

REPUBLIQUE DU SENEGAL

UNIVERSITE ASSANE SECK DE ZIGUINCHOR



L'excellence, ma référence

UFR SCIENCES ET TECHNOLOGIES

Département Informatique

Mémoire de Master

Mention : Informatique

Spécialité : Réseaux et Systèmes

Sujet

Mise en place d'une plateforme de gestion des alertes dans les systèmes de surveillance :
Cas de la coupe de bois

Présenté et soutenu publiquement par : Mlle Maty Seck

Le 21 Décembre 2024 à la salle Cicco.

Sous la direction de :

Dr Abel DIATTA

Encadrant

Sous la supervision de Pr Youssou FAYE

Devant le jury composé de :

Président : Pr Youssou FAYE

Maître de Conférences CAMES, UASZ

Membres : Dr Edouard Ngor SARR

Maître -Assistant, UASZ

Dr El Hadji Malick NDOYE

Maître -Assistant, UASZ

Encadrant : Dr Abel DIATTA

Maître -Assistant, UASZ

Année universitaire 2023-2024

Dédicace

Je dédie ce modeste travail avec reconnaissance :

À mes chers parents,

À mon petit frère, Assane SECK, où qu'il soit,

À mon tuteur, Monsieur Mamadou Diallo,

À Monsieur Mamadou Diagne,

À mon encadreur, Monsieur Abel DIATTA,

À tous les membres de ma famille,

Ainsi qu'à toutes les personnes qui me sont chères.

Remerciements

Je tiens à exprimer toute ma gratitude envers ceux qui ont rendu possible la réalisation de ce projet.

Tout d'abord, je remercie profondément mes parents pour leur soutien constant, leur encouragement et leurs sacrifices, sans lesquels ce parcours n'aurait pas été possible. Leur amour inconditionnel et leur foi en mes capacités ont été mon moteur tout au long de cette aventure. À mes frères et sœurs, merci pour leur présence, leur écoute et leur motivation.

Je souhaite également remercier chaleureusement mon encadreur, Monsieur Abel Diatta, pour son soutien inestimable, ses conseils avisés, sa patience et son suivi rigoureux, qui m'ont permis de mener ce projet à terme. Son expertise et son accompagnement ont été essentiels à la réussite de ce travail.

Je remercie sincèrement les membres du jury pour l'attention portée à mon travail, ainsi que mes professeurs pour leurs enseignements et leur soutien. Merci également à mes camarades et amis pour leur esprit de camaraderie et à ma famille d'accueil pour son accueil attentionné et son soutien constant durant cette période importante de ma vie.

Enfin, à tous ceux qui, de près ou de loin, m'ont accompagné, conseillé et soutenu durant mes études et la réalisation de ce projet, je vous adresse mes remerciements les plus sincères.

Résumé

La coupe illégale de bois constitue une menace importante pour les écosystèmes forestiers et les ressources naturelles au Sénégal. Malgré l'existence de divers systèmes de gestion d'alertes, aucun ne se concentre spécifiquement sur la surveillance de cette activité illicite. Ce projet vise à combler cette lacune en proposant une solution innovante basée sur l'Internet des objets (IoT). Dans ce travail, nous avons développé une plateforme de gestion d'alertes, nommée "**SenForet**", capable de détecter et signaler en temps réel les activités suspectes d'abattage illégal. Ce système permet aux agents des Eaux et Forêts de recevoir des alertes instantanées, garantissant une surveillance continue et une intervention rapide. La solution utilise des capteurs IoT pour détecter les sons de tronçonneuses et localiser les activités suspectes. Les données sont transmises en temps réel au cloud via LoRa et Wifi, puis analysées pour générer des alertes accessibles sur une application web et mobile. Ce système permet une surveillance proactive et une réaction rapide pour protéger les forêts et les ressources naturelles du Sénégal. Cette initiative inédite dans le pays s'inscrit dans une démarche globale de conservation environnementale, apportant une réponse technologique avancée à un problème critique. En combinant la collecte de données en temps réel et une surveillance proactive, ce système représente une avancée significative dans la lutte contre la coupe illégale de bois.

Mots clés : Coupe illégale de bois, écosystèmes forestiers, ressources naturelles, gestion d'alertes, internet des objets (IoT), agents des Eaux et Forêts

Abstract

Illegal logging is a major threat to forest ecosystems and natural resources in Senegal. Despite the existence of various alert management systems, none of them focuses specifically on monitoring this illegal activity. This project aims to fill this gap by proposing an innovative solution based on the Internet of Things (IoT). The study aims to develop an alert management platform, called "**SenForet**", capable of detecting and reporting suspicious illegal logging activities in real time. This system enables Water and Forestry officers to receive instant alerts, guaranteeing continuous monitoring and rapid intervention. The solution uses IoT sensors to detect the sounds of chainsaws and locate suspicious activity. The data is transmitted in real time to the cloud via LoRa and Wifi, then analysed to generate alerts that can be accessed via a web and mobile application. This system enables proactive monitoring and rapid reaction to protect Senegal's forests and natural resources. This initiative, the first of its kind in the country, is part of a global approach to environmental conservation, providing an advanced technological response to a critical problem. By combining real-time data collection and proactive surveillance, this system represents a significant step forward in the fight against illegal logging.

Keywords : Illegal logging, forest ecosystems, natural resources, alert management, Internet of Things (IoT), forest rangers

Table des matières

Dédicace	ii
Remerciements.....	iii
Résumé	iv
Abstract	v
Table des matières.....	vi
Liste des tableaux	xiii
Liste des Sigles et Abréviations	xiv
Introduction Générale	1
Première Partie : État de l’art et présentation de notre contribution	3
Chapitre 1 : Types d’alertes et Surveillance de la coupe de bois au Sénégal.....	4
Introduction.....	4
I. Les types d’alertes.....	4
1. Alerte Météorologique	4
2. Alerte de sécurité publique.....	5
3. Alerte en santé publique	7
4. Alerte maritime	8
5. Alerte industrielle	9
II. La Surveillance de la Coupe de Bois au Sénégal	10
1. Situation de la coupe de bois au Sénégal	10
2. La Surveillance actuelle de la coupe de bois.....	12
3. Buts de notre système de surveillance.....	12
Conclusion	14
Chapitre 2 : Etude de l’existant et présentation de notre contribution.....	15
Introduction.....	15
I. Les travaux proposés dans la littérature des systèmes de gestion des alertes existantes	15

1. Surveillance illégale de la découpe d'arbres au moyen de dispositifs IdO intelligents à très faible puissance.....	15
2. Détection automatique de la coupe d'arbres dans les forêts à l'aide de propriétés acoustiques.....	16
3. Système de détection de l'exploitation forestière par le biais de la surveillance acoustique	17
4. SAIP « Application d'alerte et d'information des populations en cas de crise »	18
5. Le système d'alerte et de prévention du risque	19
6. Système d'alerte sur obstacle (OAS)	20
7. Système d'alerte à base de la carte IOIO	21
8. Système d'alerte en Temps Réel utilisant le Réseau Satellitaire Iridium.....	22
9. Système d'alerte médical pour la surveillance de la santé à distance utilisant des capteurs et l'informatique en nuage.....	23
II. Tableau comparatif des différents systèmes utilisés dans la littérature	24
III. Présentation de notre contribution	26
1. Contexte et Objectifs.....	26
2. Organigramme de fonctionnement de notre système	28
Conclusion	30
Deuxième Partie : Conception et Mise en œuvre de notre système	31
Chapitre 1 : Conception de notre Système	32
Introduction.....	32
I. UML et Merise.....	32
1. Principes généraux de Merise et UML.....	32
a. Énumération des acteurs	33
II. Les diagrammes utilisés	34
1. Diagramme de cas d'utilisation.....	34
2. Description textuelle des cas d'utilisation.....	35

3.	Diagramme de séquence.....	43
a.	Diagramme de séquence du cas d'utilisation s'Authentifier	44
b.	Diagramme de séquence du cas d'utilisation Gérer utilisateur (Ajouter utilisateur)	44
c.	Diagramme de séquence du cas d'utilisation Générer Alerte	45
4.	Diagramme de classe.....	45
III.	Architecture physique du système	46
IV.	Description des composants de l'architecture	47
1.	Capteurs.....	48
a.	Capteur acoustique ADMP401	48
b.	Capteur de géolocalisation Module GPS NEO-6M.....	49
2.	Microcontrôleur	50
a.	Microcontrôleur ESP8266	50
b.	Microcontrôleur Arduino Mega 2560	51
3.	Modules de Communication.....	52
a.	Transmission WiFi.....	53
b.	Transmission LoRa.....	53
	Conclusion	56
	Chapitre 2 : Développement et Mise en Œuvre de notre Application SenForêt	57
	Introduction.....	57
I.	Outils de développement utilisés	57
1.	Environnements de développement intégré (IDE)	57
a.	Astah.....	57
b.	Pycharm	57
2.	Plateformes et outils de développement mobile et web	58
a.	Django	58
b.	Plateforme IoT ThingSpeak.....	59

c.	Twilio.....	61
1.	Outils pour l'électronique et l'Internet des objets (IoT).....	62
a.	Fritzing.....	62
b.	IDE Arduino	62
2.	Bases de données et Langages de programmation	63
a.	Python.....	63
b.	SQLite.....	64
II.	Développement du Système	65
1.	Codage Arduino.....	65
a.	Introduction au matériel utilisé (ESP8266, capteurs).....	65
b.	Montage du matériel avec Fritzing.....	66
c.	Gestion des seuils et envoi des données	67
2.	Développement de l'Application Web	69
a.	Environnement de développement (Django, Python, HTML, CSS, JavaScript)	69
b.	Modèles de données (modèles Django, bases de données)	70
c.	Intégration avec ThingSpeak et Twilio pour la gestion des alertes	71
d.	Intégration avec Django pour le déclenchement d'alertes en temps réel.....	72
III.	Déploiement et Tests de l'Architecture.....	73
1.	Réalisation de l'Architecture de Test	73
2.	Résultats : Courbes de Données sur ThingSpeak	73
IV.	Fonctionnement de notre Application SenForet	74
1.	Présentation de l'application Web	74
2.	Configuration de Twilio	77
	Conclusion	78
	Conclusion Générale.....	79
	Bibliographie et webographie	81

Annexe 1 : Code avec ESP8266	86
Annexe 2 : Code avec Lora	89

Liste des figures

Figure 1 : La carte de vigilance	5
Figure 2 : SAIP	7
Figure 3 : Application TousAntiCovid	8
Figure 4 : Architecture du système d’alerte Maritime.....	9
Figure 5 : FR-Alert.....	10
Figure 6 : Système de surveillance illégale de la découpe d'arbres	16
<i>Figure 7 : Système à l'aide de propriétés acoustiques</i>	<i>17</i>
<i>Figure 8 : Système de détection par le biais de la surveillance acoustique.....</i>	<i>18</i>
Figure 9 : SAIP	19
Figure 10 : La génération d’alerte au véhicule communicant.....	20
Figure 11 : Schéma général du système d’alerte d’obstacles (OAS)	21
Figure 12 : Système d’alerte d'accident à base de la carte IOIO	22
Figure 13 : Système d’alerte via le réseau satellitaire iridium	23
Figure 14 : Système d'alerte utilisant des capteurs et l'informatique en nuage.....	24
Figure 15 : Architecture logique.....	27
Figure 16 : Organigramme de notre système	28
Figure 17: Diagramme de cas d'utilisation de notre système	35
Figure 18: : Diagramme de séquence du cas d’utilisation s’Authentifier	44
Figure 19 : Diagramme de classe de notre Système.....	46
Figure 20 : Architecture Global du Système	47
<i>Figure 21 : Capteur acoustique ADMP401</i>	<i>48</i>
Figure 22 : Capteur de géolocalisation GPS NEO-6M	50
Figure 23 : Microcontrôleur ESP8266	51
Figure 24 : Arduino Méga 2560	52
Figure 25 : Dragino Lora Shield	54
Figure 26 : Passerelle Dragino LG01-N.....	55
Figure 27 : Astah	57
Figure 28 : Pycham	58
Figure 29 : Django.....	59
Figure 30 : ThingSpeak	61
Figure 31 : TWILIO	62
Figure 32: Fritzing.....	62

Figure 33 : Logiciel IDE Arduino	63
Figure 34 : Python	64
Figure 35 : SQLite.....	65
Figure 36 : Connexion Microcontrôleur - ADMP401	66
Figure 37: Connexion Microcontrôleur - GPS (NEO-6M)	66
Figure 38 : Connexion Microcontrôleur - GPS (NEO-6M) - ADMP401	67
Figure 39 : Modèle de Données Alerte.....	70
Figure 40 : Configuration de la Base de Données SQLite	70
Figure 41 : Intégration avec ThingSpeak pour la Gestion des Alertes	71
Figure 42 : Configuration de twilio.....	72
Figure 43 : Fonction envoyer alerte	72
Figure 44 : Architecture de Test	73
Figure 45 : Les informations de la géolocalisation	74
Figure 46 : Page de connexion	75
Figure 47 : Page d'accueil.....	76
Figure 48 : Interface de la liste de notification des alertes	76
Figure 49 : Liste des Utilisateurs.....	77
Figure 50 : Alerte Reçu	78
Figure 51 : Importation des bibliothèques.....	86
Figure 52 : Déclaration des constantes.....	86
Figure 53 : fonction setup.....	87
Figure 54 : Fonction Loop.....	88
Figure 55 : Construction de l'URL et envoi des données	88
Figure 56 : Affichage des Données dans le Moniteur Série	89

Liste des tableaux

Tableau 1 : Tableau Comparatif des Systèmes d'alertes.....	24
Tableau 2: Description textuelle Détecter Activités	36
Tableau 3: Description textuelle Transmettre données.....	37
Tableau 4 : Description textuelle Analyser et Stocker les données.....	38
Tableau 5: Description textuelle Générer Alertes.....	39
Tableau 6 : Description détaillée Recevoir Alertes	40
Tableau 7 : Description détaillée Intervenir	41
Tableau 8: Description détaillée Configurer Système.....	41
Tableau 9 : Description Détaillée Gérer Autorités (Utilisateurs)	42
Tableau 10 : Description détaillé S'authentifier	43
Tableau 11 : Caractéristiques ADMP401	49
Tableau 12 : Caractéristiques GPS NEO-6M.....	50
Tableau 13 : Caractéristiques ESP8266.....	51
Tableau 14 : Caractéristiques Arduino Mega 2560	52
Tableau 15 : Tableau de comparaison des modes de transmission.....	56
Tableau 16: Tableau de Connexion Microcontrôleur - ADMP401	66
Tableau 17 : Connexion Microcontrôleur - GPS (NEO-6M).....	66
Tableau 18 : Analyse des sons par période et validation du seuil	68

Liste des Sigles et Abréviations

IA : Intelligence Artificielle

IoT : Internet Of Things

FAO : Organisation pour l'alimentation et l'agriculture

UML : Unified Modeling Language

VHF : Abréviation de l'anglais Very High Frequencies

GPS : Global Positioning System

API : Application Programming Interface

SMS : Short Message Service

CSS : Cascading Style Sheets

DRF : Django Rest Framework

MCD : Modèle Conceptuel de Données

MCT : Modèle Conceptuel des Traitements

IDE : Environnements de développement intégré

MVC : Modèle-Vue-Contrôleur

MTV : Modèle-Template-Vue

ORM : Object-Relational Mapping

CSRF : Cross-Site Request Forgery

HTTP : Hypertext Transfer Protocol

MQTT : Message Queuing Telemetry Transport

Wi-Fi : Wireless Fidelity

MEMS : Micro-Electro-Mechanical Systems

SQLite : Structured Query Language (SQL) Lite

LoRaWAN : Long Range Wide-area network

Introduction Générale

Ces dernières années, les forêts ont suscité une attention croissante en raison de leur rôle essentiel dans la lutte contre la désertification, la protection de la couche d'ozone et la préservation des écosystèmes face à l'exploitation croissante du bois. Riches en biodiversité, elles abritent une multitude d'espèces végétales et animales [1] tout en fournissant des ressources indispensables à la subsistance et à l'économie de nombreuses communautés à travers le monde, notamment en Afrique.

Cependant, la demande en bois continue de croître sous l'effet de la croissance démographique et du développement économique accéléré, provoquant une pression accrue sur les ressources forestières. La coupe excessive de bois entraîne une érosion des sols, une dégradation des terres, et une perturbation des cycles hydrologiques, exacerbant les impacts de la déforestation et de la surexploitation des ressources naturelles. Cette situation pose de graves menaces pour la biodiversité, le climat mondial, et la stabilité socio-économique des régions dépendantes des forêts. C'est dans ce contexte que des organisations, comme la FAO, mènent des programmes comparant les méthodes classiques d'exploitation forestière avec des pratiques plus respectueuses de l'environnement et économiquement viables en Amérique latine, en Asie et en Afrique [2].

Face à ces nombreuses menaces, la surveillance devient ainsi un outil incontournable pour une gestion durable des ressources forestières et la protection de l'environnement. En effet, en matière de coupe de bois, une surveillance efficace permet de trouver un équilibre entre les impératifs économiques et écologiques, tout en respectant les droits des communautés locales et autochtones.

L'objectif principal de notre projet de plateforme de gestion d'alerte pour la surveillance de la coupe de bois est de développer un système avancé, réactif et efficace pour protéger les ressources forestières du Sénégal et promouvoir une gestion durable des forêts. Notre démarche s'appuie sur l'utilisation d'objets connectés pour surveiller en continu les activités de coupe de bois et les conditions environnementales des zones forestières, permettant ainsi une détection rapide des activités suspectes ou illégales. Les données collectées sont centralisées et traitées dans le cloud pour identifier des tendances, des anomalies et des indicateurs de comportements illicites. En générant des alertes en temps réel via des applications web et mobiles, cette plateforme fournit des informations précises et actuelles, facilitant la prise de décision et

permettant aux autorités compétentes de répondre rapidement et efficacement à toute activité anormale détectée. Ce mémoire est organisé comme suit :

Première Partie : ÉTAT DE L'ART ET PRÉSENTATION DE NOTRE CONTRIBUTION

❖ Chapitre 1 : Types d'alertes et Surveillance de la coupe de bois au Sénégal

Dans ce chapitre, nous examinons les différents types d'alertes, suivis d'un aperçu de la surveillance de la coupe de bois au Sénégal.

❖ Chapitre 2 : Etude de l'existant et présentation de notre contribution

Le chapitre 2 est consacré à l'examen des systèmes de gestion d'alerte existants dans la littérature, ainsi qu'à la présentation de notre contribution. Celle-ci vise à combler les lacunes identifiées et à proposer un modèle amélioré et adapté aux besoins actuels.

Deuxième Partie : CONCEPTION ET MISE EN OEUVRE

❖ Chapitre 1 : Conception de notre système

Nous détaillons ici la conception de notre système de gestion d'alerte pour la surveillance des coupes de bois, en nous appuyant sur des méthodologies de modélisation éprouvées. Nous commençons par une présentation des principes généraux des méthodologies UML et Merise, qui permettent de structurer et de formaliser notre conception, puis nous focalisons sur l'application d'UML pour la conception de notre système.

❖ Chapitre 2 : Développement et Mise en Œuvre

La partie dédiée à la mise en œuvre de notre système expose les étapes pratiques qui transforment la conception théorique en un système fonctionnel. Nous y présentons les outils et technologies utilisés, les configurations nécessaires, ainsi que les défis rencontrés et les solutions adoptées pour les surmonter.

Première Partie : État de l'art et présentation de notre contribution

Chapitre 1 : Types d'alertes et Surveillance de la coupe de bois au Sénégal

Introduction

Une alerte, en général, est un système de notification automatique qui informe les utilisateurs de nouvelles informations ou de changements spécifiques [3] . Elle est souvent utilisée pour maintenir une bonne réactivité face aux informations pertinentes et aux événements importants. La gestion des alertes est essentielle pour garantir une réponse rapide et appropriée aux informations critiques, tout en évitant la surcharge d'informations non essentielles. Ce processus implique la surveillance, la détection et la réponse aux événements critiques ou inhabituels pouvant affecter une organisation [4] . Ces alertes peuvent provenir de différentes sources, telles que des capteurs, des systèmes de surveillance, des employés ou des clients.

Dans ce chapitre nous allons parler des types d'alertes puis de donner un aperçu sur le Système de Surveillance de la Coupe de Bois au Sénégal.

I. Les types d'alertes

Dans notre société moderne, les alertes peuvent prévenir ou limiter les dommages humains et matériels, à condition qu'elles soient émises rapidement et qu'elles permettent aux intervenants et à la population concernée de se préparer adéquatement à la crise imminente [5]. Voici une présentation succincte de quelques types d'alertes couramment utilisées :

1. Alerte Météorologique

Une alerte météorologique est un message diffusé par les agences météorologiques pour prévenir le public et les autorités de conditions météorologiques dangereuses. Elle peut signaler des vents forts, des pluies torrentielles, des tempêtes, des orages violents, des tornades ou encore des brouillards denses. Ces phénomènes peuvent avoir des impacts significatifs sur les personnes et l'environnement. Ces alertes peuvent couvrir divers phénomènes, tels que les tempêtes, les ouragans, les tornades, les inondations, les vagues de chaleur, les vagues de froid, les tempêtes de neige ou les vents violents[6] . L'alerte météo indique les comportements à adopter et les mesures de sécurité à prendre selon le phénomène météorologique. Les pouvoirs publics, avertis par cette alerte, doivent relayer les informations localement et prendre toutes les précautions nécessaires. Ces alertes sont généralement classées par niveaux de gravité (par

exemple, vigilance jaune, orange ou rouge) pour indiquer le niveau de risque et les mesures à prendre. Elles sont diffusées par divers moyens, tels que la télévision, la radio, les sites web météorologiques et les applications mobiles, pour assurer une préparation adéquate de la population face aux risques météorologiques.

Au Sénégal, comme dans de nombreux autres pays, la vigilance météorologique est un système mis en place pour informer et alerter la population et les autorités sur les conditions météorologiques dangereuses ou potentiellement dangereuses. L'Agence Nationale de l'Aviation Civile et de la Météorologie (**ANACIM**¹) est l'organisme responsable de la surveillance et de la diffusion des alertes météorologiques au Sénégal. Le système de vigilance météorologique au Sénégal utilise généralement une classification par couleurs pour indiquer le niveau de danger [7] [3] .



Figure 1 : La carte de vigilance

2. Alerte de sécurité publique

Une alerte de sécurité publique est un avis émis par les autorités compétentes pour informer et avertir la population d'une situation de danger imminent ou potentiel qui pourrait menacer la sécurité publique. Ces alertes sont destinées à protéger les citoyens en leur fournissant des

¹ ANACIM <https://www.anacim.sn/>

informations essentielles et des instructions sur les mesures à prendre pour se protéger. Elles peuvent couvrir un large éventail de situations, telles que :

- **Catastrophes naturelles** : Tremblements de terre, inondations, ouragans, incendies de forêt ;
- **Incidents industriels** : Fuites de produits chimiques, explosions, accidents dans des installations nucléaires ;
- **Situations de violence ou de terrorisme** : Attaques terroristes, fusillades, prises d'otages ;
- **Urgences de santé publique** : épidémies, pandémies, contaminations alimentaires ou hydriques ;
- **Menaces environnementales** : Pollutions graves, marées noires.

