Méthodes de prédiction des flux énergétiques

2.1 Introduction

Notre outil doit être capable de fournir des prédictions et des classifications de chaque flux énergétique. L'objectif final est le pilotage des flux énergétiques et la stabilisation du réseau électrique au niveau local. Cela demande une sureté dans la qualité de la prédiction des différents flux énergétiques d'autant plus au niveau d'un système tel qu'une maison

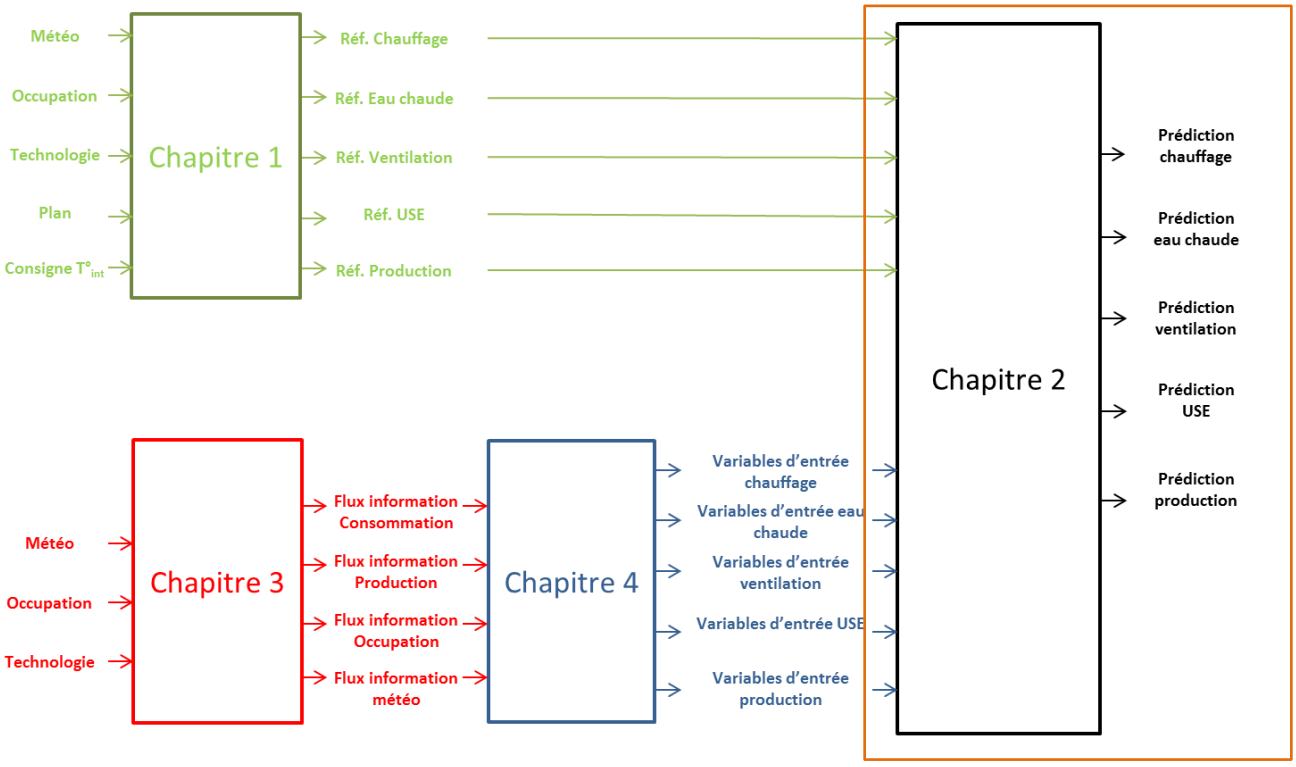

Figure 26 : Aide pour la lecture du document associée aux différents chapitres et au processus générale d'aide à la décision

Comme décrit dans le chapitre 3 pour la définition des acteurs, dans certains cas, l'objectif n'est pas d'avoir seulement une information la plus précise en termes de prédictions énergétiques mais d'avoir par exemple un ensemble de charges à éteindre à un instant donné.

