle B^{ee} testé sur ee .(f) Modèle B testé sur ee .**Figure X.30** – Comparaison des modèles entraînés par canal ($\mu\mu$, $e\mu$, ee) au modèle B .

Le canal $\tau_h \tau_h$, seul de son groupe, est ainsi déjà traité dans la section précédente.

La figure X.31 compare le modèle $B^{\ell\tau_h}$, entraîné sur les canaux semi-leptoniques, à B utilisé sur ces mêmes événements. Pour des masses supérieures à 300 GeV, les deux modèles sont équivalents en termes de réponse et de résolution relative. En revanche, à basse masse, le modèle B a une réponse de 1,06 contre 1,10 pour $B^{\ell\tau_h}$.



(a) Modèle $B^{\ell\tau_h}$ testé sur $\ell\tau_h$.

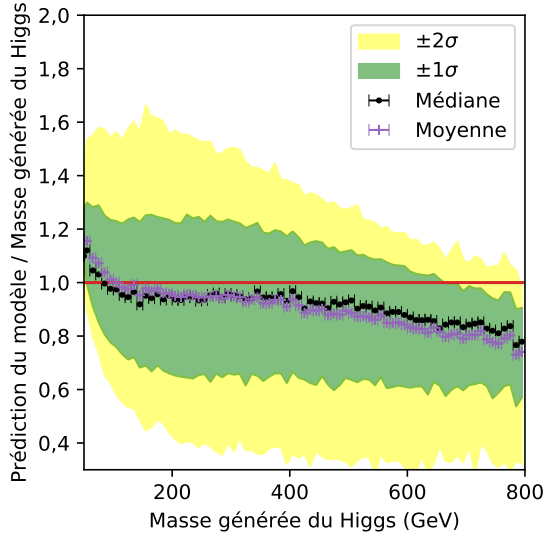


(b) Modèle B testé sur $\ell\tau_h$.

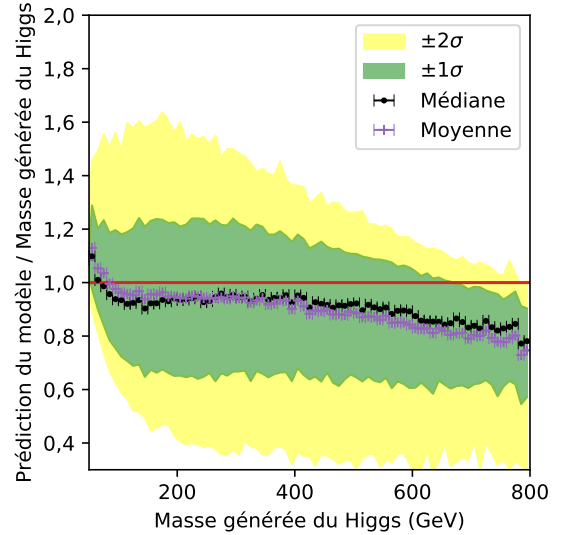
Figure X.31 – Comparaison de $B^{\ell\tau_h}$ à B.

La figure X.32 compare le modèle $B^{\ell\ell}$, entraîné sur les canaux leptoniques, à B utilisé sur ces mêmes événements.

continue here



(a) Modèle $B^{\ell\ell}$ testé sur $\ell\ell$.



(b) Modèle B testé sur $\ell\ell$.

Figure X.32 – Comparaison de $B^{\ell\ell}$ à B.

6.5 Effet de la définition de E_T^{miss}

6.6 Effet de l'intervalle de masse

6.6.1 Gamme de masse

6.6.2 Effet de bord

use the custom loss with boundaries cuts (basically all the report 2021-02-04)

Follow report from 2021-02-04 but for section 3 : We saw that predictions come out too low, which already is a motivation to put larger weights on higher masses, i.e. to weight by truth. Choosing $\text{sqrt}(\text{truth})$ is of course just a guess then

extend up to 1TeV using the tails

6.7 Modèle final

DEEPTAU
1 TeV
all inputs
activation softplus
loss mapesqrt_b
opti Adam
glorot uniform
3 layers of 1000 neurons
show reponses and 2d histo

7 Utilisation du modèle dans les analyses CMS


show distributions of $m_{T\text{tot}}$ and ML predictions

discuss

show limits

discuss

8 Conclusion

 To cite :