Les alertes de sécurité publique incluent des informations sur la nature du danger, la zone affectée, les actions recommandées (telles que l'évacuation ou le confinement), et des conseils pratiques pour minimiser les risques. Ces alertes sont diffusées par divers canaux de communication pour atteindre rapidement et efficacement le public, y compris : la télévision, la radio, les téléphones mobiles, les sirènes et systèmes d'alarme publics, les sites web, les réseaux sociaux, les emails et systèmes de messagerie. Par exemple en France on a le Système d'Alerte et d'informations aux populations (**SAIP**) est un ensemble d'outils qui permet d'avertir la population d'une zone donnée, d'un danger imminent et de l'informer sur la nature du risque et le comportement à tenir. Lorsque les sirènes SAIP sonnent, des comportements réflexes de sauvegarde sont à adopter immédiatement : se mettre en sécurité, s'informer, ne pas aller chercher ses enfants à l'école et ne téléphoner qu'en cas d'urgence vitale [5]. Donc on peut dire que les alertes de sécurité publique sont importantes pour assurer une réponse rapide et coordonnée aux situations d'urgence, permettant ainsi de sauver des vies et de réduire les dommages.[8] [4]



3. Alerte en santé publique

Une alerte sanitaire est un avis émis par les autorités de santé publique pour informer la population et les professionnels de santé d'une menace imminente ou potentielle pour la santé publique. Ces alertes sont déclenchées pour des situations variées qui peuvent mettre en danger la santé des individus ou des communautés. L'alerte en santé publique vise à informer les décideurs d'un signal vérifié qui représente une menace pour la santé des populations, afin qu'ils puissent prendre les mesures appropriées pour l'investigation, l'évaluation, le contrôle et la prévention. [9] Voici quelques exemples de situations pouvant déclencher une alerte sanitaire :

- **Épidémies et Pandémies :**
 - Maladies infectieuses comme la grippe, le COVID-19, Ebola, etc.
 - Épidémies de maladies contagieuses nécessitant une vigilance accrue et des mesures de prévention.
- **Contaminations Alimentaires :**
 - Présence de pathogènes, de toxines ou de substances chimiques dans les aliments pouvant entraîner des intoxications alimentaires.
 - Retraits de produits alimentaires contaminés du marché.
- **Contaminations de l'Eau :**
 - Pollution de l'eau potable par des agents pathogènes, des produits chimiques ou des toxines.
 - Recommandations de ne pas consommer l'eau ou d'utiliser des alternatives sûres.
- **Incidents liés à des Produits Médicaux ou Pharmaceutiques :**
 - Retraits de médicaments ou de dispositifs médicaux dangereux ou défectueux.
 - Alerte sur les effets secondaires graves de certains traitements.
- **Risques Environnementaux :**
 - Exposition à des substances chimiques toxiques ou à des radiations.
 - Pollution de l'air, entraînant des risques pour la santé respiratoire.
- **Incidents Biologiques ou Chimiques :**
 - Fuites ou déversements de substances dangereuses dans l'environnement.
 - Attaques terroristes ou chimiques.

Les alertes sanitaires peuvent être transmises par divers canaux de communication, tels que les médias, Internet et les réseaux sociaux [10]. L'application **TousAntiCovid** est un outil complet d'alerte et d'information qui offre des conseils pratiques. Son objectif est de faciliter l'information des personnes en contact avec une personne testée positive à la COVID-19 et d'accélérer leur prise en charge, en complément de l'intervention des médecins et de l'Assurance Maladie. Elle permet également de suivre l'évolution de l'épidémie et de connaître les mesures à adopter, ce qui aide à rester vigilant et à adopter les bons gestes. En outre, elle simplifie l'accès à d'autres outils disponibles pour les citoyens souhaitant participer activement à la lutte contre l'épidémie [6].



Figure 3 : Application TousAntiCovid

4. Alerte maritime

Une alerte maritime est un avis émis par les autorités compétentes pour informer et avertir les navigateurs, les pêcheurs, et les populations côtières de conditions dangereuses en mer ou sur les côtes. Ces alertes visent à protéger la vie humaine, les biens et l'environnement marin en fournissant des informations capitales et des recommandations sur les actions à entreprendre. Voici quelques exemples de situations pouvant déclencher une alerte maritime :

- **Tempêtes** : Prévisions de tempêtes violentes pouvant provoquer des vents forts, des vagues hautes et des conditions de navigation dangereuses ;
- **Vagues et marées** : vagues scélérates, marées exceptionnelles ou raz-de-marée pouvant entraîner des inondations côtières et des risques pour les navires ;
- **Courants marins** : Courants de déchirure ou courants forts pouvant représenter un danger pour les nageurs et les embarcations légères ;

- **Naufrages et Collisions** : Informations sur des naufrages, des collisions entre navires ou des débris flottants pouvant constituer des dangers pour la navigation ;
- **Pollutions et Déversements** : Déversements d'hydrocarbures, fuites de produits chimiques ou autres polluants affectant les eaux marines et côtières ;
- **Incidents de sécurité** : attaques de piraterie, activités de contrebande ou autres menaces à la sécurité maritime.

Les alertes maritimes sont diffusées par différents moyens de communication pour toucher les personnes concernées, notamment via la radio VHF, les bulletins météorologiques, les panneaux d'affichage dans les ports, Internet et des applications mobiles. Ces alertes permettent de prendre des mesures préventives, comme modifier les plans de navigation, éviter certaines zones ou adopter des précautions spécifiques pour garantir la sécurité en mer [11] .

Le système d’alerte et de suivi AIS-SART pour les marins au Sénégal vise à assurer un secours d’urgence en cas de danger. Ainsi, un marin peut alerter les centres de contrôle ainsi que tous les bateaux équipés du système AIS à proximité.

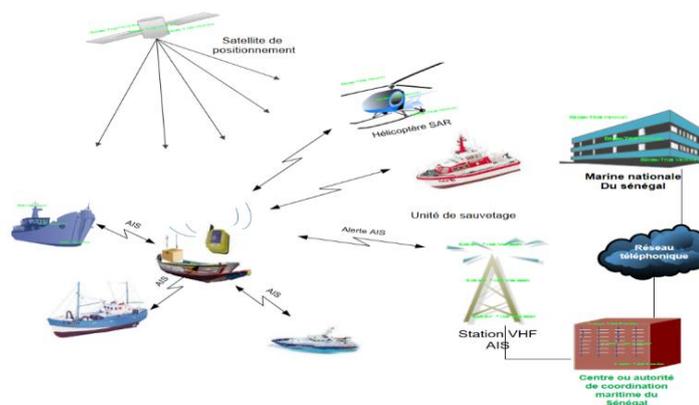


Figure 4 : Architecture du système d'alerte Maritime

5. Alerte industrielle

Une alerte industrielle est une notification émise dans un contexte industriel pour signaler des événements critiques ou anormaux. Ces alertes peuvent être liées à des problèmes de sécurité, des incidents techniques, des défauts de produits, ou des situations d'urgence. L'objectif est de permettre une réaction rapide pour minimiser les risques, protéger les employés, prévenir les

dommages matériels, et maintenir la continuité des opérations. Les alertes industrielles jouent un rôle clé dans la gestion des risques et la conformité aux normes de sécurité et de qualité.

FR-Alert est le nouveau système d'alerte et d'information des populations mis en place par les autorités françaises. Il permet de diffuser rapidement des alertes en cas d'accident industriel majeur, de catastrophe naturelle, d'attentat, etc. Ce dispositif informe sur la nature de l'événement, sa localisation, ainsi que les consignes et mesures de sécurité à suivre [12] . Il permet également de prévenir en temps réel toute personne possédant un téléphone portable se trouvant dans une zone de danger, en lui indiquant les comportements à adopter pour se protéger. En cas d'accident, une notification accompagnée d'un signal sonore spécifique sera envoyée, même si le téléphone est en mode silencieux [13] [8] .



Figure 5 : FR-Alert

II. La Surveillance de la Coupe de Bois au Sénégal

1. Situation de la coupe de bois au Sénégal

La coupe illégale de bois constitue une grave menace pour les écosystèmes forestiers du Sénégal, notamment en Casamance, une région réputée pour ses forêts denses et sa biodiversité. Alimenté [14] par une forte demande internationale, ce phénomène met en danger les ressources naturelles, les moyens de subsistance des communautés locales et les initiatives nationales pour la gestion durable des forêts. Les impacts environnementaux et socio-économiques de la coupe illégale de bois au Sénégal sont considérables. Chaque année, des milliers d'hectares de forêts disparaissent, entraînant la perte d'habitats pour la faune et l'érosion des sols, ce qui accélère l'avancée de la désertification. Cette déforestation [15] massive perturbe également les

écosystèmes, car la réduction des forêts diminue leur capacité à absorber le dioxyde de carbone, contribuant ainsi au changement climatique. Sur le plan économique, le Sénégal perd des revenus importants issus des taxes forestières, tandis que les communautés locales voient leurs ressources naturelles s'épuiser. Par ailleurs, la coupe illégale de bois engendre une déstabilisation sociale, car l'implication de réseaux criminels dans cette activité illicite crée des tensions avec les autorités et les populations locales.

Les espaces forestiers du Sénégal sont de plus en plus exploités pour la production de bois d'énergie. En effet, le Sénégal est considéré comme un pays boisé, avec des savanes arbustives et arborées au nord (Louga, Saint Louis et Matam) et au centre (Kaolack, Fatick, Kaffrine) et des forêts au sud (Casamance), le Sénégal voit son couvert végétal diminuer chaque année [16]. Au Sénégal, les populations des régions du Sud et du Sud-Est dépendent largement des forêts pour leur subsistance, notamment à travers la chasse, l'exploitation des produits forestiers ligneux (bois d'œuvre, d'artisanat, de service, bois énergie) et non ligneux (plantes médicinales, fruits forestiers), ainsi que l'accès à l'eau [17]. Les forêts jouent un rôle crucial dans la sécurité alimentaire et la réduction de la pauvreté. Cependant, l'exploitation abusive des forêts est de plus en plus courante. Les principaux enjeux de la gestion forestière sont désormais : maintenir la biodiversité et l'équilibre socio-écologique, limiter la dégradation des sols, stocker le carbone, et lutter contre l'exploitation abusive des ressources forestières [18].

Au Sénégal, plus de 80 % des ménages utilisent le bois et le charbon de bois comme principales sources d'énergie. Le bois est principalement utilisé dans les villages tandis que les ménages urbains modestes ont recours au charbon. La coupe intensive de bois, combinée aux feux de brousse fréquents durant la saison sèche, réduit la disponibilité de bois et de charbon pour la population. Les saisons sèches, de plus en plus intenses en raison des changements climatiques, empêchent certains types d'arbres de survivre ou de se régénérer naturellement [19]. Les coupes non contrôlées du bois et les défrichements à des fins agricoles représentent les deux principales causes directes du déboisement et de la déforestation. Un trafic illégal international du bois s'est développé en Casamance. La coupe illégale de bois au Sénégal, particulièrement en Casamance, a entraîné une déforestation significative ces dernières décennies. Entre 1987 et 2018, le couvert forestier de la zone frontalière entre la région de Kolda et la Gambie est passé de 80 728,4 hectares à 16 276,4 hectares, soit une réduction de près de 80 % [20]. Pour faire face à ce problème, des mesures courageuses ont été prises par les organisations de défense de l'environnement.

2. La Surveillance actuelle de la coupe de bois

Selon l'enquête que j'ai menée auprès du **Service régional des Eaux et Forêts de Ziguinchor** La dégradation des forêts en Casamance est le résultat de plusieurs menaces interconnectées, telles que les feux de brousse, la salinisation, l'acidification des sols, la coupe illicite de bois, l'expansion des activités agricoles et l'urbanisation. **Les agents des eaux et forêts** jouent un rôle important dans la surveillance des forêts. Ils mènent des campagnes de sensibilisation auprès des populations locales, effectuent des patrouilles (polices forestières) régulières pour prévenir et détecter les infractions, et adoptent des mesures préventives contre les feux de brousse, telles que l'ouverture de pare-feu et la pratique de feux précoces. En cas de feux actifs, ils interviennent avec des outils traditionnels, des unités de lutte et des techniques comme les batte feux. Bien que leur système de surveillance soit jugé efficace, il repose principalement sur l'utilisation du GPS et n'intègre pas encore les nouvelles technologies telles que les drones, les satellites ou l'Internet des objets (IoT). Pour limiter les exploitations frauduleuses et le trafic illicite de bois et dérivés, les agents des eaux et forêts effectuent régulièrement des missions de terrains de façons inopiné ou sur la base des informations en cas de constatations d'une infraction, un procès-verbal (PV) est dressé contre les contrevenants Dans la région de Ziguinchor l'exploitation des produits contingenté est fermée par arrêté. Cependant, des autorisations exceptionnelles d'abattage peuvent être délivrées par le service des eaux et forêts après avis préalable du maire concerné pour des raisons de sécurité ou de construction.

3. Buts de notre système de surveillance

Le système de surveillance de la coupe de bois au Sénégal est conçu pour apporter une solution technologique avancée à un problème environnemental critique. Il aide à prévenir la déforestation illégale, à protéger les forêts existantes, et à maintenir la biodiversité et les écosystèmes locaux. Les forêts jouent un rôle essentiel dans la séquestration du carbone, contribuant ainsi à la lutte contre le changement climatique. Ce système soutient également les communautés locales en garantissant une utilisation durable des ressources forestières, telles que le bois de chauffage et les produits forestiers non ligneux. En Casamance, par exemple, la coupe illégale de bois est liée à des conflits, et une meilleure surveillance peut aider à réduire ces tensions.

L'objectif principal de ce système est de mettre en place une plateforme de gestion des alertes pour améliorer l'efficacité des systèmes de surveillance. Grâce à l'utilisation de technologies avancées telles que les objets connectés, le système peut détecter rapidement les activités de

coupe illégale et permettre une intervention rapide des autorités compétentes. La plateforme centralisée de gestion des alertes permettra de suivre en temps réel les incidents de déforestation, et les agents des eaux et forêts pourront recevoir des notifications instantanées et accéder à des données précises pour planifier leurs interventions. Les technologies de traitement des données et les algorithmes peuvent analyser les tendances de déforestation et fournir des informations précieuses pour la prise de décision, aidant ainsi le gouvernement à élaborer des politiques plus efficaces et à cibler les zones à risque. En rendant les données de surveillance accessibles, le système renforce la transparence et la responsabilité, permettant aux communautés locales et aux organisations de la société civile de participer à la surveillance. En somme, l'intégration de technologies avancées dans le système de surveillance de la coupe de bois au Sénégal permettra de protéger les ressources naturelles, de soutenir les communautés locales et de renforcer les capacités du gouvernement et des agents des eaux et forêts à gérer durablement les forêts.

Voici les objectifs spécifiques de ce système :

- **Protection des Ressources Forestières :**

Prévention de la déforestation : Détecter et empêcher les activités illégales de coupe de bois pour préserver les forêts.

Conservation de la Biodiversité : Protéger les habitats naturels et la biodiversité en réduisant la déforestation et la dégradation des terres.

- **Renforcement des Capacités de Surveillance :**

Technologie Avancée : Utiliser des technologies avancées telles que l'Internet des Objets (IoT), et le cloud pour améliorer la surveillance et la détection des activités de coupe de bois.

- **Réactivité et Coordination :**

Automatisation des alertes : Mettre en place un système d'alerte automatisé pour signaler rapidement les activités suspectes et permettre une intervention rapide.

Coordination Interinstitutionnelle : Faciliter la coopération entre les différentes parties prenantes, y compris les autorités gouvernementales, les ONG, les communautés locales et les forces de l'ordre, pour une gestion efficace des forêts.

- **Transparence et Responsabilité :**

Suivi et Évaluation : Suivre et évaluer les activités de coupe de bois et les interventions, en assurant une transparence et une responsabilité accrues dans la gestion forestière.

- **Soutien au Développement Durable :**

Économies Locales : Soutenir les économies locales en garantissant l'accès durable aux ressources forestières pour les communautés dépendantes des forêts.

Atténuation du Changement Climatique : Contribuer à la lutte contre le changement climatique en réduisant les émissions de gaz à effet de serre liées à la déforestation.

- **Fonctionnalités de l'Application :**

Mettre en place une plateforme de gestion des alertes dédiée à la surveillance forestière pour détecter les activités de coupe illégale de bois.

Envoyer des alertes en temps réel aux utilisateurs via des interfaces web et mobile pour permettre une intervention rapide.

Archiver les alertes pour assurer un suivi et une traçabilité des incidents.

Identifier les complices impliqués dans les activités illégales.

En résumé, l'objectif du système de surveillance de la coupe de bois au Sénégal est de protéger et de gérer durablement les ressources forestières, de renforcer les capacités de surveillance et de réactivité, de promouvoir la transparence et la responsabilité, et de soutenir le développement durable et la lutte contre le changement climatique. Ce système vise à garantir que les forêts sénégalaises continuent de fournir des bénéfices écologiques, économiques et sociaux pour les générations actuelles et futures.

Conclusion

En conclusion, ce chapitre a offert une vue d'ensemble essentielle de la Généralités sur la gestion des alertes dans le système de la surveillance dans le domaine de la coupe de bois. Dans la première partie, nous avons exploré les différents types d'alertes qui existent et la seconde partie a fourni un aperçu du système de surveillance de la coupe de bois au Sénégal, mettant en lumière les outils, les technologies, et les stratégies actuellement en place pour surveiller et gérer les ressources forestières du pays.

Chapitre 2 : Etude de l'existant et présentation de notre contribution

Introduction

Plusieurs études ont proposé des systèmes de gestion d'alerte, chacun cherchant à améliorer la rapidité, la précision et l'efficacité des notifications. Dans le chapitre 2 nous allons examiner les systèmes de gestion d'alertes existantes dans la littérature et présenter notre contribution, qui vise à combler les lacunes identifiées et à proposer un modèle amélioré et adapté aux besoins actuels.

I. Les travaux proposés dans la littérature des systèmes de gestion des alertes existantes

Dans cette partie, nous aborderons les différentes approches de gestion des alertes en utilisant diverses technologies. Nous examinons aussi comment chaque technologie contribue à l'identification, la communication et la résolution des problèmes potentiels de manière efficace et dynamique.

1. Surveillance illégale de la découpe d'arbres au moyen de dispositifs IdO intelligents à très faible puissance

Le système développé par Alessandro et ses collaborateurs allie efficacité énergétique et surveillance à grande échelle grâce à l'utilisation de la technologie LPWAN et des réseaux de neurones convolutifs (CNN). Il détecte automatiquement les activités illégales de coupe de bois et autres dangers via des nœuds IoT à faible consommation énergétique. Ces nœuds surveillent l'environnement sonore, distinguent les bruits naturels des sons suspects et transmettent les données via des passerelles LoRa au serveur réseau LoRaWAN pour analyse et gestion efficace. Les alertes générées sont basées sur des sons spécifiques tels que les scies à main, les tronçonneuses et les feux, et sont transmises aux opérateurs pour assurer la protection de l'environnement et la préservation des forêts. La figure ci-dessous illustre l'architecture du système proposé.

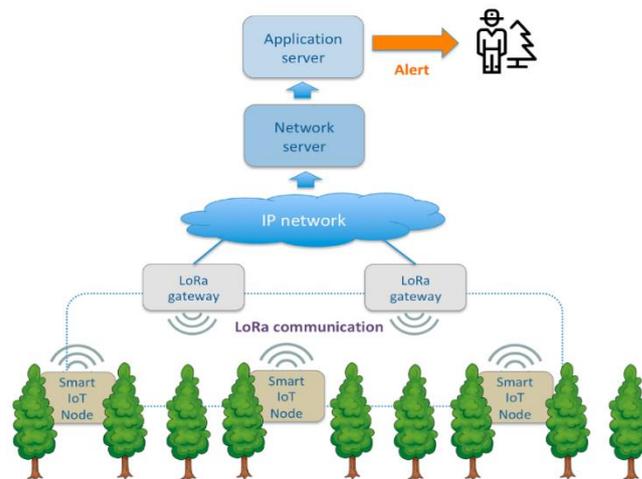


Figure 6 : Système de surveillance illégale de la découpe d'arbres

En résumé, ce système analyse les sons captés dans les forêts, distingue les bruits suspects et transmet les alertes via LoRaWAN. Cependant, il présente des limites telles que la couverture restreinte dans les zones très denses ou éloignées, la dépendance à la qualité des capteurs acoustiques, et le risque d'alertes faussement positives ou négatives dues à la variété des bruits environnementaux.

2. Détection automatique de la coupe d'arbres dans les forêts à l'aide de propriétés acoustiques

Ce système innovant lutte contre la déforestation illégale en détectant précocement les activités de coupe de bois grâce à un algorithme avancé combinant les techniques de clustering K-means, GMM et PCA. [21] Il analyse les signaux acoustiques dans les forêts pour identifier les sons caractéristiques de la coupe d'arbres, permettant une intervention rapide avec une précision de détection de 92 %. Afin de différencier les sons de coupe des autres bruits environnementaux, le système examine des enregistrements acoustiques comprenant 60 sons de coupe et 60 sons similaires (ballons éclatés, claps, creusement, martelage), classés en 5 catégories et répartis entre des ensembles d'entraînement et de test. Cette approche garantit une détection fiable et précise des activités suspectes, contribuant à la préservation des forêts et de leur biodiversité.

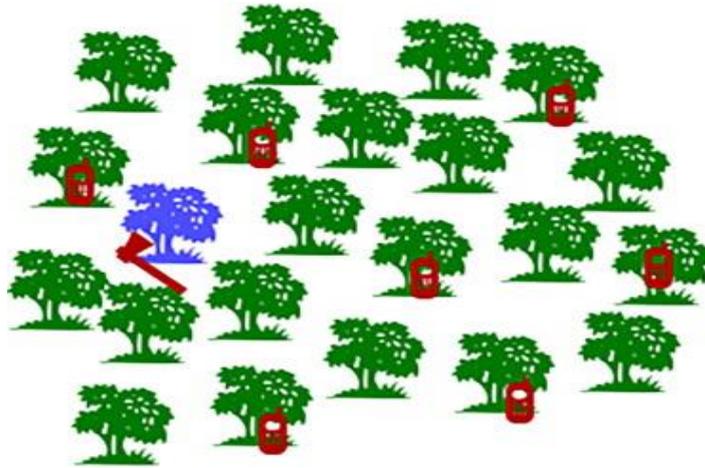


Figure 7 : Système à l'aide de propriétés acoustiques

Ce système offre une solution prometteuse pour la détection précoce de la déforestation illégale en analysant les signaux acoustiques. Cependant, ses performances peuvent être affectées par des facteurs externes tels que la qualité du son dans des environnements bruyants ou denses, ainsi que la possibilité de fausses alertes dues à la similitude des sons. Malgré ces limites, il constitue un outil puissant pour renforcer la surveillance des forêts et la préservation de la biodiversité.

3. Système de détection de l'exploitation forestière par le biais de la surveillance acoustique

Le système de surveillance forestière utilise un microphone pour capturer les sons liés à l'exploitation forestière et autres sons environnementaux. Ces échantillons audio sont transmis sans fil à un serveur pour traitement et analyse. L'alimentation de la station de surveillance est assurée par un panneau solaire, et la communication avec le serveur utilise des technologies comme le Wi-Fi, Zigbee ou un réseau mobile, selon les conditions de la zone forestière.

Au serveur, le signal audio est prétraité avant d'être analysé à l'aide de méthodes d'apprentissage automatique pour détecter les sons d'exploitation forestière. [22] La classification est binaire, distinguant les sons liés à l'exploitation des autres bruits. En cas de détection d'activité illégale, une alarme est déclenchée pour avertir les autorités via un système de gestion, SMS ou appel téléphonique aux patrouilles. Le système modulaire permet d'adapter les composants selon les besoins des gestionnaires forestiers, tout en maintenant une surveillance efficace.

