UNIVERSITÉ DU QUÉBEC

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

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE APPLIQUÉES

PAR
Kalidou Moussa SOW

MODÉLISATION STATISTIQUE ET PRÉDICTION PAR APPRENTISSAGE AUTOMATIQUE : LA DÉGRADATION PAR APPLICATION À LA CORROSION D'UN PIPELINE

Décembre 2024

Résumé

La question des défaillances des pipelines a suscité un vif intérêt au sein de diverses communautés de chercheurs en raison de ses conséquences significatives sur l'économie mondiale, ainsi que des risques associés aux fuites, aux explosions et aux coûteuses périodes d'immobilisation. Afin de mieux détecter et prévenir ces défaillances, nous proposons dans ce mémoire deux approches d'analyses.

La première développe un modèle de classification et de prédiction de la dégradation par corrosion d'un tuyau utilisé pour le transport de l'eau dans les mines, développé par le Centre de métallurgie du Québec. À cette fin, deux types de modèles ont été développés : trois modèles de classification binaire (SVM, FA et KNN) respectivement, et un modèle de réseau de neurones, Long Short-Term Memory (LSTM), permettant de prédire les variations moyennes de l'épaisseur de la canalisation sur une période de 63 jours.

La deuxième approche est une méthode multivariée permettant d'étudier l'évolution de l'épaisseur du pipeline minier à l'aide de réseaux de neurones artificiels de type LSTM, en vue d'implémenter un modèle prédictif. Le LSTM est une architecture spécifique de réseau neuronal récurrent (RNN) conçue pour modéliser des séquences temporelles. Le modèle prédictif proposé donne de très bons résultats et permet de prévoir les variations de huit mesures d'épaisseur sur une période de cent jours.

Kalidou Moussa SOW

Nadia GHAZZALI

Abstract

The issue of pipeline failures has sparked significant interest among various research communities due to its substantial impact on the global economy, as well as the associated risks of leaks, explosions, and costly downtime. To better detect and prevent these failures, this thesis proposes two analytical approaches :

The first develops a classification and prediction model for corrosion degradation of a pipe used for water transport in mines, developed by the Centre de métallurgie du Québec. For this purpose, two types of models have been developed : three binary classification models (SVM, RF, and KNN) respectively, and a Long Short-Term Memory (LSTM) neural network model, allowing for the prediction of average pipeline thickness variations over a period of 63 days.

The second is a multivariate approach to study the evolution of mining pipeline thickness using artificial neural networks LSTM to implement a predictive model. LSTM is a specific architecture of recurrent neural networks (RNN) designed to model temporal sequences. The proposed predictive model yields very good results and allows for predicting variations in eight thickness measurements over a period of one hundred days.

Remerciements

Je tiens à adresser mes sincères remerciements à ma directrice de recherche Mme Nadia Ghazzali pour avoir accepté de diriger mon mémoire. Sa disponibilité, son appui tant de côté financier que pédagogie, et ses conseils m'ont permis de surmonter les défits et alimenter mes connaissances indispensables à la conduite de cette recherche. Merci infiniment pour tous les efforts consentis.

Je remercie également le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), et le Fonds de recherche nature et technologies du Québec (FRQNT) pour leur appui financier, ainsi que le Centre Métallurgie du Québec (CMQ) et au Agnico Eagle Goldex Mine pour la disponibilité des données.

Je dédie ce mémoire à ma mère, Aminata Sow, pour son soutien constant et son encouragement, à mes frères et soeurs, à mes amis et à tous ceux qui ont contribué de près ou de loin à la rédaction de ce mémoire.

À ma femme, Salamata Sow, merci pour ton amour inconditionnel, ta force et ton incroyable capacité à me soutenir comme personne d'autre ne le peut. Chaque jour à tes côtés est un cadeau, une source d'inspiration et de sérénité.

Table des matières

Table des figures

Liste des tableaux

Introduction

Depuis de nombreuses décennies, les pipelines constituent le moyen le plus efficace et le plus sûr pour transporter des matériaux à l'échelle mondiale. Toute défaillance des systèmes de transport par pipeline affecte directement l'économie de l'industrie des matériaux [6]. Au fil des ans, les chercheurs ont étudié les modèles de ces défaillances [7], [8]. La majorité des études portant sur l'évaluation des différents types de défaillance des oléoducs et gazoducs indiquent que la corrosion est l'une des causes les plus fréquentes de défaillance des systèmes de transport. Dans [9], les causes de la corrosion ont été classées en trois catégories : 1) les facteurs environnementaux, tels que les propriétés du sol, les conditions extérieures et les courants vagabonds (des courants électriques non désirés circulant à travers des matériaux conducteurs, comme le sol ou les structures métalliques, en dehors de leur circuit prévu) ; 2) les facteurs liés aux canalisations ; et 3) les facteurs opérationnels. Ils ont utilisé le processus hiérarchique analytique flou pour représenter l'impact des trois facteurs sur la corrosion des pipelines, et ont constaté que les facteurs opérationnels avaient le poids relatif le plus élevé (0,428), suivis par les facteurs environnementaux (0,337). Dans [10], les auteurs expliquent que 60 % des défaillances des systèmes mexicains de transport de pétrole et de gaz sont causées par la corrosion par piqûres, un type de corrosion localisée formant de petites cavités ou "piqûres" à la surface du métal, souvent difficile à détecter mais pouvant entraîner des défaillances graves. Le nombre élevé d'incidents dus à la corrosion s'explique par la complexité de l'environnement entourant les pipelines, notamment la grande diversité des propriétés du sol, de l'eau

et des produits transportés par le pipeline (pétrole ou gaz). Grâce aux progrès de l'apprentissage automatique (ML) et de l'apprentissage profond (DL), les méthodes de détection basées sur des modèles et guidées par les données pour la surveillance de l'érosion et de la corrosion des pipelines ont suscité un vif intérêt. Le réseau neuronal artificiel (ANN), en particulier le réseau neuronal à rétropropagation (BPNN), est largement utilisé pour prédire le taux d'érosion-corrosion dans les pipelines [11]. Dans [12], les auteurs ont adopté le modèle ANN pour détecter la corrosion dans les oléoducs sous-marins, selon quatre niveaux (pas de corrosion, légère, modérée, grave), en se basant sur les données collectées par capteurs ultrasoniques et capteurs de fuite de flux.

Ce document a pour objectif de développer un modèle prédictif de la dégradation par corrosion d'un pipeline utilisé pour transporter de l'eau dans la région minière du centre métallurgique du Québec. Des travaux ont déjà été réalisés sur les mêmes données que celles utilisées dans ce mémoire. Dans [13], les auteurs ont appliqué un réseau neuronal non supervisé, les cartes auto-organisatrices (SOM), pour analyser l'impact de la corrosion, évaluée à partir d'inspections ultrasoniques périodiques. Ils ont combiné les SOM avec un regroupement hiérarchique pour détecter l'étendue de la corrosion dans un pipeline minier. Dans [14], les mêmes auteurs ont utilisé plusieurs méthodes d'apprentissage automatique, telles que l'élimination récursive des caractéristiques (RFE), l'analyse en composantes principales (PCA), le boosting par gradient (GBM), les machines à vecteurs de support (SVM), les forêts aléatoires (RF), les k-plus proches voisins (KNN) et le perceptron multicouche (MLP) pour estimer la perte d'épaisseur d'un pipeline transportant une boue soumise à l'érosion-corrosion. Ils ont constaté que le modèle SVM fournissait la meilleure estimation, avec une racine de l'erreur quadratique moyenne (RMSE) de 0,011 et un coefficient de détermination (R^2) de 0,83.

Ce travail s'inscrit dans la continuité des études de [13] et [14] en analysant l'approche univariée des données et en introduisant une nouvelle méthode intégrant l'ana-

lyse multivariée et les réseaux LSTM. Pour mettre en œuvre ces nouvelles approches, deux articles scientifiques ont été publiés. Ainsi, dans [15], nous étudions la nature multivariée des données et utilisons le LSTM pour prédire l'évolution de l'épaisseur des pipelines. Dans [16], nous présentons une analyse univariée des données en calculant la moyenne mobile des épaisseurs et en utilisant des modèles d'apprentissage automatique (KNN, SVM et RF) ainsi que le LSTM pour évaluer la dégradation par corrosion des pipelines miniers.

Ce mémoire sera divisé en quatre chapitres. Le premier sera consacré à une revue de la littérature sur les modèles d'apprentissage automatique et d'apprentissage profond utilisés dans le domaine de la corrosion. Le deuxième traitera la méthodologie utilisée dans les deux articles publiés mis en annexe. Ensuite, avant de conclure, nous aborderons les données utilisées dans ce mémoire dans le chapitre 3 et les résultats obtenus par les deux modèles dans le chapitre 4.

Chapitre 1

Revue de littérature

1.1 Revue de littérature sur l'étude de la corrosion

Au cours des dernières années, d'importants efforts ont été déployés pour s'attaquer au problème de la corrosion des pipelines en utilisant des modèles statistiques, y compris diverses techniques d'apprentissage automatique. Les avancées en apprentissage automatique (ML) et en apprentissage profond (DL) ont suscité un vif intérêt pour les méthodes de détection basées sur des modèles de données, dans le but de surveiller l'érosion et la corrosion des pipelines. Aghaaminiha et al. [17] utilisent des méthodes d'apprentissage automatique supervisé pour modéliser les mesures des taux de corrosion de l'acier au carbone en fonction du temps. Ils ont comparé différents modèles d'apprentissage automatique et ont conclu que la méthode basée sur les forêts aléatoires (FA) était la plus performante sur leurs données, avec une erreur quadratique moyenne comprise entre 0,005 et 0,093. Sheikh et al. [18] ont employé une technique hybride qui combine la détection de la corrosion à partir des signaux d'émission acoustique issus des essais de corrosion accélérée avec des techniques d'apprentissage automatique pour prédire avec précision les niveaux de gravité de la corrosion. Ils

ont appliqué des arbres de décision, un réseau neuronal à rétropropagation et un réseau neuronal à fonction de base radiale à leurs données, obtenant respectivement des précisions de 90,4 %, 94,57 % et 100 %. Hendi et al. [19] ont mis en œuvre un modèle de réseau neuronal à rétropropagation pour minimiser la corrosion du béton dans le système d'égouts en utilisant des billes de verre en substitution, ainsi que pour prédire la perte de masse et la perte de volume dans les échantillons. Ils ont obtenu une erreur quadratique moyenne de 0,44 pour la perte de masse et de 1,18 pour la perte de volume.

1.2 Revue de littérature sur le Long Short Terme Memory (LSTM)

La mémoire à long terme et court terme (LSTM) est une architecture spécifique de réseau de neurones récurrent (RNN) conçue pour modéliser des séquences temporelles. Développée en 1997 par Sepp Hochreiter et Jürgen Schmidhuber, le LSTM a été conçue pour résoudre le problème de la disparition du gradient présent dans les RNN traditionnels. Sa relative insensibilité à la longueur de l'intervalle constitue un avantage par rapport aux autres RNN, aux modèles de Markov cachés et à d'autres méthodes d'apprentissage de séquences. Le LSTM est particulièrement efficace pour prédire des séries temporelles. Elle peut extraire des motifs à partir de données séquentielles et stocker ces motifs dans des variables d'état internes. Chaque cellule LSTM peut conserver des informations importantes pendant une période prolongée lorsqu'elle est utilisée.

Cette propriété d'information permet au LSTM d'exceller dans la classification, le traitement ou la prédiction de séquences dynamiques complexes[20], ce qui en fait un modèle très utilisé dans la littérature. Salman et al.[21] ont appliqué un modèle LSTM aux données des variables météorologiques collectées par Weather Underground à Hang Nadim, en Indonésie. Ils ont ajouté un signal de variable intermédiaire dans

la cellule du bloc mémoire de la LSTM, et leur modèle a surpassé les autres modèles LSTM avec une précision de 0,8060 et une RMSE de 0,0775. Nelson et al.[22] ont utilisé des réseaux LSTM pour prédire les tendances futures des prix des actions en se basant sur l'historique des prix, ainsi que sur des indicateurs d'analyse technique, atteignant une précision moyenne de 55,9 % pour prédire si le prix d'une action donnée allait augmenter prochainement. Di Persio et Honchar[23] ont comparé les performances du LSTM et du Perception Multicouche (MLP) à leur propre méthode proposée, fondée sur une combinaison d'ondelettes et de réseaux neuronaux convolutifs (CNN), qui surpasse les deux autres mais dont les résultats sont très proches de ceux du réseau LSTM. Karmiani et al.[24] ont comparé le LSTM avec la Machine à Vecteurs de Support (SVM), la rétropropagation et le filtre de Kalman pour le marché boursier, en variant le nombre d'époques de 10 à 100. Il a été constaté que le LSTM présentait une grande précision et une faible variance. Chen et al.[25] ont utilisé un modèle LSTM sur les données historiques du marché boursier chinois. Ils ont entrainé le modèle LSTM sur 900 000 séquences et l'ont testé en utilisant les 311 361 autres séquences. Comparé à la méthode de prédiction aléatoire, leur modèle LSTM a amélioré la précision de la prédiction des rendements boursiers de 14,3 % à 27,2 %. Ces efforts ont démontré la puissance du LSTM dans la prédiction d'un marché boursier chinois dynamique et hautement imprévisible.

Le modèle LSTM a également prouvé son efficacité dans l'analyse et la prédiction de la corrosion des pipelines, puisque les données collectées dans ce domaine peuvent être considérées comme des séries temporelles. Li et al.[26] ont combiné un nouvel algorithme d'optimisation basé sur l'intelligence des essaims, appelé SSA, avec un modèle LSTM pour prédire la profondeur maximale de corrosion par piqûre des pipelines sous-marins. La comparaison de leur méthode SSA-LSTM avec le LSTM seul montre que le nouveau modèle SSA-LSTM a obtenu de meilleures performances en termes de précision, de prédiction et de robustesse. Ils ont utilisé la RMSE, l'erreur absolue moyenne (MAE), l'erreur quadratique moyenne (MSE) et l'erreur absolue en pourcentage moyenne (MAPE) comme paramètres d'évaluation pour mesurer la performance

de leur modèle. Le modèle hybride proposé (SSA-LSTM) a obtenu les meilleures performances avec les plus petites valeurs des paramètres d'évaluation (RMSE = 0,0607, MAE = 8,84 %, MSE = 0,36 %, MAPE = 9,58 %).

Chapitre 2

Approche méthodolique

2.1 Analyse en composantes principales

Cette section s'inspire de [27].

L'analyse en composantes principales (ACP) est une méthode statistique utilisée pour réduire la dimensionnalité d'un jeu de données tout en conservant le maximum d'information possible. Elle repose sur la transformation linéaire des variables originales en de nouvelles variables appelées composantes principales, qui sont non corrélées et ordonnées de manière à capturer le maximum de variance dans les données.

Voici les étapes principales de l'ACP basée sur la matrice corrélation :

Standardisation

Si les variables dans votre jeu de données sont mesurées dans des unités différentes, il est souvent nécessaire de standardiser les données avant de procéder à l'ACP. Cela signifie que chaque variable X_i est transformée de manière à avoir une moyenne nulle et une variance unitaire, à l'aide de la formule suivante :

$$Z_{ij} = \frac{X_{ij} - \bar{X}_i}{\sigma_i},$$

où :

X_{ij} est une valeur brute dans la matrice des données, représentant une mesure pour une observation donnée,

Z_{ij} est la valeur standardisée de l'élément j de la variable i,

$\bar{X}_i$ est la moyenne de la variable i,

σ_i est l'écart-type de la variable i.

Matrice de corrélation

La matrice de corrélation R est une matrice carrée de dimension pp, où p est le nombre de variables.

Chaque élément r_{ij} de la matrice R représente le coefficient de corrélation de Pearson entre les variables i et j.

La formule pour r_{ij} entre deux variables standardisées Z_i et Z_j est donnée par :

$$r_{ij} = \frac{1}{n-1} \sum_{k=1}^{n} Z_{ki} Z_{kj},$$

où n est le nombre d'observations.

En d'autres termes, R peut être calculée comme :

$$R = \frac{1}{n-1} Z^T,$$

où Z est la matrice des données standardisées.

Calcul des valeurs propres et des vecteurs propres

L'étape suivante consiste à diagonaliser la matrice de corrélation R c'est-à-dire à calculer ses valeurs propres et vecteurs propres.

Cela revient à résoudre l'équation suivante :

$$Rv = \lambda v.$$

Les valeurs propres λ_1, λ_2, ...,λ_p représentent la quantité de variance expliquée par chaque composante principale.

Les vecteurs propres v associés aux valeurs propres indiquent les directions dans les-

quelles les données varient le plus (ces vecteurs propres sont les composantes principales).

Projection des données dans le nouvel espace

Une fois que les vecteurs propres ont été trouvés, les données originales sont projetées dans le nouvel espace formé par les composantes principales (PC). Les nouvelles coordonnées des données dans cet espace sont appelées scores des composantes principales. Pour chaque observation X_i, le score pour la première composante principale PC_1 est donné par :

$$PC_1 = X_i v_1,$$

où v_1 est le vecteur propre correspondant à la première valeur propre.

De même, les autres composantes principales PC_2, PC_3, PC_4...sont obtenues en projetant les données sur les vecteurs propres correspondants.

2.2 Apprentissage automatique

2.2.1 Forêts aléatoires (FA)

Les forêts aléatoires (FA) est un modèle basé sur l'apprentissage supervisé qui se compose d'un grand nombre d'arbres de décision. Les arbres de décision sont les classificateurs de base des FA et fonctionnent en tant qu'ensemble. Ainsi, les forêts aléatoires sont également appelées ensembles de classificateurs. Elles ont été conceptualisées pour la première fois en 2001 par Breiman [28]. La caractéristique principale de ce modèle est que chaque arbre de décision est construit sur un ensemble de paramètres choisis aléatoirement. Cette randomisation dans la sélection des paramètres introduit de la diversité dans l'ensemble des classificateurs de base. Ainsi, chaque arbre de décision est indépendant des autres, c'est-à-dire qu'ils ne sont pas corrélés.

Le modèle de la forêt aléatoire repose sur des concepts de statistiques et d'apprentis-

sage automatique, en particulier sur les arbres de décision et l'agrégation. Voici une description détaillée de la façon dont cela fonctionne, incluant les étapes clés :

1. Arbre de Décision

Un arbre de décision est construit en suivant un processus récursif de division des données en fonction des caractéristiques. Voici quelques éléments clés associés à un arbre de décision :

Splitting Criterion : À chaque nœud de l'arbre, une caractéristique X_j est choisie pour diviser le jeu de données en deux sous-ensembles. Cette division est déterminée par un critère de qualité, tel que l'impureté de Gini ou l'entropie.