Les différents flux énergétiques seront caractérisés par des classes représentées par une gaussienne qu'il s'agira de prédire pour l'heure suivante.

Par exemple, deux classes allumées et éteintes suffiraient à caractériser l'eau chaude et le chauffage pour l'heure suivante ou encore une caractérisation précise du pic de puissance liée à la consommation électrique si l'entreprise connectée est soumise à ce système de facturation. Une étude spécifique du pic de puissance lié à la production permettrait également de prévenir les moments les moins fréquents mais les plus importants pour la stabilisation du micro-réseau.

Si la classe est connue au préalable et que l'opération de classement consiste à analyser les caractéristiques des individus pour les placer dans une classe, la méthode est dite «supervisée». Dans le cas contraire, la méthode est «non-supervisée», ce vocabulaire étant issu de l'apprentissage automatique tiré des modèles non linéaires.

2.2 Apprendre du passé pour prédire l'avenir

La consommation d'électricité est historiquement enregistrée à l'aide de compteurs électromécaniques à simple ou double tarif. La lecture des compteurs s'effectue chez le consommateur, une ou deux fois par année.

Aujourd'hui, les compteurs numériques sont capables de relever la consommation, la production ainsi que tout paramètre lié à l'état du réseau à chaque seconde près. En France, 35000 compteurs intelligents seront installés d'ici 2021.

En Suisse romande, le projet Green E-Value **[GRE-2014]** propose un portail accessible aux propriétaires des immeubles et aux locataires qui leur fournit les informations de consommation et propose des conseils d'économie d'énergie. Dans les locaux communs, des écrans indiquent les consommations moyennes des logements et permettent aux locataires de s'autoévaluer et d'améliorer leur comportement.

Ainsi les données relatives à la consommation et à la production sont ou seront collectées dans quelques années sur des millions de logements. Associées à des bases de données météorologiques déjà existantes depuis des dizaines d'années comme par Météo France ou Météo Suisse, il sera possible de créer des bases de données conséquentes relatives à un système comme une maison. Ces bases de données créées dans le cadre de cette thèse sont développées dans le chapitre 3. Cela permet à partir d'extraction de connaissances développées dans le chapitre 4 de modéliser sur plusieurs mois ou sur plusieurs années le comportement énergétique d'un système. Dans notre cas, une modélisation est propre à chaque flux et un modèle spécifique est créé pour modéliser le besoin en chauffage, en eau chaude sanitaire, en USE et la production solaire.

Pour chacun de ces flux, une base de données est créée et une partie de ces données est utilisée pour **entraîner un modèle** linéaire ou non linéaire. Une autre partie de ces données est utilisée pour **tester le modèle** entraîné. Il est possible également de rapprendre les modèles au bout d'une période prédéfinie.

2.3 Méthodes linéaires

L'une des méthodes les plus classiques est une **régression linéaire.** Avec y la variable aléatoire réelle à expliquer (variable endogène, dépendante ou réponse) et x la variable explicative ou effet fixe (exogène). Elle est toujours utilisée pour des prédictions de la consommation d'énergie totale **[ZAM-2013]**.

Nous avons :

$$y = f(x) = \beta_0 + \beta_1 x + \varepsilon \qquad \textbf{(2.1)}$$

En statistique, les modèles ARMA (modèles autorégressifs et moyenne mobile), ou aussi modèles de Box-Jenkins, sont les principaux modèles de séries temporelles. Les modèles ARIMA permettent de traiter les séries non stationnaires après avoir déterminé le niveau d'intégration. Le modèle ARIMA ou ARMA souffre d'une lacune majeure puisqu'il est incapable de traiter plusieurs variables.