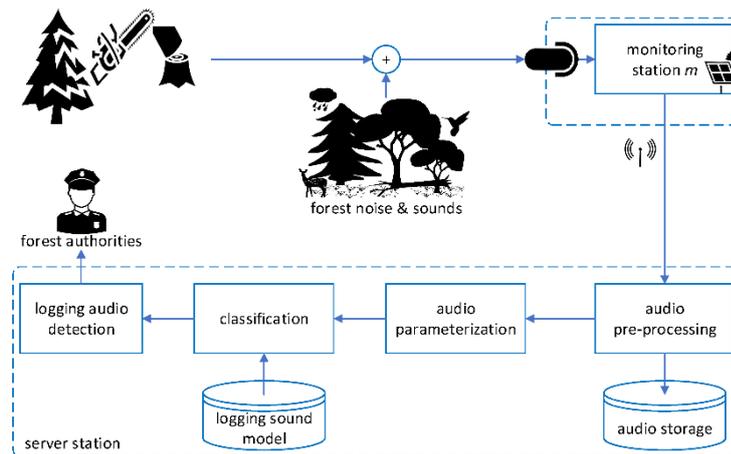


Figure 8 : Système de détection par le biais de la surveillance acoustique

On peut déduire que ce système de surveillance forestière, utilisant des microphones pour capter les sons d'exploitation forestière et les transmettre à un serveur pour analyse avec des méthodes d'apprentissage automatique. Bien que flexible et efficace, il présente des limites liées aux conditions de transmission sans fil et aux risques de fausses détections en raison des bruits environnementaux.

4. SAIP « Application d'alerte et d'information des populations en cas de crise »

ABDMEZIANE Choumeysa a développé un **système intelligent** permettant de communiquer directement avec la police algérienne en cas de danger et d'évaluer le niveau de risque dans les cités ou les régions. [23] Ce système vise à aider les forces de l'ordre à mieux gérer la sécurité en fournissant des informations cruciales sur la dangerosité des situations et des zones. SAIP² « Système d'alerte et d'information des populations » il s'agit de l'application d'alerte géolocalisée du gouvernement, conçue pour informer les utilisateurs en cas de crise majeure, d'événement important ou de situation exceptionnelle. Dans sa phase de lancement, elle offre aux utilisateurs un service gratuit pour recevoir des alertes ciblées en fonction de leur localisation, ainsi que des notifications pour d'autres zones. Donc les fonctionnalités de l'application sont :

- Notifications géolocalisées en cas d'alerte
- Informations et consignes comportementales

² <https://w>

[Owww.bouches-du-rhone.gouv.fr/Actions-de-l-Etat/Prevention-des-risques/Exercices-de-gestion-de-crise/Les-sirenes-SAIP](https://www.bouches-du-rhone.gouv.fr/Actions-de-l-Etat/Prevention-des-risques/Exercices-de-gestion-de-crise/Les-sirenes-SAIP)

- Jusqu'à 8 zones géographiques (communes/codes postaux) à suivre
- Viralisation des messages sur les réseaux sociaux (Twitter/Facebook)



Figure 9 : SAIP

Ce système est conçu pour être évolutif, avec des mises à jour régulières pour intégrer de nouvelles fonctionnalités et améliorer l'expérience utilisateur en fonction des retours des utilisateurs et des acteurs de la sécurité. Toutefois les inconvénients du système SAIP comprennent la dépendance à une connexion Internet stable, des problèmes potentiels de précision des alertes géolocalisées, et le risque de surcharge d'information pour les utilisateurs. Des préoccupations existent également concernant la protection des données personnelles, l'accès limité pour certains utilisateurs, la diffusion non vérifiée des messages sur les réseaux sociaux et une charge de travail accrue pour les forces de l'ordre.

5. Le système d'alerte et de prévention du risque

Nadeen Salameh a étudié un système d'alerte entre véhicules qui utilise des technologies de capteurs et des dispositifs informatiques. Ces technologies sont installées à la fois dans les véhicules et sur l'infrastructure routière. Le but de ce système est de détecter les obstacles ou les risques de collision et de générer des alertes pour les conducteurs. Cela permet de prévenir les accidents et d'améliorer la sécurité routière. Ce système de communication entre véhicules est crucial pour la sécurité sur les routes, car il permet une réaction rapide aux dangers potentiels. Les capteurs détectent les obstacles et les risques, tandis que les dispositifs informatiques analysent ces données et envoient des alertes aux conducteurs. Cette technologie

peut grandement réduire le nombre d'accidents en fournissant des informations en temps réel et en permettant aux conducteurs de prendre des décisions éclairées.[24]

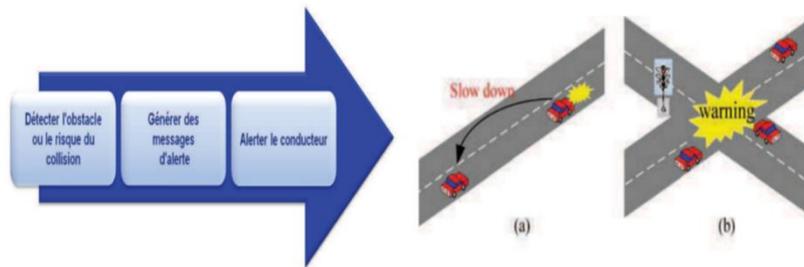


Figure 10 : La génération d'alerte au véhicule communicant

En résumé, ce système améliore la sécurité routière en permettant une détection précoce des dangers et une communication rapide entre les véhicules et l'infrastructure routière. Cependant, ils présentent des désavantages tels que des coûts élevés, une complexité technique, des problèmes de fiabilité des capteurs, des préoccupations de confidentialité et des défis liés à l'interférence et à la compatibilité. Ces aspects doivent être pris en compte pour optimiser leur efficacité.

6. Système d'alerte sur obstacle (OAS)

Dans cet article, le système d'alerte sur obstacles (OAS) est présenté comme une solution innovante pour la détection des obstacles, développée à l'aide d'un dispositif d'enregistrement de données hétérogènes. Le **système d'alerte (OAS)** est une solution conçue pour améliorer la sécurité et l'autonomie des utilisateurs de fauteuils roulants électriques (FRE). Ce système utilise des capteurs à ultrasons et infrarouges pour détecter les obstacles autour du fauteuil roulant. Lorsqu'un obstacle est détecté, le système informe l'utilisateur de sa présence et de son emplacement à travers des signaux haptiques (vibrations), visuels (écrans ou lumières) ou audio (bips ou messages vocaux), selon le choix de l'utilisateur. Les données collectées par les capteurs, ainsi que les informations sur le fauteuil roulant (position du joystick, profil de l'utilisateur, vitesse maximale, etc.), sont stockées sur une carte SD. Ces données sont ensuite transférées à une tablette et conservées dans une base de données sur le Cloud, où elles peuvent

être consultées en temps réel ou analysées ultérieurement pour améliorer le système et adapter les réglages aux besoins spécifiques de l'utilisateur.[25]

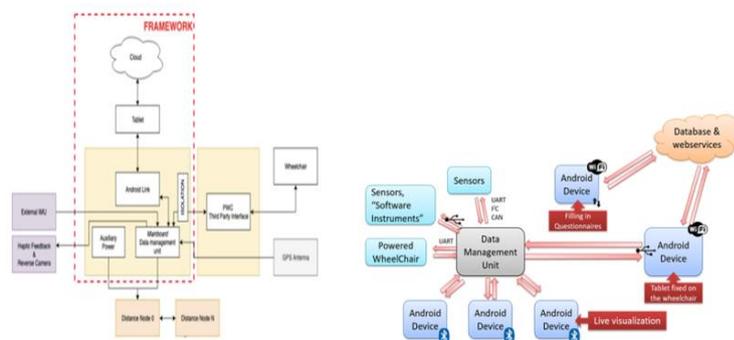


Figure 11 : Schéma général du système d'alerte d'obstacles (OAS)

Ce système modulaire et adaptable vise à offrir une meilleure autonomie et sécurité aux utilisateurs de fauteuils roulants, en leur fournissant des informations précises et en temps réel sur leur environnement. Néanmoins le système est difficile à installer et à entretenir en raison de sa complexité technique, les capteurs peuvent parfois manquer de fiabilité, il y a une forte dépendance à la technologie, des préoccupations liées à la confidentialité des données stockées sur le Cloud, et une consommation d'énergie accrue qui peut diminuer l'autonomie du fauteuil roulant.

7. Système d'alerte à base de la carte IOIO

Le système d'alerte à base de la carte IOIO, conçu par **Bennour et Chenait**, est un dispositif créatif destiné à détecter les collisions de véhicules et à envoyer automatiquement des notifications tout en enregistrant des données essentielles telles que la position GPS, la vitesse et l'accélération. Ce système, conçu pour être économique, utilise une application Android qui détecte les accidents à l'aide d'un accéléromètre externe. Les informations relatives au véhicule sont acquises et communiquées à l'appareil Android via la carte d'acquisition IOIO pour analyse. En cas d'accident, l'application enregistre les coordonnées GPS et envoie un SMS instantané à un contact d'urgence avec la position précise du véhicule. De plus, un appel automatique est passé au 1055 (gendarmerie nationale) pour dispatcher les services d'urgence et assurer la sécurité des personnes impliquées [26].

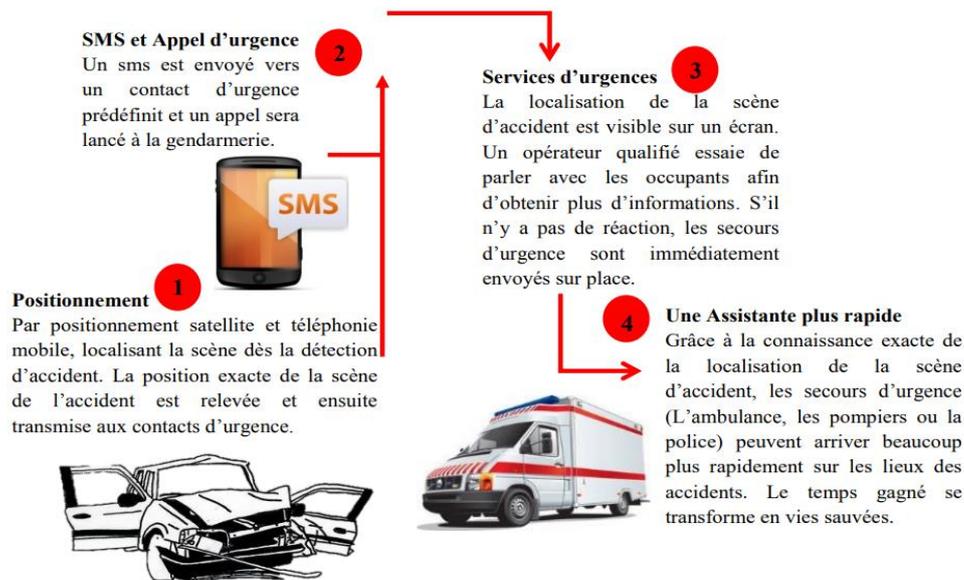


Figure 12 : Système d'alerte d'accident à base de la carte IOIO

On peut en déduire qu'il améliore significativement la sécurité des véhicules en fournissant une réponse rapide et efficace en cas d'accident. En revanche, les désavantages incluent la fiabilité des capteurs, la dépendance à la technologie, les préoccupations concernant la confidentialité des données, la complexité technique et la consommation d'énergie.

8. Système d'alerte en Temps Réel utilisant le Réseau Satellitaire Iridium

Dans le cadre de leur étude, les auteurs ont conçu un système capable de transmettre des informations fiables en temps réel en utilisant des pièges photographiques connectés au réseau satellitaire Iridium, intégrant un modèle d'intelligence artificielle, des pièges photographiques de série et du matériel spécialisé. L'objectif était de mettre en place un système robuste et fonctionnel sur le terrain en un an. Ce système repose sur un microcontrôleur Raspberry Pi qui se connecte au Wi-Fi pour récupérer les images des pièges photographiques. Une fois les images récupérées, elles sont analysées par un algorithme d'intelligence artificielle capable de classifier les espèces capturées (comme les éléphants, les humains, etc.). Après cette analyse, les données, y compris les espèces identifiées, les métadonnées telles que l'heure, la date, le lieu, ainsi que les données des capteurs de température et d'humidité, sont stockées dans une base de données. Ensuite, ces informations sont transmises via un modem au réseau satellitaire Iridium. Les données sont encodées et envoyées à une application web basée sur le cloud. Ce processus permet de fournir instantanément aux utilisateurs des messages WhatsApp et des alertes via la

plateforme web, leur permettant de réagir rapidement aux situations détectées sur le terrain [27] [6] .



Figure 13 : Système d'alerte via le réseau satellitaire iridium

Ce système offre une solution avancée pour la surveillance et la gestion de la faune, en permettant une réponse rapide et efficace grâce à la combinaison de la technologie des pièges photographiques, de l'intelligence artificielle (IA) et de la connectivité satellitaire mais il y' a une complexité technique élevée, une consommation d'énergie importante et des coûts élevés pour le matériel et l'abonnement satellite.

9. Système d'alerte médical pour la surveillance de la santé à distance utilisant des capteurs et l'informatique en nuage

Indumathy N et Dr. Kiran Kumari Patil ont développé un système novateur utilisant l'informatique en nuage pour collecter les paramètres vitaux de santé des patients et générer des alertes pour les soignants et les médecins en cas d'urgence. Ce système repose sur des capteurs qui surveillent en continu des indicateurs tels que la pression artérielle, le rythme cardiaque et la détection des chutes. Les données collectées par ces capteurs sont transmises en temps réel à un serveur en nuage, où elles sont stockées et analysées pour une surveillance à long terme. L'informatique en nuage permet aux soignants et aux médecins d'accéder aux données de santé des patients à tout moment et de n'importe où via Internet. Cela facilite une surveillance continue et proactive de l'état de santé des patients, permettant une intervention rapide en cas de détection d'anomalies ou de situations d'urgence. Les alertes générées par le système sont

envoyées immédiatement aux soignants et aux médecins [28] , assurant ainsi une réponse rapide et efficace [7].

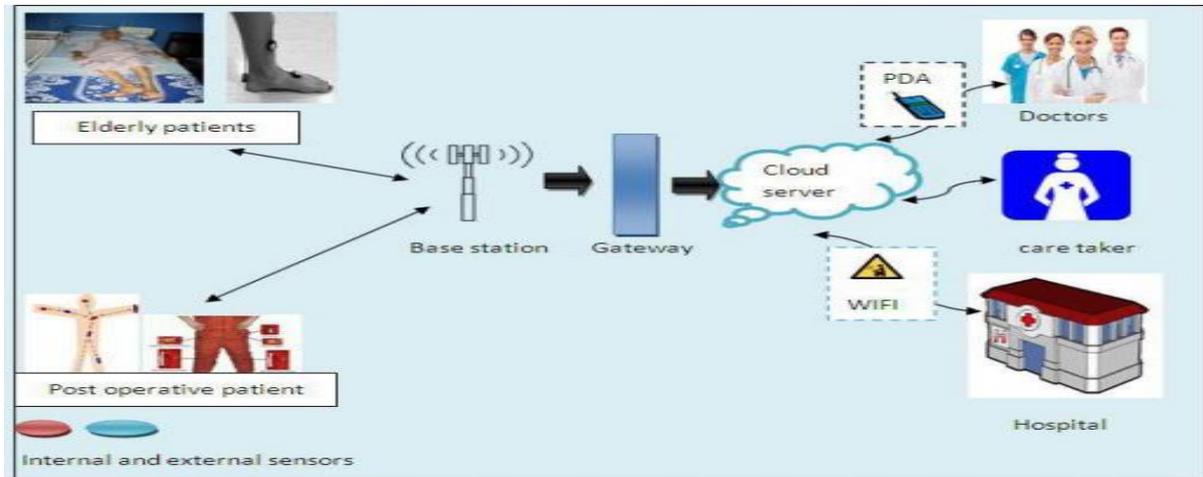


Figure 14 : Système d'alerte utilisant des capteurs et l'informatique en nuage

Ce système propose une solution avancée pour la gestion de la santé à distance, en améliorant la qualité des soins et la sécurité des patients grâce à une surveillance continue et un accès facile aux données médicales pour les professionnels de la santé. Cependant les lacunes du système d'alerte médicale comprennent des préoccupations en matière de sécurité et de confidentialité des données, une dépendance à une connexion Internet stable, des problèmes de précision des capteurs, et la nécessité d'une gestion efficace des alertes. Les coûts élevés, ainsi que les besoins en formation et en maintenance, peuvent également poser des défis.

II. Tableau comparatif des différents systèmes utilisés dans la littérature

Dans cette section, nous allons présenter un tableau comparatif des différents systèmes d'alerte proposés dans la littérature. Ces systèmes, développés pour divers contextes et publics, utilisent des technologies avancées pour améliorer la sécurité et la gestion des incidents. Le tableau ci-dessous résume les principales caractéristiques et objectifs de chaque système, facilitant ainsi une compréhension globale de leurs fonctionnalités et applications.

Les solutions existantes pour la surveillance des forêts présentent indéniablement des avantages, telles que l'utilisation de technologies avancées pour la détection acoustique ou l'intégration de réseaux satellites

Critères	Système de Surveillance intelligents à très faible puissance	Détection automatique de la coupe d'arbres à l'aide de propriétés acoustiques	Alerte en temps réel via réseau satellitaire Iridium	Système de détection de la surveillance acoustique
Méthode de détection	IoT à faible consommation d'énergie (nœuds de détection acoustique)	Algorithme avancé (K-means, GMM, PCA) pour analyser les signaux acoustiques	Analyse d'images de pièges photographiques via IA pour identifier les espèces	Apprentissage automatique pour classifier les sons d'exploitation forestière
Signaux analysés	Sons de coupe d'arbres et autres bruits environnementaux	Signaux de coupe d'arbres et autres sons environnementaux	Images capturées par pièges photographiques et classification des espèces	Sons d'exploitation forestière et autres bruits environnementaux
Transmission des données	LoRaWAN pour relier capteurs et serveur	LoRaWAN	Transmission via réseau satellitaire Iridium à une application web cloud	Wi-Fi, Zigbee ou réseau mobile selon les conditions de la zone
Avantages	<ul style="list-style-type: none"> - Haute précision de détection - Utilisation de techniques avancées pour une détection précise. 	<ul style="list-style-type: none"> - Solution modulaire et adaptable à différents besoins - Transmission sans fil flexible selon les conditions locales 	<ul style="list-style-type: none"> - Transmission instantanée via réseau Iridium - Alerte rapide via WhatsApp, permettant des réponses immédiates 	<ul style="list-style-type: none"> - Autonomie énergétique grâce au panneau solaire - Détection à distance dans des zones isolées
Limites	<ul style="list-style-type: none"> - Dépendance à l'algorithme pour des signaux spécifiques - Risque de faux positifs ou négatifs selon la variabilité du son 	<ul style="list-style-type: none"> - Risque de faux positifs ou négatifs selon la variabilité du son - Risque de faux positifs et dépendance à l'algorithme pour la détection des sons spécifiques 	<ul style="list-style-type: none"> - Dépendance au réseau satellitaire Iridium et aux pièges photographiques - Coûts et Complexité technique - Consommation d'énergie 	<ul style="list-style-type: none"> - Sensibilité aux bruits environnementaux perturbateurs - Efficacité limitée dans des zones très reculées ou sans réseau

pour une communication à distance. Cependant, elles souffrent également de certaines limites, notamment des coûts élevés, une complexité d'installation, ou une adaptation limitée au contexte local. Pour répondre à ces limites, nous proposons une solution innovante spécifiquement conçue pour les besoins de surveillance des forêts au Sénégal. Notre système, qui se base sur la détection des sons de scies et de tronçonneuses grâce à des seuils acoustiques, vise à surmonter ces défis en offrant une solution efficace et adaptée. Nous détaillerons cette approche dans les sections suivantes. Face à ces défis, notre solution innovante vise à combler ces lacunes et à offrir une alternative plus efficace, mieux adaptée aux besoins actuels.

III. Présentation de notre contribution

1. Contexte et Objectifs

Le Sénégal dépend fortement de ses forêts, qui sont essentielles pour fournir de l'énergie, des matériaux de construction et soutenir les activités agricoles et pastorales. Au-delà de leur valeur économique, elles sont cruciales pour la préservation de la biodiversité, la régulation climatique et la lutte contre la désertification, ce qui est vital face au changement climatique. Cependant, des régions comme la Casamance sont particulièrement vulnérables. L'insécurité et la pauvreté limitent les alternatives économiques, poussant les populations locales à exploiter intensivement les ressources forestières, souvent illégalement, menaçant ainsi la durabilité des forêts et les moyens de subsistance locaux. Il est donc impératif de promouvoir une gestion durable des ressources forestières, développer des sources d'énergie alternatives et diversifier les opportunités économiques rurales. Les forêts sénégalaises, en fournissant du bois de chauffe et du bois d'œuvre, soutiennent de nombreux emplois et contribuent significativement à l'économie locale. De ce fait, elles sont gravement menacées par la coupe illégale et les abattages frauduleux [29]. Pour relever ces défis, il est crucial de renforcer les outils de surveillance et de gestion des forêts. Bien que divers systèmes de gestion et d'alerte existent déjà à l'échelle mondiale, ils sont souvent coûteux, complexes à installer ou inadaptés aux réalités locales. Pour relever ces défis, il est crucial de renforcer les outils de surveillance et de gestion des forêts. Bien que divers systèmes de gestion et d'alerte existent déjà à l'échelle mondiale, ils sont souvent coûteux, complexes à installer ou inadaptés aux réalités locales.

Notre solution se distingue comme la première au Sénégal à se concentrer spécifiquement sur la surveillance et la gestion des alertes liées à la coupe illégale de bois. En intégrant des technologies avancées de l'Internet des Objets (IoT), notre système offre une surveillance en temps réel des forêts, détectant et signalant les activités suspectes d'abattage illégal dès leur

survenue. Ce dispositif permet une surveillance proactive, garantissant une réponse rapide et ciblée aux infractions. Il inclut des fonctionnalités telles que la collecte et l'analyse des données en temps réel, la génération d'alertes instantanées via des interfaces web et mobiles, et l'archivage des incidents pour un suivi précis et une meilleure traçabilité.

En résumé, notre système de gestion d'alerte pour la surveillance forestière comble les lacunes des systèmes actuels avec une solution innovante, efficace et adaptée au contexte sénégalais. Il vise à protéger les forêts et les ressources naturelles, préserver l'équilibre écologique et assurer un avenir durable pour les communautés locales.

Architecture logique de notre Système

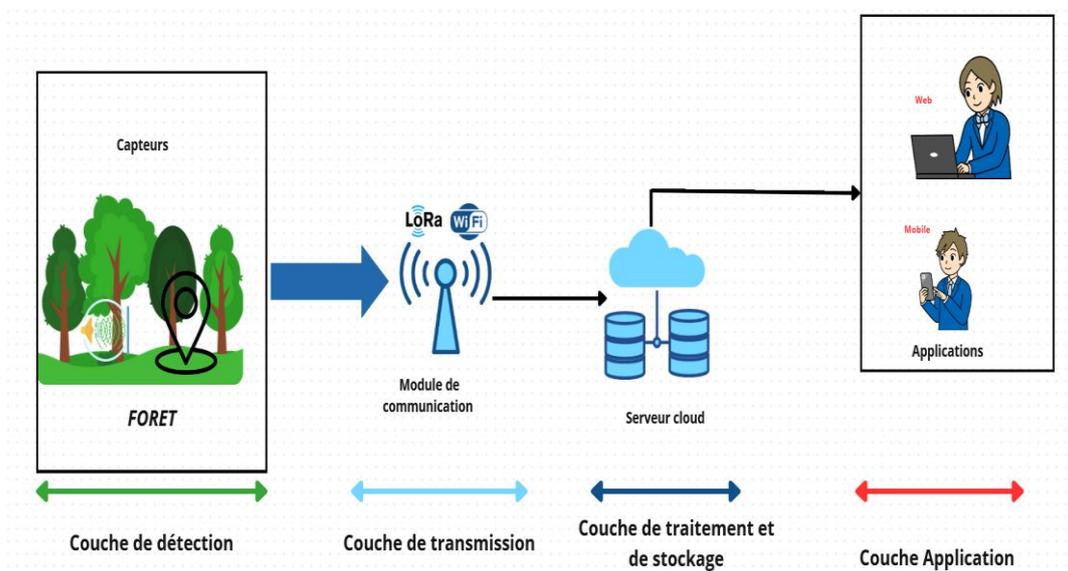


Figure 15 : Architecture logique

Description de l'architecture logique

Cette section décrit l'architecture logique du système de gestion d'alerte destiné à la surveillance de la coupe illégale de bois au Sénégal. Le système est organisé en plusieurs couches distinctes :

- **Couche de Détection (Capteurs) :**

Cette première couche est responsable de la détection des événements dans la forêt. Elle utilise des capteurs pour recueillir des données telles que des sons, de la géolocalisation et d'autres

signaux environnementaux. Les capteurs sont déployés sur le terrain pour identifier les activités suspectes liées à la coupe illégale de bois.