Impureté de Gini :

$$Gini(D) = 1 - \sum_{k=1}^{K} p_k^2,$$

où p_k est la proportion d'observations de la classe k dans le sous-ensemble D.

Entropie :

$$Entropie(D) = -\sum_{k=1}^{K} p_k \log_2 p_k,$$

où $\log_2$ représente le logarithme en base 2. **2. Bagging (Bootstrap Aggregating)** La forêt aléatoire utilise le bagging pour construire chaque arbre. Cela consiste à créer plusieurs échantillons de données à partir de l'ensemble de données d'origine.

Échantillonnage Bootstrap : Pour chaque arbre t, on crée un échantillon D_t de taille N (même taille que l'ensemble de données d'origine), en tirant au hasard avec remplacement.

3. Construction des Arbres Pour chaque arbre t, on crée un sous-ensemble D_t en utilisant le tirage bootstrap.

À chaque nœud, on sélectionne un sous-ensemble aléatoire de caractéristiques F (généralement beaucoup plus petit que le nombre total de caractéristiques P).

Pour chaque nœud n, on choisit la meilleure division basée sur les critères de séparation (impureté de Gini ou entropie) parmi les m caractéristiques choisies au hasard.

4. Prédiction

Une fois que tous les arbres sont construits, la forêt aléatoire fait des prédictions :

Pour une observation x, chaque arbre t prédit une classe $\hat{y}_t(x)$.
La prédiction finale est la classe qui reçoit le plus de votes :

$$\hat{y}(x) = argmax_k \left(\sum_{t=1}^{T} Ind[\hat{y}_t(x) = k] \right),$$

où Ind est la fonction indicatrice qui vaut 1 si $\hat{y}_t(x)$ est égal à k et 0 sinon, et T est le nombre total d'arbres.

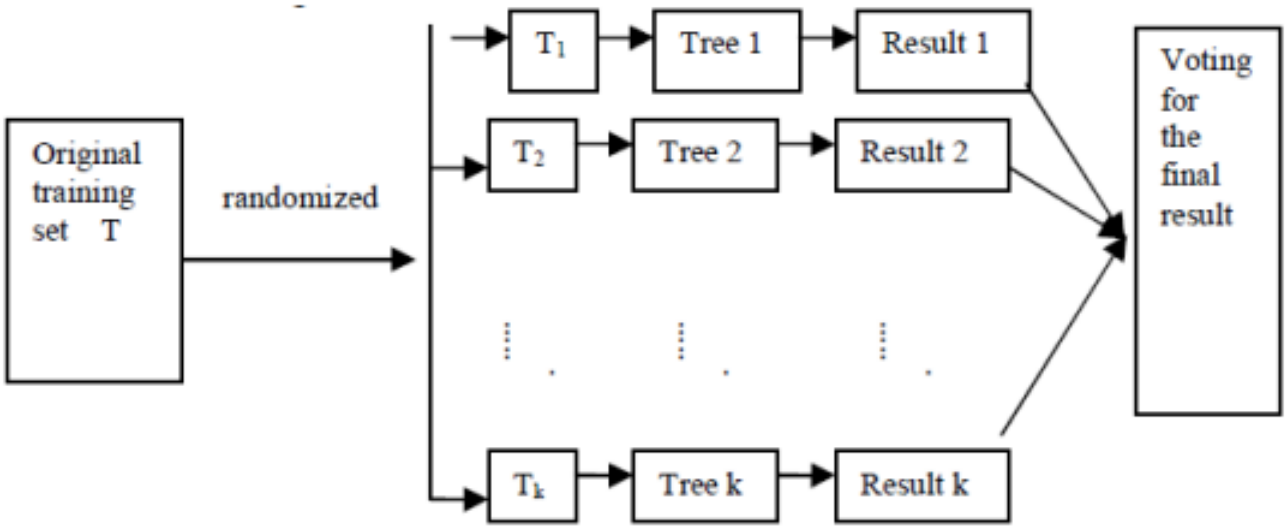

FIGURE 2.1 – *Illustration du fonctionnement d'un FA [1]*

2.2.2 Les K plus proches voisins (KNN)

L'algorithme des K plus proches voisins (KNN)[29] est une méthode d'apprentissage automatique appartenant a la catégorie des algorithmes d'apprentissage supervisé. Il est simple a mettre en oeuvre et peut être utilisé pour aborder des problèmes de classification et de régression.
Étant donné l'ensemble de données d'entraînement :

$$(x_1, y_1), (x_2, y_2), ..., (x_n, y_n).$$

Étape 1 : Stocker l'ensemble de données d'entraînement.

Étape 2 : Pour chaque nouvelle donnée non étiquetée (voir figure 2.2) :

A.Calculer la distance euclidienne avec tous les points de données d'entraînement en utilisant la formule :

$$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}.$$

On peut également utiliser d'autres types de calcul de distance, tels que la **distance de Manhattan** et la **distance de Minkowski**, selon la nature des données.

B. Trouver les K plus proches voisins

C. Attribuer la classe contenant le plus grand nombre de voisins les plus proches.

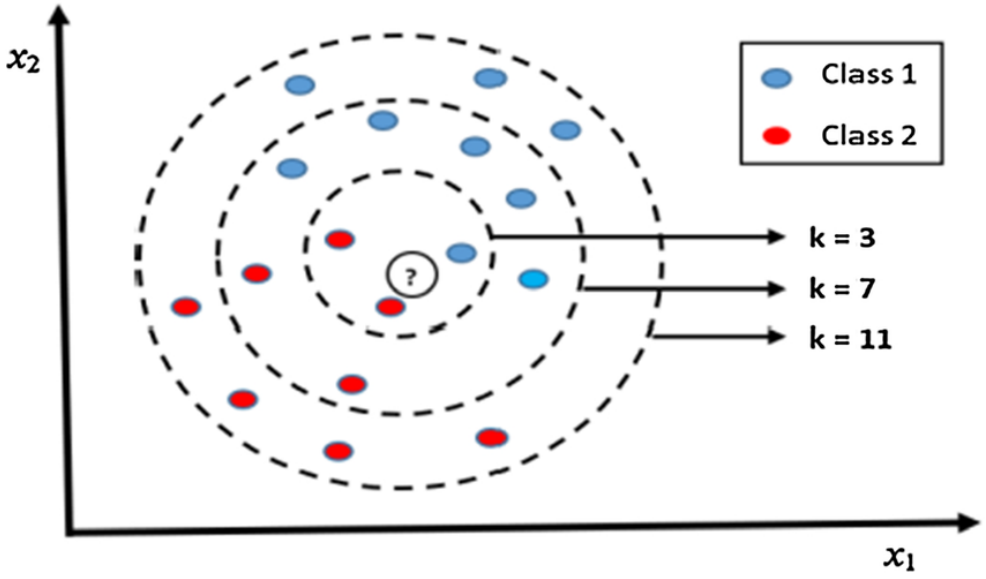

FIGURE 2.2 – *Illustration du fonctionnement d'un KNN[2]*

Le choix de la valeur de K dans l'algorithme KNN est crucial car il influence directement les performances du modèle en termes de précision, de généralisation, et de résistance au bruit.

Voici les principaux points à prendre en compte pour choisir la valeur de K :

1. Petit K (K = 1, 3, 5) :

Sensibilité au bruit : Un petit K (comme $K = 1$) peut rendre l'algorithme très sensible au bruit, car il prend en compte uniquement les voisins les plus proches, même s'ils sont des valeurs aberrantes.

Variance élevée : L'algorithme est plus susceptible de surajustement (overfitting), car il peut trop coller aux points particuliers du jeu d'entraînement.

Décision rapide : Cependant, un petit K peut capturer des détails fins dans la classification, ce qui peut être bénéfique dans des ensembles de données très distincts.

2. Grand K (K élevé) :

Lissage des décisions : Un plus grand K (comme $K = 10$ ou $K = 20$) réduit l'impact des valeurs aberrantes et fait en sorte que l'algorithme prenne des décisions plus généralisées.

Moins de bruit : L'effet des points bruyants diminue, car la décision repose sur un plus grand nombre de voisins.

Biais élevé : Cependant, un K trop grand peut conduire à un sous-ajustement (underfitting), car il dilue l'influence des points proches et peut mélanger des points appartenant à différentes classes, rendant les frontières entre les classes moins précises.

3. Choisir K par la méthode de Validation Croisée :

Une approche courante consiste à choisir K en utilisant la validation croisée. Cette méthode consiste à diviser l'ensemble de données en sous-ensembles et à tester différentes valeurs de K pour voir laquelle donne la meilleure performance sur les données de validation. L'erreur quadratique moyenne peut être utilisée pour évaluer les performances.

2.2.3 Support vecteur machine (SVM)

SVM est une méthode de classification proposée par Vapnik 1982[30] et visant à chercher un hyperplan séparateur tout en maximisant la marge entre les deux classes. Dans ce scénario à deux dimensions, avec X_1 et X_2 comme caractéristiques et une variable dépendante binaire (cercles ou carrés), l'objectif est de trouver une frontière

de décision — une ligne — qui sépare au mieux les deux classes (voir figure 2.3).
Voici comment les SVM abordent ce problème :

Hyperplan optimal : Parmi toutes les lignes possibles (ou hyperplans) qui séparent les classes, le SVM sélectionne celle qui maximise la marge, c'est-à-dire la distance entre les points les plus proches de chaque classe (appelés vecteurs de support) et l'hyperplan. Cette maximisation de la marge réduit les erreurs de classification et renforce la robustesse du modèle.

Maximisation de la marge : L'algorithme SVM optimise la position de l'hyperplan pour maximiser cette marge. Mathématiquement, cela est réalisé en résolvant un problème d'optimisation convexe, garantissant que l'hyperplan est à égale distance des points les plus proches de chaque classe.

Données linéairement séparables : Si les données sont parfaitement séparables de manière linéaire, le SVM trouvera une droite qui sépare tous les points rouges et bleus avec une marge maximale. Si les données ne sont pas linéairement séparables, le SVM peut utiliser des astuces de noyau (ou kernel tricks) pour projeter les données dans un espace de dimension supérieure où un hyperplan séparateur peut exister. Pour

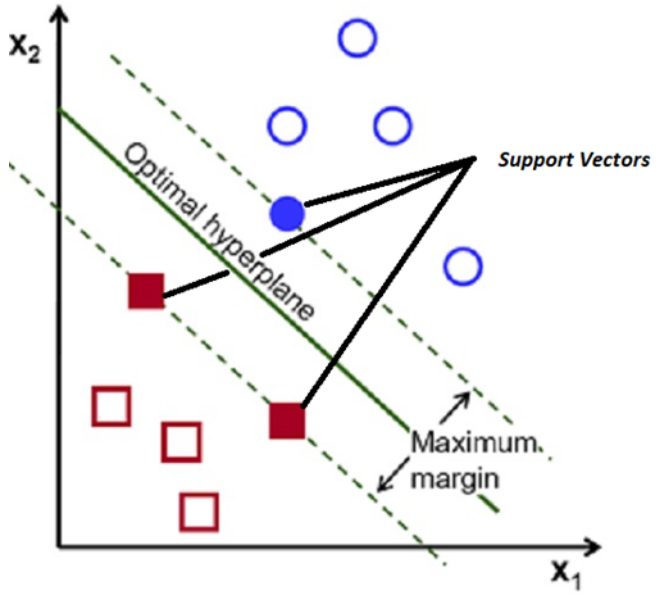

FIGURE 2.3 – *Illustration du fonction d'un SVM [3]*

expliquer le fonctionnement du SVM, considérons un ensemble de données d'entrée

$X = (x_i)$, où les $x_i \in \mathbb{R}^p$, et une étiquette de classe $y_i \in \{-1, +1\}$.
L'équation de l'hyperplan est définit comme suite :

$$\omega^T X + b = 0. \tag{2.1}$$

Le vecteur ω, de dimension p, représente le vecteur normal à l'hyperplan, b est le terme de biais (ou d'interception) qui détermine la position de l'hyperplan dans l'espace.
La distance entre un point de données x_i et la limite de décision peut être calculée comme suit :

$$d_i = \frac{\omega^T x_i + b}{|\omega|}, \tag{2.2}$$

où $|\omega|$ represente la norme euclidienne du vecteur ω.
Pour un modéle de SVM linéaire, on cherche à optimiser l'expression :

$$\min_{\omega,b} \frac{1}{2}\omega^T\omega = \min_{\omega,b} \frac{1}{2}|\omega|^2, \tag{2.3}$$

sous contrainte :

$$y_i(\omega^T x_i + b) \geq 1,$$

ce qui revient à minimiser l'équation de Lagrage suivant :

$$L(\omega, b, \lambda) = \frac{1}{2}|\omega|^2 - \sum_i^m \lambda_i(y_i(\omega^T x_i + b) - 1)), \tag{2.4}$$

sous contrainte :

$$\lambda_i \geq 0.$$

Pour prédire une nouvelle données x_{nouv}, on determine son signe à l'aide de la formule suivante :

$$\hat{y} = sign(\omega^T x_{nouv} + b), \tag{2.5}$$

où

$$\omega = \sum_{i}^{m} \lambda_i y_i x_i$$

et

$$b = y_i - \omega^T x_i$$

et *sign* renvoie +1 ou -1 selon le signe de l'expréssion.

Dans le cas de non séparabilité linéaire (Voir figure 2.4), SVM est incapable de trouver l'hyperplan séparateur permettant de séparer les deux classes.On utilise donc une fonction de noyau qui transforme les données en des données séparables linéairement. Les noyaux couramment utilisés incluent :

Noyau Polynomial :

$$K(x_i, x_j) = (x_i.x_j + c)^d,$$

où d est le degré du polynôme et c est un paramètre constant qui permet de contrôler l'influence des termes linéaires et d'ajuster la flexibilité du modèle.

Noyau RBF (Radial Basis Function) :

$$K(x_i, x_j) = \exp\left\{-\frac{||x_i - x_j||^2}{2\sigma^2}\right\},$$

où σ est un paramètre qui détermine l'ampleur de l'influence d'un point de données sur ses voisins dans l'espace des caractéristiques.

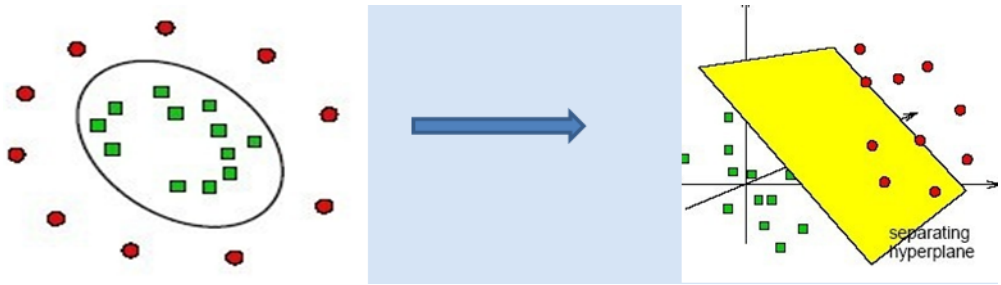

FIGURE 2.4 – *Illustration d'un SVM non linéaire [4]*

2.3 Réseau de neuronnes réccurents (RNN)

Les réseaux de neurones récurrents ou Recurrent Neural Network (RNN) sont des modèles d'apprentissage automatique puissants qui permettent d'analyser des séquences de données, telles que du texte, de la parole ou des séries temporelles. Ces réseaux permettent aux machines de « se souvenir » des informations passées et de les utiliser pour prendre des décisions en temps réel.

Un RNN se compose généralement de plusieurs neurones organisés en couches, avec

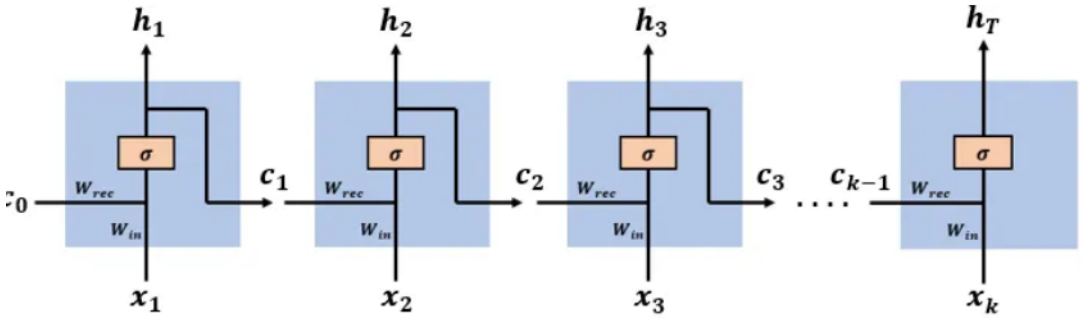

FIGURE 2.5 – *Architecture d'un réseau de neurones réccurents (RNN)[5]*

des connexions récurrentes permettant à l'information de circuler d'un neurone à l'autre à travers le temps (voir figure 2.5).

À chaque étape k, un vecteur d'entrée x_k est fourni au réseau.

Chaque neurone a un état caché c_k qui est mis à jour à chaque instant en fonction de l'entrée actuelle et de l'état caché précédent :

$$c_k = \sigma(W_{rec}c_{k-1} + W_{in}x_k),$$

où W_{rec} et W_{in} sont les poids associés aux états précédents et aux entrées respectivement, et σ est la fonction d'activation sigma (on peut également utilisé la tangente hyperbolique).

À chaque instant t, le RNN peut également produire une sortie h_t.

2.3.1 Fonctions d'activation

Cette fonction détermine si un neurone artificiel doit être activé ou pas et, dans le premier cas, le degré de cette activation. Il existe plusieurs fonctions apportant chacune des comportements différents (sigmoïde, tangente hyperbolique, etc.). Le LSTM n'utilise que deux fonctions d'activations : la sigmoïde et la tangente hyperbolique (tanh) (voir figure 2.6).

Fonction d'activation sigmoïde :

Dans un réseau de neurones artificiels, la fonction d'activation sigmoïde est souvent utilisée dans les couches cachées pour introduire une non-linéarité. Cela permet au réseau de capturer des relations complexes dans les données. Soit ζ l'entrée pondérée pour un neurone particulier, calculée comme suit :

$$\zeta = \sum_i \omega_i x_i + b_i,$$

où ω_i sont les poids, x_i sont les entrées, et b_i est le biais. L'entrée pondérée ζ est ensuite passée à travers la fonction sigmoïde pour obtenir la sortie a du neurone :

$$a = \sigma(\zeta)$$

avec

$$\sigma(\zeta) = \frac{1}{1 + \exp\{-\zeta\}}.$$

Fonction d'activation tangente hyperbolique (tanh)

La tangente hyperbolique, souvent notée tanh, est une autre fonction d'activation couramment utilisée dans les réseaux de neurones artificiels. Elle est définie comme suit :

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

La sortie de la fonction tangente hyperbolique varie entre -1 et 1, ce qui lui permet

de produire des sorties centrées autour de zéro. Cela peut aider à la convergence plus rapide lors de l'entraînement des réseaux de neurones.

