**Plan provisoire du mémoire de recherche**

**Titre du mémoire :** Approche DevOps et surveillance distribuée des applications basées sur les architectures microservices

**Resumé/Abstract**

Dans un contexte où les architectures microservices deviennent la norme pour les systèmes applicatifs modernes, la surveillance distribuée joue un rôle central dans l’identification précoce des comportements anormaux. Ce mémoire explore une approche comparative de différents modèles de détection d’anomalies, appliqués à des données collectées dans un environnement simulé de microservices. Plusieurs techniques ont été implémentées et évaluées, notamment un perceptron multicouche (MLP) supervisé, un auto-encodeur LSTM, un modèle non supervisé basé sur Isolation Forest, ainsi qu’un modèle hybride combinant des représentations séquentielles et statiques. Les résultats expérimentaux montrent que, malgré l’augmentation de la complexité des architectures, les performances globales restent relativement stables. Ce constat soulève des interrogations sur la pertinence d'utiliser des modèles complexes dans des contextes de surveillance aux données peu bruitées ou peu diversifiées.

Cette étude propose ainsi une réflexion sur le choix des techniques d’apprentissage dans un cadre DevOps, et ouvre la voie à de futures améliorations basées sur l’enrichissement des données et l’intégration de mécanismes d’attention ou d’approches en ligne.

**1. Introduction**

* Contexte général : évolution vers les architectures microservices
* Problématique : la complexité croissante de la surveillance
* Justification du choix du sujet
* Objectifs du mémoire
* Structure du document

**2. Cadre théorique et état de l'art**

* Architectures microservices et défis de surveillance
* Notions de monitoring distribué dans les environnements DevOps
* Catégories de métriques surveillées (logs, traces, métriques système)
* Outils classiques de monitoring (Prometheus, Grafana, Jaeger, etc.)
* Détection d'anomalies : définitions, types (point, contextuelle, collective)
* Revue des modèles : MLP, LSTM Autoencodeur, Isolation Forest, Modèles hybrides

**3. Problématique et questions de recherche**

* Problème : les outils classiques signalent, mais n'expliquent pas toujours
* Question principale : Quelle approche offre les meilleurs compromis pour la détection d'anomalies dans les environnements microservices ?
* Sous-question:
  + L'augmentation de la complexité du modèle amène-t-elle une meilleure détection ?

**4. Méthodologie**

* Description du dataset utilisé (temps réel, anonymisé, etc.)
* Pipeline de traitement des données : nettoyage, normalisation, séquençage
* Implémentation des modèles :
  + MLP classique (Sklearn)
  + LSTM Autoencodeur (Keras)
  + Isolation Forest (non supervisé)
  + Modèle Hybride MLP + LSTM
* Critères d’évaluation : accuracy, recall, precision, F1-score

**5. Résultats expérimentaux**

* Tableau comparatif des performances
* Visualisations des anomalies détectées
* Analyse des cas de détection réussie/ratée
* Interprétation des résultats obtenus

**6. Discussion**

* Forces et faiblesses des approches testées
* Pourquoi les performances plafonnent-elles ?
* Limites liées au dataset et aux features
* Recommandations pour de futurs travaux

**7. Conclusion et perspectives**

* Bilan des approches testées
* Apports du travail par rapport au thème DevOps et monitoring
* Ouverture vers de nouvelles pistes (Mécanisme d’attention, …)