- DELPHES 3.4.2 [19, 20]?
- CMS Fast Simulation (FASTSIM) [21-24]
- PYTHIA 8.235 [29]
- FASTJET [45, 46]
- KERAS [47]
- TENSORFLOW [48]
- XGBOOST [33]
- W. SARLE. « Neural Networks and Statistical Models ». 1994. URL : https://people.orie.cornell.edu/davidr/or474/nn_sas.pdf
- P. BÄRTSCHI & coll. « Reconstruction of τ lepton pair invariant mass using an artificial neural network ». Nuclear Instruments and Methods in Physics Research **A929** (2019), p. 29-33. DOI : [10.1016/j.nima.2019.03.029](https://doi.org/10.1016/j.nima.2019.03.029). URL : <http://www.sciencedirect.com/science/article/pii/S0168900219303377>
- SVFIT [17]

L. TORTEROTOT, E. AŞILAR & C. BERNET. *Reconstruction of di-tau mass using Machine Learning*. URL : https://github.com/lucastorterotot/DL_for_HTT_mass

Références

- [1] DEEPMIND. *AlphaGo*. URL : <https://www.deepmind.com/research/case-studies/alphago-the-story-so-far>.

- [2] C. BERNET. *The Data Frog – Image Recognition : Dogs vs Cats !* URL : <https://thedatafrog.com/en/articles/dogs-vs-cats/>.
- [3] M. MIR. *House Prices Prediction Using Deep Learning*. URL : <https://towardsdatascience.com/house-prices-prediction-using-deep-learning-dea265cc3154>.
- [4] G. TOUQUET. « Search for an additional neutral MSSM Higgs boson decaying to tau leptons with the CMS experiment ». Thèse de doct. Université Claude Bernard Lyon 1, oct. 2019. URL : <https://hal.archives-ouvertes.fr/tel-02526393>.
- [5] M. SCHAM. « Standard Model $H \rightarrow \tau\tau$ Analysis with a Neural Network Trained on a Mix of Simulation and Data Samples ». Mém. de mast. Fakultät für Physik des Karlsruher Instituts für Technologie (KIT), juin 2020. URL : <https://publish.etp.kit.edu/record/21993>.
- [6] T. KOPF. « Recoil Calibration as a Neural Network Task ». Mém. de mast. Fakultät für Physik des Karlsruher Instituts für Technologie (KIT), fév. 2019. URL : <https://publish.etp.kit.edu/record/21500>.
- [7] P. BALDI, P. SADOWSKI & D. WHITESON. « Enhanced Higgs Boson to $\tau^+\tau^-$ Search with Deep Learning ». *Physical Review Letters* **114**.11 (mar. 2015). DOI : [10.1103/physrevlett.114.111801](https://doi.org/10.1103/physrevlett.114.111801).
- [8] D. GUEST & coll. « Jet flavor classification in high-energy physics with deep neural networks ». *Physical Review* **D94**.11 (déc. 2016). DOI : [10.1103/physrevd.94.112002](https://doi.org/10.1103/physrevd.94.112002).
- [9] The CMS Collaboration. « Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV ». *Journal of Instrumentation* **13**.05 (mai 2018). DOI : [10.1088/1748-0221/13/05/p05011](https://doi.org/10.1088/1748-0221/13/05/p05011).
- [10] The CMS Collaboration. *DeepJet : deep learning based on physics objects for jet reconstruction*. URL : <https://twiki.cern.ch/twiki/bin/viewauth/CMS/DeepFlavour>.
- [11] The CMS Collaboration. « Performance of the DeepTau algorithm for the discrimination of taus against jets, electron, and muons » (oct. 2019). URL : <https://cds.cern.ch/record/2694158>.
- [12] J. ANDREJKOVIC & coll. « Measurement of Higgs(125) boson properties in decays to a pair of tau leptons with full Run II data using Machine-Learning techniques ». *CMS analysis Note* (sept. 2020). URL : https://cms.cern.ch/iCMS/jsp/db_notes/noteInfo.jsp?cmsnoteid=CMS%5C%20AN-2019/177.
- [13] J. ANDREJKOVIC & coll. « Multi-class neural network architecture and training for measurements of Higgs(125) boson decays to two tau leptons on full Run II data ». *CMS analysis Note* (mai 2020). URL : https://cms.cern.ch/iCMS/jsp/db_notes/noteInfo.jsp?cmsnoteid=CMS%5C%20AN-2019/178.
- [14] A. ELAGIN & coll. « A new mass reconstruction technique for resonances decaying to $\tau\tau$ ». *Nuclear Instruments and Methods in Physics Research* **A654**.1 (2011), p. 481-489. DOI : [10.1016/j.nima.2011.07.009](https://doi.org/10.1016/j.nima.2011.07.009).
- [15] A. J. BARR & coll. « Speedy Higgs boson discovery in decays to tau lepton pairs : $h \rightarrow \tau\tau$ ». *Journal of High Energy Physics* **2011**.10 (oct. 2011). DOI : [10.1007/JHEP10\(2011\)080](https://doi.org/10.1007/JHEP10(2011)080).
- [16] B. GRIPAIOS & coll. « Reconstruction of Higgs bosons in the di-tau channel via 3-prong decay ». *Journal of High Energy Physics* **2013**.3 (mar. 2013). DOI : [10.1007/JHEP03\(2013\)106](https://doi.org/10.1007/JHEP03(2013)106).
- [17] L. BIANCHINI & coll. « Reconstruction of the Higgs mass in $H \rightarrow \tau\tau$ Events by Dynamical Likelihood techniques ». *Journal of Physics : Conference Series* **513**.2 (juin 2014). DOI : [10.1088/1742-6596/513/2/022035](https://doi.org/10.1088/1742-6596/513/2/022035).
- [18] P. BÄRTSCHI & coll. « Reconstruction of τ lepton pair invariant mass using an artificial neural network ». *Nuclear Instruments and Methods in Physics Research* **A929** (2019), p. 29-33. DOI : [10.1016/j.nima.2019.03.029](https://doi.org/10.1016/j.nima.2019.03.029). URL : <http://www.sciencedirect.com/science/article/pii/S0168900219303377>.
- [19] J. de FAVEREAU & coll. « DELPHES 3 : a modular framework for fast simulation of a generic collider experiment ». *Journal of High Energy Physics* **2** (fév. 2014). DOI : [10.1007/jhep02\(2014\)057](https://doi.org/10.1007/jhep02(2014)057).

