Ministère de l’Enseignement Supérieur, de la Recherche et de l’Innovation

**ECOLE POLYTECHNIQUE DE THIES**



**DEPARTEMENT GENIE INFORMATIQUE ET TÉLÉCOMMUNICATIONS**

**Projet de Fin d’Études en vue de l’obtention du Diplôme d’Ingénieur de Conception**

Présenté par : Papa Omar DIOP

**Sujet :**

**Conception et Réalisation d'un système de détection et de gestion automatisé des anomalies au niveau des transactions de la DerApp**

\

Soutenu publiquement le ……. devant le jury composé de :

|  |  |  |  |
| --- | --- | --- | --- |
| Président |  |  |  |
| Examinateurs |  |  |  |
|  |  |  |
| Encadrant |  |  |  |
|  |  |  |

**PRÉFACE**

**DEDICACES**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**REMERCIEMENTS**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**Résumé**

When it comes to threat detection, how many times have you heard someone say “It is all in my head, just ask me if you have any questions!” or “Only he/she/they know(s) how to do it!” Plenty of times, right? Not documenting, standardizing, or sharing how to analyze data to detect potential intrusions in a network is more common than you think, especially when the team is very diverse from a technical and expertise perspective. It does not only affect your detection strategies but also the dynamics of your team.

Now, how many times have you also thought about a more efficient, intuitive or creative way to analyze the security events your organization collects, but you feel limited to the capabilities of a one language-dependent search bar?

In software development, there is an absolute requirement to ensure that a system functions at its best throughout its lifetime once developed. Application log data is critical to maintaining application performance and thus techniques to parse, understand and detect anomalies in application log data are critical to ensuring efficiency in software development. While initially hampered by limited hardware and lack of quality datasets, anomaly detection techniques have recently increased interest with advancements in machine learning technology, especially neural networks. This paper explores anomaly detection, historical techniques to detect anomalies, and recent advancements in neural networks, which promise to revolutionize anomaly detection in application log data. Further, we analyze the most promising anomaly detection techniques and propose a hybrid model combining LSTM Neural Network and Auto Encoder which improves upon existing techniques.

Index Terms – **Anomaly Detection, Application logs, Machine Learning, Neural Networks**

**Abstract**

When it comes to threat detection, how many times have you heard someone say “It is all in my head, just ask me if you have any questions!” or “Only he/she/they know(s) how to do it!” Plenty of times, right? Not documenting, standardizing, or sharing how to analyze data to detect potential intrusions in a network is more common than you think, especially when the team is very diverse from a technical and expertise perspective. It does not only affect your detection strategies but also the dynamics of your team.

Now, how many times have you also thought about a more efficient, intuitive or creative way to analyze the security events your organization collects, but you feel limited to the capabilities of a one language-dependent search bar?

In software development, there is an absolute requirement to ensure that a system functions at its best throughout its lifetime once developed. Application log data is critical to maintaining application performance and thus techniques to parse, understand and detect anomalies in application log data are critical to ensuring efficiency in software development. While initially hampered by limited hardware and lack of quality datasets, anomaly detection techniques have recently increased interest with advancements in machine learning technology, especially neural networks. This paper explores anomaly detection, historical techniques to detect anomalies, and recent advancements in neural networks, which promise to revolutionize anomaly detection in application log data. Further, we analyze the most promising anomaly detection techniques and propose a hybrid model combining LSTM Neural Network and Auto Encoder which improves upon existing techniques.

Index Terms – **Anomaly Detection, Application logs, Machine Learning, Neural Networks**

**TABLE DES MATIERES**

**Résumé**

**Abstract**

**Liste des Figures**

**Liste des Tableaux**

**Liste des Sigles et Abréviations**

1. **Introduction générale**
   1. Présentation de la structure d’accueil ………………………………………………

1.1.1. Structure d’accueil ……………………………………………………………

1.1.2. Contexte de l’immersion ………………………………………………………

* 1. Problématique …………………………………………………………………………
  2. Objectifs du projet ……………………………………………………………………
  3. Plan du mémoire ………………………………………………………………………

1. **Généralités sur la détection d’anomalies**
   1. Mise en contexte …………………………………………………………………………

1.1.1. Logs ……………………………………………………………………………….

1.1.2. Anomalies …………………………………………………………………………

* 1. Les types d’anomalies ……………………………………………………………………

1.1.1. Les anomalies ponctuelles…………………………………………………………

1.1.2. Les anomalies contextuelles ………………………………………………………

1.1.3. Les anomalies collectives …………………………………………………………

* 1. Les challenges ……………………………………………………………………………

1.1.1. Données non structurées ……………………………………………………………

1.1.2. Informations d'exécution redondantes ………………………………………………

1.1.3. Données déséquilibrées …………………………………………………………….

1. **Etude de solution**
   1. Évolution des techniques de détection d’anomalies ………………………………………

3.1.1. Détection d'anomalies basée sur la statistique ……………………………………… 3.1.2. Détection d'anomalies basée sur la profondeur …………………………………… 3.1.3. Détection d'anomalies basée sur le regroupement (apprentissage non supervisé)