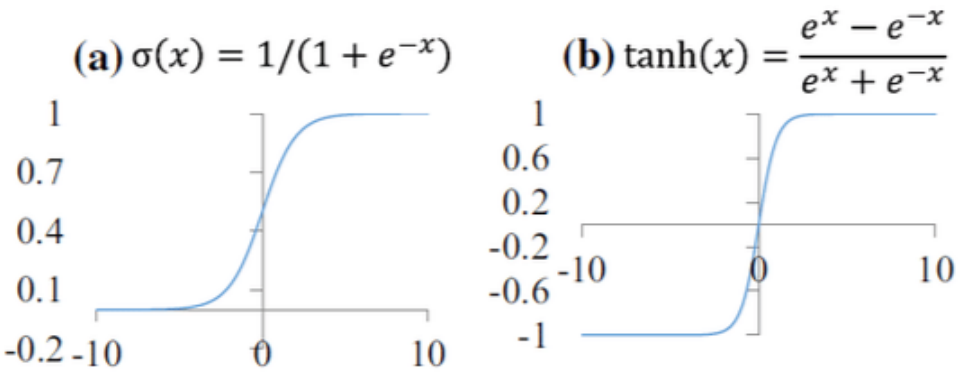

FIGURE 2.6 – *Fonction d'activation (a) Sigmoide et (b) Tangente hyperbolique*

2.3.2 Problème du "Vanishing gradient"

Le problème du "Vanishing gradient" est un exemple de comportement instable qu'on peut rencontrer lors de la formation d'un réseau neuronal. Il décrit la situation dans laquelle un réseau de rétroaction multicouche profond ou un réseau neuronal récurrent est incapable de propager des informations de gradient utiles depuis l'extrémité de sortie du modèle vers les couches proches de l'extrémité d'entrée du modèle.

En effet, après que le modèle RNN a généré le vecteur de sortie h_T,l'algorithme de "Backward propagation" est utilisé pour calculer le gradient.

Le gradient est utilisé pour mettre à jour les paramètres du modèle comme suite :

$$W \leftarrow W - \alpha \frac{\partial E}{\partial W}, \tag{2.6}$$

où $\frac{\partial E}{\partial W}$ est l'erreur de prédiction du modèle : $\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial E_t}{\partial W}$. On continue ce processus de propagation du gradient jusqu'à la fin de l'apprentissage.

Supposons qu'on ait T temps d'apprentissage, donc pour chaque étape k d'apprentissage, le gradient est donné par :

$$\begin{aligned}\frac{\partial E_k}{\partial W} =& \frac{\partial E}{\partial h_k}\frac{\partial h_k}{\partial c_k}....\frac{\partial c_2}{\partial c_1}\frac{\partial c_1}{\partial W},\\ =& \frac{\partial E}{\partial h_k}\frac{\partial h_k}{\partial c_k}\left(\Pi_{t=2}^{k}\frac{\partial c_t}{\partial c_{t-1}}\right)\frac{\partial c_1}{\partial W}.\end{aligned} \tag{2.7}$$

Et puisque :

$$c_k = \alpha(W_{rec}c_{k-1} + W_{in}x_k),$$

et donc la dérivée de c_k est égale à :

$$\begin{aligned}\frac{\partial c_k}{\partial c_{k-1}} =& \alpha'(W_{rec}c_{k-1} + W_{in}x_k)\frac{\partial}{\partial c_{k-1}}(W_{rec}c_{k-1} + W_{in}x_k),\\ =& \alpha'(W_{rec}c_{k-1} + W_{in}x_k)W_{rec}.\end{aligned} \tag{2.8}$$

D'aprés l'équation (2.7) et (2.8) on trouve le formule du gradient :

$$\frac{\partial E_k}{\partial W} = \frac{\partial E}{\partial h_k}\frac{\partial h_k}{\partial c_k}\left(\Pi_{t=2}^{k}\alpha'(W_{rec}c_{t-1} + W_{in}x_t)\right)W_{rec})\frac{\partial c_1}{\partial W}. \tag{2.9}$$

On remarque pour chaque itération k on multiplie par le poids W_{rec}.
Alors si W_{rec} est inférieur à 1 on aura :

$$\Pi_{t=2}^{t=k}\alpha'(W_{rec}c_{t-1} + W_{in}x_t)W_{rec} \to 0,$$

donc

$$\frac{\partial E_t}{\partial W} \to 0,$$

d'où

$$W \leftarrow W - \alpha\frac{\partial E}{\partial W} \simeq W.$$

Il n'y aura donc pas de mise à jour des poids du réseau si k est grand. Pour pallier le problème du "vanishing gradient", S. Hochreiter et al.[31] ont proposé le LSTM en 1997, puis le modèle de LSTM a été amélioré dans l'article de Gers et al.[32] en 2000.

2.3.3 Long short-term memory (LSTM)

Les réseaux LSTM permettent d'apprendre les dépendances à long terme. Ils sont explicitement conçus pour éviter le problème de la dépendance à long terme. Un réseau LSTM possède trois portes qui mettent à jour et contrôlent les états des cellules : la porte d'oubli, la porte d'entrée et la porte de sortie. La porte d'oubli (voir la figure 2.7) est responsable de la décision de laisser passer l'information.
Elle détermine quelles informations de l'état de cellule précédent h_{t-1} doivent être supprimées ou conservées dans l'état de cellule actuel h_t. Elle est définie par l'équation suivante :

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \tag{2.10}$$

où :

- L'élément W_f est la matrice de poids pour la porte d'oubli.
- L'élément $[h_{t-1}, x_t]$ désigne la concaténation de l'entrée actuelle x_t et de l'état caché précédent h_{t-1}.
- b_f est le biais associé de la porte d'oubli.
- σ est la fonction sigmoïde.

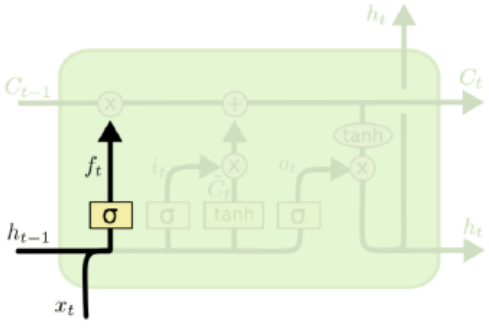

FIGURE 2.7 – *Porte d'oubli [5]*

La porte d'entrée (voir la figure 2.8) contrôle les nouvelles informations qui seront encodées dans l'état de la cellule, compte tenu des nouvelles informations d'entrée.

L'information est régulée à l'aide de la fonction sigmoïde et filtre les valeurs à retenir de la même manière que la porte d'oubli, en utilisant les entrées h_{t-1} et x_t. La porte d'entrée est calculée en appliquant une fonction d'activation sigmoïde à une combinaison de l'entrée actuelle x_t et de l'état caché précédent h_{t-1}. Cela produit une sortie i_t qui varie entre 0 et 1.
La formule pour la porte d'entrée est :

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \tag{2.11}$$

où

- W_i est la matrice de poids pour la porte d'entrée.
- $[h_{t-1}, x_t]$ désigne la concaténation de l'entrée actuelle x_t et de l'état caché précédent h_{t-1}.
- b_i est le biais associé de la porte d'entrée.
- σ est la fonction sigmoïde.

En plus de la porte d'entrée, un vecteur d'état de cellule candidat $\hat{C}_t$ est calculé pour déterminer quelles nouvelles informations peuvent être ajoutées. Ce vecteur est obtenu en appliquant la fonction tangente hyperbolique à la combinaison.

$$\hat{C}_t = \tanh\left(W_c[h_{t-1}, x_t] + b_c\right), \tag{2.12}$$

où W_c et b_c sont respectivement les poids et le biais associés au cellule candidate.
Une fois que les valeurs de la porte d'entrée i_t et de l'état de cellule candidat $\hat{C}_t$ ont été calculées, l'état de cellule c_t est mis à jour en combinant ces informations :

$$c_t = f_t c_{t-1} + i_t \hat{C}_t, \tag{2.13}$$

où f_t est la sortie de la porte d'oubli.

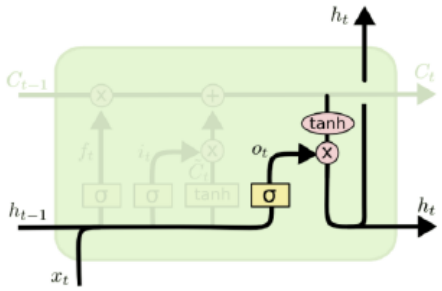

FIGURE 2.8 – *Porte d'entrée [5]*

La porte de sortie (voir la figure 2.9) contrôle les informations codées dans l'état de la cellule qui sont envoyées au réseau d'entrée au prochain pas de temps, par l'intermédiaire du vecteur de sortie h_t. Tout d'abord, un vecteur est généré en appliquant la fonction tanh à la cellule. Ensuite, l'information est régulée à l'aide de la fonction sigmoïde et filtrée par les valeurs à retenir à l'aide des entrées h_{t-1} et x_t. Enfin, les valeurs vectorielles et les valeurs régulées sont multipliées et envoyées en tant que sortie et entrée à la cellule suivante. L'équation de la porte de sortie est la suivante :

$$\begin{aligned} h_t =& o_t \otimes \tanh(c_t), \\ \text{où } o_t =& \sigma(W_o[h_{t-1}, x_t] + b_o). \end{aligned} \tag{2.14}$$

Où

— L'élément W_o représente la matrice de poids associée à la porte de sortie.
— L'élément $[h_{t-1}, x_t]$ désigne la concaténation de l'entrée actuelle x_t et de l'état caché précédent h_{t-1}.
— L'élément b_o est le biais associé à la porte de sortie.
— c_t est la cellue actuel qui contient des informations cumulées au fil du temps, servant de mémoire à long terme pour le LSTM.

— $\otimes$ est le produit élément par élément.

— σ est la fonction sigmoïde.

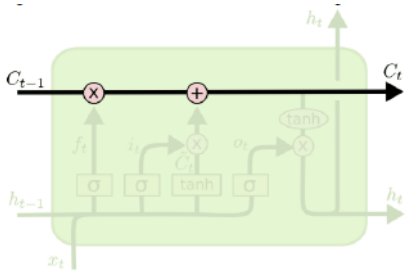

FIGURE 2.9 – *Porte de sortie [5]*

2.4 Fonction d'évaluation des modèles

L'évaluation des modèles d'apprentissage automatique et d'apprentissahe pronfond est une étape cruciale pour mesurer la performance d'un modèle et comprendre sa capacité à généraliser sur des données non vues. Il existe plusieurs fonctions d'évaluation selon les types de tâches (classification ou régression) et les caractéristiques du modèle.

Accuracy

L'*Accuracy* est le pourcentage de prédictions correctes parmi toutes les prédictions effectuées.

$$Accuracy = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}}.$$

Elle est simple à comprendre et à utiliser mais elle peut être trompeuse dans des cas de données déséquilibrées (où une classe domine fortement les autres).

Précision, Rappel et F1-score

Ces métriques sont particulièrement utiles lorsque les classes sont déséquilibrées.

La précision est la proportion des données classées positivement qui le sont réellement.

$$Précision = \frac{\text{Vrais PositifS}}{\text{Vrais Positifs} + \text{Faux Positifs}}.$$

Le rappel est la proportion des données positives correctement identifiées par le modèle.

$$Rappel = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Négatifs}}.$$

Le F1-Support correspond à la moyenne harmonique entre la précision et le rappel. Il équilibre la précision et le rappel, particulièrement utile lorsqu'on a des classes déséquilibrées.

$$F1 - Support = 2\frac{Précision * Rappel}{Précision + Rappel}.$$

Fonction de perte (Loss)

Elle mesure la différence entre les prédictions d'un modèle et les valeurs réelles. L'objectif de l'entraînement est de minimiser la perte afin d'améliorer la performance du modèle.

La fonction de perte utilisé dans ce mémoire est l'erreur quadratique moyenne (MSE). Elle est definit comme suit :

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2.$$

La MSE pénalise les erreurs importantes en les élevant au carré, ce qui signifie qu'elle est très sensible aux valeurs aberrantes (outliers) et son objectif est de réduire cette erreur à zéro.

Chapitre 3

Analyse des données

3.1 Description des données

Les données de cette étude ont été obtenues auprès du Centre de métallurgie du Québec en collaboration avec Agnico Eagle Mine Goldex. Elles s'étendent de 2016 à 2023.

La base de données est composée de seize variables de processus qui sont collectées toutes les cinq minutes et de huit variables d'épaisseur de conduite qui sont mesurées à l'aide d'une sonde installée dans la conduite et recollectées dans un délai de 24 heures ou plus. Le tableau 3.1 montre la distribution des données de processus ainsi que leur moyenne et leur écart-type. Le nombre d'enregistrements reçus pour les seize variables est de 179989 et de 635 enregistrements des mesures d'épaisseur de chacun des huit pipelines. L'épaisseur nominale du pipeline est comprise entre 5,5 et 6,5 mm. La détermination de la variation de l'épaisseur du tuyau a été effectuée grâce à l'analyse par ultrasons. Les mesures ont été effectuées à l'aide de la jauge ultrasonique de mesure d'épaisseur MMX-6 DL de Dakota Ultrasonics, fournie par Stone Tucker (http ://www.stone-tucker.com/en/ultrasonics/dakota-mmx6-ultrasonic-thickness-gauge).

Avant utilisation, la jauge a été calibrée en utilisant un bloc étalon de différentes épaisseurs. Pour effectuer les mesures d'épaisseur, la circonférence du tuyau a été di-

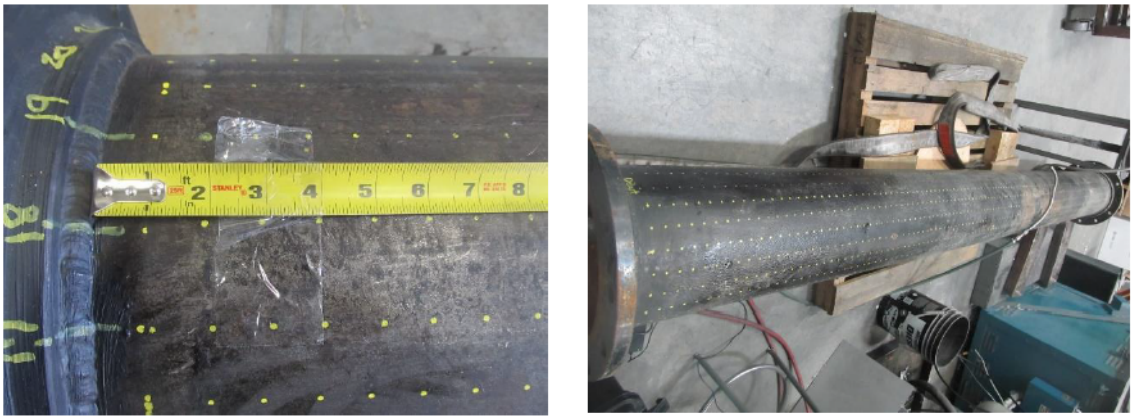

FIGURE 3.1 – *Étape de marquage (gauche) et vue d'ensemble (droite) du tuyau témoin*

visée en 24 repères équidistants, à partir desquels des lignes ont été tracées sur toute la longueur du tuyau. Des points de repérage espacés d'un pouce ont été marqués le long de ces 24 lignes. Au total, 125 points de repérage ont été marqués sur chaque ligne, numérotés de 1 (début) à 125 (fin), soit un total de 3000 points de repérage (125 x 24) sur la surface du tuyau pour les mesures d'épaisseur (voir figure 3.1).

Secteur	Paramètre	Moyenne	Ecart-type
Alimentation	Tonnage Sag	337,88	61,74
Secteur de Flotation	temperature pulpe flotation	25,4	5,89
	pH flottation	9,08	0,26
Pipeline	Debit residue	431,14	113,4
	% solid residue	24,98	15,7
	Residue TPH calcul	156,25	132,02
	Pression Km 0	2095	737,3
	Temperature Km 0	18,89	6,7
	Pression Km 14	430,36	446,66
	Temperature Km 14	18,09	19,87
Rivière du Thompson	flow rate m3/h	182,6	106,65
	Temperature	11,01	6,59
Bassin Sédimentaire	flow rate m3/h	70,27	15,99
	Temperature	12,78	7,55
Parc du sud	flow rate m3/h	166,4	101,6
	Temperature	7,8	6,3

TABLE 3.1 – Les seize variables du processus, leur moyenne et leur écart-type

La base de données contient des erreurs et des valeurs manquantes. Pour utiliser un modèle de l'apprentissage automatique ou de l'apprentissage profond, on va d'abord faire une étape de prétraitement des données.

3.2 Analyse descriptive

La figure 3.2 montre un diagramme en boîte de chacune des huit épaisseurs. Les données varient entre 4 et 7 mm, à l'exception du diagramme en boîte pour l'épaisseur 1, qui présente des mesures d'épaisseur supérieures à 40 et d'autres égales à 0 (Voir figure3.3, ce qui indique qu'il y a des erreurs d'entrée pour l'épaisseur 1. Pour traiter ces valeurs manquantes ou aberrantes, nous les remplacerons par une moyenne mobile afin de réduire le bruit et de maintenir la tendance des données d'épaisseur récues.