- **Couche de Transmission** (Module de communication) :

Une fois les données captées par les capteurs, elles doivent être transmises à un serveur central. Cette tâche est assurée par le module de communication, qui peut utiliser des technologies telles que LoRa (Long Range) ou Wi-Fi pour envoyer les données collectées vers le cloud.

- **Couche de Traitement et de Stockage** (Serveur cloud) :

Les données transmises par le module de communication arrivent dans le serveur cloud. Ici, elles sont traitées et stockées. Le traitement des données implique l'analyse et l'interprétation des informations pour détecter les activités illégales. Le stockage garantit que les données sont conservées de manière sécurisée et peuvent être consultées ultérieurement si nécessaire.

- **Couche d'Application** (Applications) :

La dernière couche comprend les interfaces utilisateur, notamment les applications web et mobiles. Ces interfaces permettent aux utilisateurs finaux d'accéder aux données traitées, de visualiser les alertes et de prendre les mesures appropriées. Elles offrent une interaction directe avec le système, facilitant la gestion et la surveillance des activités illégales.

Cette architecture en couches permet une surveillance efficace et organisée, de la détection des événements à l'action des utilisateurs via des applications conviviales.

2. Organigramme de fonctionnement de notre système

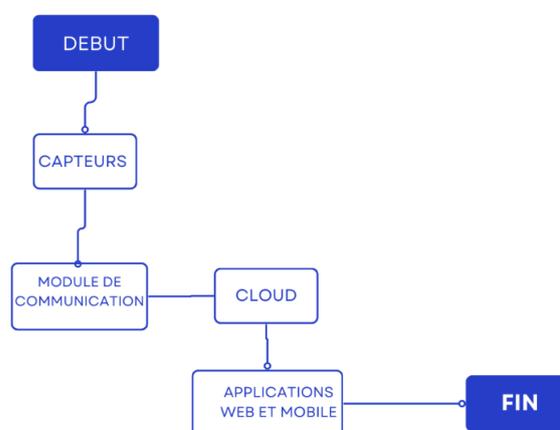


Figure 16 : Organigramme de notre système

Description des Étapes

Nous utilisons les technologies **LoRa** et **Wi-Fi** pour transmettre les données dans notre système de surveillance forestière, assurant ainsi une communication fiable et à faible consommation d'énergie entre les dispositifs IoT dans les zones isolées. Les capteurs compatibles LoRa détectent divers événements, tels que les bruits de tronçonneuses et les mouvements suspects dans la forêt. Ces capteurs collectent des informations essentielles pour une surveillance en temps réel, notamment des données de géolocalisation. Les données sont ensuite transmises soit directement via LoRa pour assurer une couverture étendue, soit via Wi-Fi dans les zones disposant d'une connexion locale.

Les informations recueillies sont envoyées vers une plateforme cloud pour y être analysées et stockées. Sur cette plateforme, des services de traitement analysent les données en temps réel et génèrent automatiquement des alertes en cas de détection d'activités suspectes. Ces alertes sont accessibles via une application web et mobile, offrant une interface utilisateur intuitive aux agents de surveillance, leur permettant ainsi de réagir rapidement et efficacement aux menaces de déforestation illégales. Ce système assure donc une couverture fiable avec une infrastructure adaptée aux diverses conditions des zones surveillées.

Le système débute par la configuration et l'installation des capteurs, incluant un capteur acoustique ADMP pour détecter les sons de tronçonneuses et un module GPS pour localiser précisément les activités suspectes. Deux configurations sont possibles pour la transmission des données :

Transmission via ESP8266 et Wi-Fi : Le microcontrôleur ESP8266, une fois installé et configuré, collecte les données des capteurs. Lorsque le capteur acoustique atteint un seuil prédéfini indiquant une activité suspecte, l'ESP8266 traite ces données et transmet la localisation de l'événement en temps réel à ThingSpeak via Wi-Fi.

Transmission via Arduino Mega et module Dragino LoRa Shield : Dans les zones où la connectivité Wi-Fi est limitée, le système utilise une configuration Arduino Mega associée au module Dragino LoRa Shield. Les données collectées sont envoyées via LoRa à la passerelle LG01-N, qui les transmet ensuite à la plateforme ThingSpeak. Ce mode de transmission permet une couverture étendue avec une faible consommation d'énergie.

Dans le cloud, les données sont stockées de manière sécurisée et mises à disposition pour les applications connectées. Dès réception des données, l'application web et mobile génère une

alerte automatique contenant des informations détaillées, incluant un lien vers l'emplacement suspecté, accessible en temps réel par les utilisateurs. Les autorités compétentes, notamment les agents des eaux et forêts, reçoivent ces alertes en temps réel, leur permettant ainsi d'organiser rapidement une intervention appropriée.

En complément, le système intègre une composante de sensibilisation des communautés locales sur l'importance de la conservation des forêts et les impacts de la coupe illégale de bois. Enfin, le système est régulièrement surveillé et maintenu pour garantir son bon fonctionnement. Les algorithmes d'analyse sont continuellement améliorés pour accroître la précision des détections et réduire les faux positifs, assurant ainsi une surveillance proactive et efficace des ressources forestières.

Conclusion

Dans ce chapitre, nous avons recensé les solutions proposées dans la littérature scientifique pour les systèmes de gestion d'alertes. Bien que de nombreuses approches existent, aucune n'est spécifiquement dédiée à la surveillance des coupes de bois à l'aide d'objets connectés. Nous avons donc conçu notre propre solution, qui offre deux options de transmission pour s'adapter aux différentes contraintes de connectivité sur le terrain. Cette flexibilité permet une surveillance efficace des activités illégales, tout en améliorant la fiabilité, réduisant la dépendance technologique, renforçant la confidentialité des données, augmentant l'efficacité énergétique, et réduisant les coûts. Ces avantages contribuent à l'implémentation d'un système de gestion d'alertes performant et économique pour lutter contre la coupe illégale de bois.

Deuxième Partie : Conception et Mise en œuvre de notre système

Chapitre 1 : Conception de notre Système

Introduction

Dans ce chapitre, nous présentons la conception de notre système de gestion des alertes pour la surveillance de la coupe de bois, en utilisant des méthodologies de modélisation éprouvées. Pour structurer et formaliser notre conception, nous expliquons d'abord les principes généraux de UML et de Merise, avant de nous concentrer sur UML pour la conception de notre système. Nous proposons les différents diagrammes et modèles utilisés, illustrant chaque étape avec des exemples concrets tirés de notre projet. Ce faisant, nous fournissons une vue d'ensemble complète de notre processus de conception, mettant en lumière les choix techniques et méthodologiques qui sous-tendent notre solution de gestion d'alerte pour la surveillance de la coupe de bois.

I. UML et Merise

Dans cette partie on va parler des principes généraux de UML et de la méthode MERISE, ainsi qu'une mention des acteurs.

1. Principes généraux de Merise et UML

UML est un langage de modélisation standardisé, largement utilisé pour visualiser, spécifier, construire et documenter les artefacts d'un système logiciel [30]. Il propose divers diagrammes pour capturer les aspects statiques et dynamiques d'un système, tels que les diagrammes de cas d'utilisation, de classes, de séquence et d'activités. Ces outils graphiques facilitent la communication entre les différentes parties prenantes du projet et assurent une compréhension commune des exigences et des solutions proposées.

Merise, quant à elle, est une méthode de modélisation et de conception de systèmes d'information très utilisée en France. Elle se distingue par son approche systématique et rigoureuse, séparant les phases de conception en modèles conceptuels, logiques et physiques [31]. Merise met l'accent sur la compréhension des besoins organisationnels avant de traduire ces besoins en spécifications techniques détaillées. Les modèles conceptuels des données (MCD) et des traitements (MCT), ainsi que leurs dérivés logiques et physiques, constituent le cœur de cette méthode [32].

Pour la conception de notre système de gestion d'alerte, nous avons choisi d'utiliser UML en raison de sa flexibilité et de son adoption internationale. En suivant les étapes de modélisation UML, nous détaillons les différents aspects de notre solution, en commençant par les cas d'utilisation et les interactions avec les acteurs, puis en décrivant les structures de données et les comportements dynamiques. Cette approche nous permet de concevoir un système robuste, bien documenté et facile à maintenir, répondant ainsi aux défis spécifiques de la surveillance des coupes de bois.

a. Enumération des acteurs

Un acteur est une entité externe qui interagit avec le système pour atteindre un objectif spécifique. Cette interaction peut être directe ou indirecte et est généralement représentée dans des diagrammes de cas d'utilisation en UML (Unified Modeling Language). Dans le cadre de notre système, les acteurs identifiés sont les suivants :

Administrateur : L'administrateur du système de gestion de surveillance de la coupe de bois est chargé de la configuration, de la maintenance, de la sécurité et de l'assistance du système. Il installe et configure les capteurs IoT, les modules de transmission, la plateforme cloud et les applications pour garantir une surveillance efficace. Il gère les utilisateurs et leurs autorisations, surveille le fonctionnement du système, résout les problèmes techniques et effectue les mises à jour. En outre, il met en place des mesures de sécurité pour protéger les données, fournit un support technique et forme les utilisateurs.

Utilisateurs : Les utilisateurs (c'est-à-dire les autorités, les Agents des Eaux et Forêts, Forces de l'ordre ou Agences de protection de l'environnement) sont responsables de gérer les alertes générées par le système de surveillance de la coupe de bois. Elles reçoivent les alertes, visualisent les données associées pour évaluer la situation, et interviennent sur le terrain pour stopper les activités suspectes et protéger les forêts. Leur rôle est important pour assurer une réponse rapide et efficace aux incidents détectés.

Dispositifs : Les dispositifs sont des entités matérielles et logicielles conçues pour détecter les activités de coupe de bois et transmettre les informations aux systèmes de surveillance. Ils se composent de plusieurs éléments clés : des capteurs IoT, qui repèrent les mouvements, les vibrations, et localisent précisément les lieux d'activités suspectes, ainsi que des modules de transmission qui envoient les données captées par les capteurs au cloud, via WiFi ou LoRa, pour analyse.

Service de notification : Le service de notification est chargé de la génération des alertes, de l'analyse des données et du stockage des informations collectées. Il comprend deux principales fonctions : le service d'analyse et de stockage des données, qui analyse les données transmises par les capteurs pour identifier les activités suspectes et génère des alertes basées sur ces résultats, tout en stockant les données pour une utilisation future et le service de génération d'alertes, qui envoie des notifications aux autorités compétentes lorsqu'une activité suspecte est détectée.

II. Les diagrammes utilisés

Dans cette section, nous allons énumérer les principaux diagrammes utilisés pour modéliser notre système. Dans le cadre de ce projet, deux options de transmission de données ont été envisagées : **la transmission via LoRa** et **la transmission via WiFi**. La transmission LoRa, adaptée aux environnements éloignés et nécessitant une faible consommation d'énergie, permet une couverture étendue, idéale pour une utilisation en zones forestières. En revanche, la transmission via WiFi, bien que plus limitée en portée, offre un débit de données plus élevé et une configuration simplifiée pour des environnements équipés d'infrastructures réseau locales.

Pour le **déploiement**, nous avons opté pour la transmission via **WiFi** et **Lora** afin de faciliter les tests et le suivi des données en temps réel sur la plateforme ThingSpeak. Cette approche permet une intégration rapide et une surveillance directe de l'architecture, permettant ainsi de valider la fonctionnalité du système.

1. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est un diagramme UML qui illustre les interactions entre les acteurs externes et les fonctionnalités du système. Il permet d'identifier ces acteurs, tels que les utilisateurs finaux, les administrateurs ou d'autres systèmes, et de décrire les cas d'utilisation, c'est-à-dire les services que le système offre pour répondre aux besoins des acteurs. Chaque cas d'utilisation correspond à une série d'actions permettant d'accomplir une tâche spécifique. Le diagramme montre aussi les interactions entre les acteurs et les cas d'utilisation à l'aide de flèches, illustrant comment le système répond aux demandes des acteurs.

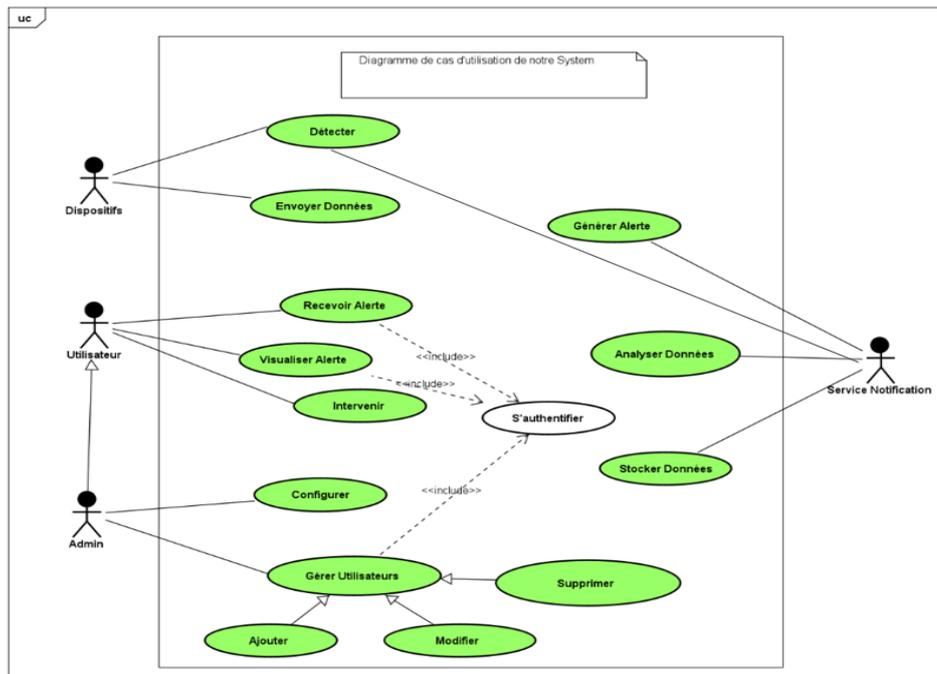


Figure 17: Diagramme de cas d'utilisation de notre système

2. Description textuelle des cas d'utilisation

Description textuelle « Détecter Activités »

Cas d'utilisation	Détecter activités
Acteurs Principaux	Les dispositifs
Objectifs	<ul style="list-style-type: none"> ✓ Détecter en temps réel les activités suspectes de coupe de bois illégale ✓ Notifier les parties prenantes pour une intervention rapide.
Préconditions	<ul style="list-style-type: none"> ✓ Les dispositifs IoT sont installés et opérationnels dans les zones ciblées. ✓ Les réseaux de communication sont fonctionnels.
Postconditions	<ul style="list-style-type: none"> ✓ Les activités suspectes de coupe de bois sont détectées et enregistrées. ✓ Des alertes sont envoyées aux utilisateurs concernés ✓ Les utilisateurs reçoivent les informations nécessaires pour intervenir rapidement.

Tableau 2: Description textuelle Détecter Activités

Description textuelle « Envoyer données »

Cas d'utilisation	Envoyer données
Acteurs	Dispositifs (capteurs de sons et GPS)
Objectifs	<ul style="list-style-type: none"> ✓ Assurer la transmission des données détectées par les dispositifs IoT vers le cloud pour analyse et gestion des alertes.
Préconditions	<ul style="list-style-type: none"> ✓ Les dispositifs IoT ont détecté des données pertinentes (vibrations et localisations). ✓ Le réseau est opérationnel ✓ Les serveurs cloud sont prêts à recevoir et traiter les données.
Postconditions	<ul style="list-style-type: none"> ✓ Les données détectées par les dispositifs IoT sont correctement transmises via le réseau wifi ou lora. ✓ Les données sont reçues et stockées dans le cloud pour une analyse ultérieure.

Tableau 3: Description textuelle Transmettre données

Description textuelle « Analyser et Stocker données »

Cas d'utilisation	Analyser et Stocker données
Acteurs	Service de notification (Cloud)
Objectifs	<ul style="list-style-type: none"> ✓ Analyser les données en temps réel pour détecter les activités suspectes. ✓ Stocker les données pour une consultation et une analyse ultérieure.
Préconditions	<ul style="list-style-type: none"> ✓ Les capteurs IoT doivent être installés et opérationnels dans les zones forestières ciblées. ✓ La connectivité via lora ou wifi doit être en place pour relayer les données des capteurs aux serveurs cloud. ✓ Le service de notification doit être configuré et capable de recevoir, traiter, et stocker les données. ✓ Les administrateurs doivent avoir accès à l'interface de gestion pour configurer les paramètres d'analyse et de stockage.
Postconditions	<ul style="list-style-type: none"> ✓ Les données des capteurs sont analysées et les résultats de l'analyse sont stockés de manière sécurisée sur le serveur cloud. ✓ Les alertes pertinentes sont générées et transmises aux utilisateurs via les applications web et mobiles. ✓ Les utilisateurs peuvent consulter les rapports et les historiques des activités détectées. ✓ Les administrateurs peuvent accéder aux données stockées pour des analyses avancées et des ajustements de configuration.

Tableau 4 : Description textuelle Analyser et Stocker les données

Description textuelle « Générer Alertes »

Cas d'utilisation	Générer Alertes
Acteurs	Service de notification (Cloud)
Objectifs	<ul style="list-style-type: none"> ✓ Générer des alertes en temps réel basées sur les résultats de l'analyse des données des capteurs. ✓ Notifier les utilisateurs des activités suspectes détectées pour permettre une intervention rapide.
Préconditions	<ul style="list-style-type: none"> ✓ Les données des capteurs doivent être reçues et analysées par le service de notification. ✓ Les seuils et critères de génération d'alertes doivent être configurés par l'administrateur de la plateforme. ✓ La connectivité pour la transmission des alertes doit être établie ✓ Les utilisateurs doivent être enregistrés et avoir accès aux applications web et mobiles pour recevoir les alertes.
Postconditions	<ul style="list-style-type: none"> ✓ Les alertes sont générées et envoyées aux utilisateurs en temps réel. ✓ Les utilisateurs sont informés des activités suspectes et peuvent prendre des mesures appropriées. ✓ Les alertes générées sont stockées pour une consultation et une analyse ultérieure ✓ L'administrateur peut examiner les alertes et ajuster les paramètres de génération si nécessaire

Tableau 5: Description textuelle Générer Alertes

Description textuelle « Recevoir alertes »

Cas d'utilisation	Recevoir alertes
Acteurs	Utilisateurs
Objectifs	<ul style="list-style-type: none"> ✓ Assurer que les autorités reçoivent les alertes en temps réel pour permettre une réaction rapide aux activités suspectes. ✓ Fournir des informations détaillées et pertinentes dans les alertes pour faciliter l'évaluation de la situation et la prise de décision.
Préconditions	<ul style="list-style-type: none"> ✓ Les alertes doivent être générées par le service de notification suite à l'analyse des données des capteurs. ✓ Les autorités doivent être enregistrées dans le système et avoir accès aux applications web et mobiles. ✓ Les canaux de communication (notifications SMS) doivent être configurés et opérationnels. ✓ Les critères de réception d'alertes doivent être définis pour les différentes autorités en fonction de leurs responsabilités et zones de surveillance.
Postconditions	<ul style="list-style-type: none"> ✓ Les utilisateurs reçoivent les alertes en temps réel avec toutes les informations nécessaires. ✓ Les alertes sont disponibles pour consultation ultérieure dans les applications web et mobiles. ✓ Les autorités peuvent réagir rapidement et efficacement aux alertes reçues. ✓ Les autorités peuvent utiliser les alertes reçues pour évaluer les tendances et ajuster leurs stratégies de surveillance et d'intervention.

Tableau 6 : Description détaillée Recevoir Alertes

Description textuelle « Intervenir sur le terrain »

Cas d'utilisation	Intervenir sur le terrain
Acteurs	Utilisateurs (Autorités)
Objectifs	<ul style="list-style-type: none"> ✓ Réagir rapidement et efficacement aux alertes générées par le système pour vérifier et contrer les activités illégales. ✓ Assurer la sécurité des zones forestières en intervenant sur les lieux des activités suspectes. ✓ Collecter des preuves et prendre des mesures contre les personnes impliquées dans les activités illégales.
Préconditions	<ul style="list-style-type: none"> ✓ Les alertes doivent être reçues par les autorités via les applications web et mobiles. ✓ Les équipes d'intervention doivent être disponibles et prêtes à agir en réponse aux alertes. ✓ Les informations de l'alerte (localisation, nature de l'activité, heure, etc.) doivent être complètes et précises pour faciliter l'intervention.
Postconditions	<ul style="list-style-type: none"> ✓ Les équipes d'intervention se rendent sur le terrain et prennent les mesures nécessaires en réponse aux alertes. ✓ Les résultats de l'intervention sont documentés et mis à jour dans le système. ✓ Les autorités utilisent les informations collectées pour évaluer l'efficacité des interventions et ajuster les stratégies de surveillance et d'intervention.

Tableau 7 : Description détaillée Intervenir

Description textuelle « Configurer système »

Cas d'utilisation	Configurer système
Acteurs	Administrateur
Objectifs	<ul style="list-style-type: none"> ✓ Configurer les paramètres du système de surveillance pour assurer une détection et une gestion efficaces des activités illégales. ✓ Définir les critères de génération d'alertes basés sur les données des capteurs. ✓ Assurer la connectivité et l'intégrité des capteurs IoT et du service de notification. ✓ Gérer les utilisateurs et les rôles pour contrôler l'accès et les permissions au sein du système.
Préconditions	<ul style="list-style-type: none"> ✓ Le système doit être installé et accessible par l'administrateur. ✓ Les capteurs IoT doivent être installés et opérationnels dans les zones surveillées. ✓ L'infrastructure de serveurs cloud doit être en place et fonctionnelle.
Postconditions	<ul style="list-style-type: none"> ✓ Le système de surveillance est correctement configuré et opérationnel. ✓ Les critères et seuils de génération d'alertes sont définis et optimisés. ✓ Les capteurs IoT sont connectés et intégrés au système de notification. ✓ Les utilisateurs et leurs rôles sont gérés et les accès sont sécurisés.

Tableau 8: Description détaillée Configurer Système

Description textuelle « Gérer utilisateurs »

Cas d'utilisation	Gérer utilisateurs (Autorités)
Acteurs	Utilisateurs
Objectifs	<ul style="list-style-type: none"> ✓ Créer, modifier et supprimer des comptes utilisateurs pour contrôler l'accès au système. ✓ Définir et attribuer des rôles et des permissions pour chaque utilisateur afin d'assurer une utilisation sécurisée et appropriée du système. ✓ Maintenir la sécurité et l'intégrité des données et des fonctionnalités du système en gérant les accès des utilisateurs.
Préconditions	<ul style="list-style-type: none"> ✓ L'administrateur doit avoir des droits d'accès suffisants pour gérer les utilisateurs et leurs permissions. ✓ Le système doit disposer d'une interface de gestion des utilisateurs accessible à l'administrateur ✓ Les utilisateurs doivent être enregistrés ou en cours d'enregistrement dans le système
Postconditions	<ul style="list-style-type: none"> ✓ Les comptes utilisateurs sont créés, modifiés ou supprimés selon les besoins. ✓ Les rôles et permissions sont correctement définis et attribués, garantissant un accès sécurisé et approprié. ✓ Les utilisateurs peuvent accéder aux fonctionnalités et aux données du système en fonction de leurs rôles et permissions.