Pour contourner ce problème, il faut pouvoir généraliser le modèle ARIMA dans le cas à plusieurs variables décrit dans **[SIM-1980]** nommé VAR pour (Vector Auto Regressive). Mais, contrairement au modèle structurel à plusieurs variables, dans les modèles VAR, toutes les variables sont endogènes. Ces modèles (ARIMA et VAR) ne permettent de traiter que des phénomènes qui sont linéaires ou approximativement (par exemple le PIB) mais ne permettent pas d'identifier les propriétés des phénomènes qui sont non linéaires.

2.4 Méthodes non linéaires

2.4.1 Réseau de neurones

Un réseau de neurones artificiels est un ensemble d'algorithmes dont la conception est à l'origine très schématiquement inspirée du fonctionnement des neurones biologiques. Nous retrouvons plusieurs types de réseau de neurones qui ont chacun leurs spécificités, avantages et inconvénients.

Un réseau de neurones est en général composé d'une succession de couches dont chacune prend ses entrées sur les sorties de la précédente. À chaque synapse est associé un poids synaptique, de sorte que les neurones de la couche précédente sont multipliés par ce poids, puis additionnés par les neurones de niveau i. Le neurone reçoit des neurones en amont un certain nombre de valeurs via ses connexions synaptiques, et il produit une certaine valeur en utilisant une fonction d'activation appelée aussi fonction de combinaison. Cette fonction peut être formalisée comme étant une fonction vecteur-à-scalaire, notamment :

- ⇨ Les réseaux de type **MLP** (multi-layer perceptron) calculent une combinaison linéaire des entrées, c'est-à-dire que la fonction de combinaison renvoie le produit scalaire entre le vecteur des entrées et le vecteur des poids synaptiques.

⇨ Les réseaux de type **RBF** (radial basis function) calculent la distance entre les entrées, c'est-à-dire que la fonction de combinaison renvoie la norme euclidienne du vecteur issu de la différence vectorielle entre les vecteurs d'entrées.

Nous pouvons citer les plus utilisés dans le domaine de la prédiction énergétique comme la famille des ANN (Artificial Neural Network) avec l'un des plus polaires, le **MLP (Multi Layer Perceptron) [BER-2010]**.

Dans le **MLP**, les neurones d'une couche sont reliés à la totalité des neurones des couches adjacentes. Ces liaisons sont soumises à un coefficient altérant l'effet de l'information sur le neurone de destination (Figure 27). Le poids de chacune de ces liaisons est l'élément clef du fonctionnement du réseau : la mise en place d'un Perceptron multicouche pour résoudre un problème passe donc par la détermination des meilleurs poids applicables à chacune des connexions inter-neuronales **[ONO-2000] [BER-2010]**.

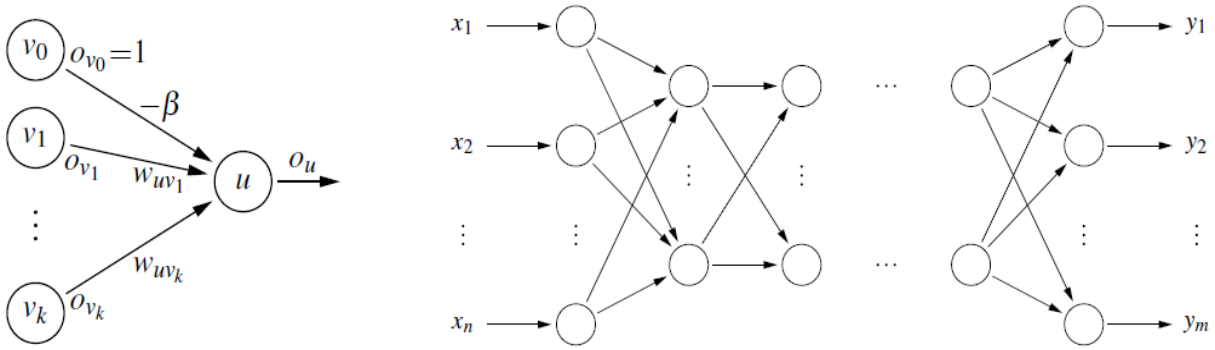

Figure 27 : Représentation d'un réseau de neurones [BER-2010]