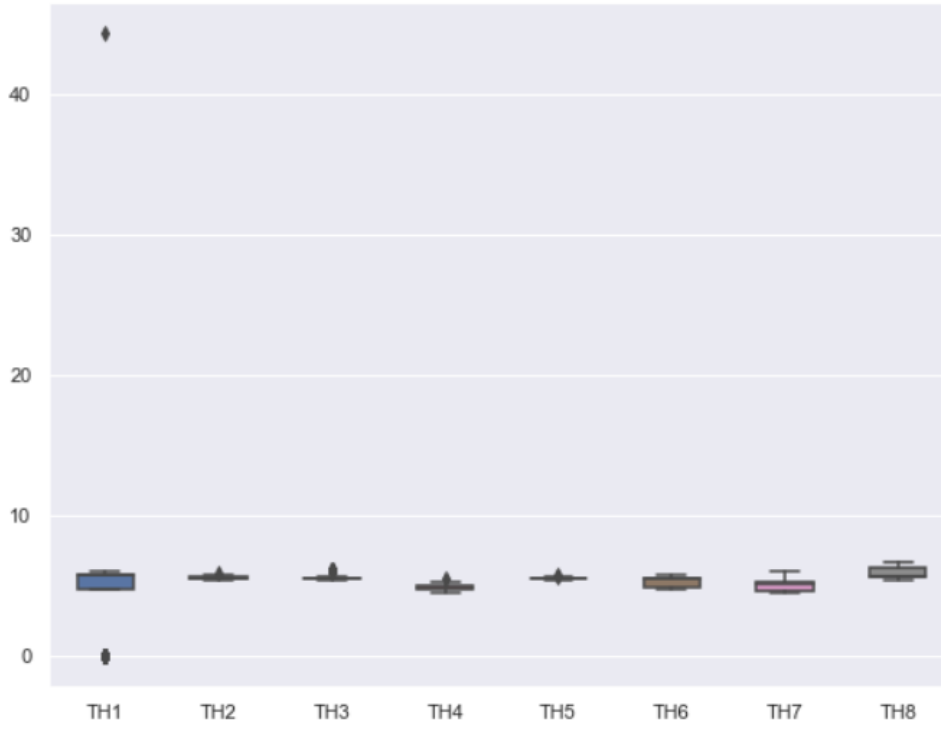

FIGURE 3.2 – *Diagramme en boîte de l'épaisseur de huit*

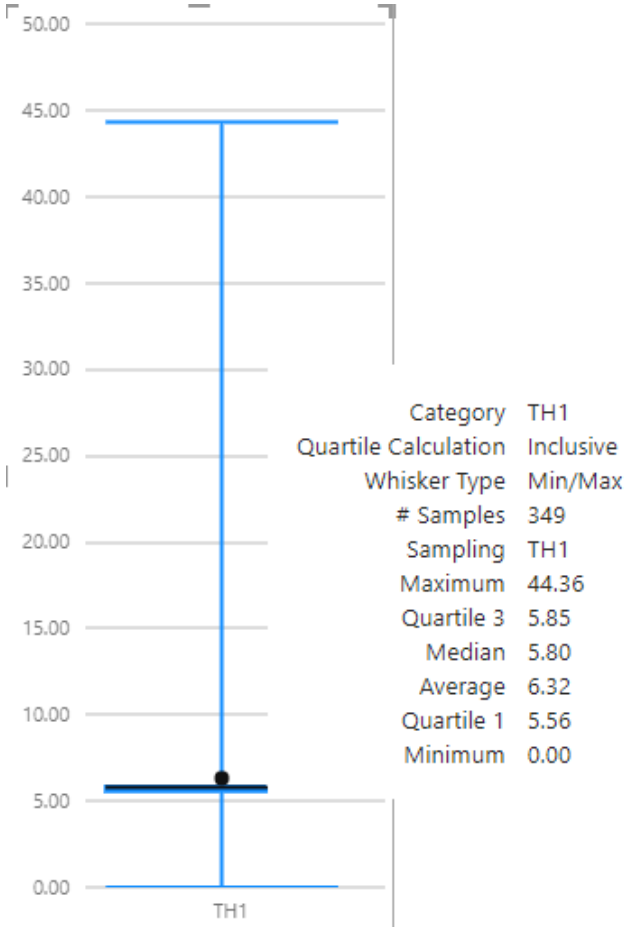

FIGURE 3.3 – *Diagramme en boîte de l'épaisseur de 1*

La figure 3.4 montre l'évolution des données d'épaisseur en fonction des dates de collecte. Après application des différentes transformations, les valeurs obtenues varient entre 4mm et 7mm. Ce qui correspond à la variation des autres épaisseurs.

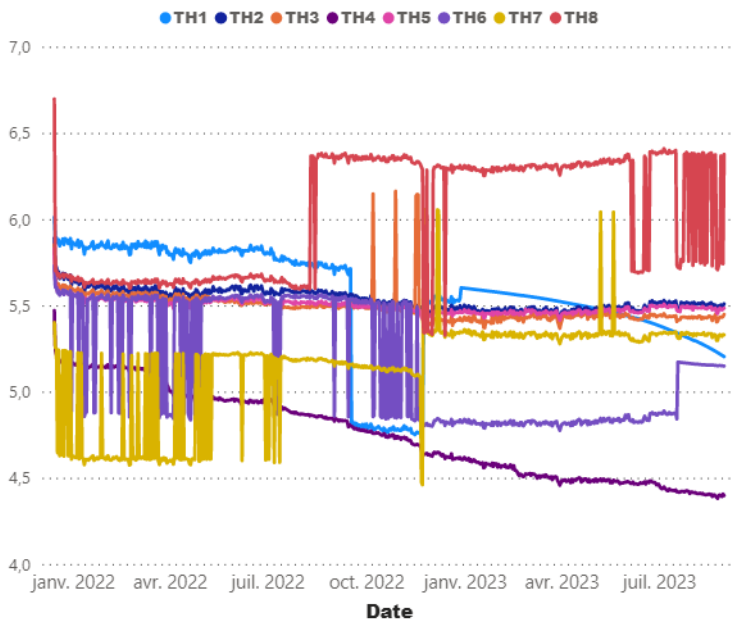

FIGURE 3.4 – *L'évolution des mesures d'épaisseur après l'application des transformations*

3.3 Analyse en Composantes Principales

La matrice de corrélation (voir Figure 3.5) montre que certaines mesures sont fortement corrélées. Par exemple, le pourcentage de solides dans le résidu (solide résidu) et les Tonnes Par Heure du résidu calculé (TPH résidu calculé) présentent une corrélation positive avec un coefficient de Pearson de 0,96. Il est donc intéressant de réduire les données en réalisant une analyse en composantes principales (ACP).

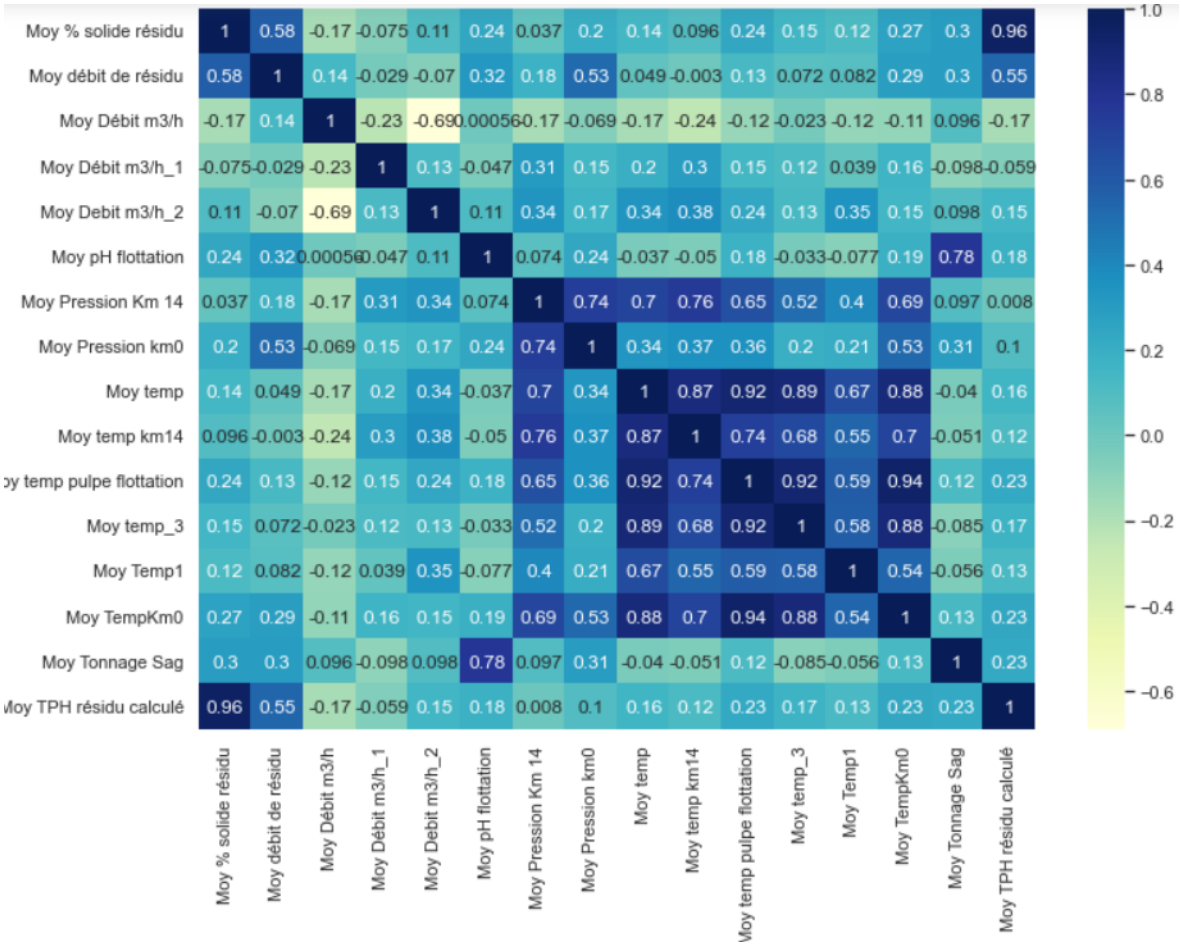

FIGURE 3.5 – Matrice de corrélation des seize variables

Après avoir appliqué l'ACP, l'analyse du pourcentage de variation expliquée montre qu'à partir de la neuvième dimension, nous avons déjà atteint 100% de la variance expliquée. Par conséquent, nous n'utiliserons que neuf dimensions pour le reste de l'analyse pour conserver un maximum d'information sur les données d'origine(voir figure 3.6).

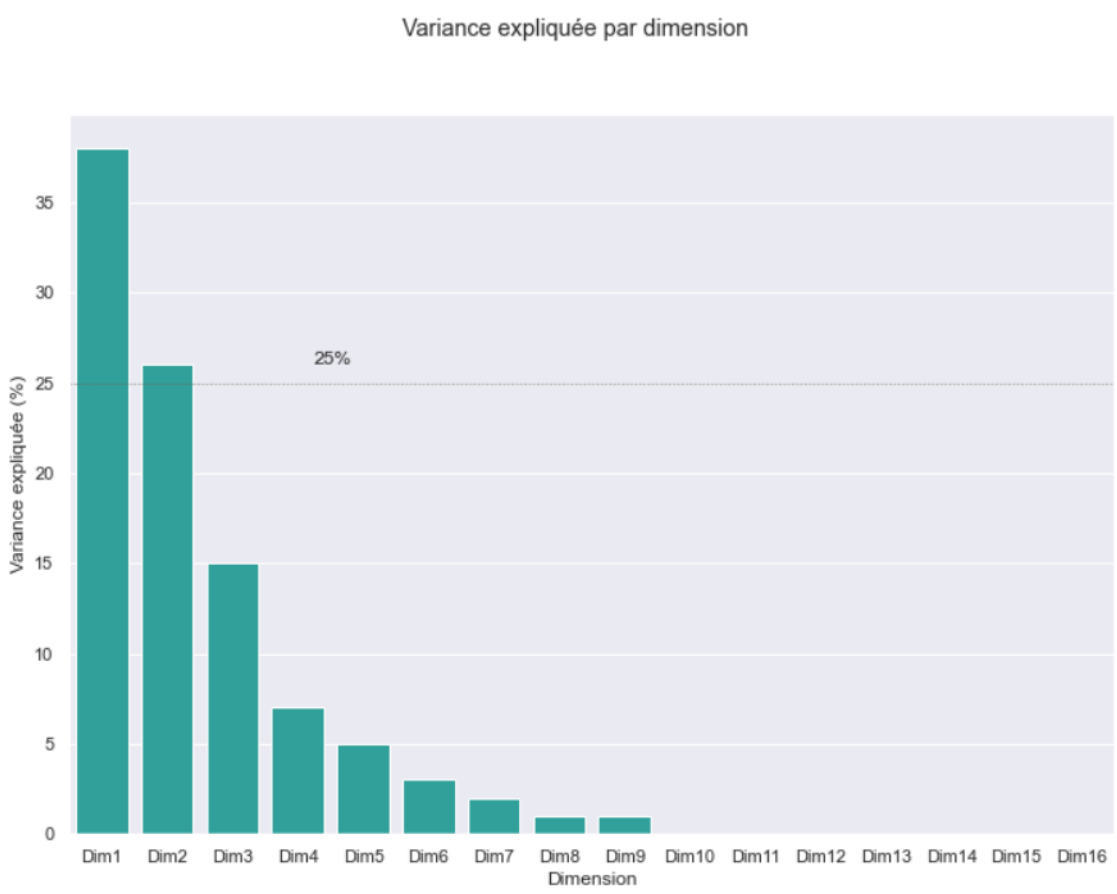

FIGURE 3.6 –

Histogramme du pourcentage de la variation expliquée par la dimension

Chapitre 4

Résultats et discussions

Ce chapitre présente les résultats des deux modèles univarié et multivarié issues de l'approche méthodologique.Ils font référence aux deux articles déja publiés [16] et [15]. Les deux modèles ont été entraînés sur 100 "epochs". ces dernières correspondent au nombre de fois où l'algorithme d'apprentissage parcourt l'intégralité du jeu de données d'entraînement afin d'ajuster les poids du réseau.

4.1 Analyse Univariée

Cette section se référe aux résultats de [16].
La première étape consiste à transformer les huit mesures d'épaisseur en une seule valeur pour travailler sur un modèle univarié. Pour cela, une méthode d'agrégation est utilisée, en calculant la moyenne quotidienne des huit épaisseurs (voir figure 4.1). Afin de garantir que le nombre d'enregistrements des variables de processus corresponde à celui des mesures d'épaisseur, nous agrégons également les variables de processus en calculant leurs moyennes journalières. Cela génère 635 enregistrements pour chaque

variable.

Après cette transformation, les modèles de classification (KNN, SVM et FA) sont appliqués pour classifier la moyenne des mesures des pipelines, tandis qu'un modèle LSTM est utilisé pour prédire la variation moyenne des huit épaisseurs sur les 63 prochains jours.

Pour effectuer une étude de classification de la moyenne des huit épaisseurs, une colonne "label" est créée et définie comme suit :

Si la moyenne des huit épaisseurs est inférieure à 5,5, label prend la valeur 1 ; Sinon, label prend la valeur 0.

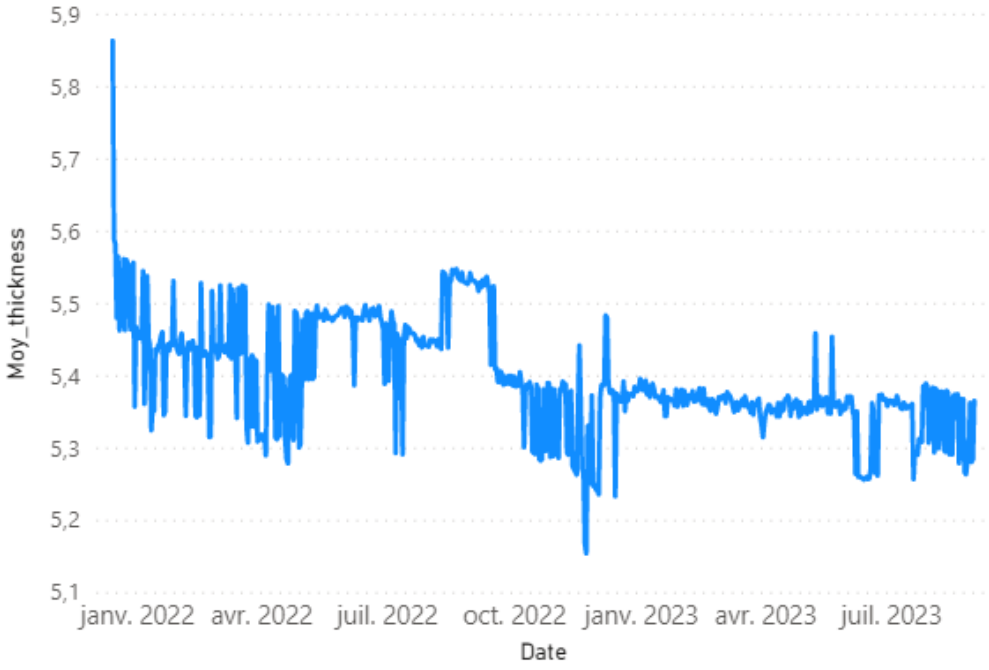

FIGURE 4.1 – *Évolution de la moyenne des huit épaisseurs en fonction du temps*

Le tableau 4.1 montre les performances des différents modèles d'apprentissage automatique mis en œuvre. Le modèle KNN offre une meilleure précision sur l'ensemble des données que les modèles SVM et RF. Cela signifie que le modèle KNN offre une meilleure classification des pipelines corrodés. Le rappel du modèle FA est égal à 1,00, ce qui signifie qu'il n'a produit aucun faux négatif. Cela s'explique par le fait que les

données issues des mesures de canalisations recoupées ne sont pas équilibrées. Il y a plus de 1 que de 0. C'est pourquoi, pour comparer les performances de ces modèles, nous allons nous baser sur le F1-Support, qui est une combinaison de la précision et du rappel. La Forêt Aléatoire présente le meilleur F1-Support, ce qui indique qu'elle est plus performante sur nos données que les deux autres modèles.
On observe que la différence entre le F1-Support du modèle de Forêt Aléatoire et celui du modèle SVM est de 0,001. Bien que cette différence soit minime, elle pourrait être significative en raison du nombre petit d'enregistrements disponibles.

Models	Precision	Rappel	F1-Support	Accuracy
SVM	0,964	0,973	0,968	0,941
FA	0,940	1,000	0,969	0,941
KNN	0,981	0,945	0,945	0,933

TABLE 4.1 – Comparaison des differents modèles d'apprentissage automatique

La figure 4.2 montre l'évolution de la fonction de perte du modèle LSTM au fil du temps. La diminution de la fonction de perte indique que le modèle LSTM utilisé a minimisé les erreurs de prédiction au cours de la formation.