Tableau 9 : Description Détaillée Gérer Autorités (Utilisateurs)

Description Détaillée « S'authentifier »

Cas d'utilisation	S'authentifier
Acteurs	Administrateur, Autorités
Objectifs	<ul style="list-style-type: none">✓ Vérifier l'identité de l'utilisateur pour garantir que seules les personnes✓ Autorisées accèdent au système.✓ Fournir un accès sécurisé aux fonctionnalités et aux données du système.✓ Protéger les informations sensibles contre les accès non autorisés.
Préconditions	<ul style="list-style-type: none">✓ L'utilisateur doit disposer d'un compte enregistré dans le système avec des identifiants valides (nom d'utilisateur et mot de passe).✓ Le système d'authentification doit être opérationnel et accessible.✓ Les informations de connexion doivent être protégées par des mesures de sécurité appropriées (chiffrement, protocoles sécurisés, etc.).
Postconditions	<ul style="list-style-type: none">✓ L'utilisateur est authentifié et a accès aux fonctionnalités et aux données du système en fonction de ses rôles et permissions.✓ En cas d'échec de l'authentification, l'utilisateur reçoit un message d'erreur approprié et n'est pas autorisé à accéder au système.✓ Les tentatives de connexion réussies et échouées sont enregistrées pour des audits de sécurité.

Tableau 10 : Description détaillé S'authentifier

3. Diagramme de séquence

Un diagramme de séquence est un outil UML qui modélise les interactions entre les composants d'un système au fil du temps, en montrant comment les objets ou acteurs échangent des messages pour accomplir une tâche. Les participants sont représentés par des lignes de vie avec des flèches indiquant les messages échangés. Il peut également modéliser des conditions et des

boucles. La sémantique statique d'un diagramme de séquence est de vérifier s'il est cohérent avec la déclaration de diagramme de classe ainsi qu'avec sa structure arborescente bien formée.[33] Dans notre projet, il nous aide à concevoir et valider les interactions entre capteurs, cloud et interfaces utilisateur pour détecter et signaler les activités illégales, assurant ainsi la clarté et l'efficacité du système.

a. Diagramme de séquence du cas d'utilisation s'Authentifier

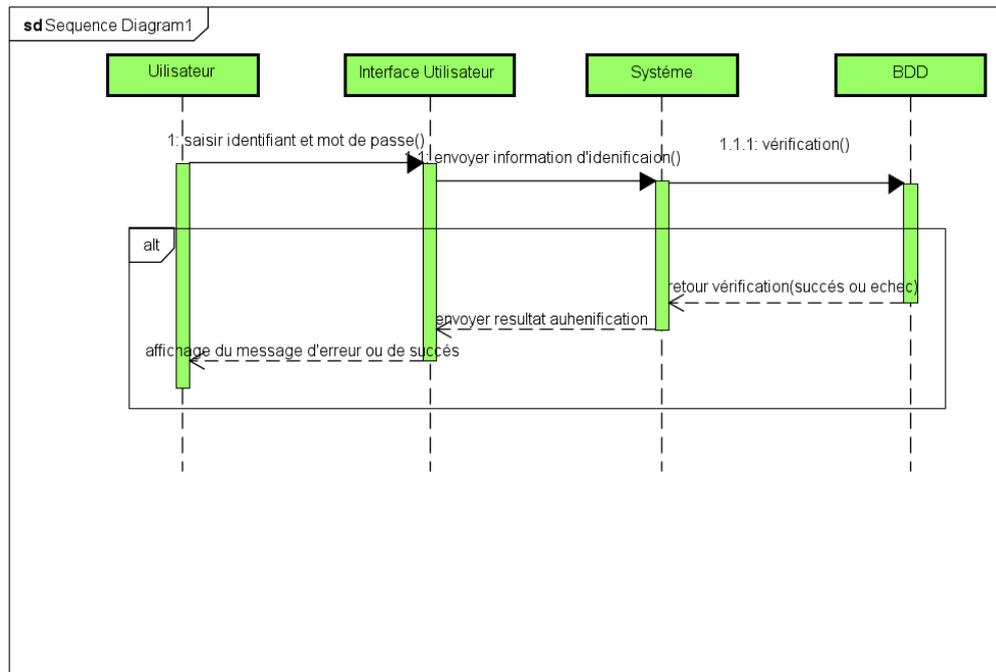


Figure 18: : Diagramme de séquence du cas d'utilisation s'Authentifier

b. Diagramme de séquence du cas d'utilisation Gérer utilisateur (Ajouter utilisateur)

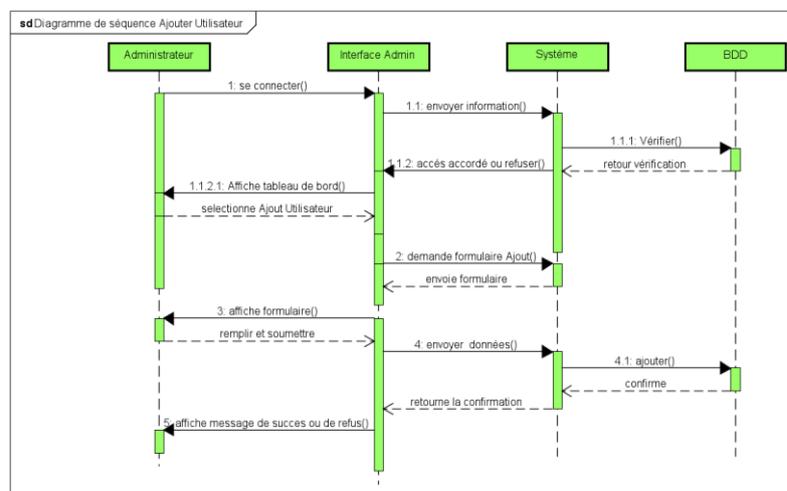


Figure 17 : Diagramme de séquence du cas d'utilisation Ajouter Utilisateur

c. Diagramme de séquence du cas d'utilisation Générer Alerte

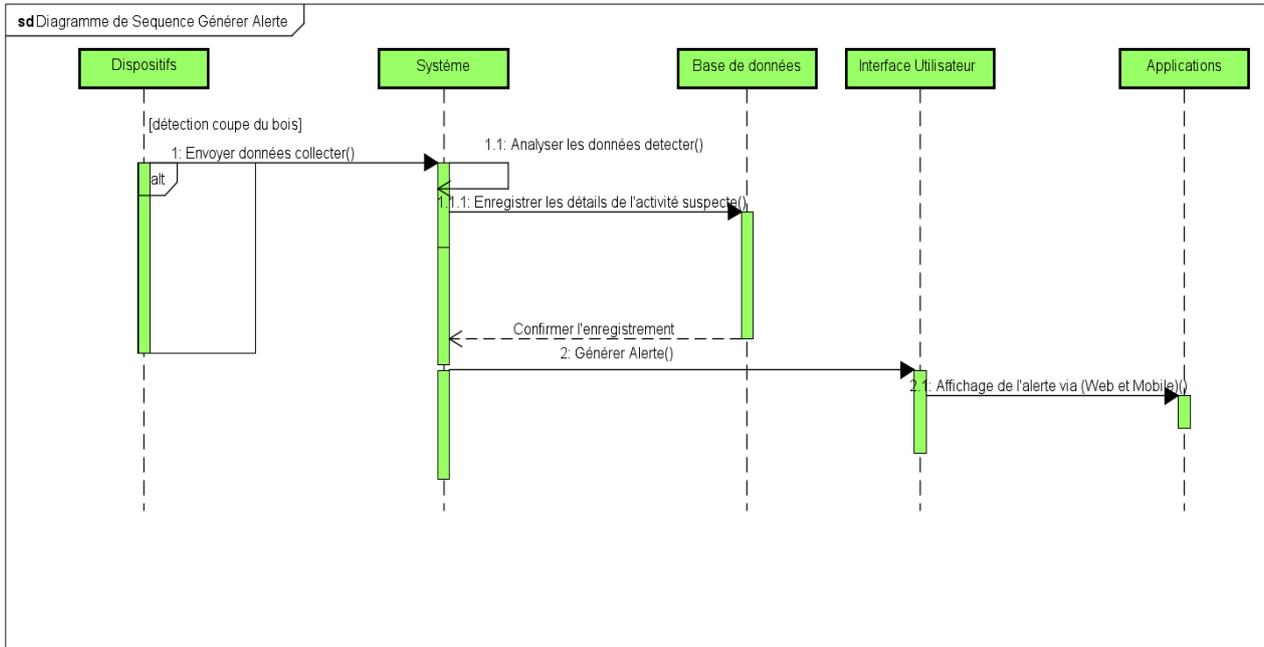


Figure 18 : Diagramme de séquence du cas d'utilisation Générer Alerte

4. Diagramme de classe

Un **diagramme de classe** est un type de diagramme utilisé dans la modélisation de systèmes orientés objet, en particulier dans le cadre du langage UML (Unified Modeling Language) qui facilite, indépendamment de tout langage de programmation, la conception de programmes, ainsi que leur description pour des non-informaticiens.[34] Il sert à représenter la structure statique d'un système en illustrant les classes qui le composent, leurs attributs, leurs méthodes, ainsi que les relations entre elles.

Voici les principaux éléments d'un diagramme de classe :

Classes : Représentées par des rectangles divisés en trois parties (nom, attributs, méthodes).

Attributs : Caractéristiques ou propriétés des classes.

Méthodes : Fonctions ou opérations que les classes peuvent effectuer.

Relations : Liaisons entre les classes, comme l'héritage, l'association, et la composition.

Les diagrammes de classes sont essentiels pour visualiser et planifier la structure d'un système orienté objet, facilitant ainsi la compréhension et la communication entre les membres d'une équipe de développement[35]. Voilà notre diagramme de classe :

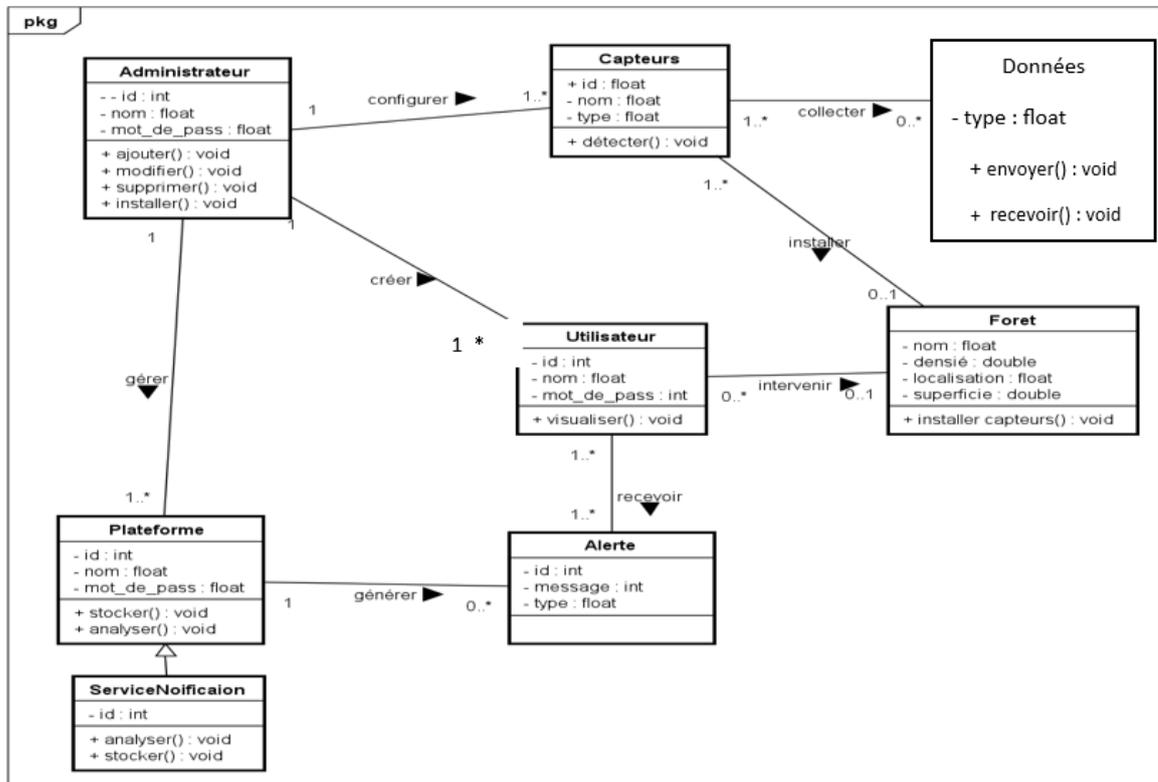


Figure 19 : Diagramme de classe de notre Système

III. Architecture physique du système

Notre système de surveillance forestière repose sur une architecture hybride intégrant deux modes de transmission de données : **LoRa** et **Wi-Fi**. Les capteurs IoT sont utilisés pour détecter les activités illégales, telles que les sons de tronçonneuses, ainsi que pour obtenir des données de géolocalisation précises. Ces capteurs transmettent les données au microcontrôleur pour traitement et envoi. Lorsque le Wi-Fi n'est pas disponible, les données collectées sont transmises par un **module LoRa (Dragino LoRa Shield)** vers une **passerelle LoRa LG01-N**. Cette passerelle, grâce aux protocoles **HTTP** ou **MQTT**, envoie ensuite les données au cloud, ce qui est idéal pour les zones forestières éloignées, car cette technologie permet une communication longue portée tout en étant économe en énergie. En revanche, dans les zones couvertes par un réseau Wi-Fi, le module **ESP8266** permet une transmission directe des

données vers le serveur cloud **ThingSpeak** en utilisant le protocole **HTTP POST**, assurant ainsi une gestion rapide et en temps réel des données collectées.

Une fois dans le cloud, les données transmises par LoRa et Wi-Fi sont agrégées et analysées via la plateforme **ThingSpeak**. En cas de détection d'activités suspectes, des alertes sont générées automatiquement. Ces alertes sont ensuite accessibles via des plateformes web et mobiles, permettant aux autorités de visualiser les informations en temps réel et de prendre des décisions rapides pour une intervention sur le terrain. Cette architecture repose sur l'utilisation de **technologies avancées de l'IoT** pour garantir une **surveillance en temps réel** et continue des forêts, assurant ainsi une couverture optimale et une flexibilité adaptée aux environnements variés.

On a la représentation de notre architecture :

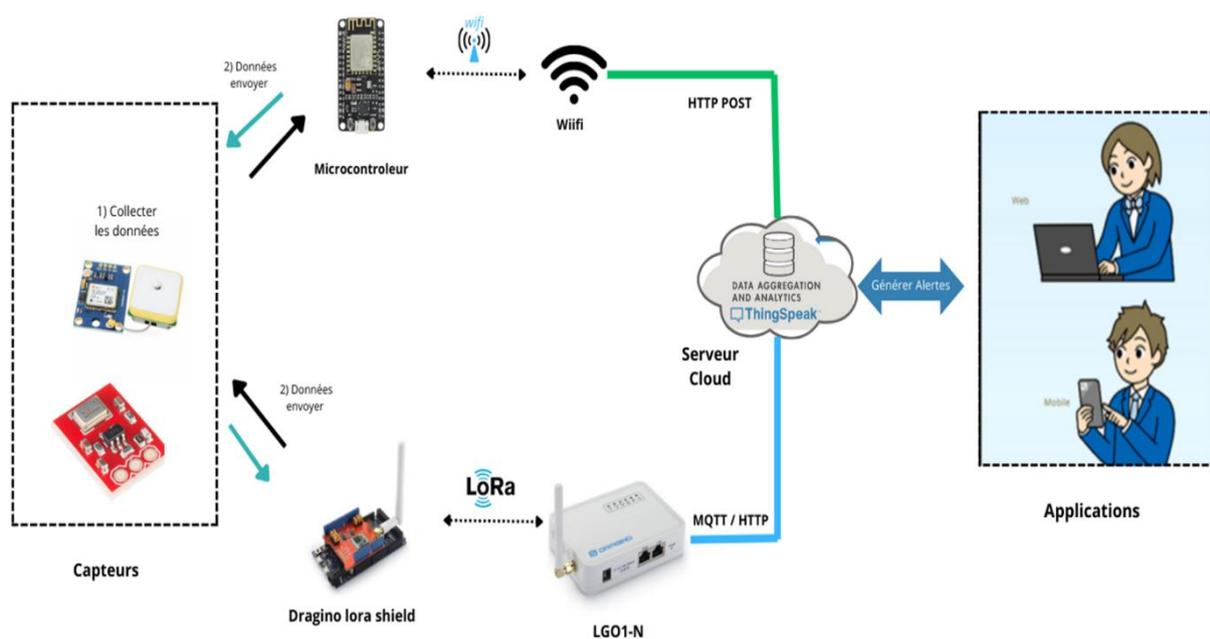


Figure 20 : Architecture Global du Système

IV. Description des composants de l'architecture

Dans cette partie, nous allons détailler les différents composants de l'architecture du système de gestion d'alerte pour la surveillance de la coupe illégale de bois au Sénégal. Cette description inclura les types de capteurs utilisés pour la détection des activités illégales, l'infrastructure de transmission des données via le microcontrôleur, ainsi que les deux modes de communication

disponibles : le réseau **LoRa** et le **Wi-Fi**. Nous expliquerons comment la **passerelle LoRa (LG01-N)** assure l'envoi des données sur de longues distances lorsque le Wi-Fi n'est pas disponible, et comment le **module ESP8266** permet la transmission directe vers le cloud en zones couvertes. Les données transmises sont ensuite agrégées et analysées en temps réel sur la plateforme cloud **ThingSpeak** pour générer des alertes en cas d'activités suspectes. Enfin, nous aborderons les interfaces utilisateur, à travers les applications web et mobiles, qui permettent une visualisation et une gestion efficace des alertes générées, facilitant ainsi l'intervention rapide des autorités locales.

1. Capteurs

Les capteurs IoT sont essentiels pour la collecte de données sur le terrain. Ils détectent les activités spécifiques liées à la coupe de bois. Dans notre système, nous utilisons :

a. Capteur acoustique ADMP401

Le capteur ADMP401 est un microphone MEMS (Micro-Electro-Mechanical Systems) analogique, essentiel pour la détection du son grâce à sa capacité à capter une large gamme de fréquences, de 100 Hz à 15 kHz, avec une sensibilité typique de -42 dBV/Pa. Il convertit les ondes sonores en signaux analogiques, permettant une détection précise même des sons faibles. Sa faible consommation d'énergie et sa petite taille le rendent particulièrement adapté aux dispositifs portables et aux systèmes intégrés, où la détection fiable du son est cruciale pour des applications telles que la reconnaissance vocale, la surveillance acoustique, et les systèmes d'enregistrement audio[36] .

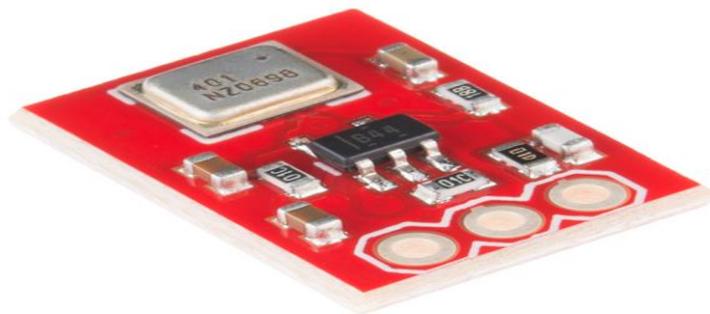


Figure 21 : Capteur acoustique ADMP401

Les caractéristiques principales du ADMP401 sont les suivantes :

Caractéristiques	ADMP401
Type	Microphone MEMS analogique
Plage de fréquences	100 Hz à 15 kHz
Sensibilité	- 38 dB ± 3 dB
Signal de rapport/bruit (SNR)	62 dB
Sortie	Signal analogique
Tension d'alimentation	1,5V à 3,3V
Consommation de courant	250 µA
Plage dynamique	100 dB SPL
Directivité	Omnidirectionnel
Dimensions	4.72 mm x 3.76 mm x 1.0 mm

Tableau 11 : Caractéristiques ADMP401

Ces caractéristiques permettent au ADMP401 de fonctionner efficacement dans diverses applications, y compris la capture de l'audio, la reconnaissance vocale, et les systèmes de surveillance sonore.

b. Capteur de géolocalisation Module GPS NEO-6M

Le module GPS NEO-6M est un récepteur compact utilisant le chipset u-blox NEO-6M, conçu pour fournir des données de localisation précises en temps réel. Il intègre une antenne GPS active et utilise une interface série UART pour communiquer avec des microcontrôleurs, fonctionnant sous une alimentation de 3,3V à 5V. Offrant une précision de localisation de 2,5 à 5 mètres, il peut se connecter à plusieurs satellites pour optimiser la précision et met à jour la position chaque seconde. Ce module est particulièrement utile pour améliorer l'efficacité du suivi et de la gestion des données géographiques, contribuant à des interventions plus précises dans la surveillance forestière.[37]



Figure 22 : Capteur de géolocalisation GPS NEO-6M

Les caractéristiques principales du GPS NEO-6M sont les suivantes [30]

Caractéristiques	GPS NEO-6M
Chipset	U-blox NEO-6M
Précision de localisation	2,5 à 5 mètres
Fréquence de mise à jour	1 Hz (configurable)
Interface de communication	Série UART (TX/RX)
Alimentation	3,3V à 5V
Dimensions	Compact et facile à intégrer
Temps de démarrage à chaud	Quelques secondes
Temps de démarrage à froid	Quelques minutes

Tableau 12 : Caractéristiques GPS NEO-6M

Ces caractéristiques font du NEO-6M un choix populaire pour des applications de géolocalisation dans divers projets électroniques et systèmes de suivi.

2. Microcontrôleur

Les microcontrôleurs intègrent processeur, mémoire et périphériques sur une seule puce, et sont essentiels pour les dispositifs IoT. Après avoir évalué la performance, la connectivité, la consommation d'énergie et le support, nous avons choisi l'ESP8266 et l'Arduino Mega 2560 pour répondre aux besoins de notre projet.

a. Microcontrôleur ESP8266

Le microcontrôleur ESP8266 est une puce de la société Espressif Systems qui est principalement utilisée pour ajouter des capacités de connexion Wi-Fi à des projets de l'Internet

des objets (IoT). Compact et peu coûteux, il intègre un processeur, une mémoire RAM et une interface Wi-Fi, ce qui en fait une solution tout-en-un pour des applications nécessitant une connexion Internet [38].

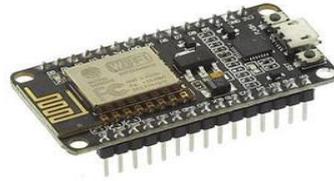


Figure 23 : Microcontrôleur ESP8266

Principales caractéristiques du ESP8266

Caractéristiques	ESP8266
Processeur	32 bits
Fréquence	80 MHz
Mémoire	96 Ko de RAM et 4 Mo de mémoire flash
Connectivité	Wi-Fi 802.11 b/g/n
Interfaces	UART, SPI et I2C
Tension d'Alimentation	3.0 V à 3.3 V
Consommation d'Énergie	Basse consommation
Programmable	Arduino IDE et PlatformIoT

Tableau 13 : Caracteristiques ESP8266

Ces caractéristiques en font un choix idéal pour notre projet, car elles permettent une collecte et un traitement efficaces des données. Les données collectées peuvent ensuite être transmises via wifi ou Lora pour une analyse en temps réel, renforçant ainsi la fiabilité et l'efficacité de notre système de surveillance forestière.

b. Microcontrôleur Arduino Mega 2560

Le microcontrôleur **Arduino Mega 2560** est une carte de développement très populaire pour les projets nécessitant plus de ports et de mémoire que les modèles Arduino de base, comme

l'Arduino Uno. L'Arduino Méga 2560 est un choix judicieux pour notre projet de surveillance forestière en raison de ses nombreuses broches et de sa grande capacité mémoire [39].