3.1.4. Détection d'anomalies basée sur la distance ……………………………………

3.1.5. Détection d'anomalies basée sur la densité (apprentissage non supervisé) ………

3.1.6. Décomposition spectrale …………………………………………………………

3.1.7. Détection d'anomalies basée sur l'apprentissage automatique supervisé …………

3.1.8. Détection d'anomalies par réseau neuronal classique ……………………………

* 1. Analyse des données fournies ……………………………………………………………

3.2.1. Nature des données ……………………………………………………………

3.2.2. Types d’anomalies ………………………………………………………………

3.2.3. Etiquettes des données ……………………………………………………………

3.2.4. Sortie de la détection d'anomalies ………………………………………………

* 1. Discussion sur la meillleure approche pour ces données …………………………………

(Discussion expliquer le pourquoi on choisit telle méthode d’apprentissage par rapport à un autre ect regarder dans tes notes)

1. **Etude conceptuelle de la solution**

4.1. Analyse des besoins ………………………………………………………………

4.1.1. Les besoins fonctionnels ………………………………………………………

4.1.2. Les besoins non fonctionnels ………………………………………………………

4.2. Modélisation UML ………………………………………………………………

4.2.1. Diagramme des cas d’utilisation (Uses Cases) ……………………………………

4.2.3. Diagramme des classes ……………………………………………………………

4.3. Architecture du projet ………………………………………………………………

4.4. Workflow global du système ……………………………………………………………

4.5. Collecte de données ………………………………………………………………

4.6. Description du jeu de données ……………………………………………………………

4.7. Workflow de la détection ………………………………………………………………

4.8. Conclusion ………………………………………………………………

1. **Implémentation et déploiement**

5.1. Environnement de développement et technologies

5.1.1. Machine Learning Pipeline ………………………………………………………

5.1.1. Docker ……………………………………………………………………………

5.1.1.1 Containers and Virtual Machines ………………………………………

5.1.1.2 Images …………………………………………………………………

5.1.1.3 Containers Lifecycle …………………………………………………

5.1.1.4 Networking ……………………………………………………………

5.1.1.5 Docker Compose …………………………………………………………

5.1.2. Apache Kafka …………………………………………………

5.1.2.1 Kafka architecture …………………………………………………

5.1.2.2 Topic structure …………………………………………………

5.1.2.3 Clustered deployment …………………………………………………

5.1.2.4 Parallel processing of the topic ………………………………………

5.1.3. Logstash …………………………………………………

5.1.3.1 Data extraction ……………………………………………

5.1.3.2 Data transformation …………………………………………………

5.1.3.3 Data load …………………………………………………

5.1.3.4 Parallel processing …………………………………………………

5.1.4. Elasticsearch ………………………………………………

5.1.4.1 Elasticsearch architecture …………………………………………………

5.1.4.2 Difference from a database ………………………………………………

5.1.4.3 Clustered deployment …………………………………………………

5.1.4.4 Search and aggregation ………………………………………………

5.1.5. Kibana ………………………………………………

5.1.5.1 Widget and Dashboard …………………………………………………

5.1.5.2 Discover ………………………………………

5.1.5.3 ELK stack monitor and management ………………………………………

5.1.6. Beats ………………………………………………

5.1.7.1 Filebeat and Metricbeat…………………………………………

5.1.7. Apache Spark ………………………………………………

5.1.7.1 Challenges in big data processing …………………………………………

5.1.7.2 Spark cluster …………………………………………

5.1.7.3 Handling distributed data …………………………………………

5.1.7.4 Broadcast variable …………………………………………

5.1.7.5 Low-level unstructured Spark API ………………………………………

5.1.7.6 Structured Spark API …………………………………………

5.1.7.7 Distributed execution of jobs …………………………………………

5.1.8. Summary ………………………………………………

5.2. Présentation de la solution ……………………………………………………………

5.3. Analyse et évaluation des résultats obtenus ………………………………………………

5.4. Déploiement de la solution ………………………………………………………………

1. **Conclusion générale**

6.1. Synthèse des travaux

6.2. Apport du stage

6.3. Difficultés rencontrées

6.4. Perspectives

**References**

## RÉFÉRENCES

1. Anomaly Detection [En Ligne] Disponible : <https://www.gavstech.com/anomaly-detection-in-aiops/>
2. What is Elasticsearch? [En Ligne] Disponible: <https://www.elastic.co/guide/en/elasticsearch/reference/8.1/elasticsearch-intro.html>
3. Elastic Machine Learning [En Ligne] Disponible: <https://www.elastic.co/guide/en/machine-learning/current/machine-learning-intro.html#machine-learning-intro>
4. Nginx [En Ligne]. Disponible : <http://nginx.org/en/docs/http/ngx_http_log_module.html#log_format>
5. Mohiuddin Solaimani, “Anomaly Detection for Application Log Data” (2015). Mémoires de maîtrise et recherches de troisième cycle, University of Texas at Dallas, Disponible : <https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1640&context=etd_projects>