Le nombre de couches, le nombre de neurones dans chaque couche, ainsi que d'autres paramètres, déterminent ce que l'on nomme la « structure » du réseau qui définissent nos paramètres de configuration :

⇨ **Le nombre de couches cachées**

⇨ **Le nombre de neurones par couche cachée**

⇨ **Le nombre maximum d'itérations pour l'apprentissage**

Au sein de cette structure, chaque neurone est connecté à l'ensemble des neurones des deux couches voisines (supérieure et inférieure). Ces connections entre neurones sont dotées de poids qui sont déterminés lors de la phase d'entraînement, phase dans laquelle nous présentons au réseau des données d'entrée, telles que des statistiques sur la consommation électrique, des données météorologiques, météos, ainsi que la sortie désirée, telle qu'une valeur de prédiction horaire. Une fois la phase d'entraînement terminée et les poids déterminés, le réseau de neurones est prêt pour la phase de prédiction basée sur les données de test.

Dans la famille des réseaux de neurones, nous trouvons également le PNN (Perceptron Neural Network) ou encore le SVM (Support Vector Machine).

Concernant les PNN, cet algorithme génère des règles basées sur des données numériques. Chaque règle est définie comme une fonction gaussienne de grande dimension qui est ajustée par deux seuils, **theta moins et theta plus**, qui décrivent l'aire de conflit (Figure 28). Chaque fonction gaussienne est définie par un vecteur central (à partir de la première instance couverte) et un écart type qui est ajusté pendant la formation pour couvrir uniquement les instances non conflictuelles.

L'une des forces majeurs des réseaux de neurones est la technique de **back propagation** qui est fondamentale pour l'entraînement et est le secret du « deep learning » : depuis la couche de sortie, l'erreur d'estimation qui est faîte durant l'entraînement est propagée en arrière dans chaque couche de neurone et corrige les poids des connections pour minimiser l'erreur.

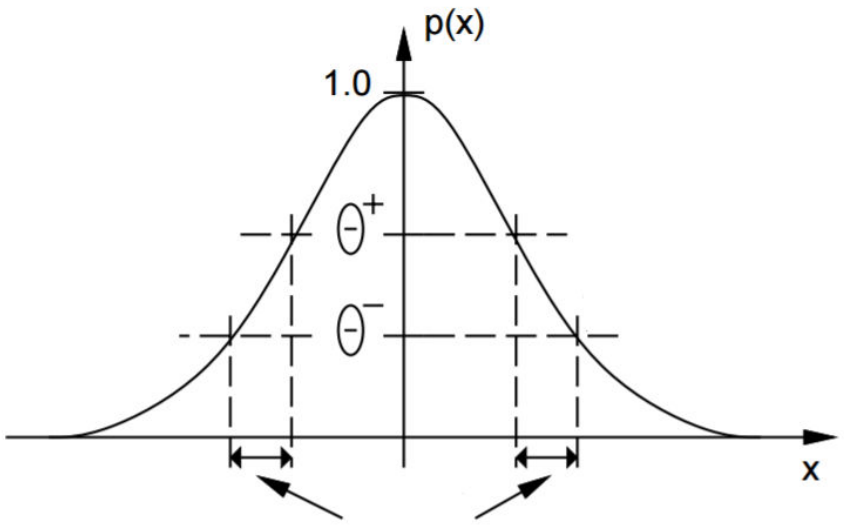

Figure 28: Paramètres de configuration de l'algorithme PNN qui correspondent à la délimitation de la zone de conflit [BER-2010]