Figure 24 : Arduino Méga 2560

Caractéristiques	Arduino Mega 2560
Tension d'entrée (recommandée)	7-12V
Tension d'entrée (limite)	6-20V
Entrées numériques/sorties numériques	54 (dont 15 fournissent la production de PWM)
Mémoire flash	Wi-Fi 802.11 b/g/n
Interfaces	256 KB, dont 8 KB utilisés par le chargeur de chaussures
SRAM	8 ko
EEPROM	4 KB
Programmable	Arduino IDE, PlatformIoT et Atmel Studio

Tableau 14 : Caractéristiques Arduino Mega 2560

Dans notre projet, cette carte joue un rôle central en permettant une connexion simultanée avec nos capteurs. Grâce à ses capacités de gestion de plusieurs dispositifs, l'Arduino Mega 2560 facilite la collecte de données à partir de divers capteurs qui détectent les activités illégales, puis transmet ces informations via Lora pour une analyse approfondie dans le cloud Thingspeak.

3. Modules de Communication

Dans notre architecture, les modes de transmission jouent un rôle crucial pour assurer la connectivité des données capturées par les capteurs vers le serveur cloud en fonction de la disponibilité des réseaux. Voici un aperçu détaillé de chacun de ces modes :

a. Transmission WiFi

Le mode de transmission WiFi utilise le microcontrôleur **ESP8266**, qui est équipé de capacités WiFi, pour transmettre les données vers le cloud lorsque le réseau WiFi est disponible.

- **Fonctionnement** : Lorsque le réseau WiFi est accessible, l'ESP8266 collecte les données des capteurs (acoustiques, géolocalisation, etc.) et les envoie au cloud, spécifiquement à **ThingSpeak**, via une requête **HTTP POST**. ThingSpeak est une plateforme IoT qui permet l'agrégation, l'analyse et la visualisation des données en temps réel.
- **Avantages** :

Vitesse de transmission élevée : Le WiFi permet une transmission rapide des données vers le cloud, rendant ce mode idéal pour les environnements où une connexion rapide est nécessaire.

Facilité d'intégration : Avec le WiFi, les données peuvent être transmises directement au cloud sans passer par des dispositifs intermédiaires, ce qui simplifie l'architecture.

- **Limitations** :

Dépendance au réseau WiFi : Ce mode nécessite une couverture WiFi, ce qui peut être un problème dans des zones reculées où cette connectivité est limitée ou inexistante.

Consommation d'énergie : Le WiFi peut consommer plus d'énergie, ce qui pourrait réduire la durée de vie de la batterie si le système est alimenté par des sources d'énergie limitées.

b. Transmission LoRa

La transmission LoRa (Long Range) est utilisée pour transmettre les données sur de longues distances lorsque le WiFi n'est pas disponible. Cette technologie utilise le **Dragino Lora Shield** et la passerelle **Dragino LG01-N** pour relayer les informations vers le serveur cloud.

Dragino LoRa Shield

Le Dragino LoRa Shield est un module qui permet d'ajouter des capacités de communication LoRa à des cartes de développement comme l'Arduino. [40] Il est conçu pour être facile à utiliser et à intégrer dans des projets IoT. Voici quelques caractéristiques clés :

- **Fonctionnalité** : Permet la communication sans fil sur de longues distances grâce à la technologie LoRa.

- Interface : Il se connecte à un Arduino via des broches GPIO, ce qui le rend compatible avec divers projets de développement.
- Fréquences : Disponible dans plusieurs bandes de fréquences, notamment 433 MHz, 868 MHz et 915 MHz, selon la région.
- Portée : Offre une portée de communication pouvant atteindre plusieurs kilomètres, en fonction de l'environnement et de l'antenne utilisée.
- Consommation d'énergie : Conçu pour une faible consommation d'énergie, idéal pour les applications sur batterie.

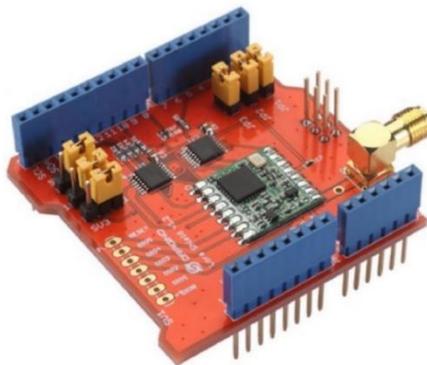


Figure 25 : Dragino Lora Shield

Dragino LG01-N

Le Dragino LG01-N est une passerelle LoRaWAN qui permet de relayer les données entre les capteurs LoRa et un serveur cloud. Voici ses caractéristiques principales :

- Fonctionnalité : Sert de passerelle pour connecter les dispositifs LoRa à Internet via le réseau LoRaWAN.
- Connectivité : Inclut des ports Ethernet et Wi-Fi, permettant une connexion flexible à Internet.
- Compatibilité : Supporte les protocoles LoRaWAN, ce qui facilite l'intégration avec divers capteurs et dispositifs IoT.
- Gestion des données : Peut collecter des données de plusieurs capteurs LoRa et les transmettre à un serveur distant pour analyse.

- Alimentation : Alimentée par une source externe ou par un adaptateur secteur, ce qui le rend adapté pour une utilisation en extérieur.
- Intégration dans notre projet

En combinant ces deux composants, nous avons créé un système robuste pour la surveillance et la détection des activités illégales dans les forêts. La passerelle **Dragino LG01-N** [41] reçoit les données LoRa et les transfère ensuite vers le serveur cloud (ThingSpeak) en utilisant le protocole **MQTT** qui est un protocole de communication léger, optimisé pour le transfert de données dans les réseaux où la bande passante est limitée et où la fiabilité et la rapidité sont cruciales. [42] Créé par IBM dans les années 1990 et aujourd'hui largement utilisé dans l'Internet des objets (IoT), il permet aux appareils connectés (capteurs, actionneurs, etc.) de communiquer entre eux ou avec des serveurs centralisés via un modèle de messagerie basé sur le principe publish/subscribe. Ou bien **HTTP** (Hypertext Transfer Protocol) est un protocole de communication utilisé principalement pour transférer des données sur le Web. Créé en 1989 par Tim Berners-Lee et l'équipe du CERN, HTTP permet aux navigateurs web et aux serveurs de communiquer pour l'échange de documents hypertextes (pages web), ce qui est la base de la navigation sur internet.[43] HTTP est un protocole **stateless** (sans état), ce qui signifie que chaque requête du client vers le serveur est traitée indépendamment des précédentes. Ce processus permet aux données d'atteindre le cloud même lorsque le WiFi n'est pas disponible sur le site de collecte.



Figure 26 : Passerelle Dragino LG01-N

Comparaison des deux modes

Critère	Transmission WiFi	Transmission LoRa
Distance de couverture	Courte portée (nécessite une couverture WiFi)	Longue portée (plusieurs kilomètres)
Consommation d'énergie	Moyenne à élevée	Faible
Vitesse de transmission	Rapide	Faible
Capacité de données	Haute (peut transmettre des données volumineuses)	Faible (adapté aux petits paquets de données)
Utilisation	Zones avec WiFi	Zones sans WiFi ou éloignées

Tableau 15 : Tableau de comparaison des modes de transmission

En conclusion, l'architecture utilise le WiFi pour les zones où la connectivité est accessible et favorise LoRa pour les zones éloignées et dépourvues de réseau WiFi. Ce système hybride optimise ainsi la couverture et la fiabilité du transfert de données dans divers environnements.

Conclusion

La conception de notre système de surveillance des activités illégales en forêt repose sur une architecture robuste et intégrée, combinant les technologies de l'Internet des Objets (IoT) et des communications via wifi et Lora. En mettant l'accent sur l'utilisation de capteurs intelligents, d'un module de traitement efficace, et d'une infrastructure de communication fiable, nous avons créé un système capable de détecter, analyser, et signaler en temps réel toute activité suspecte ou illégale.

Chapitre 2 : Développement et Mise en Œuvre de notre Application SenForêt

Introduction

Ce chapitre se concentre sur la mise en œuvre de notre projet, en mettant en avant les outils et technologies employés pour la conception du projet de la base de données et le développement des applications. Cette phase est essentielle car elle permet de transformer les concepts et spécifications en une solution opérationnelle et efficiente.

I. Outils de développement utilisés

Dans cette section nous allons détailler les logiciels utilisés dans notre projet, en expliquant leur rôle et leur importance dans chaque étape de la réalisation.

1. Environnements de développement intégré (IDE)

a. Astah

Astah est un logiciel de modélisation UML (Unified Modeling Language) utilisé pour créer des diagrammes qui visualisent et conçoivent les structures et comportements d'un système. Il est essentiel pour documenter et planifier des projets logiciels, facilitant la communication entre développeurs, architectes et parties prenantes. [44] Dans notre projet, nous avons utilisé Astah pour créer des diagrammes de cas d'utilisation, de classes et de séquence. Astah nous aide ainsi à planifier, organiser et communiquer les aspects clés de notre système de manière visuelle et structurée [45].



Figure 27 : Astah

b. Pycharm

PyCharm est un environnement de développement intégré (IDE) spécialement conçu pour le développement en Python. [46] Développé par JetBrains, PyCharm offre de nombreuses fonctionnalités qui facilitent la programmation, notamment :

- **Éditeur de Code Intelligent** : PyCharm propose une auto-complétion de code, la mise en évidence de la syntaxe, des refactorisations de code, et bien plus encore.
- **Débogage** : Il permet de déboguer notre code en posant des points d'arrêt, en examinant les variables et en suivant l'exécution du programme étape par étape.
- **Intégration avec les Outils de Développement** : PyCharm s'intègre facilement avec des outils de versionnage (comme Git), des gestionnaires de paquets (comme pip), des environnements virtuels, et des frameworks comme Django et Flask.
- **Support de Base de Données** : Il fournit des outils pour travailler avec les bases de données, y compris la possibilité d'interroger et de gérer les bases de données directement depuis l'IDE.
- **Développement Web** : PyCharm inclut des fonctionnalités spécifiques pour le développement de projets web, comme le support des langages front-end (HTML, CSS, JavaScript) et l'intégration avec les frameworks web Python.
- **Tests Unités** : Il permet d'écrire, d'exécuter et de déboguer des tests unitaires facilement.

En somme, PyCharm est un atout majeur dans notre projet, facilitant non seulement le développement, mais aussi la gestion et la validation de notre solution complète de surveillance forestière [47].

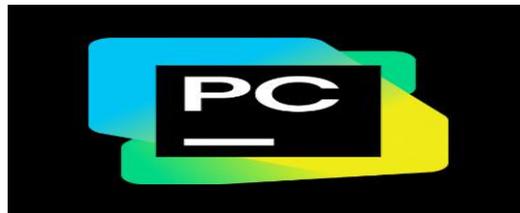


Figure 28 : Pycham

2. Plateformes et outils de développement mobile et web

a. Django

Django est un Framework web avancé pour Python, conçu pour accélérer le développement d'applications web robustes, sécurisées et évolutives. Il adopte le modèle MVC (Modèle-Vue-Contrôleur), ou plus précisément son équivalent en Django, le modèle MTV (Modèle-Template-View), qui permet de séparer la logique de l'application, la présentation et les données [48] .

Caractéristiques principales de Django :

- **ORM intégré** : Django fournit un Object-Relational Mapping (ORM) qui permet de manipuler les bases de données de manière simple et intuitive à l'aide de modèles Python.
- **Sécurité** : Django inclut par défaut de nombreuses protections contre les menaces courantes, telles que les attaques par injection SQL, les scripts intersites (XSS), et les attaques CSRF (Cross-Site Request Forgery).
- **Administration automatique** : Django génère automatiquement une interface d'administration pour gérer les données de l'application, ce qui permet de gagner du temps lors du développement.
- **Extensibilité** : Django dispose d'un écosystème riche en bibliothèques et modules supplémentaires, facilitant l'ajout de nouvelles fonctionnalités à l'application.
- **Communauté et documentation** : Django bénéficie d'une large communauté de développeurs et d'une documentation complète, rendant l'apprentissage et l'utilisation du Framework plus accessibles.

Dans notre projet, nous allons utiliser Django pour la création de notre application web. Django est particulièrement adapté pour les projets nécessitant une structure solide et une gestion efficace des données, comme le nôtre. Grâce à son ORM, nous pourrions gérer facilement les bases de données et générer les alertes en temps réel. De plus, son interface d'administration nous permettra de gérer les utilisateurs et l'historique des alertes efficacement [49] .



Figure 29 : Django

b. Plateforme IoT ThingSpeak

[50] ThingSpeak est une plateforme IoT (Internet of Things) qui permet de collecter, stocker, analyser et visualiser des données en temps réel à partir de dispositifs connectés. Elle est particulièrement utilisée pour les projets de capteurs, l'automatisation, et les applications de surveillance.

Fonctionnalités Clés de ThingSpeak :

- **Collecte de données** : ThingSpeak peut recevoir des données de capteurs ou de dispositifs connectés via HTTP ou MQTT. Ces données peuvent inclure des mesures de température, d'humidité, de pression, de géolocalisation, etc.
- **Stocks de données** : La plateforme stocke les données collectées dans des canaux (channels), où chaque canal est associé à une application ou un projet spécifique. Les données peuvent être consultées et analysées ultérieurement.
- **Analyse et visualisation** : ThingSpeak propose des outils pour analyser les données en temps réel ou après leur collecte. On peut créer des graphiques, des tableaux de bord, et appliquer des algorithmes d'analyse pour interpréter les données. La plateforme prend également en charge l'intégration avec MATLAB pour des analyses plus complexes.
- **Alertes et Notifications** : ThingSpeak peut générer des alertes ou des notifications basées sur des conditions définies par l'utilisateur. Par exemple, vous pouvez configurer une alerte pour être notifié si une certaine valeur de capteur dépasse un seuil prédéfini.
- **API RESTful** : ThingSpeak fournit une API RESTful, permettant aux développeurs d'intégrer facilement la plateforme avec d'autres systèmes, applications ou services cloud.
- **Support Multi-Plateforme** : La plateforme est accessible via le web et peut être utilisée avec divers dispositifs IoT, y compris ceux basés sur Arduino, Raspberry Pi, et ESP8266.

Importance de ThingSpeak dans notre projet

Pour notre projet, ThingSpeak joue un rôle crucial en tant que plateforme de backend pour gérer les données de nos capteurs IoT. Voici comment il est important :

- **Collecte et Stockage des Données** : ThingSpeak collecte les données de nos capteurs, comme pour les utilisés surveiller l'exploitation forestière illégale, et les stocke de mise au point.
- **Analyse et Génération d'Alertes** : Il permet d'analyser les données en temps réel et de générer des alertes lorsqu'une activité suspecte ou illégale est détectée.
- **Intégration avec nos applications** : On peut intégrer ThingSpeak avec notre application Django pour visualiser les alertes et données en temps réel sur notre interface mobile ou web.

- **Automatisation** : ThingSpeak facilite l'automatisation des tâches de surveillance et d'alerte, ce qui est essentiel pour notre système de surveillance en forêt.

En somme, ThingSpeak est la plateforme qui gère l'ensemble du flux de données de notre système IoT, du capteur à l'interface utilisateur. Parmi les différentes plateformes IoT disponibles, nous avons choisi d'utiliser ThingSpeak pour notre projet. Cette plateforme permet de collecter, stocker, et analyser les données en temps réel provenant de nos capteurs. ThingSpeak est également responsable de la génération des alertes, lesquelles sont transmises à nos applications web et mobile. Son intégration pour des analyses avancées et sa capacité à gérer les données de manière centralisée en font une solution adaptée pour optimiser la surveillance et la gestion des activités illégales de la coupe du bois dans notre système [51].



Figure 30 : ThingSpeak

c. Twilio

Twilio est une plateforme cloud qui permet aux développeurs d'intégrer des fonctionnalités de communication, comme l'envoi de SMS, de courriels, d'appels vocaux, ou encore de vidéos, directement dans leurs applications via des API (interfaces de programmation). Twilio simplifie l'ajout de services de communication en temps réel, ce qui est utile dans des applications de notification, d'authentification à deux facteurs (2FA), de support client, et bien plus encore [52].

Principales fonctionnalités de Twilio :

- **SMS** : Envoi et réception de messages texte.
- **Vocal** : Gestion des appels téléphoniques (entrants et sortants).
- **Email** : Envoi et suivi des courriels via **Twilio SendGrid**.
- **Vidéo** : Intégration de vidéo en direct dans les applications (appels vidéo, visioconférence).
- **Authy** : Authentification à deux facteurs et validation des utilisateurs.

Twilio est utilisé dans des applications nécessitant des communications automatisées, comme les alertes de sécurité, les notifications transactionnelles ou les services clients [53] .



Figure 31 : TWILIO

1. Outils pour l'électronique et l'Internet des objets (IoT)

a. Fritzing

Fritzing est un outil open-source idéal pour notre projet, permettant de créer des schémas de circuits électroniques en représentant les connexions entre composants comme les microcontrôleurs, capteurs et modules de transmission. [54] Il facilite le prototypage sur breadboard en simulant la disposition physique des composants, et offre des outils pour concevoir des circuits imprimés (PCB), organiser les composants, tracer les pistes et générer les fichiers Gerber pour la fabrication. Fritzing permet également de documenter les configurations matérielles de manière claire et visuelle, facilitant la communication au sein de l'équipe et la documentation du projet. Enfin, il permet d'exporter les fichiers nécessaires pour la fabrication des PCB ou de partager les schémas et diagrammes avec d'autres [55].



Figure 32: Fritzing

b. IDE Arduino

L'IDE Arduino est un logiciel open source et gratuit, essentiel pour les développeurs et amateurs travaillant avec des cartes Arduino et des microcontrôleurs comme l'ESP8266 et Arduino Mega 2560. Il offre un éditeur de code avec coloration syntaxique et autocomplétions, un compilateur intégré, et la possibilité de téléverser des programmes via USB.[56] Il permet également d'intégrer facilement des bibliothèques pour ajouter des fonctionnalités spécifiques, et dispose d'un moniteur série pour la communication en temps réel. Facile à utiliser, même

pour les débutants, l'IDE Arduino est compatible avec Windows, macOS, et Linux, et est soutenu par une large communauté, ce qui en fait un outil indispensable pour écrire, tester, et déboguer du code sur les microcontrôleurs [57].

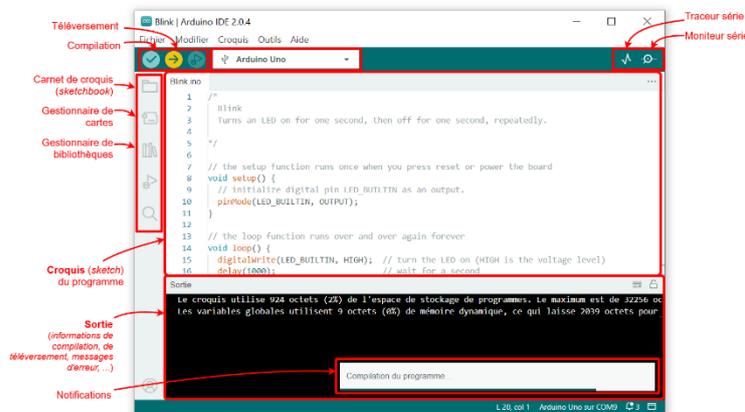


Figure 33 : Logiciel IDE Arduino

2. Bases de données et Langages de programmation

a. Python

Python est un langage de programmation interprété, de haut niveau, et polyvalent, connu pour sa syntaxe claire et lisible.[58] Il est largement utilisé dans divers domaines tels que le développement web, l'analyse de données, l'intelligence artificielle, l'automatisation, et bien d'autres. Python est apprécié pour sa facilité d'apprentissage, son vaste écosystème de bibliothèques et Framework, ainsi que pour sa communauté active [59].

Quelques caractéristiques de Python :

- **Syntaxe simple et lisible** : Python utilise une syntaxe claire qui permet de coder plus rapidement et de maintenir plus facilement le code ;
- **Langage interprété** : Python est exécuté par un interpréteur, ce qui signifie qu'il n'a pas besoin d'être compilé avant d'être exécuté ;
- **Polyvalent** : Python peut être utilisé pour le développement web (avec des Framework comme Django et Flask), l'analyse de données (avec des bibliothèques comme Pandas et NumPy), le machine Learning (avec TensorFlow, PyTorch), et bien plus encore ;
- **Extensible** : Python peut être étendu avec des modules écrits dans d'autres langages comme C ou C++ ;
- **Large communauté** : Il existe une vaste communauté d'utilisateurs et de développeurs Python, ce qui facilite l'accès à des ressources d'apprentissage, des bibliothèques et des outils.

En somme, Python constitue un choix judicieux pour notre projet de gestion d'alertes forestières grâce à sa simplicité, son vaste écosystème, et sa compatibilité avec les technologies IoT et Django, ce qui facilite la création d'un système performant et facile à maintenir grâce à sa vaste bibliothèque standard et à ses nombreux modules tiers [60].



Figure 34 : Python

b. SQLite

SQLite est un système de gestion de base de données relationnelle léger, autonome, et intégré dans de nombreuses applications.[61] Contrairement à d'autres bases de données comme MySQL ou PostgreSQL, SQLite n'a pas besoin d'un serveur séparé pour fonctionner, ce qui en fait une solution idéale pour des applications légères, des tests, ou des environnements avec des ressources limitées. Voici un aperçu détaillé de **SQLite**, ses caractéristiques, et son utilisation avec **Django**.

Caractéristiques de SQLite :

- **Autonome** : SQLite est livré sous forme de bibliothèque C, intégrée directement dans l'application. Il n'y a pas besoin d'installer ou de configurer un serveur de base de données séparé ;
- **Fichier unique** : Toutes les données sont stockées dans un seul fichier (fichier avec extension .sqlite ou .db), ce qui rend la base de données facilement transportable ;
- **Léger** : Il consomme peu de mémoire et est très rapide pour des projets de petite à moyenne envergure ;
- **ACID-compliant** : SQLite assure les propriétés ACID (Atomicité, Cohérence, Isolation, Durabilité), garantissant ainsi l'intégrité des transactions ;
- **Multiplateforme** : Il fonctionne sur presque tous les systèmes d'exploitation, dont Linux, Windows, macOS, et même des environnements embarqués.

Dans notre projet, Django adopte par défaut SQLite comme base de données pour les projets en développement. Cela s'explique par la simplicité de SQLite et son absence de configuration complexe. SQLite est une solution simple et légère pour gérer des bases de données, particulièrement utile dans les phases de développement et pour des applications de petite à moyenne envergure [62].



Figure 35 : SQLite

II. Développement du Système

1. Codage Arduino

a. Introduction au matériel utilisé (ESP8266, capteurs)

Dans le cadre de notre projet de surveillance forestière, la détection d'activités illégales, telles que la coupe de bois, repose sur l'utilisation de plusieurs composants matériels essentiels. La sélection de ces équipements a été effectuée en fonction de leur capacité à fonctionner dans des environnements isolés avec une connectivité limitée, tout en garantissant une transmission fiable des données. Chaque élément joue un rôle clé dans la collecte d'informations en temps réel, la transmission à longue distance et la visualisation des alertes.

Le **capteur acoustique ADMP401** est conçu pour détecter des sons spécifiques, comme ceux émis par les tronçonneuses. Le **module GPS NEO-6M** permet de localiser avec précision les événements suspects. L'**ESP8266**, un module Wi-Fi, offre des capacités de communication sans fil, facilitant l'envoi des données collectées vers le cloud pour une analyse ultérieure. L'**Arduino Mega** gère le traitement et l'intégration des données provenant des capteurs avant leur transmission.

Le **Dragino LoRa Shield** est intégré pour permettre une communication sans fil à longue distance via la technologie LoRa, essentielle dans les zones forestières dépourvues d'infrastructures réseau traditionnelles. Enfin, la **passerelle LG01** sert de relais pour transmettre les données des capteurs vers un serveur cloud, assurant une connectivité stable et efficace.