- [20] A. MERTENS. « New features in DELPHES 3 ». *Journal of Physics : Conference Series* **608.1** (2015). Sous la dir. de L. FIALA, M. LOKAJICEK & N. TUMOVA. DOI : [10.1088/1742-6596/608/1/012045](https://doi.org/10.1088/1742-6596/608/1/012045).
- [21] S. ABDULLIN & coll. « The Fast Simulation of the CMS Detector at LHC ». *Journal of Physics : Conference Series* **331.3** (déc. 2011). DOI : [10.1088/1742-6596/331/3/032049](https://doi.org/10.1088/1742-6596/331/3/032049).
- [22] A. GIAMMANCO. « The Fast Simulation of the CMS Experiment ». *Journal of Physics : Conference Series* **513.2** (juin 2014). DOI : [10.1088/1742-6596/513/2/022012](https://doi.org/10.1088/1742-6596/513/2/022012).
- [23] M. KOMM. « Fast emulation of track reconstruction in the CMS simulation ». *Journal of Physics : Conference Series* **898** (oct. 2017). DOI : [10.1088/1742-6596/898/4/042034](https://doi.org/10.1088/1742-6596/898/4/042034).
- [24] S. SEKMEN. *Recent Developments in CMS Fast Simulation*. 2017. arXiv : [1701.03850](https://arxiv.org/abs/1701.03850).
- [25] S. AGOSTINELLI & coll. « GEANT4 – A simulation toolkit ». *Nuclear Instruments and Methods in Physics Research* **A506.3** (2003), p. 250-303. DOI : [10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8). URL : <http://www.sciencedirect.com/science/article/pii/S0168900203013688>.
- [26] J. ALLISON & coll. « GEANT4 developments and applications ». *IEEE Transactions on Nuclear Science* **53.1** (fév. 2006), p. 270-278. DOI : [10.1109/tns.2006.869826](https://doi.org/10.1109/tns.2006.869826).
- [27] J. ALLISON & coll. « Recent developments in GEANT4 ». *Nuclear Instruments and Methods in Physics Research* **A835** (2016), p. 186-225. DOI : [10.1016/j.nima.2016.06.125](https://doi.org/10.1016/j.nima.2016.06.125). URL : <http://www.sciencedirect.com/science/article/pii/S0168900216306957>.
- [28] E. AŞILAR. *How to produce nanoAOD events of $h \rightarrow \tau\tau$ where Higgs has a 130 GeV mass*. URL : https://github.com/easilar/cmssw/blob/from-CMSSW_10_2_22/README.
- [29] T. SJÖSTRAND & coll. « An Introduction to PYTHIA 8.2 ». *Computer Physics Communications* **191** (2015), p. 159-177. DOI : [10.1016/j.cpc.2015.01.024](https://doi.org/10.1016/j.cpc.2015.01.024). arXiv : [1410.3012](https://arxiv.org/abs/1410.3012) [hep-ph].
- [30] The CMS Collaboration. « Event generator tunes obtained from underlying event and multiparton scattering measurements ». *European Physical Journal* **C76.3** (2016). DOI : [10.1140/epjc/s10052-016-3988-x](https://doi.org/10.1140/epjc/s10052-016-3988-x). arXiv : [1512.00815](https://arxiv.org/abs/1512.00815) [hep-ex].
- [31] The CMS Collaboration. « Extraction and validation of a new set of CMS PYTHIA 8 tunes from underlying-event measurements ». *European Physical Journal* **C80** (mar. 2019). DOI : [10.1140/epjc/s10052-019-7499-4](https://doi.org/10.1140/epjc/s10052-019-7499-4). URL : <https://cds.cern.ch/record/2669320>.
- [32] LHC Higgs Cross Section Working Group. « Higgs Properties ». *Handbook of LHC Higgs Cross Sections. 3*. CERN Yellow Reports : Monographs. Geneva : CERN, 2013. DOI : [10.5170/CERN-2013-004](https://doi.org/10.5170/CERN-2013-004). URL : <https://cds.cern.ch/record/1559921>.
- [33] T. CHEN & C. GUESTIN. « XGBOOST : A Scalable Tree Boosting System ». *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (août 2016). DOI : [10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785).
- [34] *Kaggle Competitions*. URL : <https://www.kaggle.com/competitions>.
- [35] I. GOODFELLOW, Y. BENGIO & A. COURVILLE. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [36] X. GLOROT & Y. BENGIO. « Understanding the difficulty of training deep feedforward neural networks ». *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Sous la dir. d'Y. W. TEH & M. TITTERINGTON. **9**. Proceedings of Machine Learning Research. PMLR, mai 2010, p. 249-256. URL : <http://proceedings.mlr.press/v9/glorot10a.html>.
- [37] A. CAUCHY. « Méthode générale pour la résolution des systèmes d'équations simultanées ». *Comptes rendus hebdomadaires des séances de l'Académie des Sciences*. **25**. 1847, p. 536-538. URL : <https://gallica.bnf.fr/ark:/12148/bpt6k2982c/f3.item>.
- [38] L. BOTTOU. « Online Algorithms and Stochastic Approximations ». *Online Learning and Neural Networks*. Sous la dir. de D. SAAD. Cambridge, UK : Cambridge University Press, 1998. URL : <http://leon.bottou.org/papers/bottou-98x>.