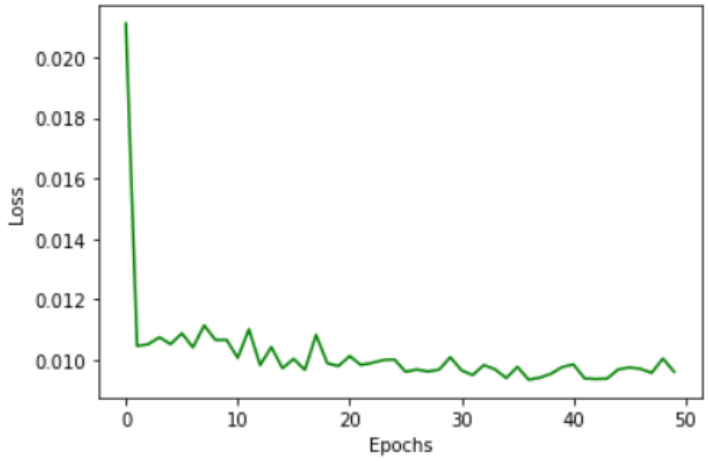

FIGURE 4.2 – *Fonction de perte du modèle LSTM univarié*

Les mesures d'épaisseur prédites par le modèle LSTM, illustrées a la figure 4.3 sont au dessous de l'intervalle d'épaisseur nominale [5,5 mm ; 6,5mm], ce qui signifie que la canalisation est déjà corrodée. Comme le LSTM surveille l'évolution des mesures passées, les canalisations correspondantes devront être remplacées ou traitées. (voir Figure 4.3)

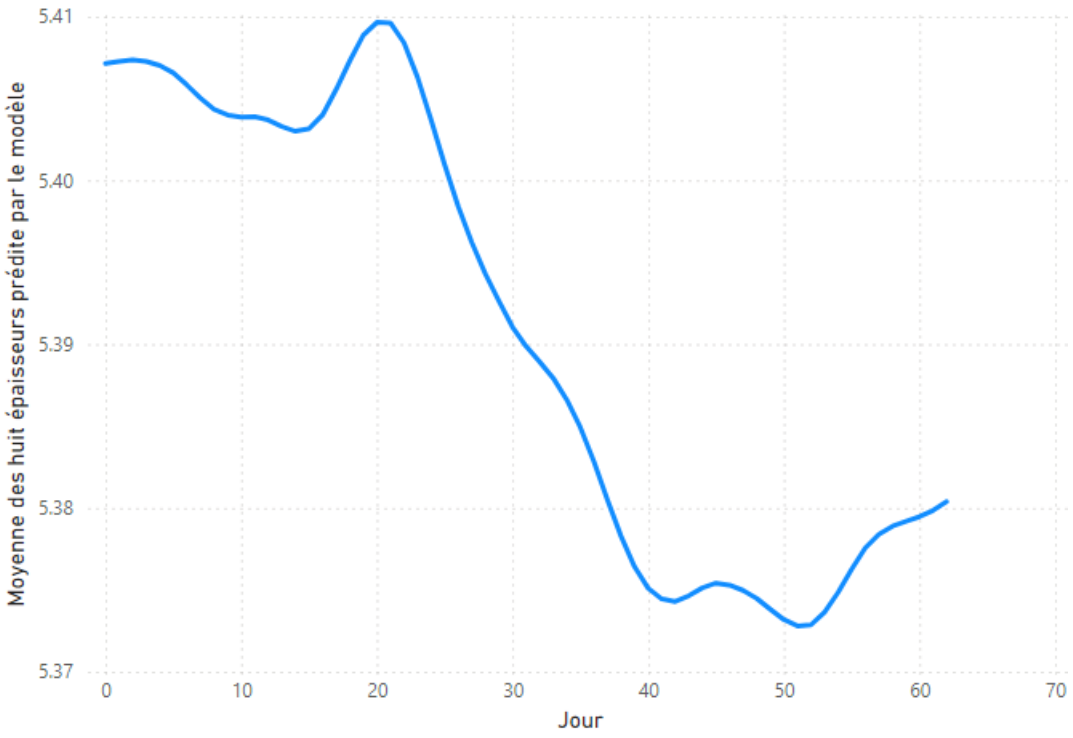

FIGURE 4.3 – *Mesure de l'épaisseur moyenne prédite par le modèle LSTM univarié*

4.2 Analyse Multivariée

Cette section se référe aux résultats obtenus dans [15].
Les huit mesures d'épaisseur des pipelines sont prises en compte, et un modèle LSTM multivarié est utilisé pour prédire la variation des épaisseurs des huit pipelines sur les 100 prochains jours.

Le modèle utilise l'intégralité des données d'origine, soit 179 989 enregistrements pour les variables de processus et 635 enregistrements pour les mesures d'épaisseur de chacun des huit pipelines.

4.2.1 Évaluation du modèle multivarié

La figure 4.4 montre l'évolution de la fonction de perte (Loss) du modèle LSTM au fil du temps. La diminution de cette fonction de perte indique que le modèle LSTM a réussi à minimiser les erreurs de prédiction au cours de l'apprentissage. La performance du modèle peut également être évaluée à travers sa précision.

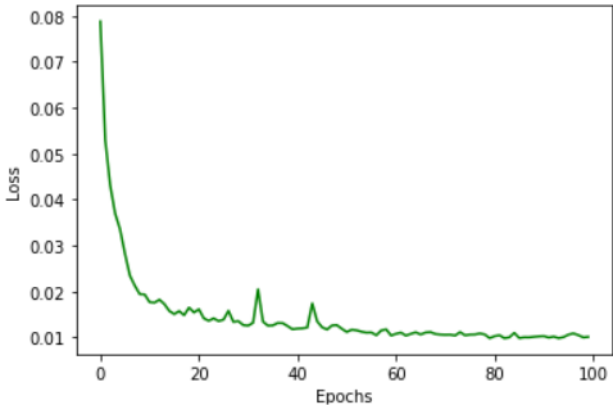

FIGURE 4.4 – *Fonction de perte du modèle LSTM*

Au cours de la formation, le modèle utilisé a atteint une précision de 80 %, ce qui signifie que 80% des valeurs prédites par le modèle sont correctes (figure 4.5).

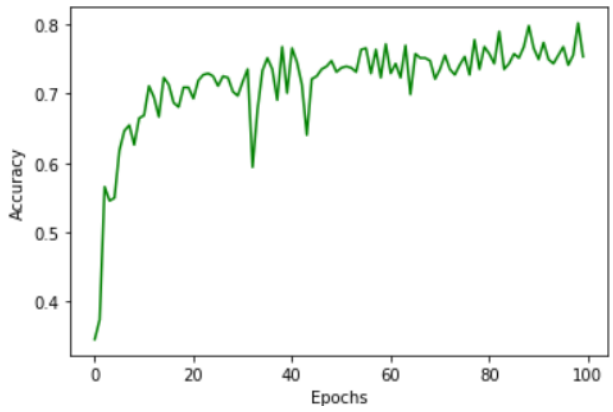

FIGURE 4.5 – *Evolution de l'accuracy du modèle de LSTM*

4.2.2 Résultats et discussion

Le tableau 4.2 montre que l'application de l'ACP sur les seize variables du procédé améliore les performances du modèle. L' accuracy passe de 76 % à 80 %, et la perte diminue à 0,100. Cela s'explique par le fait que l'ACP permet de sélectionner les composantes les plus significatives, ce qui aide le modèle LSTM à se concentrer sur les caractéristiques les plus pertinentes des données, améliorant ainsi la qualité des prédictions tout en réduisant le bruit. En se focalisant sur un sous-ensemble plus pertinent de caractéristiques, le LSTM peut obtenir une meilleure précision tout en diminuant le risque de surajustement (overfitting), en évitant de traiter des variables redondantes ou peu informatives.

Model	Accuracy(%)	Loss
LSTM	76	0,0132
ACP+LSTM	80	0,0100

TABLE 4.2 – Comparaison du modele LSTM utilisé sur les deux bases de données

Une fois le modèle entraîné, nous le testerons sur les données de test. Pour ce

faire, nous allons prédire l'évolution des épaisseurs au cours des 100 prochains jours. La figure 4.6 montre l'évolution de l'épaisseur des pipelines 1, 2, 3, 5 et 6 prédite par le modèle LSTM pour les cent prochains jours. Les valeurs d'épaisseur des pipelines 1 et 6 sont inférieures à la valeur d'épaisseur nominale (entre 5,5 et 6,5) avant les 40 jours, ce qui signifie que le pipeline correspondant est affecté par la corrosion et doit être surveillé. Après 40 jours, les mesures d'épaisseur des conduites 1 et 6 augmentent, ce qui signifie que les deux conduites (1 et 6) doivent être traitées.

La courbe d'évolution des mesures d'épaisseur prédites par le modèle LSTM pour les pipelines (2, 3 et 5) montre une variation presque constante entre 5,4 mm et 5,6 mm, ce qui est juste au-dessus de l'épaisseur minimale de 5,5 mm. Cela suggère que des inspections régulières devraient être effectuées au cours des 100 prochains jours.

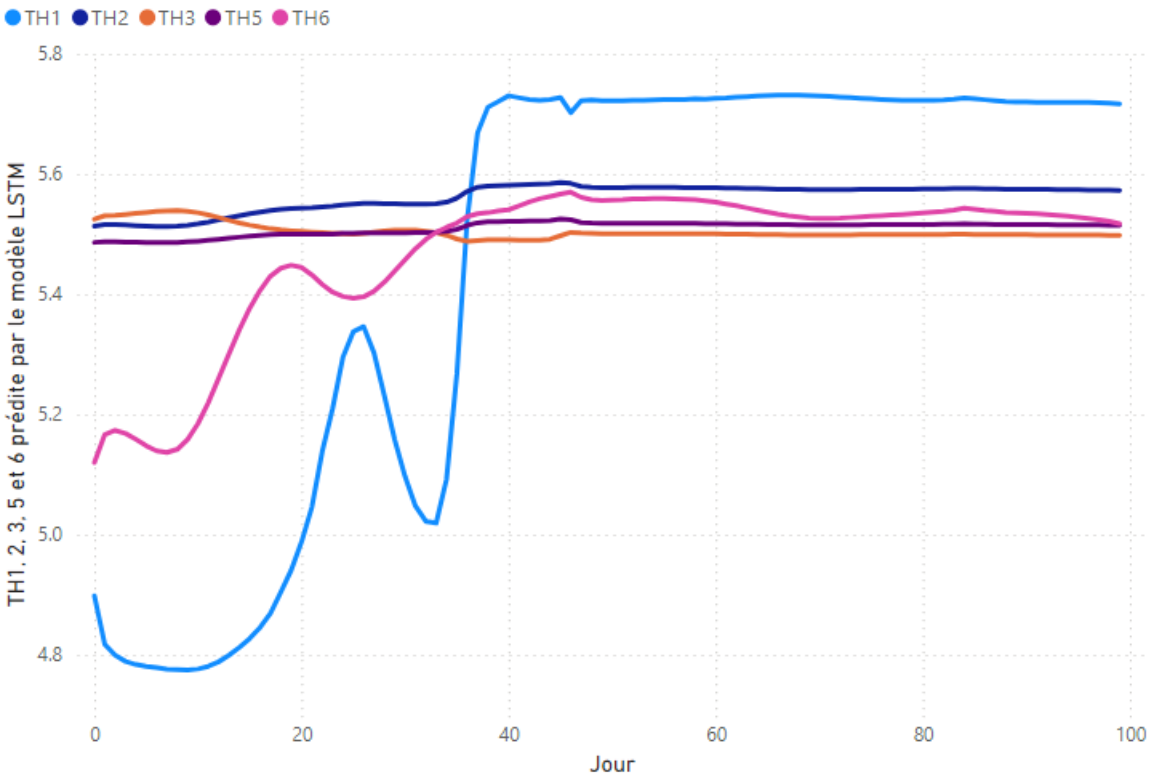

FIGURE 4.6 – *Épaisseur des tuyaux 1, 2, 3, 5 et 6 prédite par le modèle multivarié pour les 100 prochains jours*

Les mesures d'épaisseur prédites par le modèle pour les conduites 4 et 7 (voir figure 4.7) sont supérieures à l'épaisseur nominale (5,5mm), ce qui signifie que les conduites 4 et 7 sont déjà corrodées, puisque le LSTM surveille l'évolution des mesures passées, les conduites correspondantes devront être remplacées ou traitées.

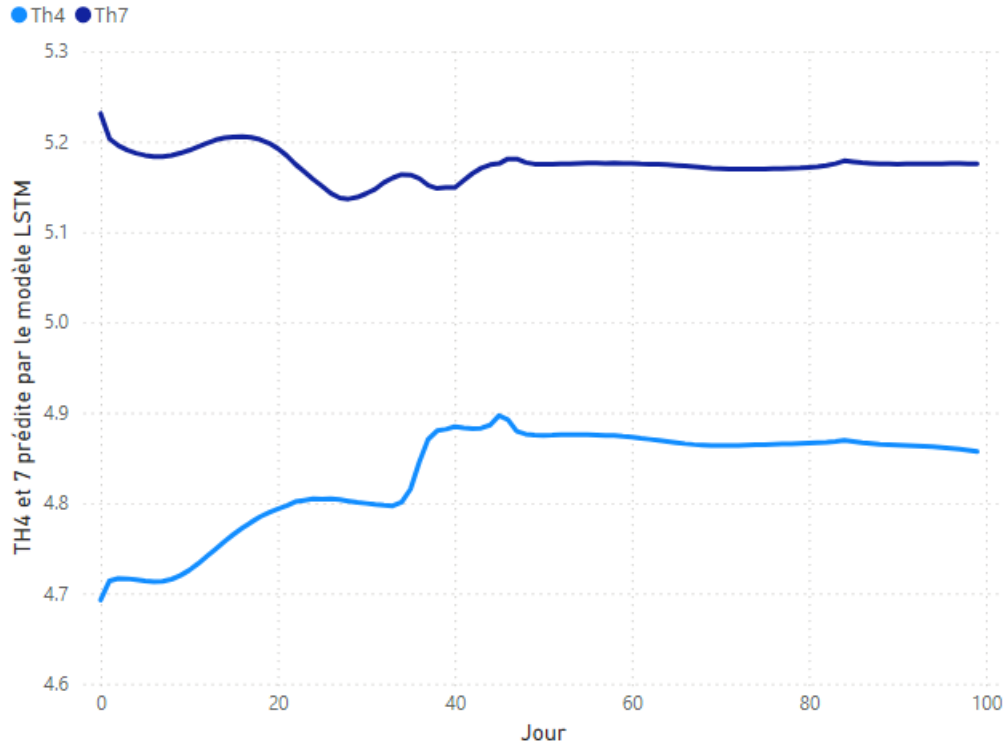

FIGURE 4.7 – *Epaisseur des tuyaux 4 et 7 prédite par le modèle multivarié*

L'évolution de la courbe d'épaisseur 8 prévue par le modèle montre une chute rapide de l'épaisseur 8 jusqu'au 40ème jour, ce qui signifie qu'il y a un risque de perte d'épaisseur due à la corrosion correspondante de la canalisation. Même si les mesures d'épaisseur de la canalisation correspondante se situent entre les épaisseurs nominales, elle doit être traitée avant les 40 jours pour éviter l'augmentation de la corrosion qui correspondra à l'augmentation de la mesure de l'épaisseur de la canalisation 8.

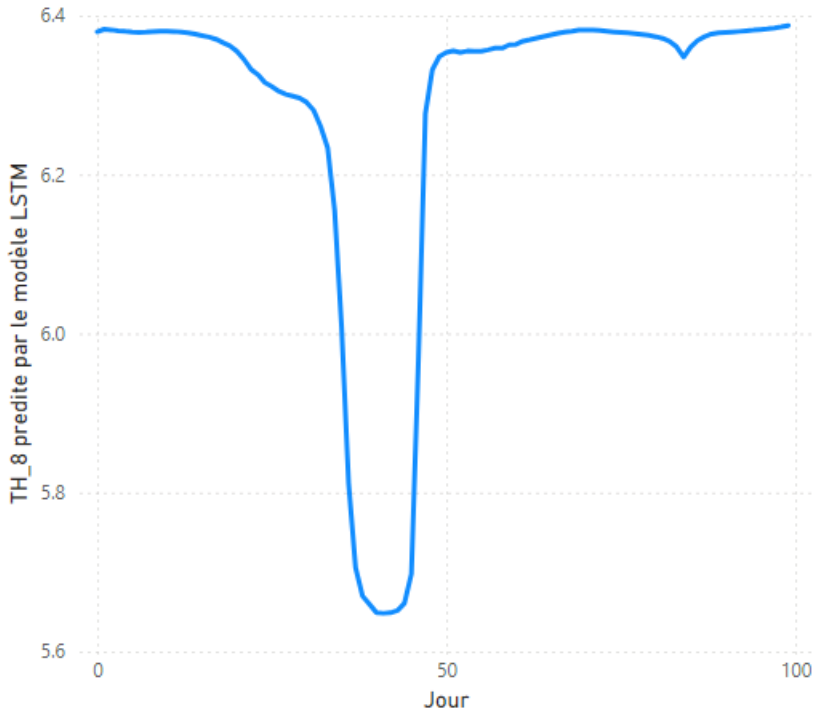

FIGURE 4.8 – *Epaisseur des tuyaux 8 prédite par le modèle multivarié*

Conclusion et perspectives

La corrosion des pipelines engendre d'importants défis économiques et environnementaux. En raison de la complexité inhérente des données relatives à la corrosion, les approches de modélisation statistique simplistes peinent à offrir une analyse exhaustive et rend difficile sa représentation dans un modèle mathématique unique. La défaillance des pipelines a captivé l'attention de multiples communautés de chercheurs en raison de son impact majeur sur l'économie mondiale, les risques de fuites, d'explosions et les périodes coûteuses d'arrêts. De nombreux efforts ont été déployés pour élaborer des modèles de prédiction de la corrosion, entraînant une abondance de publications.

L'analyse des données collectées de 2016 à 2023 sur les pipelines a permis de mettre en lumière les différentes variables influençant la corrosion et l'épaisseur des conduites. Après un prétraitement rigoureux des données pour analyser les erreurs et les valeurs manquantes, des modèles d'apprentissage automatique et d'apprentissage profond ont été appliqués pour prédire et classifier la variation de l'épaisseur des pipelines.

Dans notre étude, nous avons examiné deux approches : une approche univariée et une approche multivariée.

Dans l'approche univariée, l'agrégation des huit mesures d'épaisseur en une seule valeur quotidienne a permis de simplifier le modèle et d'appliquer des algorithmes de classification tels que KNN, SVM, et FA. Ces modèles ont montré des performances encourageantes, notamment le KNN, qui a offert une meilleure précision dans la classification des pipelines corrodés. Le modèle LSTM univarié a également été efficace

pour prédire la variation moyenne de l'épaisseur des pipelines sur une période de 63 jours, révélant que plusieurs pipelines nécessitent une intervention immédiate en raison de leur état de corrosion avancée.

L'approche multivariée a approfondi cette analyse en tenant compte des huit mesures d'épaisseur, et l'application de l'ACP a permis d'améliorer la performance du modèle LSTM en réduisant la dimensionnalité des données tout en conservant un maximum d'information pertinente. Cette méthode a conduit à une meilleure précision et à une réduction des erreurs de prédiction. Les résultats obtenus grâce au modèle LSTM multivarié indiquent que certains pipelines sont fortement corrodés et nécessitent un suivi régulier et des actions de maintenance pour éviter des défaillances critiques.

Cette étude montre que les modèles d'apprentissage automatique, en particulier les modèles LSTM combinés avec l'ACP, permettent de réaliser des prédictions précises sur l'état des pipelines. Ces outils offrent une méthode efficace pour surveiller l'évolution de la corrosion et planifier les interventions nécessaires afin de prévenir des défaillances graves dans les systèmes de conduite.