L'intégration de ces composants permet de créer un système de surveillance robuste et efficace, garantissant une couverture et un suivi en temps réel dans des régions éloignées.

b. Montage du matériel avec Fritzing

Dans cette section, nous allons détailler l'installation de chaque capteur utilisé dans notre projet de surveillance forestière. Nous expliquons les connexions et le câblage de chaque composant avant de réaliser l'assemblage global du matériel avec l'outil Fritzing. Ce schéma visuel nous permet d'illustrer clairement l'intégration des capteurs et du microcontrôleur ESP8266 ou Arduino mega.

Montage Arduino Méga et le capteur acoustique ADMP401

Microcontrôleur ADMP401

5V	VCC
GND	GND
A0	OUT

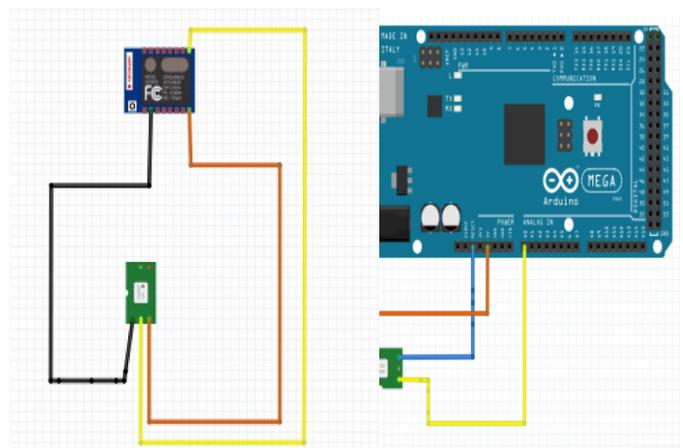


Tableau 16: Tableau de Connexion Microcontrôleur - ADMP401

Figure 36 : Connexion Microcontrôleur - ADMP401

Montage Arduino Méga et le capteur GPS (NEO-6M)

Microcontrôleur GPS (NEO-6M)

5V	VCC
GND	GND
RX1	TX1
TX1	RX1

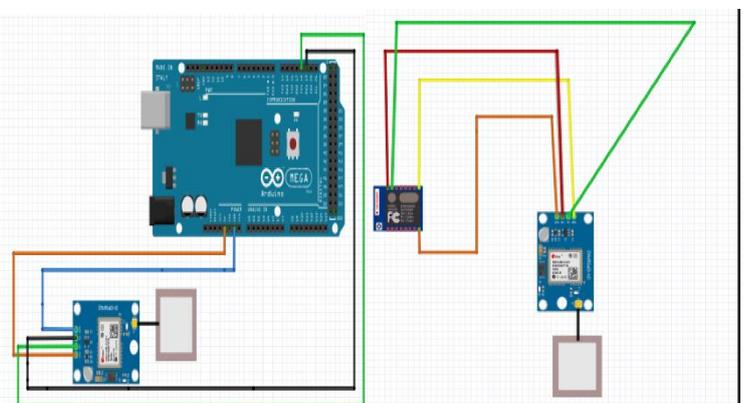


Tableau 17 : Connexion Microcontrôleur - GPS (NEO-6M)

Figure 37: Connexion Microcontrôleur - GPS (NEO-6M)

✚ Montage finale ESP8266 ADMP401 et GPS 6 NEO

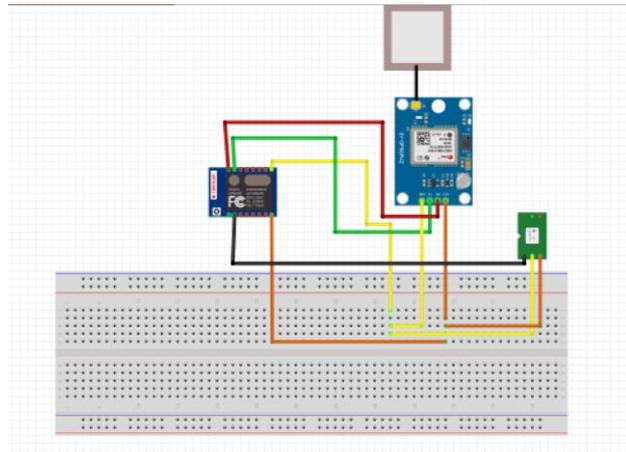


Figure 38 : Connexion Microcontrôleur - GPS (NEO-6M) - ADMP401

c. Gestion des seuils et envoi des données

Installation et Configuration de l'IDE Arduino :

Pour configurer notre environnement de développement Arduino, suivons ces étapes :

- **Lancer l'IDE Arduino** : Ouvrez le logiciel Arduino IDE sur notre ordinateur ;
- **Sélection de la carte** : Dans le menu **Outils**, allons à **Type de carte** et choisissons le modèle de carte que nous utilisons, comme ESP8266 ou Arduino Mega ;
- **Choisir le port série** : Toujours dans le menu **Outils**, allons à **Port** et sélectionnons le port série auquel notre carte Arduino est connectée.

Ces étapes permettent de configurer notre Arduino IDE pour qu'il puisse communiquer correctement avec notre carte Arduino, facilitant ainsi le développement et le téléchargement de nos programmes.

Avant d'envoyer les données vers **ThingSpeak**, il était essentiel de définir une **valeur seuil** permettant d'identifier les sons caractéristiques d'une scie ou d'une tronçonneuse. Cette étape a été réalisée de manière méthodique pour garantir une détection fiable :

Le capteur **ADMP401** a été choisi pour mesurer les niveaux sonores. Ce capteur peut enregistrer des valeurs allant jusqu'à **1055**, représentant son amplitude maximale.

Collecte des données en conditions réelles :

Le capteur **ADMP401** a été choisi pour mesurer les niveaux sonores. Ce capteur peut enregistrer des valeurs allant jusqu'à **1055**, représentant son amplitude maximale.

Collecte des données en conditions réelles :

- **Le matin** : Les données sonores ont été collectées lorsque l'environnement est généralement plus calme.
- **Le soir** : Les mesures ont été reprises dans un contexte où les bruits ambiants sont plus présents, en raison de l'activité humaine accrue.

Observation des variations sonores :

Les tests ont révélé que :

- **Le matin**, les valeurs captées sont relativement faibles.
- **Le soir**, les valeurs sont significativement plus élevées, principalement à cause du bruit ambiant.

Fixation de la valeur seuil :

Afin de tenir compte de ces différences tout en minimisant les erreurs de détection (faux positifs et faux négatifs), une valeur seuil de **500** a été fixée. Cette valeur correspond à une estimation moyenne permettant de détecter les sons spécifiques des scies ou tronçonneuses tout en filtrant les bruits ambiants.

Justification du seuil de 500 :

- Il s'agit d'une moyenne calculée à partir des tests réalisés dans des environnements différents (matin et soir).
- Ce seuil garantit que seuls les sons suffisamment puissants et caractéristiques sont pris en compte pour générer une alerte.

Période	Valeur du son	Seuil atteint
Matin	44	Non
Matin	365	Non
Matin	500	Oui
Soir	56	Non
Soir	877	Oui
Soir	500	Oui

Tableau 18 : Analyse des sons par période et validation du seuil

1. Code avec ESP8266 (Annexe 1)

2. Code avec LoRa (Annexe 2)

2. Développement de l'Application Web

Dans cette section, nous détaillons les étapes du développement de l'application : présentation de l'environnement de développement (Django, Python, HTML, CSS, JavaScript), conception des modèles de données avec Django, développement des API REST via Django Rest Framework, intégration avec ThingSpeak pour les alertes, et description de l'interface utilisateur, notamment le tableau de bord pour la gestion des alertes et des utilisateurs.

a. Environnement de développement (Django, Python, HTML, CSS, JavaScript)

Django et python : Dans ce projet, nous avons utilisé **Django**, un Framework web écrit en **Python**, pour le développement de l'application web. Django permet une gestion simplifiée des bases de données, une architecture MVC (Modèle-Vue-Contrôleur) claire, et une sécurité renforcée avec des mécanismes intégrés comme la protection contre les attaques CSRF (Cross-Site Request Forgery). Le langage **Python**, étant au cœur de Django, est utilisé pour la logique métier, le traitement des données, et l'interaction avec la base de données via l'ORM (Object-Relational Mapping) de Django.

HTML (HyperText Markup Language) : est utilisé pour structurer les pages web de l'application. Il permet de définir les éléments de base tels que les titres, les paragraphes, les tableaux, et les formulaires que les utilisateurs interagissent dans le navigateur.

CSS (Cascading Style Sheets) : est utilisé pour la mise en forme des éléments HTML. Il permet de personnaliser l'apparence visuelle de l'application en définissant des styles comme les couleurs, les polices, les marges, et la disposition des éléments sur la page.

JavaScript : est utilisé pour ajouter des fonctionnalités interactives et dynamiques à l'application. Cela inclut des comportements comme la validation de formulaires en temps réel, la mise à jour de contenu sans rechargement de la page grâce à l'AJAX, ou encore des animations et effets visuels pour améliorer l'expérience utilisateur.

En résumé, Django et Python sont utilisés pour la partie back-end (serveur et logique), tandis que HTML, CSS et JavaScript sont employés pour le front-end, offrant une interface utilisateur attrayante et réactive.

b. Modèles de données (modèles Django, bases de données)

Les modèles de données dans Django définissent la structure de la base de données et permettent une interaction facile avec celle-ci via l'ORM de Django.

- ✚ **Modèle Alerte** : Ce modèle représente les informations liées aux alertes générées par le système, comme le type d'alerte, la description, la date de création, et le statut. Il est crucial d'enregistrer et de suivre les événements détectés.

```
# Create your models here.
from django.db import models

# Create your models here.
from django.db import models
class Alerte(models.Model):
    type = models.CharField(max_length=50, primary_key=True)
    description = models.TextField(max_length=250)
    date_creation = models.DateTimeField(auto_now_add=True)
    statut = models.CharField(max_length=50)

    utilisateur = models.ManyToManyField(Utilisateur, related_name='alerte')

    def __str__(self):
        return f" {self.type} "
```

Figure 39 : Modèle de Données Alerte

Modèle Utilisateur : Ce modèle gère les utilisateurs de l'application, stockant des informations telles que leur nom, email, rôle, et les alertes qu'ils sont habilités à consulter. Il assure la gestion des droits et de l'accès aux alertes.

- ✚ **Configuration de la Base de Données SQLite** : Dans le fichier de configuration `settings.py` de notre projet Django, la configuration de la base de données SQLite est définie comme suit :

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

Figure 40 : Configuration de la Base de Données SQLite

Nous avons défini plusieurs modèles de données essentiels, dont les **modèles Alerte** et **Utilisateur**, qui sont stockés et gérés dans la base de données **SQLite**. Ces modèles sont directement connectés à la base de données **SQLite** via l'ORM de Django, ce qui permet de manipuler et interagir avec les données sans écrire de requêtes SQL complexes.

Dans cette section, nous avons détaillé aussi la mise en place des API REST avec Django Rest Framework (DRF) dans notre projet. DRF est une bibliothèque puissante et flexible qui simplifie le développement d'API RESTful. Elle facilite la création de points d'accès (endpoints), permettant aux clients (applications mobiles, web, etc.) d'interagir de manière structurée avec le serveur via HTTP.

c. Intégration avec ThingSpeak et Twilio pour la gestion des alertes

Dans cette partie, nous avons intégré **ThingSpeak** pour gérer les alertes liées à la détection d'activités suspectes dans notre projet. ThingSpeak, une plateforme IoT, reçoit les données des capteurs via des requêtes **HTTP POST** et les analyse en temps réel. Les alertes, incluant des informations comme la géolocalisation et le statut, sont ensuite récupérées via des **API GET** et affichées dans l'application Django. Cette intégration permet une gestion centralisée et en temps réel des alertes, facilitant ainsi la surveillance et la réactivité face aux incidents.

```
class ThingSpeakDataView(APIView): 2 usages
    def get(self, request, *args, **kwargs):
        # URL de l'API externe
        url = "https://api.thingspeak.com/apps/thinghttp/send-request"

        # Paramètres de la requête
        params = {
            'api-key': 'PR2NQ6D1U1B8RNB',
        }

        # Effectuer la requête GET
        response = requests.get(url, params=params)

        # Vérifier si la requête a réussi
        if response.status_code == 200:
            data = response.json()

            # Créer ou mettre à jour une alerte dans la base de données
            alerte_data = {
                'type': data.get('type', 'N/A'),
                'description': data.get('description', 'No description'),
                'statut': data.get('statut', 'N/A')
            }
```

Figure 41 : Intégration avec ThingSpeak pour la Gestion des Alertes

Configuration de Twilio pour envoyer des SMS et notifications WhatsApp

Pour envoyer des alertes via Twilio, on commence par créer un compte et récupérer nos identifiants API (Account SID et Auth Token). On achète ensuite un numéro de téléphone

Twilio pour l'envoi des SMS puis on ajoute des numéros d'utilisateurs pour la réception d'alerte pour afin qu'il soit déclenché automatiquement dès qu'une activité suspecte est détectée.

d. Intégration avec Django pour le déclenchement d'alertes en temps réel

Cette section traite l'intégration de Twilio avec **Django** pour déclencher l'envoi d'alertes automatiquement lorsque le système détecte une activité suspecte, Django envoie immédiatement les informations à Twilio, qui se charge de transmettre les alertes aux destinataires via SMS ou WhatsApp avec l'url vers les coordonnées GPS du lieu où se passe l'activité suspecte.

✚ Configuration de Twilio

Nous configurons nos identifiants Twilio dans notre projet Django de la manière suivante :

```
TWILIO_ACCOUNT_SID = 'notre_account_sid'
TWILIO_AUTH_TOKEN = 'notre_auth_token'
TWILIO_PHONE_NUMBER = '+notre_numero_twilio'
```

Figure 42 : Configuration de twilio

La fonction `send_alert_sms` permet d'envoyer des messages SMS aux utilisateurs lorsqu'une alerte est déclenchée :

```
from twilio.rest import Client
from django.conf import settings

def send_alert_sms(phone_numbers, message_body):
    client = Client(settings.TWILIO_ACCOUNT_SID, settings.TWILIO_AUTH_TOKEN)
    for phone_number in phone_numbers:
        try:
            message = client.messages.create(
                body=message_body,
                from_=settings.TWILIO_PHONE_NUMBER, # Remplacez par votre numéro Twilio
                to=phone_number
            )
            print(f"Message envoyé à {phone_number}: {message.sid}")
        except Exception as e:
            print(f"Erreur lors de l'envoi à {phone_number}: {str(e)}")
```

Figure 43 : Fonction envoyer alerte

L'intégration de l'envoi d'alertes SMS via Twilio est cruciale pour notre système de surveillance, permettant une communication rapide et efficace avec les utilisateurs en les informant en temps réel des activités suspectes. Cette fonctionnalité améliore la réactivité, la sécurité et la gestion des incidents, tout en instaurant un climat de confiance parmi les utilisateurs grâce à une intervention rapide. Ainsi, les alertes SMS constituent un élément clé de notre projet, garantissant un système de surveillance moderne, sécurisé et efficace.

III. Déploiement et Tests de l'Architecture

Dans cette partie, nous présentons le montage de l'architecture matérielle et logicielle ainsi que les résultats obtenus sous forme de courbes de données sur la plateforme ThingSpeak.

1. Réalisation de l'Architecture de Test



Figure 44 : Architecture de Test

2. Résultats : Courbes de Données sur ThingSpeak

Après la collecte et la transmission des données depuis le microcontrôleur, ThingSpeak permet de visualiser les informations sous forme de courbes en temps réel. Cette plateforme analyse et affiche les données envoyées par les capteurs, facilitant ainsi l'observation des tendances et la détection d'activités suspectes ou d'anomalies. Les courbes fournissent une vue graphique des

données mesurées, confirmant que la transmission est réussie et que le système fonctionne comme prévu.

Latitude : Le premier graphique montre les variations de latitude enregistrées au fil du temps, indiquant les positions géographiques collectées par le système.

Longitude : Le deuxième graphique illustre les changements de longitude sur la même période, permettant d'obtenir une vue complète des coordonnées de localisation.

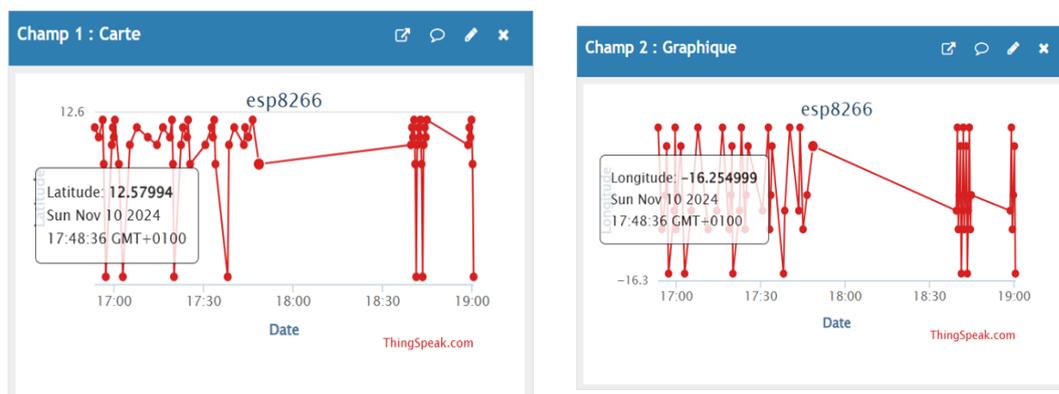


Figure 45 : Les informations de la géolocalisation

Les courbes montrent que le système fonctionne correctement, avec des données de géolocalisation reçues en temps réel. Lorsque le seuil d'activité suspecte est atteint, le microcontrôleur envoie ces données à ThingSpeak, qui les transmet ensuite à notre application pour générer une alerte.

IV. Fonctionnement de notre Application SenForet

Dans cette section, nous allons présenter en détail l'interface web de SenForet, en décrivant ses principales fonctionnalités, son système d'administration, et son architecture intuitive qui facilitent la gestion et le suivi des alertes liées aux activités illégales en forêt.

1. Présentation de l'application Web

L'application web **SenForet** a été développée avec le Framework **Django** et est conçue pour surveiller les forêts et prévenir les activités illégales telles que la coupe de bois non autorisée. L'interface utilisateur permet de gérer efficacement les alertes générées par les capteurs IoT et

de superviser l'ensemble du système à travers un tableau de bord intuitif. En parallèle, l'interface d'administration de Django offre une gestion complète des bases de données de l'application.

🚦 Interface de Connexion

L'application débute par une page de connexion assurant la sécurité des données et permettant aux utilisateurs d'accéder aux diverses fonctionnalités selon leur rôle, qu'ils soient administrateurs ou simples utilisateurs. Cette page comporte des champs où l'utilisateur doit saisir son nom d'utilisateur et son mot de passe. Un bouton de connexion est également présent pour valider les informations d'authentification et accéder au tableau de bord de l'application.



Connexion

Nom d'utilisateur

Mot de passe

Se connecter

Figure 46 : Page de connexion

🚦 Page d'Accueil (Tableau de Bord)

La **page d'accueil** de l'application **SenForet** sert de **tableau de bord** centralisé, offrant aux utilisateurs une vue d'ensemble sur les alertes en temps réel et l'évolution des événements dans les zones forestières surveillées.



Figure 47 : Page d'accueil

L'Historique des Alertes est présentée sous forme de graphiques dynamiques, permettant de visualiser les tendances des alertes sur des périodes comme un jour, une semaine ou un mois. Ces graphiques aident à suivre l'évolution des activités illégales dans les forêts surveillées. Les utilisateurs peuvent observer le nombre d'alertes mensuelles et comparer les périodes d'activité. Différents types de graphes, comme des barres, permettent de comparer les alertes par type d'incident (tronçonneuse, mouvement suspect). Ces visualisations facilitent une analyse rapide et la prise de décisions préventives ou correctives.

Notifications d'Alerte : Cette section affiche en temps réel les alertes générées par les capteurs IoT, telles que l'utilisation illégale de tronçonneuses ou la présence suspecte dans la forêt. Chaque alerte comprend des informations comme la date, l'heure, et la localisation précise de la coupe de bois, permettant aux utilisateurs de prendre des décisions rapidement.



Type	Description	Date de création
35	Alerte d'urgence : détection d'une activité suspecte. Voici le lien vers le lieu de l'action : 12.579940,-16.254999	10-11-2024 16:48
34	Alerte d'urgence : détection d'une activité suspecte. Voici le lien vers le lieu de l'action : 12.58533,-16.27111	10-11-2024 15:48
33	Alerte d'urgence détection d'une activité suspecte.Voici le lien vers le lieu de l'action :12.585330,-16.271111	01-11-2024 16:26
19	Alerte d'urgence détection d'une activité suspecte.Voici le lien vers le lieu de l'action : 48.8566,2.3522	25-10-2024 19:28

Figure 48 : Interface de la liste de notification des alertes

Liste des Utilisateurs : les administrateurs doivent pouvoir visualiser une liste qui affiche pour chaque utilisateur le nom complet, l'adresse email, ainsi que le rôle attribué. Le rôle de l'utilisateur peut être soit "Utilisateur simple", qui a accès à la plateforme et peut recevoir les alertes, soit "Administrateur", qui possède des privilèges supplémentaires, comme la gestion des utilisateurs et des alertes. Ces informations permettent aux administrateurs de gérer les accès et les permissions des utilisateurs au sein du système.

Dashboard Deconnexion

Liste des utilisateurs

Nom d'utilisateur	Email	Rôle
Mamadou Diagne	diagne@gmail.com	user
Maty Seck	m.seck20160634@zig.univ.sn	admin
Amy thiandoum	a.thiandoum20160901@zig.univ.sn	user

Figure 49 : Liste des Utilisateurs

Un bouton de déconnexion est intégré, permettant à l'utilisateur de se déconnecter en un clic. Lorsque ce bouton est cliqué, il se redirige automatiquement vers la page de connexion, garantissant ainsi la sécurité de la session et permettant à l'utilisateur de quitter la plateforme de manière sécurisée.

2. Configuration de Twilio

Twilio est une plateforme de communication cloud qui permet d'intégrer facilement des services de messagerie dans des applications. Dans notre projet, nous avons choisi Twilio pour sa fiabilité et sa capacité à transmettre des notifications critiques par SMS, afin de garantir que les alertes soient reçues immédiatement par les utilisateurs, où qu'ils se trouvent.

Utilisation de Twilio dans notre Système d'Alerte

Lorsque notre application Django détecte une activité suspecte (comme une coupe de bois illégale), une alerte est générée. Cette alerte est ensuite transmise à Twilio via une API, qui se charge d'envoyer un SMS directement au téléphone de l'utilisateur.

Ce système assure une réactivité maximale en notifiant en temps réel les agents ou responsables de la sécurité sur le terrain. Grâce à la flexibilité de Twilio, nous avons pu configurer des messages d'alerte personnalisés, incluant les détails de l'incident pour une prise de décision rapide.

Fonctionnement de l'Intégration :

- **Détection de l'alerte** : Une alerte est générée dans l'application Django lors de la détection d'une activité suspecte ;
- **Transmission à Twilio** : L'alerte est envoyée à Twilio via l'API, incluant un message personnalisé et le numéro de téléphone de l'utilisateur ;
- **Notification SMS** : Twilio délivre l'alerte par SMS directement au téléphone de l'utilisateur en quelques secondes.

Résultat

Les utilisateurs reçoivent une notification immédiate sous forme de SMS, incluant un lien les redirigeant vers le lieu exact de l'action suspecte. Ce système renforce la réactivité face aux incidents, permettant une intervention rapide et contribuant ainsi à une meilleure protection des zones surveillées.