Les Support Vector Machine nommé **SVM** sont des classificateurs qui permettent de traiter des problèmes de discrimination non linéaire **[BER-2010]**. Le cas simple est le cas d'une fonction discriminante linéaire, obtenue par combinaison linéaire du vecteur d'entrée $x = (x_1, \ldots\ldots, x_n)^T$, avec un vecteur de poids $w = (w_1, \ldots\ldots, w_n)^T$

$$h(x) = w^T x + w_0 \qquad \textbf{(2.2)}$$

Il est introduit dans les SVM la notion de marge maximale qui est la distance entre la frontière de séparation et les échantillons les plus proches (Figure 29). Ces derniers sont appelés des vecteurs supports. La frontière de séparation est choisie comme celle qui maximise la marge.

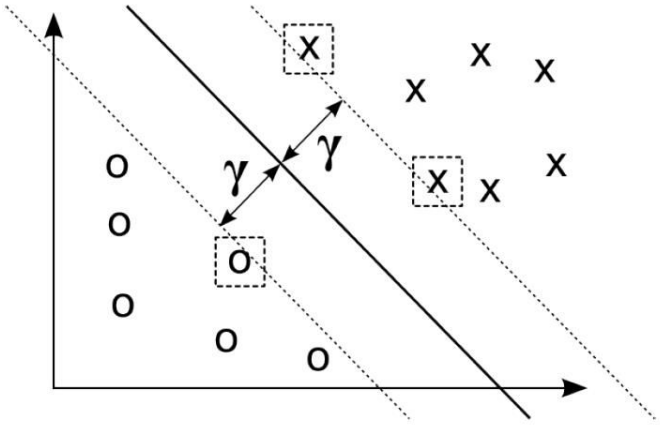

Figure 29 : Représentation de la notion de marge ϒ dans le cas de séparation de deux classes [BER-2010]

Pour résoudre des problèmes non linéaires, l'idée des SVM est de reconsidérer le problème dans un espace de dimension supérieure, éventuellement de dimension infinie. Il est appliqué aux vecteurs d'entrée une transformation non-linéaire appelé fonction noyau. Plusieurs **fonctions noyaux** sont intégrées dans des outils de prédictions comme les **RBF, Hyper tangentes ou polynomiales** décrites dans **[BER-2010]**.

La fonction noyau RBF pour Radial Basic Fonction a été retenue pour l'ensemble de nos classifications qui nous a fourni nos meilleurs résultats en test.

2.4.2 Les arbres de décision

Les arbres de décision sont l'une des structures de données majeures de l'apprentissage statistique. Leur structure arborescente les rend **lisibles par un être humain**, contrairement à d'autres approches où le prédicteur construit est une « boîte noire » comme les réseaux de neurones. Cette faculté de lecture est possible puisqu'un arbre de décision fait une sélection explicite des paramètres qu'il utilise en exploitant un critère de quantité d'information **[CAR-2010]**.

Il modélise une hiérarchie de tests sur les valeurs d'un ensemble de variables appelées attributs. À l'issue de ces tests, le prédicteur produit une valeur numérique ou choisit un élément dans un ensemble discret de conclusions. On parle de régression dans le premier cas et de classification dans le second **[BRE-2001]**.

L'apprentissage d'un arbre de décision se fait sur un ensemble d'instances T = {(**x**, y)} appelé « ensemble d'entraînement ». Un ensemble de valeurs pour les différents attributs est appelé une « instance ».

Si toutes ou la majorité des instances ont la même classe $c \in \{1, \ldots, C\}$, c apparaît comme la meilleure prédiction possible par vote majoritaire. Un arbre de classification est d'autant meilleur que les instances de ses feuilles sont de classes homogènes. Une mesure de l'hétérogénéité d'une feuille est

calculée selon différentes méthodes comme le critère de Gini **[BER-2010]**. Il s'agit du taux d'erreur sur l'ensemble d'entraînement qui retourne la classe c avec une probabilité p_c (au lieu de toujours retourner la classe c^*). La mesure d'erreur est utilisée de même pour choisir le meilleur attribut de partage. De manière heuristique, l'attribut qui minimise cette erreur est considéré comme le meilleur choix possible.