Les futures recherches se focaliseront sur l'intégration de données reçues et la combinaison d'autres modèles de traitement des séries temporelles avec la LSTM. Notre étude incorporera des modèles statistiques temporels tels que ARIMA et SARIMA, les intégrant à des techniques d'apprentissage profond pour améliorer les capacités prédictives du modèle.

Bibliographie

[1] A. Coşkuner, Ö. Rençber et Z. Çelik, « The factors that affects the profitability in real estate investment trust companies : Comparison of turkey and malaysia », *Trends in Business and Economics*, vol. 38, no. 1, p. 2–11, 2024.

[2] A. Binbusayyis, H. Alaskar, T. Vaiyapuri et M. Dinesh, « An investigation and comparison of machine learning approaches for intrusion detection in iomt network », *The Journal of Supercomputing*, vol. 78, no. 15, p. 17403–17422, 2022.

[3] G. M. Hadjidemetriou, « Roadway pavement condition assessment utilizing computer vision, machine learning and entropy », 2018.

[4] J. Cheng et S. Yang, « Data mining applications in evaluating mine ventilation system », *Safety science*, vol. 50, no. 4, p. 918–922, 2012.

[5] Oinkina et Hakyll, « Comprendre les réseaux lstm », 2015.

[6] M. S. El-Abbasy, A. Senouci, T. Zayed, F. Mirahadi et L. Parvizsedghy, « Artificial neural network models for predicting condition of offshore oil and gas pipelines », *Automation in Construction*, vol. 45, p. 50–65, 2014.

[7] C. Lam et W. Zhou, « Statistical analyses of incidents on onshore gas transmission pipelines based on phmsa database », *International Journal of Pressure Vessels and Piping*, vol. 145, p. 29–40, 2016.

[8] A. Valor, F. Caleyo, J. M. Hallen et J. C. Velázquez, « Reliability assessment of buried pipelines based on different corrosion rate models », *Corrosion Science*, vol. 66, p. 78–87, 2013.

[9] H. M. Hussein Farh, M. E. A. Ben Seghier, R. Taiwo et T. Zayed, « Analysis and ranking of corrosion causes for water pipelines : a critical review », *npj Clean Water*, vol. 6, no. 1, p. 65, 2023.

[10] J. R. Scully, N. D. Budiansky, Y. Tiwary, A. S. Mikhailov et J. L. Hudson, « An alternate explanation for the abrupt current increase at the pitting potential », *Corrosion Science*, vol. 50, no. 2, p. 316–324, 2008.

[11] L. Zhang, Y. Du et A. Cao, « The design of natural gas pipeline inspection robot system », *in 2015 IEEE International Conference on Information and Automation*, p. 843–846, IEEE, 2015.

[12] J. Tian, M. Gao et J. Li, « Corrosion detection system for oil pipelines based on multi-sensor data fusion by improved simulated annealing neural network », *in 2006 International Conference on Communication Technology*, p. 1–5, IEEE, 2006.

[13] A. K. Dia, N. Ghazzali et A. Gambou Bosca, « Unsupervised neural network for data-driven corrosion detection of a mining pipeline », *in The International FLAIRS Conference Proceedings*, vol. 35, 2022.

[14] A. K. Dia, A. G. Bosca et N. Ghazzali, « Walk-through corrosion assessment of slurry pipeline using machine learning », *International Journal of Corrosion*, vol. 2024, no. 1, p. 9427747, 2024.

[15] K. M. Sow et N. Ghazzali, « Developing a predictive model using multivariate analysis and long short-term memory (lstm) to assess corrosion degradation in mining pipeline thickness. », *in The International FLAIRS Conference Proceedings*, vol. 37, 2024.

[16] K. M. Sow et N. GHAZALLI, « Machine learning-based classification and prediction to assess corrosion degradation in mining pipelines », *in 18th International Federation of Classification Societies*, 2024.

[17] M. Aghaaminiha, R. Mehrani, M. Colahan, B. Brown, M. Singer, S. Nesic, S. M. Vargas et S. Sharma, « Machine learning modeling of time-dependent corrosion rates of carbon steel in presence of corrosion inhibitors », *Corrosion Science*, vol. 193, p. 109904, 2021.

[18] M. F. Sheikh, K. Kamal, F. Rafique, S. Sabir, H. Zaheer et K. Khan, « Corrosion detection and severity level prediction using acoustic emission and machine learning based approach », *Ain Shams Engineering Journal*, vol. 12, no. 4, p. 3891–3903, 2021.

[19] A. Hendi, A. Behravan, D. Mostofinejad, S. M. Moshtaghi et K. Rezayi, « Implementing ann to minimize sewage systems concrete corrosion with glass beads substitution », *Construction and Building Materials*, vol. 138, p. 441–454, 2017.

[20] B. C. Mateus, M. Mendes, J. T. Farinha, R. Assis et A. M. Cardoso, « Comparing lstm and gru models to predict the condition of a pulp paper press », *Energies*, vol. 14, no. 21, p. 6958, 2021.

[21] A. G. Salman, Y. Heryadi, E. Abdurahman et W. Suparta, « Single layer & multi-layer long short-term memory (lstm) model with intermediate variables for weather forecasting », *Procedia Computer Science*, vol. 135, p. 89–98, 2018.

[22] D. M. Nelson, A. C. Pereira et R. A. De Oliveira, « Stock market's price movement prediction with lstm neural networks », *in 2017 International joint conference on neural networks (IJCNN)*, p. 1419–1426, Ieee, 2017.

[23] L. Di Persio et O. Honchar, « Artificial neural networks approach to the forecast of stock market price movements », *International Journal of Economics and Management Systems*, vol. 1, 2016.

[24] D. Karmiani, R. Kazi, A. Nambisan, A. Shah et V. Kamble, « Comparison of predictive algorithms : backpropagation, svm, lstm and kalman filter for stock market », *in 2019 amity international conference on artificial intelligence (AICAI)*, p. 228–234, IEEE, 2019.

[25] K. Chen, Y. Zhou et F. Dai, « A lstm-based method for stock returns prediction : A case study of china stock market », *in 2015 IEEE international conference on big data (big data)*, p. 2823–2824, IEEE, 2015.

[26] X. Li, M. Guo, R. Zhang et G. Chen, « A data-driven prediction model for maximum pitting corrosion depth of subsea oil pipelines using ssa-lstm approach », *Ocean Engineering*, vol. 261, p. 112062, 2022.

[27] N. Ghazzali, « Notes de cours sur les méthodes d'analyse des données », *Université de Québec à Trois-Rivières*, 2022.

[28] L. Breiman, « Random forests », *Machine learning*, vol. 45, p. 5–32, 2001.

[29] K. Taunk, S. De, S. Verma et A. Swetapadma, « A brief review of nearest neighbor algorithm for learning and classification », *in 2019 international conference on intelligent computing and control systems (ICCS)*, p. 1255–1260, IEEE, 2019.

[30] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[31] S. Hochreiter et J. Schmidhuber, « Long short-term memory », *Neural computation*, vol. 9, no. 8, p. 1735–1780, 1997.

[32] F. A. Gers, J. Schmidhuber et F. Cummins, « Learning to forget : Continual prediction with lstm », *Neural computation*, vol. 12, no. 10, p. 2451–2471, 2000.

Annexe A

Article scientifique I

Machine Learning-Based Classification and Prediction to Assess Corrosion Degradation in Mining Pipelines

SOW Kalidou Moussa and GHAZZALI Nadia

Abstract The issue of pipeline failure has garnered considerable interest from various research communities due to its notable repercussions on the worldwide economy, as well as the risks associated with leaks, explosions, and expensive periods of downtime. This paper aims to build a model for classifying and predicting the corrosion degradation of a pipe used to transport water in mines by the Quebec Metallurgy Center. To this end, two types of models were developed: three binary classification models: SVM, RF, and KNN, yielding F1-measurements of 0.968, 0.969, and 0.945 respectively, and a time series model, LSTM, which, with a loss of less than 0.01, was able to predict average variations in pipeline thickness for 63 days.

Keywords: Machine Learning, Classification, Prediction, Pipeline Corrosion

1 Introduction

In the last ten years, substantial endeavors have been undertaken to address the challenge of pipeline corrosion through the application of statistical modeling, incorporating various machine-learning techniques.
As a result of the advancements in machine learning (ML) and deep learning (DL), there has been significant interest in data-driven model-based detection methods for pipeline erosion-corrosion monitoring.
Aghaaminiha et al.[1] use supervised machine learning methods to model measure-

SOW Kalidou Moussa
University of Quebec at Trois-Rivières, 3351 Bd des Forges, Trois-Rivières, QC G8Z 4M3, Canada, e-mail: Kalidou.Moussa.Sow@uqtr.ca

GHAZZALI Nadia
University of Quebec at Trois-Rivières, 3351 Bd des Forges, Trois-Rivières, QC G8Z 4M3, Canada,e-mail: Nadia.Ghazzali@uqtr.ca

ments of carbon steel corrosion rates as a time function. They compared different machine learning models and concluded that Random Forest performed better on their data with the mean squared error ranging from 0.005 to 0.093. Sheikh et al.[2] uses a hybrid technique that combines the detection of corrosion through acoustic emission signals from accelerated corrosion testing with machine learning techniques to accurately predict the corrosion severity levels. They applied decision trees, back propagation neural network, and radial basis function neural network on their data and obtained an accuracy of 90.4%, 94.57%, and 100% respectively. Hendi et al.[3] implemented a back propagation neural network model to minimize sewage system's concrete corrosion with glass beads substitution and to predict the mass-loss and volume-loss in the specimens. They obtained a mean error squared of 0.44 for the mass-loss and 1.18 for the volume-loss. Dia et al.[4] have applied an unsupervised neural network, self-organizing maps (SOM), to study the impact of corrosion assessed by periodic ultrasonic inspections. They combined SOM and hierarchical clustering to detect the extent of corrosion in a mining pipeline. Li et al.[5] combined the swarm intelligence optimization algorithm (SSA) and a LSTM model to predict the maximum pitting corro- sion depth of subsea oil pipelines. the comparison of their SSA-LSTM method with the LSTM alone shows that the new model SSA-LSTM performed superior in prediction ac- curacy and robustness which evaluation parameters are the smallest values in these models.
In a prior investigation, in our paper (Sow and al.)[6] accepted in 2024, our emphasis was on the multivariate aspect of the data outlined in section 3. However, in this study, our focus will shift to the univariate model of the data.

2 Theory and Formulation

2.1 Support Vector Machine (SVM)

SVM (Support Vector Machines) is a classification method proposed by Vapnik 1982 and aimed at finding a separating hyperplane while maximizing the margin between the two classes. To explain how SVM works, we consider a binary classification problem where the labels are defined as -1 and 1. We have a dataset composed of input feature vectors X and their corresponding class labels Y. The hyperplane equation is defined as follows:

$$\omega^T x + b = 0 \tag{1}$$

The vector ω represents the normal vector to the hyperplane, the parameter b in the equation represents the offset or distance of the hyperplane from the origin along the normal vector ω. The distance between a data point x_i and the decision boundary can be calculated as follows:

$$d_i = \frac{\omega^T x_i + b}{||\omega||} \tag{2}$$

where $||\omega||$ represents the Euclidean norm of the vector ω. For a linear SVM model, we seek to optimize the expression:

$$\min_{\omega,b} \frac{1}{2}\omega^T\omega = \min_{\omega,b} \frac{1}{2}||\omega||^2 \tag{3}$$

under constraint:

$$y_i(\omega^T x_i + b) \geq 1$$

. This is equivalent to minimizing the following Lagrage equation:

$$L(\omega, b, \lambda) = \frac{1}{2}||\omega||^2 - \sum_i^N \lambda_i(y_i(\omega^T x_i + b) - 1)) \tag{4}$$

U.C $\lambda_i \geq 0$

To predict new data, we determine its sign using the following formula:

$$\hat{y} = sign(\omega^T x_{new} + b) \tag{5}$$

Where $\omega = \sum_i^{NSV} \lambda_i y_i x_i$, $b = average(y_i - \omega^T x_i)$ and NSV represents the Number of support vectors.

2.2 Random Forest (RF)

The "random forest" algorithm was proposed by Leo Breiman and Adèle Cutler in 2001 [7]. It combines several decision trees in parallel, in a bagging-type approach, which reduces the variance of predictions from a single decision tree. This technique is simple to implement and delivers good results in terms of prediction quality on complex data, and in the presence of a large number of explanatory variables. A random forest is an aggregation of a large number of classification or regression trees. The randomness of the algorithm comes from the fact that the trees are built based on bootstrap samples. Bootstrap samples are generally obtained by drawing n observations from n in the initial sample N. In particular, another random aspect is introduced in the selection of variables at each stage in the construction of these trees (at each node, a subset of the variables is selected to determine the cut-off).

A random forest is a collection of decision tree classifiers $h_k(x, \theta_k), k = 1..N$ where the θ_k are randomly generated trees. The final result of this tree system is obtained by majority vote:

$$H = \arg\max_Y \sum_{i=1}^{i=k} I(h_i(x = Y))$$

2.3 k-nearest neighbors (KNN)

The k-nearest neighbors algorithm is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

2.4 Long Short-Term Memory (LSTM)

Long Short-Term Memory Networks (LSTM)[8] allow to learn long-term dependencies. They are explicitly designed to avoid the long-term dependency problem.
An LSTM network has three gates that update and control the states of the cells: the Forget gate, the Input gate, and the Output gate.
In the equations listed under the forget gate, input gate, and output gate in the diagram, h_{t-1} is the previous hidden state, x_t is the current input, W is the weight matrix, b is the bias, σ is the sigmoid function, $tanh$ is the hyperbolic tangent function, and $\otimes$ represents vector multiplication.
The Forget gate is responsible for deciding to let information pass. State 0 corresponds to "keep complete information" while state 1 represents "Totally get rid of the information". It is defined by the following equation:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{6}$$

The Input gate controls what new information will be encoded in the cell state, given the new input information. The information is regulated using the sigmoid function and filters the values to be retained in the same way as the forgetting gate, using the inputs h_{t-1} and x_t. Next, a vector is created using the tanh function, which gives an output from -1 to +1, containing all possible values of h_{t-1} and x_t. Finally, the vector values and the regulated values are multiplied to obtain useful information.
The input gate equation is as follows:

$$\tanh(W_c[h_{t-1}, x_t] + b_c) \otimes \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{7}$$

The Output gate controls which information encoded in the cell state is sent to the input network at the next time step, this is done via the output vector h_t. First, a vector is generated by applying the tanh function to the cell. Next, the information is regulated using the sigmoid function and filtered by the values to be retained using the inputs h_{t-1} and x_t. Finally, the vector values and the regulated values are multiplied and sent as output and input to the next cell. The output gate equation is as follows:

$$\begin{aligned} h_t &= o_t \otimes \tanh(c_t) \\ \text{where } o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \end{aligned} \tag{8}$$

3 Data Analysis

The study utilized data provided by the Quebec Metallurgy Centre in collaboration with Agnico Eagle Mine Goldex, covering the period from 2016 to 2023. The dataset comprises sixteen process variables recorded every five minutes and eight pipe thickness variables measured with a probe installed in the pipe. Table 1 shows the distribution of process data as well as their mean and standard deviation (std) Thickness measurements are collected over 24 hours or more.
To have the same number of records for process variables and thickness measures, we aggregate the process variables by calculating their average per day. This results in 635 records for each variable. We also calculate the average of the eight thicknesses.

Table 1 The process data, their mean and their standard deviation

Area	Parameter	Mean	Std
Alimentation	Tonnage Sag	337.88	61.74
Flotation sector	pulp flotation temperature	25.4	5.89
	pH flottation	9.08	0.26
Pipeline	residue flow	431.14	113.4
	% solid residue	24.98	15.7
	Calculated residual TPH	156.25	132.02
	Pressure Km 0	2095	737.3
	Temperature Km 0	18.89	6.7
	Pressure Km 14	430.36	446.66
	Temperature Km 14	18.09	19.87
Thompson River	flow rate m3/h	182.6	106.65
	Temperature	11.01	6.59
Sedimentary Basin	flow rate m3/h	70.27	15.99
	Temperature	12.78	7.55
South Park	flow rate m3/h	166.4	101.6
	Temperature	7.8	6.3

Figure1 represents the evolution of the average thickness as a function of time. Dates are represented in French (janv. January, avr. April, juil. July, and oct. October). Pipelines are affected by corrosion if the measure of their thickness is less than 5.5. First, we'll perform a binary classification with machine learning models (SVM, KNN, and RF) on the risk of corrosion by encoding the labels: 1 if the measure of thickness is less than 5.5 and 0 if the measure of thickness is greater than 5.5.

In the second part, we'll predict the evolution of the average thickness measure with the LSTM, taking the data as a time series.

Fig. 1 Average thickness as a function of time

4 Results

Table 2 shows the performance of the different machine learning models implemented. SVM and RF provide higher dataset accuracy than KNN. This indicates that SVM and RF offer better classification of corroded pipelines.
The recall of the RF is equal to 1.00, which means that it didn't produce any false negatives. This is because the data from re-cut pipeline measures are not balanced. there are more 1's than 0's. That's why, to compare the performance of these models, we're going to base it on the F1-measure, which is the harmonic mean of precision and recall. The Random Forest obtains the best F1-measure, which means that it performs better on our data than the other two models.

Table 2 Comparison of different machine learning models

Models	Precision	Recall	F1-measure	Accuracy
SVM	0.964	0.973	0.968	0.941
RF	0.940	1.000	0.969	0.941
KNN	0.981	0.945	0.945	0.933

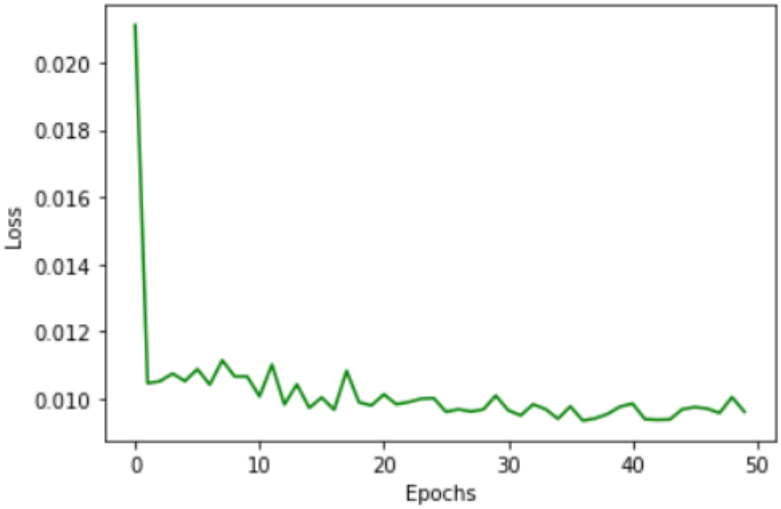

Fig. 2 Training Loss

Figure 2 shows the evolution of the loss function of the LSTM model over time. The decrease in the loss function proves that the LSTM model used has minimized the prediction errors during training.