- [39] J. DUCHI, E. HAZAN & Y. SINGER. « Adaptive Subgradient Methods for Online Learning and Stochastic Optimization ». *Journal of Machine Learning Research* **12.61** (2011), p. 2121-2159. URL : <http://jmlr.org/papers/v12/duchi11a.html>.
- [40] G. HINTON. *Neural Networks for Machine Learning*. Coursera Video Lectures, Academic Torrents. 2012. URL : [https://archive.org/search.php?query=creator%3A%22Geoffrey+Hinton%22&and\[\]=year%3A%222012%22](https://archive.org/search.php?query=creator%3A%22Geoffrey+Hinton%22&and[]=year%3A%222012%22).
- [41] D. P. KINGMA & J. BA. « Adam : A Method for Stochastic Optimization » (2017). arXiv : [1412.6980](https://arxiv.org/abs/1412.6980) [cs.LG].
- [42] J. ANDREJKOVIC & coll. « Data-driven background estimation of fake-tau backgrounds in di-tau final states with 2016 and 2017 data ». *CMS analysis Note* (oct. 2018).
- [43] J. ANDREJKOVIC & J. BECHTEL. « Data-driven background estimation of fake-tau backgrounds in di-tau final states with the full Run-II dataset ». *CMS analysis Note* (juin 2020). URL : https://cms.cern.ch/iCMS/jsp/db_notes/noteInfo.jsp?cmsnoteid=CMS%5C%20AN-2019/170.
- [44] The CMS Collaboration. « Reconstruction and identification of tau lepton decays to hadrons and tau neutrino at CMS ». *Journal of Instrumentation* **11.1** (2016). DOI : [10.1088/1748-0221/11/01/P01019](https://doi.org/10.1088/1748-0221/11/01/P01019). arXiv : [1510.07488](https://arxiv.org/abs/1510.07488) [physics.ins-det].
- [45] M. CACCIARI, G. P. SALAM & G. SOYEZ. « FASTJET user manual ». *European Physical Journal* **C72** (nov. 2012). DOI : [10.1140/epjc/s10052-012-1896-2](https://doi.org/10.1140/epjc/s10052-012-1896-2). arXiv : [1111.6097](https://arxiv.org/abs/1111.6097) [hep-ph].
- [46] M. CACCIARI & G. P. SALAM. « Dispelling the N^3 myth for the k_T jet-finder ». *Physics Letters* **B641.1** (sept. 2006), p. 57-61. DOI : [10.1016/j.physletb.2006.08.037](https://doi.org/10.1016/j.physletb.2006.08.037).
- [47] F. CHOLLET & coll. KERAS. <https://keras.io>. 2015.
- [48] M. ABADI & coll. TENSORFLOW : *Large-scale machine learning on heterogeneous distributed systems*. Software available from tensorflow.org. 2015. URL : <https://www.tensorflow.org/>.
- [49] W. SARLE. « Neural Networks and Statistical Models ». 1994. URL : https://people.orie.cornell.edu/davidr/or474/nn_sas.pdf.
- [50] L. TORTEROTOT, E. AŞILAR & C. BERNET. *Reconstruction of di-tau mass using Machine Learning*. URL : https://github.com/lucastorterotot/DL_for_HTT_mass.