Un arbre de décision est composé de nœuds de décision, qui permettent de tester les attributs. Un nœud de décision est étiqueté par un test qui peut être appliqué à chaque description d'un individu ou d'une population. Il est composé de branches qui représentent chacune une valeur de l'attribut testé. Il peut également être constitué de feuilles, c'est-à-dire les nœuds terminaux, qui indiquent la classe résultante. Les paramètres de configuration d'un arbre de décision sont :

- ⇨ **Critère de fractionnement (exemple : indice de Gini)**
- ⇨ **Nombre limite de niveaux (profondeur de l'arbre)**
- ⇨ **Taille minimale du noeud**

À toute description est associée une seule feuille de l'arbre de décision. Les arbres de décision tels qu'implémentés dans notre outil sont décrits dans **[SHA-2010].**

2.4.3 Extensions liées au boosting et bagging

Technique du « bagging »

Cela consiste à **sous-échantillonner** ou ré-échantillonner au hasard avec doublons les données d'entraînement et de faire générer à l'algorithme **un modèle pour chaque sous-échantillon**. Un ensemble de modèles est obtenu qu'il convient de **moyenner** lorsqu'il s'agit d'une régression ou de **faire voter** pour une classification.

Cette technique est utilisée particulièrement dans les arbres de décision tels que nous venons de les aborder qui présentent plusieurs limitations comme la sensibilité au bruit ou au sur-apprentissage. Proposées en 2002 par Leo BREIMAN et Adele CUTLER **[BRE-2001]**, les forêts aléatoires utilisent cette technique pour tirer de manière aléatoire un ensemble de données dans le set d'entraînement et recopie des données sous représentés.

Un échantillon d'un ensemble nommé T est l'ensemble obtenu en tirant $|T|$ fois des éléments de T uniformément au hasard et avec remise. Généralement, cet échantillon représente en moyenne $1 - e^{-1} \approx 63\%$ instances uniques différentes de T quand $|T| \gg 1$.

Pour l'aggrégation, plusieurs échantillons sont créés et chaque échantillon T'_i est utilisé pour entraîner un arbre. Étant donnée une instance $(\boldsymbol{x}, y)$, une régression est réalisée au niveau de chaque arbre, ce qui nous donne un ensemble de valeurs $y_1, \ldots, y_m$ prédites. Celles-ci sont alors agrégées en calculant leur moyenne.

Technique du « boosting »

Le boosting est une technique d'ensemble qui consiste à agréger des modèles mathématiques élaborés de manière séquentielle sur les données d'apprentissage. Les poids des individus sont corrigés au fur et à mesure. Une pondération est réalisée au niveau des modèles selon leurs performances. Ces techniques sont décrites dans **[FRE-1999] [BRE-1999] [FRE-2001].**

Cette technique est utilisée dans des algorithmes encore peu utilisés dans le domaine de la prédiction énergétique comme l'algorithme nommé **AdaBoost**. C'est un algorithme de boosting qui s'appuie sur ce principe, avec un paramètre de mise à jour adaptatif permettant de donner plus d'importance aux valeurs difficiles à prédire, donc en boostant les classifieurs qui réussissent quand d'autres ont échoué.

Nous retrouvons cette notion également dans le **gradient boosting** ou descente du gradient qui est une technique itérative qui permet d'approcher la solution d'un problème d'optimisation. En apprentissage supervisé, la construction du modèle revient souvent à déterminer les paramètres (du modèle) qui permettent d'optimiser (max ou min) une fonction objective. L'idée est d'agréger plusieurs modèles ensembles mais en les créant itérativement. Elle est employée majoritairement avec des arbres de décision nommé alors **Gradient Tree Boosting [FRE-1999]**.