The thickness measurements predicted by the model are below the nominal thickness (5.5mm), which means the pipeline is already corroded, since the LSTM is monitoring the evolution of past measurements, the corresponding pipelines will have to be replaced or treated. (see Figure3)

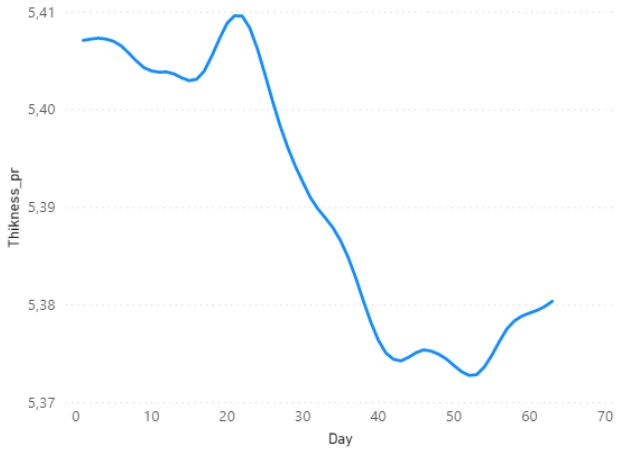

Fig. 3 Measurement of the average thickness predicted by the LSTM model

5 Conclusion

Corrosion in pipelines poses significant economic and environmental challenges. Due to the intricate nature of corrosion data, simplistic statistical modeling struggles to provide a comprehensive analysis.

In our study, we assessed three classification models – SVM, KNN, and Random Forest (RF). Our findings indicate that the random forest outperformed the others, achieving an F1-measure of 0.969. Additionally, we utilized an LSTM model to make predictions for 63 days.

Moving forward, our research will incorporate temporal statistical models like ARIMA and SARIMA, integrating them with deep learning techniques to enhance the model's predictive capabilities.

6 Acknowledgement

We thank Agnico Eagle Goldex Mine for its support. This work was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Québec Fonds de recherche nature et technologies (FRQNT).

References

1. M. Aghaaminiha, R. Mehrani, M. Colahan, B. Brown, M. Singer, S. Nesic, S. M. Vargas, and S. Sharma, "Machine learning modeling of time-dependent corrosion rates of carbon steel in presence of corrosion inhibitors," *Corrosion Science*, vol. 193, p. 109904, 2021.
2. M. F. Sheikh, K. Kamal, F. Rafique, S. Sabir, H. Zaheer, and K. Khan, "Corrosion detection and severity level prediction using acoustic emission and machine learning based approach," *Ain Shams Engineering Journal*, vol. 12, no. 4, pp. 3891–3903, 2021.
3. A. Hendi, A. Behravan, D. Mostofinejad, S. M. Moshtaghi, and K. Rezayi, "Implementing ann to minimize sewage systems concrete corrosion with glass beads substitution," *Construction and Building Materials*, vol. 138, pp. 441–454, 2017.
4. A. K. Dia, N. Ghazzali, and A. Gambou Bosca, "Unsupervised neural network for data-driven corrosion detection of a mining pipeline," in *The International FLAIRS Conference Proceedings*, vol. 35, 2022.
5. X. Li, M. Guo, R. Zhang, and G. Chen, "A data-driven prediction model for maximum pitting corrosion depth of subsea oil pipelines using ssa-lstm approach," *Ocean Engineering*, vol. 261, p. 112062, 2022.
6. K. M. Sow and N. Ghazzali, "Developing a predictive model using multivariate analysis and Long Short-Term Memory (LSTM) to assess corrosion degradation in mining pipeline thickness, 2024, accepted."
7. L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.
8. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

Annexe B

Article scientifique II

Developing a predictive model using multivariate analysis and Long Short-Term Memory (LSTM) to assess corrosion degradation in mining pipeline thickness.

Kalidou Moussa Sow, Nadia Ghazzali
University of Quebec at Trois-Rivieres, Department of Mathematics and Computer Science
Kalidou.Moussa.Sow@uqtr.ca
Nadia.Ghazzali@uqtr.ca

Abstract

Pipeline corrosion has significant impacts on the human, economic, and natural environment. To help better detect and prevent it over time, in this paper, we propose a multivariate approach using machine learning. More precisely, we propose to study the evolution of the thickness of the mining pipeline using a multivariate approach and to implement a predictive model using the Long Short-Term Memory (LSTM) artificial neural network. Indeed, LSTM is a specific recurrent neural network (RNN) architecture designed to model temporal sequences. The proposed predictive model achieved an accuracy of 80% and a loss of 0.01 and was able to predict variations in eight thickness measurements over one hundred days.

Introduction

For many decades, pipelines have served as the most efficient and secure means of transporting materials globally. Any breakdown in pipeline transmission systems directly affects the economics of the material industry (El-Abbasy et al. 2014). Over the years, researchers have delved into studying models of these failures (Lam and Zhou 2016), (Valor et al. 2013). The majority of inquiries into assessing various failure modes in oil and gas pipelines indicate that corrosion stands out as one of the most prevalent causes of failures in transmission systems.

Pipeline corrosion leads to degradation and reduced pipeline thickness. A high-performance model is needed to quickly identify the risk of corrosion and effectively halt its spread, thus restoring the pipeline's original thickness.

(Hussein Farh et al. 2023) classified the cause of corrosion into three categories: 1) environmental factors; soil, external, and stray current factors; 2) pipe factors, and 3) operational factors. They used the fuzzy analytical hierarchy process to represent the impact of the three factors on pipeline corrosion. They found that operational factors had the highest relative weight (0.428), followed by environmental factors (0.337).

(Scully et al. 2008) explained that 60% of failures in Mexican oil and gas transportation systems are caused by pitting corrosion on the external walls of pipelines. This fact of high incidents occurring due to corrosion relates to the complexity of the environment that surrounds pipelines, including a large variety of soil properties, water, and transported products using the pipeline (Oil or Gas).

As a result of the advancements in machine learning (ML) and deep learning (DL), there has been significant interest in data-driven model-based detection methods for pipeline erosion-corrosion monitoring. The artificial neural network (ANN), particularly the backpropagation neural network (BPNN), is widely employed for predicting the erosion-corrosion rate in pipelines (Zhang, Du, and Cao 2015). (Tian, Gao, and Li 2006) adopted the ANN model to detect the corrosion of the submarine oil pipeline under four degrees (no, mild, moderate, serious) based on the original input data collected by the ultrasonic sensor and flux leakage sensor.

This paper aims to build a predictive model of the corrosion degradation of a pipeline used to convey water in the mining area of the Quebec metallurgy center. Work has already been done on the same data used in this paper. (Dia, Ghazzali, and Gambou Bosca 2022) have applied an unsupervised neural network, self-organizing maps (SOM), to study the impact of corrosion assessed by periodic ultrasonic inspections. They combined SOM and hierarchical clustering to detect the extent of corrosion in a mining pipeline. This paper builds upon the research conducted by (Dia, Ghazzali, and Gambou Bosca 2022) by introducing a novel approach that incorporates multivariate analysis and LSTM.

Related Work

Long Short-Term Memory is a specific recurrent neural network (RNN) architecture that was designed to model temporal sequences. It was developed in 1997 by Sepp Hochreiter and Jürgen Schmidhuber (Hochreiter and Schmidhuber 1997) to solve the vanishing gradient problem present in traditional RNNs. Its relative insensitivity to interval length is its advantage over other RNNs, hidden Markov models, and other sequence learning methods. LSTM is very good at predicting in a time series (Lara-Benítez et al. 2020). It could extract patterns from sequential data and store these patterns in internal state variables. Each LSTM cell can retain important information for a longer period when it is used. This

information property allows the LSTM to perform well in classifying, processing, or predicting complex dynamic sequences (Mateus et al. 2021), which makes LSTM a very used model in the literature.

(Salman et al. 2018) applied an LSTM model to weather variable data collected by weather underground at Hang Nadim Indonesia. They added an intermediate variable signal on the memory block cell of LSTM and their model performs better than other LSTM models with an accuracy of 0.8060 and a Root Mean Square Error (RMSE) of 0.0775. (Nelson, Pereira, and De Oliveira 2017) used LSTM networks to predict future trends of stock prices based on the price history, alongside technical analysis indicators to an average of 55.9% accuracy when predicting if the price of a particular stock is going to go up or not shortly. (Di Persio and Honchar 2016) compares the performance of LSTM and Multilayer Perception (MLP) to their own proposed method based on a combination of wavelets and Convolutional Neural Networks (CNN), which outperforms both but has very close results to the LSTM network. (Karmiani et al. 2019) compared LSTM with Support Vector Machine (SVM), backpropagation, and Kalman filter for the stock market for different numbers of epochs varying from 10 to 100. LSTM was found to have high accuracy and low variance. (Chen, Zhou, and Dai 2015) used an LSTM model on the historical data of the Chinese stock market. They trained the LSTM model on 900000 sequences and tested it using the other 311361 sequences. Compared with the random prediction method, their LSTM model improved the accuracy of stock returns prediction from 14.3% to 27.2%. These efforts demonstrated the power of LSTM in predicting China's dynamic and highly unpredictable stock market.

LSTM has also proven itself in the analysis and prediction of corrosion pipelines Since the data collected in this area can be considered as time series. (Li et al. 2022) combined a new swarm intelligence optimization algorithm called SSA and a LSTM model to predict the maximum pitting corrosion depth of subsea oil pipelines. The comparison of their SSA-LSTM method with the LSTM alone shows that the new model SSA-LSTM performed superior in prediction accuracy and robustness. They used RMSE, Mean Absolute Error (MAE), Mean Squared Error (MSE) and Mean Absolute Percent Error (MAPE) as evaluation parameters to measure the performance of their model. The proposed hybrid model (SSA-LSTM) obtained the best performance with the smallest evaluation parameter values (RMSE = 0.0607 , MAE = 8.84% , MSE = 0.36%, MAPE = 9.58%).

Data processing

Data Collection

The data for this study were obtained from the Quebec Metallurgy Centre in collaboration with Agnico Eagle Mine Goldex. The range of data for this study was from year 2016 to year 2023. The database is composed of sixteen process variables which are collected every five minutes and eight pipe thickness variables which are measured using a probe installed in the pipe and recollected within 24 hours or more. Table 1 shows the distribution of process data as well as their mean and standard deviation (std). The number of registrations received for sixteen variables is 179989 and 635 for each thickness measurement. The nominal thickness of the pipeline is between 5.5mm and 6.5mm.

In contrast to (Dia, Ghazzali, and Gambou Bosca 2022), which aggregated the eight thicknesses by calculating their average, in this paper, the eight thicknesses and 16 process variables are considered to predict future measurements of the eight thicknesses.

Area	Parameter	Mean	Std
Alimentation	Tonnage Sag	337.88	61.74
Flotation sector	pulp flotation temperature	25.4	5.89
	pH flottation	9.08	0.26
Pipeline	residue flow	431.14	113.4
	% solid residue	24.98	15.7
	Calculated residual TPH	156.25	132.02
	Pressure Km 0	2095	737.3
	Temperature Km 0	18.89	6.7
	Pressure Km 14	430.36	446.66
	Temperature Km 14	18.09	19.87
Thompson River	flow rate m3/h	182.6	106.65
	Temperature	11.01	6.59
Sedimentary Basin	flow rate m3/h	70.27	15.99
	Temperature	12.78	7.55
South Park	flow rate m3/h	166.4	101.6
	Temperature	7.8	6.3

Tab. 1: The process data, their mean and their standard deviation

Data Analysis

Figure1 shows the distribution of the eight thicknesses. The thickness data varies between 4mm and 7mm, except for the boxplot for thickness 1, which shows thickness measurements greater than 40 and others equal to 0, indicating that there are input errors for thickness 1. To deal with these missing or outliers, we will replace them with a moving average to reduce noise and maintain the trend in thickness values.

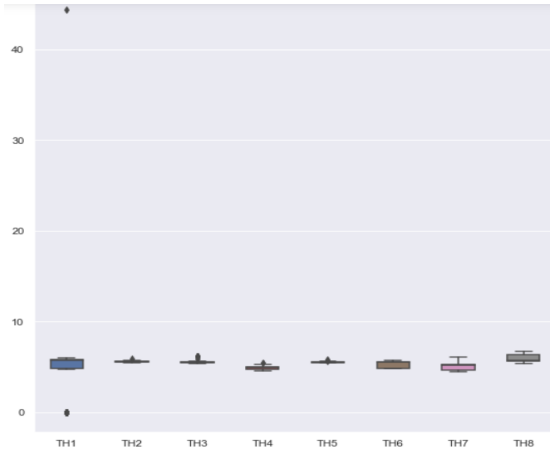

Fig. 1: Boxplot of the Eight thicknesses

Figure 2 shows the thickness data's evolution according to collection dates. Dates are represented in French (janv. January, avr. April, juil. July, and oct. October). After applying the various transformations, the values obtained vary between 4mm and 7mm.

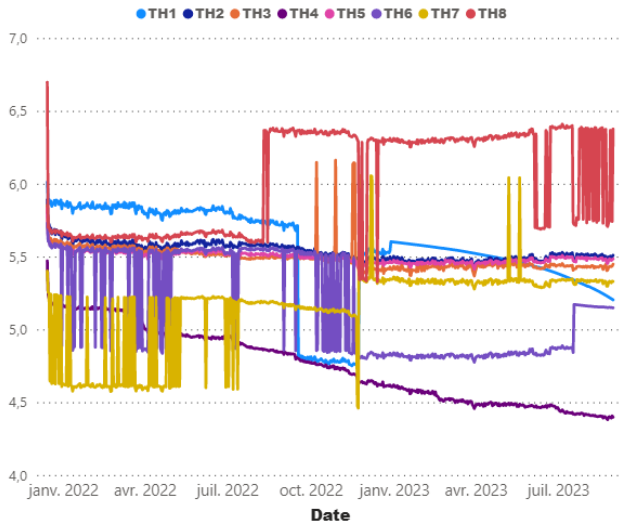

Fig. 2: The evolution of thickness measurements after the application of the transformations

The correlation matrix (see Figure 3) shows that some measurements are strongly correlated. It can be seen that the % solid of the residue and the Ton Per Hour (TPH) of the calculated residue are positively correlated with a Pearson coefficient of 0.96 and also the Ph of the flotation and the flow rate m3/h are negatively correlated. It is therefore interesting to reduce the data by performing a principal component analysis (PCA).

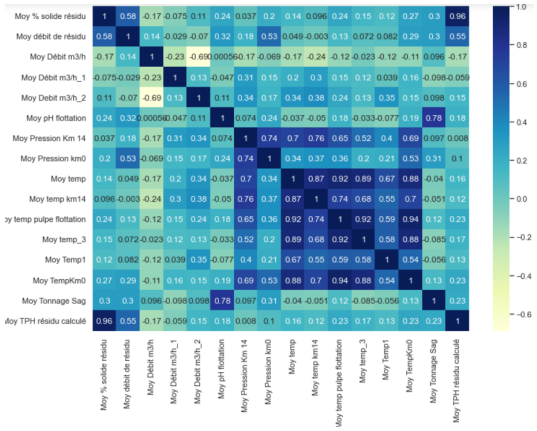

Fig. 3: Correlation matrix

After applying the PCA, analysis of the percentage of variation explained shows that from the ninth dimension onwards, we've already reached more than 99% of the variance explained (see Figure 4) so we'll keep only nine dimensions for the rest of the analysis.

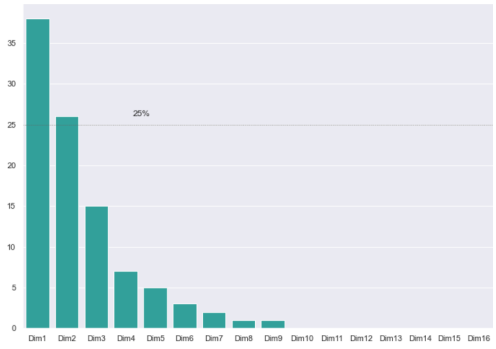

Fig. 4: percentage of variation explained by dimension

Long short-term memory

Long Short-Term Memory (LSTM) Networks allow to learn long-term dependencies. They are explicitly designed to avoid the long-term dependency problem.
An LSTM network has three gates that update and control the states of the cells: the Forget gate, the Input gate, and the Output gate.
The Forget gate (see Figure 5) is responsible for deciding to let information pass. State 0 corresponds to "keep complete information" while state 1 represents "Totally get rid of the information". It is defined by the following equation:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (1)$$

where:

- W_f represents the weight matrix associated with the Forget gate.

- $[h_{t-1}, x_t]$ designates the concatenation of the current entry and the previous hidden state.
- b_f is the bias.
- σ is the sigmoid function.

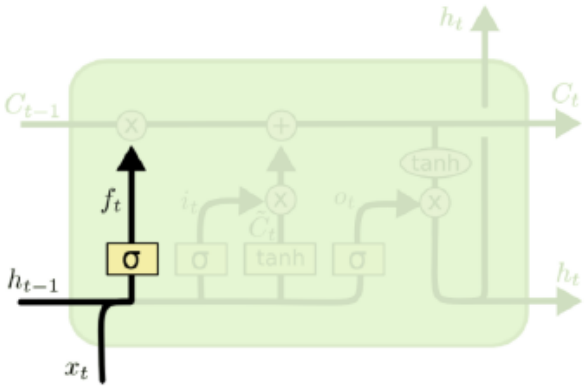

Fig. 5: Forget gate (Oinkina and Hakyll 2015)

The Input gate (see Figure 6) controls what new information will be encoded in the cell state, given the new input information. The information is regulated using the sigmoid function and filters the values to be retained in the same way as the forgetting gate, using the inputs h_{t-1} and x_t. Next, a vector is created using the tanh function, which gives an output from -1 to +1, containing all possible values of h t-1 and x_t. Finally, the vector values and the regulated values are multiplied to obtain useful information.

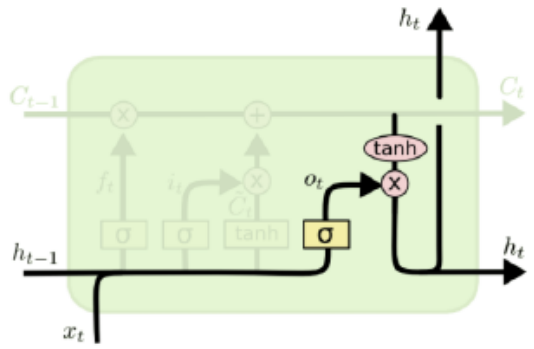

Fig. 6: Input gate (Oinkina and Hakyll 2015)

The input gate equation using the hyperbolic tangent function (tanh) is as follows:

$$\tanh(W_c[h_{t-1}, x_t] + b_c) \otimes \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

where:

- W_c represents the weight matrix associated with the Input gate.
- $[h_{t-1}, x_t]$ designates the concatenation of the current entry and the previous hidden state.
- $\otimes$ is the element-by-element product.
- σ is the sigmoid function.