Table des matières

| | | |
|----------|--|----------|
| X | Reconstruction de la masse d'une résonance grâce au <i>Machine Learning</i> | 1 |
| 1 | Introduction | 1 |
| 2 | Événements utilisés | 2 |
| 2.1 | Génération avec FASTSIM | 3 |
| 2.2 | Sélection des événements | 3 |
| 2.3 | Événements obtenus et pondération | 5 |
| 2.4 | Cible et variables d'entrée des modèles | 6 |
| 3 | Arbres de décision améliorés | 7 |
| 3.1 | Arbres de décision | 7 |
| 3.2 | <i>Gradient Boosting</i> | 8 |
| 3.3 | Fonction de coût et <i>Gradient Descent</i> | 8 |
| 3.4 | Sous-entraînement et surentraînement | 9 |
| 4 | Réseaux de neurones profonds | 9 |
| 4.1 | Neurones | 10 |
| 4.2 | Réseaux de neurones | 12 |
| 4.3 | Entraînement | 12 |
| 5 | Optimisation des hyper-paramètres et choix d'un modèle | 15 |
| 5.1 | Variables d'entrée | 17 |
| 5.2 | Type de modèle | 18 |
| 5.3 | Fonction de coût | 20 |
| 5.4 | Algorithme d'optimisation | 21 |
| 5.5 | Autres hyper-paramètres | 22 |
| 6 | Discussions | 27 |
| 6.1 | Effet de l'empilement | 29 |
| 6.2 | Effet de la reconstruction des particules | 30 |
| 6.3 | Effet des faux taus hadroniques | 31 |
| 6.4 | Effet de la séparation des canaux | 34 |
| 6.5 | Effet de la définition de E_T^{miss} | 37 |
| 6.6 | Effet de l'intervalle de masse | 37 |
| 6.7 | Modèle final | 38 |
| 7 | Utilisation du modèle dans les analyses CMS | 38 |
| 8 | Conclusion | 38 |