Intégrés dans les arbres de décision, les paramètres de configuration des arbres tels que les **Random forest** et le **Gradient bossting tree** décrits par Leo Breiman et Adele Cutler sont :

- ⇨ Critère de fractionnement (indice de Gini)
- ⇨ Nombre limite de niveaux (profondeur de l'arbre)
- ⇨ Taille minimale du noeud
- ⇨ **Nombre de modèles**
- ⇨ Échantillonnage des données

2.5 Calcul de l'erreur

2.5.1 Courbe ROC dans le cas de problème binaire

La courbe ROC (de l'anglais Receiver Operating Characteristic) est une mesure de la performance d'un classificateur binaire. Graphiquement, on représente souvent la mesure ROC sous la forme d'une courbe qui donne le taux de vrais positifs (fraction des positifs qui sont effectivement détectés) en fonction du taux de faux positifs (fraction des négatifs qui sont détectés incorrectement). Nous étudions alors un ensemble de valeurs seuil possibles et, pour chacune, on calcule différentes statistiques dont les plus simples sont :

- ⇨ Vrais positifs (VP) : nombre d'heures classées positivement par le test et qui le sont effectivement.
- ⇨ Faux positifs (FP) : nombre d'heures classées positivement par le test mais qui sont en réalité négatives.

- ⇨ Vrais négatifs (VN) : nombre d'heures classées négativement par le test et qui le sont effectivement.
- ⇨ Faux négatifs (FN) : nombre d'heures classées négativement par le test mais qui sont en réalité positives.
- ⇨ Prévalence de l'évènement : fréquence de survenance de l'évènement dans l'échantillon total (VP+FN)/N.

Plusieurs indices synthétiques ont été mis au point afin d'évaluer la performance du test à une valeur seuil donnée :

La sensibilité (aussi appelée fraction de Vrais Positifs) est la proportion d'heures positives effectivement bien détectées par le test. Le test est parfait lorsque la sensibilité vaut 1, équivalent à un tirage au hasard lorsque la sensibilité vaut 0.5. S'il est inférieur à 0.5, le test est contre-performant et on aurait intérêt à inverser la règle pour qu'il soit supérieur à 0.5 (à condition que cela n'affecte pas la spécificité).

La définition mathématique est :

$$Sensibilité = \frac{VP}{(VP+FN)} \quad \textbf{(2.3)}$$

La spécificité (aussi appelée Fraction de Vrais Négatifs) est la proportion d'heures négatives effectivement bien détectées par le test. La définition mathématique est :

$$Spécificité = \frac{VN}{(VN+FP)} \quad \textbf{(2.4)}$$

On définit également **la fraction de faux positifs (FP)** qui est la proportion de négatifs détectés comme des positifs par le test (1-Spécificité) et la **fraction de faux négatifs (FN)** : proportion de positifs détectés comme des négatifs par le test (1-Sensibilité).

2.5.2 Méthodes statistiques

Nous utilisons également des outils statistiques qui permettent de déterminer la qualité de la prédiction. Nous en avons choisi quatre qui sont retenus dans la plupart des travaux de prédictions :

Le coefficient de détermination (R^2) est une mesure de la qualité de la prédiction d'une régression linéaire. Il est défini comme 1 moins le ratio entre l'erreur avec les valeurs prédites et la variance des données :

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{\sum_{i=1}^{n}(y_i - \bar{y}_i)} \quad \textbf{(2.5)}$$

y_i les valeurs des mesures, $\widehat{y_i}$ les valeurs prédites et y_i Lorsqu'il est proche de 0, le pouvoir prédictif du modèle est faible et lorsqu'il est proche de 1, le pouvoir prédictif du modèle est fort.

L'erreur quadratique moyenne (Mean Square Error, MSE**)** est le carré moyen des erreurs ou erreur quadratique moyenne (MSE pour *Mean Square Error* ou MCE pour moyenne des carrés des erreurs) : c'est la moyenne arithmétique des carrés des écarts entre les prévisions et les observations.L'erreur type **RMSE** (root mean square deviation) est la racine carrée de la MSE.