The Output gate (see Figure 7) controls which information encoded in the cell state is sent to the input network at the next time step, this is done via the output vector h(t). First, a vector is generated by applying the tanh function to the cell. Next, the information is regulated using the sigmoid function and filtered by the values to be retained using the inputs h_{t-1} and x_t. Finally, the vector values and the regulated values are multiplied and sent as output and input to the next cell. The output gate equation is as follows:

$$h_t = o_t \otimes \tanh(c_t) \text{ where } o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (3)$$

where:

- W_o represents the weight matrix associated with the Output gate.
- $[h_{t-1}, x_t]$ designates the concatenation of the current entry and the previous hidden state.
- b_o is the bias associated with the Output gate.
- $\otimes$ is the element-by-element product.
- σ is the sigmoid function.

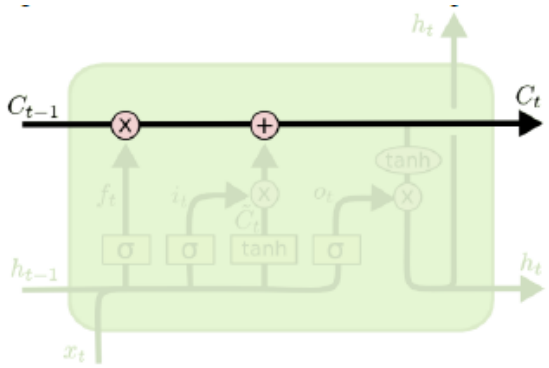

Fig. 7: Output gate (Oinkina and Hakyll 2015)

Results and Discuss

Model evaluation

Figure 8 shows the evolution of the loss function of the LSTM model over time. The decrease in the loss function proves that the LSTM model used has minimized the prediction errors during training. The performance of the model can also be measured by its accuracy.

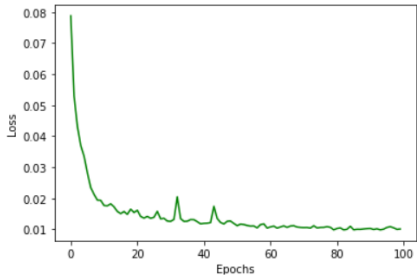

Fig. 8: Training Loss

During training, the model used achieved an accuracy score of 80%, which means that 80% of the values predicted by the model are correct (Figure 9).

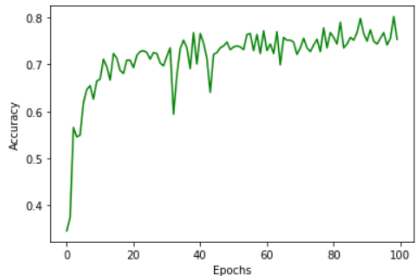

Fig. 9: Training Accuracy

Table 2 shows that using the PCA on the dataset increases the model's performance, with accuracy rising from 76% to 80 % and Loss decreasing to 0.100. This is because the PCA eliminates redundancies between variables.

Model	Accuracy(%)	Loss
LSTM	76	0.132
PCA + LSTM	80	0.100

Tab. 2: Comparison of LSTM and PCA+LSTM performance

Model predictions

Once the model has been trained, we will test it on the test data. To do this, we will predict the evolution of the thicknesses over the next 100 days. Figure 10 shows the evolution of the pipeline thickness 1, 2, 3, 5, and 6 predicted by the LSTM model for the next hundred days.
The thickness values of pipelines 1 and 6 are below the nominal thickness value (between 5.5 and 6.5) before the 40 days which means that the corresponding pipeline is affected by corrosion and should be monitored. And after 40 days, there is an increase in the thickness measurements of pipelines 1 and 6 which means that both pipelines (1 and 6) should be treated. The evolution curve of the thickness measurements predicted by the LSTM model of the pipelines (2, 3, and 5) shows an almost constant variation between 5.4mm and 5.6mm, which is just over the minimum thickness of 5.5mm. This suggests that regular inspections should be carried out over the next 100 days.

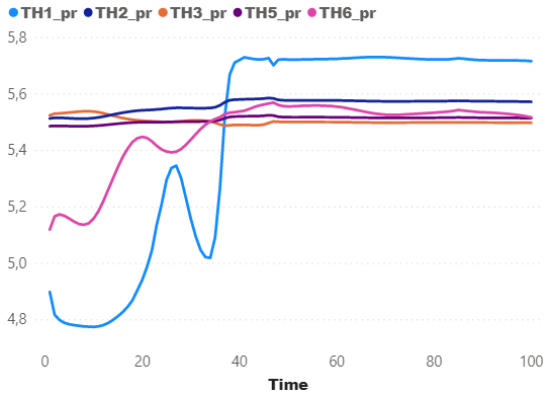

Fig. 10: Pipe thickness 1, 2, 3, 5 and 6 predicted by the model

The thickness measurements predicted by the model for pipelines 4 and 7 are above the nominal thickness (5mm), which means that pipelines 4 and 7 are already corroded, since the LSTM is monitoring the evolution of past measurements, the corresponding pipelines will have to be replaced or treated. (see Figure11)

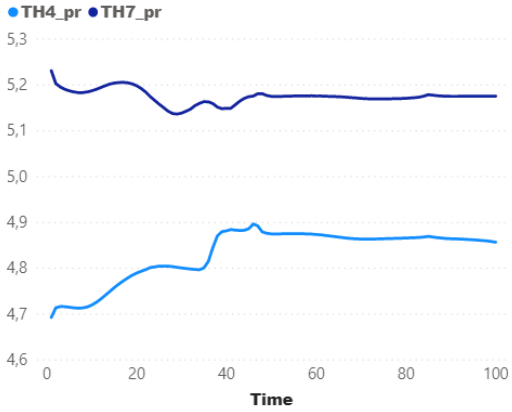

Fig. 11: Pipeline thickness 4 and 7 predicted by the model

The evolution of the thickness 8 curve predicted by the model shows a rapid fall in thickness 8 up to the 40th day, which means that there is a risk of loss of thickness due to the corresponding corrosion of the pipe. Even if the measurements of the thickness of the corresponding pipeline are between the nominal thicknesses, it must be treated before the 40 days to avoid the increase in corrosion which will correspond to the increase in the measurement of the thickness of pipeline 8.

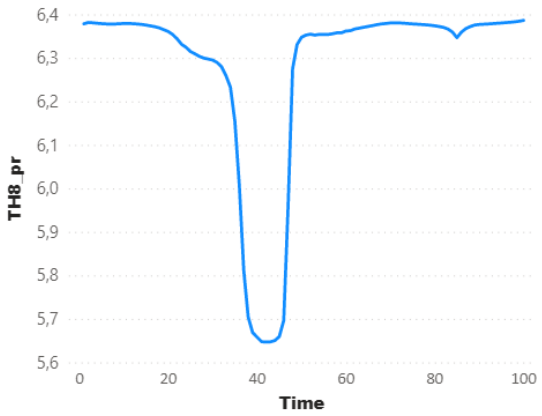

Fig. 12: Pipeline thickness 8 predicted by the model

Conclusion

Pipeline failure has attracted the attention of many research communities because of its significant impact on the global economy, leaks, explosions, and costly downtime. Many efforts to build corrosion prediction models have been proposed resulting in a vast number of publications available in the literature. However, the nature of corrosion pipeline is so complex that difficult to formulate in a single mathematical model.

In this paper, an LSTM model was used to predict the evolution and measurement of the thicknesses of an ore transport pipeline. The model developed was able to predict the evolution of different pipeline thickness values.

Future work will focus on the received data and combine other time series processing models with the LSTM.

Acknowledgement

We thank Agnico Eagle Goldex Mine for its support. This work was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Québec Fonds de recherche nature et technologies (FRQNT).

References

Chen, K.; Zhou, Y.; and Dai, F. 2015. A lstm-based method for stock returns prediction: A case study of china stock market. In *2015 IEEE international conference on big data (big data)*, 2823–2824. IEEE.

Di Persio, L., and Honchar, O. 2016. Artificial neural networks approach to the forecast of stock market price movements. *International Journal of Economics and Management Systems* 1.

Dia, A. K.; Ghazzali, N.; and Gambou Bosca, A. 2022. Unsupervised neural network for data-driven corrosion detection of a mining pipeline. In *The International FLAIRS Conference Proceedings*, volume 35.

El-Abbasy, M. S.; Senouci, A.; Zayed, T.; Mirahadi, F.; and Parvizsedghy, L. 2014. Artificial neural network models for predicting condition of offshore oil and gas pipelines. *Automation in Construction* 45:50–65.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hussein Farh, H. M.; Ben Seghier, M. E. A.; Taiwo, R.; and Zayed, T. 2023. Analysis and ranking of corrosion causes for water pipelines: a critical review. *npj Clean Water* 6(1):65.

Karmiani, D.; Kazi, R.; Nambisan, A.; Shah, A.; and Kamble, V. 2019. Comparison of predictive algorithms: backpropagation, svm, lstm and kalman filter for stock market. In *2019 amity international conference on artificial intelligence (AICAI)*, 228–234. IEEE.

Lam, C., and Zhou, W. 2016. Statistical analyses of incidents on onshore gas transmission pipelines based on phmsa database. *International Journal of Pressure Vessels and Piping* 145:29–40.

Lara-Benítez, P.; Carranza-García, M.; Luna-Romera, J. M.; and Riquelme, J. C. 2020. Temporal convolutional networks applied to energy-related time series forecasting. *applied sciences* 10(7):2322.

Li, X.; Guo, M.; Zhang, R.; and Chen, G. 2022. A data-driven prediction model for maximum pitting corrosion depth of subsea oil pipelines using ssa-lstm approach. *Ocean Engineering* 261:112062.

Mateus, B. C.; Mendes, M.; Farinha, J. T.; Assis, R.; and Cardoso, A. M. 2021. Comparing lstm and gru models to predict the condition of a pulp paper press. *Energies* 14(21):6958.

Nelson, D. M.; Pereira, A. C.; and De Oliveira, R. A. 2017. Stock market's price movement prediction with lstm neural networks. In *2017 International joint conference on neural networks (IJCNN)*, 1419–1426. Ieee.

Oinkina, and Hakyll. 2015. Comprendre les réseaux lstm.

Salman, A. G.; Heryadi, Y.; Abdurahman, E.; and Suparta, W. 2018. Single layer & multi-layer long short-term memory (lstm) model with intermediate variables for weather forecasting. *Procedia Computer Science* 135:89–98.

Scully, J. R.; Budiansky, N. D.; Tiwary, Y.; Mikhailov, A. S.; and Hudson, J. L. 2008. An alternate explanation for the abrupt current increase at the pitting potential. *Corrosion Science* 50(2):316–324.

Tian, J.; Gao, M.; and Li, J. 2006. Corrosion detection system for oil pipelines based on multi-sensor data fusion by improved simulated annealing neural network. In *2006 International Conference on Communication Technology*, 1–5. IEEE.

Valor, A.; Caleyo, F.; Hallen, J. M.; and Velázquez, J. C. 2013. Reliability assessment of buried pipelines based on different corrosion rate models. *Corrosion Science* 66:78–87.

Zhang, L.; Du, Y.; and Cao, A. 2015. The design of natural gas pipeline inspection robot system. In *2015 IEEE International Conference on Information and Automation*, 843–846. IEEE.

Annexe C

Modèle univarié du LSTM en python

Importation des modules d'analyse de données

```
numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Exportation des données

```
dataset_train = pd.read_excel('donnee1.xlsx')
dataset_train.head(5)
```

Importation des modules de LSTM

```
from keras.models import Sequential
from keras.layers import Dense, LSTM
from keras.layers import Dropout
```

```
    regressor = Sequential()
regressor.add(LSTM(units=50, return_sequences= True, input_shape = (X_train.shape[1],1)))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units=50, return_sequences= True))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units=50, return_sequences= True))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units=50))
regressor.add(Dropout(0.2))
regressor.add(Dense(units = 1))
regressor.compile(optimizer='adam' , loss = 'mean_squared_error', metrics=['accuracy'])
hist = regressor.fit(X_train, y_train, epochs = 50, batch_size = 10)
hist



    leni = len(thickness) - len(test_set)-50
inputs = thickness[leni :,]
inputs = sc.transform(inputs)



    X_test = [ ]
for i in range(50,len(inputs)) :
X_test.append(inputs[i-50 :i,0])
X_test = np.array(X_test)
len(X_test)



    X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1],1))
```

```
prediction = regressor.predict(X_test)
prediction = sc.inverse_transform(prediction)
```

Affichage de la prédiction

```
plt.plot(prediction, color ='blue')
plt.title('LSTM predict')
plt.xlabel('time')
plt.ylabel('Thickness')
plt.legend()
plt.show()
```

```
predict_value = pd.DataFrame(prediction, columns = ['Thikness_pr'])
predict_value.to_csv('predict_value.csv', index=False, encoding='utf-8')
predict_value.head(10)
```

Jour	Thickness_pr
0	5.40712
1	5.40727
2	5.40735
3	5.40720
4	5.40720
5	5.40720
6	5.40720
7	5.40720
8	5.40720
9	5.40720

TABLE C.1 – L'épaisseur moyenne prédite par le modèle LSTM univarié

Annexe D

Modèle de classification par SVM, FA et KNN en python

Importation des modules d'analyse

```
import pandas as pd
dataset = pd.read_csv('data1.csv', sep = ',')
dataset.head()
dataset.loc[dataset['Moy_thikness'] ≤ 5.5, 'Risque de coresion'] = True
dataset.loc[dataset['Moy_thikness'] ≥ 5.5, 'Risque de coresion'] = False
```

Mettre les données en format numpy

```
feature_data = dataset[['Moy Débit_m3/h', 'Moy Débit_m3/h.1', 'Moy Débit_m3/h2',
'Moy Debut_m3/h1', 'Moy ph Flot', 'Moy pourc soli', 'Moy pression KM14', 'Moy
Pression Km14.1', 'Moy Tem Flo', 'Moy temp km0', 'Moy temp km14', 'Moy temp1',
'Moy temp2', 'Moy temp3', 'Moy Tonnage', 'Moy TPH', 'Moy debit']]
import numpy as np
X = np.asarray(feature_data)
```

X

Encodage des étiquettes

```
from sklearn.preprocessing import LabelEncoder
creating instance of labelencoder
labelencoder = LabelEncoder()
dataset['Risque de coresion'] = labelencoder.fit_transform(dataset['Risque de coresion'])
dataset
```

Modèle de SVM

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size = 0.2, random_state = 4)
from sklearn import svm
classifier = svm.SVC(kernel ='linear', gamma='auto', C=2)
classifier.fit(X_train, y_train)
y_predict = classifier.predict(X_test)
```

	Precision	Recall	F1-Support
0	0.62	0.56	0.59
1	0.96	0.97	0.97
macro avg	0.79	0.76	0.78
weighted avg	0.94	0.94	0.94

TABLE D.1 – Performance du modèle de SVM

Modèle de Forêt aléatoire

```
from sklearn.ensemble import RandomForestClassifier
```

from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, ConfusionMatrixDisplay

rf = RandomForestClassifier()

rf.fit(X_train, y_train)

y_pred_rf = rf.predict(X_test)

Modèle de KNN

from sklearn.neighbors import KNeighborsClassifier

knn_model_Kn = KNeighborsClassifier(n_neighbors=3)

knn_model_Kn.fit(X_train, y_train) y_pred_kn = knn_model_Kn.predict(X_test)

Annexe E

Modèle multivarié de LSTM en python

Importation des modules import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

Importation du base de données dataset_train = pd.read_excel('Epaisseur.xlsx')

dataset_train.head(5)

dataset_train.head()

train_set = dataset_train.iloc[:400,0 :9].values

Jour	TH1	TH2	TH3	TH4	TH5	TH6	TH7	TH8
0	6.0142	5.8941	5.8561	5.4747	5.8325	5.7332	5.4043	6.7010
1	5.8914	5.7300	5.6619	5.2424	5.6239	5.6081	5.2582	5.7039
2	5.8959	5.7259	5.6377	5.2153	5.6133	5.6082	5.2514	5.7057
3	5.8669	5.6881	5.6126	5.1887	5.5831	5.5767	4.6485	5.6791
4	5.8850	5.6991	5.6279	5.1967	5.5907	5.5894	5.2449	5.6883

TABLE E.1 – Données des huit épaisseurs dans les cinq premiers jours

test_set = dataset_train.iloc[400 :,0 :9].values train_set

Prétraitement des données

from sklearn.preprocessing import MinMaxScaler

sc = MinMaxScaler(feature_range=(0,1))

train_set_sc = sc.fit_transform(train_set)

test_set_sc = sc.fit_transform(test_set)

X_train = []

y_train = []

for i in range(30, 400) :

X_train.append(train_set_sc[(i-30) :i, :])

y_train.append(train_set_sc[i, :])

X_train = np.array(X_train)

y_train = np.array(y_train)

X_train

X_train = np.reshape(X_train,(X_train.shape[0],X_train.shape[1],8))

y_train = np.reshape(y_train,(y_train.shape[0],8))

Importation des modules de LSTM from keras.models import Sequential

from keras.layers import Dense

from keras.layers import LSTM

from keras.layers import Dropout

regressor = Sequential()

regressor.add(LSTM(units =50 ,return_sequences = True, input_shape =(X_train.shape[1],8)

))

regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 50 , return_sequences= True)) regressor.add(Dropout(0.2))

regressor.add(LSTM(units =50 ,return_sequences = True))

regressor.add(Dropout(0.2))

regressor.add(LSTM(units =50 ,return_sequences = True))

regressor.add(Dropout(0.2))

regressor.add(LSTM(units =50))

regressor.add(Dropout(0.2))

Fully connected

regressor.add(Dense(units= 8))

Compilation du modele LSTM

regressor.compile(optimizer='Adam' , loss = 'mean_squared_error' , metrics=['accuracy'])

hist = regressor.fit(X_train, y_train, epochs = 100, batch_size = 15)

hist

Training Loss du modèle

plt.plot(hist.epoch, hist.history["loss"], 'g', label='Training loss')

plt.title('Training loss')

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend()

plt.show()

Training Accuracy du modèle

plt.plot(hist.epoch, hist.history["accuracy"], 'g', label='Training loss')

plt.title('Training accuracy')

```
plt.xlabel('Accuracy')
plt.ylabel('Epochs')
plt.legend()
plt.show()



    X_test = []
for i in range(60,123) :
X_test.append(inputs[i-60 :i, :])
X_test = np.array(X_test)
X_test = np.reshape(X_test,(X_test.shape[0],X_test.shape[1],8))



    predict_value = regressor.predict(X_test)
predict_value = sc.inverse_transform(predict_value)
predict_value
```