L'erreur moyenne absolue (Mean Absolute Error, MAE) est la moyenne arithmétique des valeurs absolues des écarts.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} | y_i - \overline{y_i}| \qquad \textbf{(2.6)}$$

L'écart quadratique moyen (MSD). Pour calculer l'erreur quadratique moyenne RMS (Root Mean Square), les erreurs individuelles sont tout d'abord élevées au carré, puis additionnées les unes aux autres. On divise ensuite le résultat obtenu par le nombre total d'erreurs individuelles, puis on en prend la racine carrée. Les valeurs aberrantes ont un effet plus important sur le MSD que sur le MAD. L'équation est la suivante :

$$MSD = \sqrt{\frac{\sum_{i=1}^{n} | y_i - \widehat{y_i}|^2}{n}} \qquad \textbf{(2.7)}$$

Enfin une erreur en termes de consommation énergétique sera également présentée. Elle est définie par :

$$Erreur = Réel - Prédiction \qquad \textbf{(2.8)}$$

2.6 Conclusion

Les paramètres pouvant influencée la prédiction sont tellement nombreux, qu'il est difficile d'utiliser les résultats issus d'autres études. Par exemple, dans **[ZHA-2016]**, la série temporelle de consommation est prédite à partir d'une méthode hybride qui utilise une régression suivie d'une méthode d'apprentissage avec l'algorithme SVM.

Au niveau de la prédiction de production solaire décentralisée, les différentes études concernent un large choix d'installations photovoltaïques et de type de données, (étude sur 12 cellules PV en laboratoire ou de 14 000 m^2 de panneaux) avec des modèles ARIMA ou des réseaux de neurones **[ZAM-2014] [GRA-2016] [GUL-2016]**. Basé sur un historique, les données de prédiction météorologiques sont parfois corrigées lors d'une première phase de test afin d'améliorer la prédiction de production des panneaux. Les données proviennent d'une seule station météo locale ou régionale.

Nous notons cependant que les réseaux de neurones sont aujourd'hui les modèles de prédictions les plus utilisés. Les études utilisent des PNN (Probabilistic Neural network), MLP (Multi Layer Perceptron, ANN (Artificial Neural Network), SVM (Support Vector Machine). Chaque algorithme a des paramètres spécifiques que l'on peut faire varier et représente à lui seul un domaine de spécialité à lui

seul. Il n'en reste pas moins que les réseaux de neurones sont considérés comme des boîtes noires difficilement interprétables.

L'arbre de décision est lui beaucoup plus simple et permet une lisibilité par le lecteur. Dans notre cas où plusieurs milliers de variables vont caractériser le vecteur d'entrée, seul, il est mal adapté. C'est pourquoi des techniques plus récentes ont été développées qui permettent de construire des modèles mathématiques adaptés à notre cas d'étude comme les ensembles d'arbres de décision. Les résultats statistiques issus de la construction des arbres permettent de recueillir une connaissance précieuse sur la pertinence d'une variable par rapport à une autre.

Tous les algorithmes cités ne sont pas toujours disponibles dans les différents outils de prédictions. Nous essayerons de choisir également un outil de traitement qui intégrera la majorité des algorithmes cités dans ce chapitre : les méthodes linéaires (régression linéaire et ARIMA) et des méthodes non linéaires (ANN, PNN, MLP, SVM, Random Forest, Adaboost et Gradient Boosted tree).

Cependant, l'intégration des ensembles d'arbres de décision nous semblent être un caractère indispensable pour la compréhension et la réduction de notre vecteur d'entrée.

Chapitre 3 : Mettre en place le système d'information

3.1 Introduction

L'objectif de chapitre est de mette en place un **système d'information local** au niveau de chaque système qui nous permet de collecter, stocker, traiter et distribuer les connaissances liées aux prédictions à un niveau de pilotage supérieur (Figure 30).