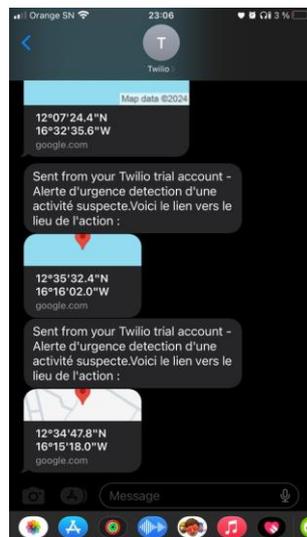


Figure 50 : Alerte Reçu

Ce message reçu est la confirmation que l'alerte a bien été transmise et que le système de notification par SMS fonctionne comme prévu, assurant ainsi un suivi efficace et une sécurité renforcée.

Conclusion

Dans ce chapitre, nous avons présenté le développement de l'application SenForêt, dédiée à la surveillance forestière et à la détection d'activités illégales. Grâce aux capteurs IoT connectés via l'ESP8266 et LoRa, les données sont collectées et transmises efficacement, même dans des zones reculées. La plateforme web, développée avec Django, permet de visualiser et gérer les

alertes en temps réel, tandis que l'intégration de Twilio assure l'envoi rapide des notifications par SMS aux agents. Ce chapitre marque une étape clé dans la réalisation de SenForêt pour la protection des ressources forestières.

Conclusion Générale

Ce projet de recherche a porté sur le développement d'une plateforme de gestion des alertes pour surveiller la coupe illégale de bois au Sénégal, contribuant ainsi à la préservation des ressources forestières. Une étude initiale a permis d'explorer les systèmes de surveillance existants et les types d'alertes possibles, révélant des lacunes en matière de couverture et de transmission en temps réel, notamment dans les zones isolées.

La solution développée repose sur une architecture hybride intégrant le Wifi et la technologie LoRa pour assurer la transmission de données dans des régions reculées et difficiles d'accès. Pour la connectivité Wifi, un ESP8266 utilisant le protocole HTTP a été mis en œuvre pour transmettre les données de manière fiable et directe, tandis qu'un Arduino Mega, associé à un module Dragino LoRa Shield, assure la couverture en LoRa. Cette combinaison répond aux besoins spécifiques des zones éloignées tout en offrant une large portée de surveillance.

Lors de la phase de tests, des configurations précises ont été mises en place pour garantir la transmission des données. Toutefois, l'interprétation des données via le protocole MQTT sur ThingSpeak présente certains défis avec la passerelle Dragino LG01. Bien que la transmission fonctionne, l'affichage des données s'est révélé limité, probablement en raison de contraintes matérielles ou de compatibilité. Dans ce contexte, la transmission via Wifi et le protocole HTTP a été privilégiée pour assurer la stabilité de la collecte et de l'affichage des données.

En définitive, ce travail a permis de concevoir et de mettre en œuvre un modèle de surveillance forestière optimisé. En capturant les données par des capteurs acoustiques et de géolocalisation, et en les traitant grâce à un ESP8266 et un Arduino Mega, la plateforme offre une visualisation

en temps réel des alertes sur ThingSpeak, avec un suivi et une gestion garantie par Django. Les tests confirment l'efficacité du système pour détecter des activités suspectes de manière fiable, bien que des limitations de couverture subsistent dans certaines régions.

Ce projet contribue donc à renforcer la lutte contre la déforestation en proposant une solution de surveillance adaptée au contexte sénégalais, avec un potentiel d'évolution et d'amélioration technologique pour élargir sa portée à l'avenir.

Pour résoudre les limitations actuelles et améliorer la plateforme, plusieurs perspectives sont envisagées :

- Amélioration de la Transmission LoRa : Étant donné que les configurations sont correctes, une exploration plus approfondie du fonctionnement de la passerelle Dragino LG01 ou l'utilisation d'une passerelle alternative pourraient résoudre les problèmes d'affichage. Tester des protocoles ou des formats de transmission alternatifs pourrait également être une piste pour surmonter les limitations actuelles ;
- Intégration d'un Système de Surveillance Hybride : En mettant en place une architecture capable de basculer entre les transmissions WiFi et LoRa selon la couverture réseau, il serait possible d'optimiser la connectivité et la fiabilité du système dans divers environnements ;
- Déploiement de drones pour la Surveillance Complémentaire : Ajouter des drones, pour une surveillance aérienne en temps réel, renforcerait la précision et la rapidité de vérification des alertes. Ces drones pourraient être activés automatiquement ou par l'utilisateur dès qu'une alerte est détectée ;
- Utilisation de Satellites Lacuna Space pour les Zones Isolées : Intégrer la transmission via satellite permettrait de couvrir des zones forestières isolées où ni WiFi ni LoRa ne sont disponibles. Cela renforcerait la capacité de détection en assurant une connexion même dans les environnements les plus éloignés.

En conclusion, ce projet constitue une avancée importante dans la surveillance des ressources forestières. Les perspectives d'amélioration permettront de surmonter les limites actuelles et d'optimiser la portée et la précision de la plateforme, la rendant ainsi plus efficace pour la détection des activités illégales.

Bibliographie et webographie

- [1] « Importance de la forêt dans la vie de l'homme en zone rurale ». Consulté le: 16 août 2024. [En ligne]. Disponible sur: <https://www.fao.org/4/XII/0525-A1.htm>
- [2] « Introduction ». Consulté le: 16 août 2024. [En ligne]. Disponible sur: <https://www.fao.org/4/t0843f/t0843f02.htm>
- [3] J. P. S. Coloma, « TEMPAS-contribution à la qualité dans un système d'alertes contextualisées adaptable », PhD Thesis, Université de Grenoble, 2014. Consulté le: 29 octobre 2024. [En ligne]. Disponible sur: <https://theses.hal.science/tel-01552629/>
- [4] A. RODRIGUEZ, « gestion alerte : comment faire ? », Systeme d"alerte par SMS, logiciel de supervision des alarmes et astreinte. Consulté le: 14 août 2024. [En ligne]. Disponible sur: <https://www.clever.fr/gestion-alerte-comment-faire/>
- [5] M. Arru, E. Negre, et C. Rosenthal-Sabroux, « Emergence d'un nouveau type de Système de Systèmes : observations et propositions à partir du système d'alerte national français », présenté à 36ème congrès INFORSID. Construire les Systèmes d'Information pour la Transformation des Organisations à l'ère de l'Innovation Numérique (INFORSID 2018), 2018, p. 143. Consulté le: 14 août 2024. [En ligne]. Disponible sur: <https://hal.science/hal-02178984>
- [6] « Les systèmes d'alertes précoces ont permis de sauver des vies face aux phénomènes météorologiques extrêmes. | CCNUCC ». Consulté le: 29 octobre 2024. [En ligne]. Disponible sur: <https://unfccc.int/fr/news/les-systemes-d-alertes-precoces-ont-permis-de-sauver-des-vies-face-aux-phenomenes-meteorologiques?form=MG0AV3>
- [7] « La vigilance | Géorisques ». Consulté le: 14 août 2024. [En ligne]. Disponible sur: <https://www.georisques.gouv.fr/minformer-sur-la-prevention-des-risques/la-vigilance>
- [8] « Le système d'alerte et d'information des populations - Ornikar ». Consulté le: 14 août 2024. [En ligne]. Disponible sur: <https://www.ornikar.com/code/cours/securite/prevention-routiere/saip>
- [9] « adsp n° 106 - Alerte en santé publique ». Consulté le: 14 août 2024. [En ligne]. Disponible sur: <https://www.hcsp.fr/explore.cgi/adsp?clef=168>

- [10] « COVID-19 », Les services de l'État en Seine-Maritime. Consulté le: 14 août 2024. [En ligne]. Disponible sur: <https://www.seine-maritime.gouv.fr/Actualites/Archives/COVID-19>
- [11] M. POUYE, A. D. KORA, A. GNANSOUNOU, et B. NIANG, « AMELIORATION DE LA NAVIGATION MARITIME ET LE CONTROLE DU PARC (BATEAUX, NAVIRES, PIROGUES) LE LONG DES COTES », Consulté le: 7 novembre 2024. [En ligne]. Disponible sur: https://www.researchgate.net/profile/Ahmed-Dooguy-Kora/publication/305687315_AMELIORATION_DE_LA_NAVIGATION_MARITIME_ET_LE_CONTROLE_DU_PARC_BATEAUX_NAVIRES_PIROGUES_LE_LONG_DES_COTES/links/5799b76808ae33e89fb2a8be/AMELIORATION-DE-LA-NAVIGATION-MARITIME-ET-LE-CONTROLE-DU-PARC-BATEAUX-NAVIRES-PIROGUES-LE-LONG-DES-COTES.pdf
- [12] « Évaluer le contenu des notifications d'alerte diffusées en France via FR-Alert® : enjeux scientifiques et retombées opérationnelles ». Consulté le: 14 août 2024. [En ligne]. Disponible sur: <https://journals.openedition.org/netcom/8740>
- [13] « Les moyens d'alerte - Les bons reflexes 2023 ». Consulté le: 14 août 2024. [En ligne]. Disponible sur: <https://www.lesbonsreflexes.com/les-moyens-dalerte/>
- [14] M. M. BALDE, A. C. DIOUF, et A. I. S. Wally, « Coupe illicite de bois et recomposition territoriale: le cas des communes de Badion et de Kandia en haute Casamance (sud du Sénégal) », *Espace Géographique et Société Marocaine*, n° 49, 2021, Consulté le: 25 décembre 2024. [En ligne]. Disponible sur: <https://revues.imist.ma/index.php/EGSM/article/download/26398/13841>
- [15] « LE REVEIL CITOYEN CONTRE LA COUPE ILLICITE DE BOIS », SenePlus. Consulté le: 25 décembre 2024. [En ligne]. Disponible sur: <https://www.senepius.com/societe/le-reveil-citoyen-contre-la-coupe-illicite-de-bois>
- [16] « Caractérisation des méthodes de coupe de bois dans les zones aménagée et non aménagée en Haute Casamance, Sénégal ». Consulté le: 14 août 2024. [En ligne]. Disponible sur: https://www.ifgdg.org/index.php?option=com_content&view=article&id=1502%3Acaracterisation-des-methodes-de-coupe-de-bois-dans-les-zones-amenagee-et-non-amenagee-en-haute-casamance-senegal&catid=160%3Adecember-2014&Itemid=564
- [17] B. Solly, E. H. B. Diéye, I. Mballo, O. Sy, T. Sane, et M. Thior, « Dynamique spatio-temporelle des paysages forestiers dans le Sud du Sénégal: cas du département de Vélingara », *Physio-Géo. Géographie physique et environnement*, n° Volume 15, p. 41-67, 2020.
- [18] « Boite à outils sur la gestion forestière – Foncier au Sénégal ». Consulté le: 14 août 2024. [En ligne]. Disponible sur: <https://foncierausenegal.info/boite-a-outils-sur-la-gestion-forestiere/>
- [19] « Sénégal : Freiner la déforestation | ULB-Coopération ». Consulté le: 14 août 2024. [En ligne]. Disponible sur: <https://www.ulb-cooperation.org/fr/projets/senegal-freiner-la-deforestation/>
- [20] « Sénégal : quand le trafic de bois alimente la rébellion en Casamance - Jeune Afrique.com », JeuneAfrique.com. Consulté le: 25 décembre 2024. [En ligne]. Disponible sur: <https://www.jeuneafrique.com/1340050/politique/senegal-quand-le-traffic-de-bois-alimente-la-rebellion-en-casamance/>
- [21] A. Andreadis, G. Giambene, et R. Zambon, « Monitoring Illegal Tree Cutting through Ultra-Low-Power Smart IoT Devices », *Sensors*, vol. 21, n° 22, Art. n° 22, janv. 2021, doi: 10.3390/s21227593.

- [22] I. Mporas, I. Perikos, V. Kelefouras, et M. Paraskevas, « Illegal Logging Detection Based on Acoustic Surveillance of Forest », *Applied Sciences*, vol. 10, n° 20, Art. n° 20, janv. 2020, doi: 10.3390/app10207379.
- [23] « Chahed: Système Intelligent Policier Pour La Gestion De La Sécurité Publique. » Consulté le: 9 décembre 2024. [En ligne]. Disponible sur: <https://theses-algerie.com/2922400986475061/memoire-de-master/centre-universitaire-abdel-hafid-boussouf---mila/chahed-syst%C3%A8me-intelligent-policier-pour-la-gestion-de-la>
- [24] N. Salameh, « Conception d'un système d'alerte embarqué basé sur les communications entre véhicules », phdthesis, INSA de Rouen, 2011. Consulté le: 14 août 2024. [En ligne]. Disponible sur: <https://theses.hal.science/tel-00722227>
- [25] G. Tagne, A. Kokosy, P. Saey, et M. Steeve, « Système d'alerte sur obstacles pour améliorer l'autonomie des personnes utilisant un fauteuil roulant », in *Conférence Handicap 2022 – 12ème édition*, Paris, France: IFRATH, juin 2022. Consulté le: 14 août 2024. [En ligne]. Disponible sur: <https://hal.science/hal-03962928>
- [26] B. N. E. Imene et C. Lamia, « Conception et réalisation d'un système d'alerte d'accident de véhicule à base de la carte IOIO. », Université Mouloud Mammeri, 2015. Consulté le: 14 août 2024. [En ligne]. Disponible sur: <https://dspace.ummo.dz/handle/ummo/12235>
- [27] R. C. Whytock *et al.*, « Real-time alerts from AI-enabled camera traps using the Iridium satellite network: a case-study in Gabon, Central Africa », 13 novembre 2021, *bioRxiv*. doi: 10.1101/2021.11.10.468078.
- [28] . I. N., « MEDICAL ALERT SYSTEM FOR REMOTE HEALTH MONITORING USING SENSORS AND CLOUD COMPUTING », *IJRET*, vol. 03, n° 04, p. 884-888, avr. 2014, doi: 10.15623/ijret.2014.0304156.
- [29] « Lutte contre le trafic de bois au Sénégal: les nouvelles mesures du Président Sall seront-elles efficaces? », ISS Africa. Consulté le: 16 août 2024. [En ligne]. Disponible sur: <https://issafrica.org/fr/iss-today/lutte-contre-le-traffic-de-bois-au-senegal-les-nouvelles-mesures-du-president-sall-seront-elles-efficaces>
- [30] L. Amir, « La Génération Automatique des Ontologies à partir des Diagrammes de classes UML », Working Paper, 2017. Consulté le: 16 août 2024. [En ligne]. Disponible sur: <http://dspace.univ-guelma.dz/jspui/handle/123456789/1475>
- [31] K. NIZAR, « MERISE », Consulté le: 31 octobre 2024. [En ligne]. Disponible sur: <http://khayatinizar.0fees.net/Docs/CoursMerise.pdf>
- [32] *Merise Guide pratique*. Consulté le: 16 août 2024. [En ligne]. Disponible sur: https://books.google.com/books/about/Merise_Guide_pratique.html?hl=fr&id=dz8IOUYpFUMC
- [33] Xiaoshan Li, Zhiming Liu, et H. Jifeng, « A formal semantics of UML sequence diagram », in *2004 Australian Software Engineering Conference. Proceedings.*, Melbourne, Vic., Australia: IEEE, 2004, p. 168-177. doi: 10.1109/ASWEC.2004.1290469.
- [34] M. Alonso, L. Auxepaules, et D. Py, « DIAGRAM, un EIAH pour l'initiation à la modélisation orientée objet avec les diagrammes de classe UML », *STICEF (Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation)*, vol. 17, p. 22 pages, 2010.
- [35] B. Ben Ammar, M. T. Bhiri, et J. Souquières, « Quelques patrons de raffinement pour le développement de diagrammes de classes UML », in *6ème atelier sur les Objets, Composants et*

Modèles dans l'ingénierie des Systèmes d'Information, OCM-SI, couplé avec le 25ème congrès INFORSID, Perros-Guirec, France, mai 2007, p. 12 pages. Consulté le: 10 novembre 2024. [En ligne]. Disponible sur: <https://hal.science/hal-00182740>

[36] « MEMS Microphone Breakout INMP401 ADMP401 SparkFun Philippines », circuitrocks. Consulté le: 16 août 2024. [En ligne]. Disponible sur: <https://circuit.rocks/products/mems-microphone-sound-sensor>

[37] « NEO-6M GPS Module », TEKPARTS. Consulté le: 16 août 2024. [En ligne]. Disponible sur: <https://www.tekparts.in/product/neo-6m-gps-module/>

[38] G. TRONIC, « Module NodeMCU ESP8266 », GO TRONIC. Consulté le: 31 octobre 2024. [En ligne]. Disponible sur: <https://www.gotronic.fr/art-module-nodemcu-esp8266-27744.htm>

[39] « Arduino Mega 2560 Rev3 », Arduino Official Store. Consulté le: 31 octobre 2024. [En ligne]. Disponible sur: <https://store.arduino.cc/products/arduino-mega-2560-rev3>

[40] A. Kurniawan, *Dragino LoRa Development Workshop*. PE Press.

[41] « LG01-N -- Single Channel LoRa IoT Gateway ». Consulté le: 30 octobre 2024. [En ligne]. Disponible sur: <https://www.dragino.com/products/lora/item/143-lg01n.html>

[42] M. GRIFFON, C. MAUGET, C. GUYOT, E. SPOTO, V. MOREL, et P. YANG, « INTERNET DES OBJETS POUR LA SMARTCITY », Consulté le: 3 novembre 2024. [En ligne]. Disponible sur: https://moodle.insa-rouen.fr/pluginfile.php/167510/mod_folder/content/0/Rapport_P6_2021_32.pdf

[43] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, et T. Berners-Lee, « Protocole de transfert hypertexte-HTTP/1.1 ». juin, 1999. Consulté le: 3 novembre 2024. [En ligne]. Disponible sur: http://sgnsl.free.fr/ftp/RFC/HTTP_rfc_2616_fr.pdf

[44] M. F. Neto, « Tutorial da ferramenta de modelagem ASTAH ». Santa. Consulté le: 31 octobre 2024. [En ligne]. Disponible sur: <http://codilon.qlix.com.br/bd/bdaula09.pdf>

[45] « Premier Diagramming, Modeling Software & Tools », Astah. Consulté le: 17 août 2024. [En ligne]. Disponible sur: <https://astah.net/>

[46] П. А., *Python, PyCharm и Django для начинающих*. БХВ-Петербург, 2021.

[47] « PyCharm, a JetBrains IDE (@pycharm) / X », X (formerly Twitter). Consulté le: 30 août 2024. [En ligne]. Disponible sur: <https://x.com/pycharm>

[48] T. S. Jalolov, « ADVANTAGES OF DJANGO FEMWORKER », *imjrd*, vol. 10, n° 12, Art. n° 12, déc. 2023, Consulté le: 31 octobre 2024. [En ligne]. Disponible sur: <http://www.ijmrd.in/index.php/imjrd/article/view/573>

[49] « Présentation de django ». Consulté le: 30 août 2024. [En ligne]. Disponible sur: <https://python.doctor/page-django-introduction-python>

[50] « IoT Analytics - ThingSpeak Internet of Things ». Consulté le: 17 août 2024. [En ligne]. Disponible sur: <https://thingspeak.com/>

[51] « Blog Search Result »: Consulté le: 17 août 2024. [En ligne]. Disponible sur: <https://www.iqhome.org/blog-search?tag=thingspeak>

- [52] R. Stringer, *Twilio Cookbook*. Packt Publishing Ltd, 2014. Consulté le: 25 septembre 2024. [En ligne]. Disponible sur: <https://books.google.com/books?hl=fr&lr=&id=LNcrAwAAQBAJ&oi=fnd&pg=PT6&dq=Twilio&ots=lclG7omGa9&sig=d9TsNnoPoputF0GjLJyeb0JAOKE>
- [53] « Logo Twilio, symbole, signification, historique, PNG, marque ». Consulté le: 25 septembre 2024. [En ligne]. Disponible sur: <https://logos-world.net/twilio-logo/>
- [54] O. M. Кривонос, O. B. Струтинська, et M. П. Кривонос, « The use of visual electronic circuits modelling and designing software Fritzing in the educational process », *Вісник Житомирського державного університету імені Івана Франка. Педагогічні науки*, n° 108, p. 198-208, 2022.
- [55] « Welcome to Fritzing ». Consulté le: 17 août 2024. [En ligne]. Disponible sur: <https://fritzing.org/>
- [56] C. Peña, *Arduino IDE: Domina la programación y controla la placa*. RedUsers, 2020.
- [57] « Logiciel : IDE – Arduino : l'essentiel ». Consulté le: 17 août 2024. [En ligne]. Disponible sur: <https://arduino.blaisepascal.fr/presentation/logiciel/>
- [58] Y. Derfoufi, « Programmation en langage Python », 2019, Consulté le: 28 août 2024. [En ligne]. Disponible sur: <https://hal.science/hal-02126596/document>
- [59] W. Python, « Python », *Python releases for windows*, vol. 24, 2021, Consulté le: 31 octobre 2024. [En ligne]. Disponible sur: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=1f2ee3831eebfc97bfafd514ca2abb7e2c5c86bb>
- [60] J. W. Flory, « Help port Python packages to Python 3 », *Fedora Magazine*. Consulté le: 14 septembre 2024. [En ligne]. Disponible sur: <https://fedoramagazine.org/help-port-python-packages-python-3/>
- [61] K. P. Gaffney, M. Prammer, L. Brasfield, D. R. Hipp, D. Kennedy, et J. M. Patel, « SQLite: past, present, and future », *Proc. VLDB Endow.*, vol. 15, n° 12, p. 3535-3547, août 2022, doi: 10.14778/3554821.3554842.
- [62] admin@dbschema.com, « DbSchema | How to Create a Database in SQLite? » Consulté le: 15 septembre 2024. [En ligne]. Disponible sur: <https://dbschema.com/>

Annexe 1 : Code avec ESP8266

Importation des bibliothèques :

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
```

Figure 51 : Importation des bibliothèques

Ces bibliothèques permettent d'utiliser le module WiFi (ESP8266WiFi), de réaliser des requêtes HTTP (ESP8266HTTPClient), de lire des données GPS (TinyGPS++), et de créer une communication série virtuelle pour le GPS (SoftwareSerial).

Déclaration des constantes et variables :

```
const char* ssid = "votre ssid";
const char* password = "votre mot de passe";
const char* apiKey = "votre Clé API"; // Clé API de ThingSpeak
const char* server = "http://api.thingspeak.com";

WiFiClient client;
TinyGPSPlus gps;
SoftwareSerial ss(D1, D2); // RX, TX pour le module GPS

const int seuil = 500; // Seuil du capteur de son
```

Figure 52 : Déclaration des constantes

Dans ce code, les informations de connexion WiFi et la clé API pour ThingSpeak sont d'abord déclarées afin d'assurer la communication avec la plateforme en ligne. Ensuite, le client WiFi, le module GPS et un port série dédié pour le GPS sont instanciés pour permettre la connexion et la récupération des données de localisation. Un seuil sonore est également défini, servant de déclencheur d'alerte lorsqu'une activité suspecte est détectée au-delà de ce niveau sonore.

Fonction setup () :

```
void setup() {
  Serial.begin(9600);
  ss.begin(9600); // Vitesse de communication du module GPS

  WiFi.begin(ssid, password);
  Serial.print("Connexion à ");
  Serial.println(ssid);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nConnecté au WiFi !");
}
```

Figure 53 : fonction setup

Le programme commence par l'initialisation des communications série, en configurant à la fois la connexion standard (Serial) et une communication SoftwareSerial pour le module GPS. Ensuite, la connexion au réseau WiFi est établie, avec une vérification en boucle de l'état de connexion jusqu'à ce que celle-ci soit effective. Une fois la connexion réussie, un message de confirmation s'affiche pour indiquer que l'appareil est désormais connecté au réseau.

Fonction loop() et Vérification du seuil :

```
void loop() {
  int soundValue = analogRead(A0); // Lire la valeur du capteur de son ADMP401
  Serial.print("Valeur du son : ");
  Serial.println(soundValue);

  // Seuil atteint, récupération des données GPS
  if (soundValue > seuil) {
    Serial.println("Seuil atteint, lecture des coordonnées GPS...");
  }
}
```

Figure 54 : Fonction Loop

Le programme lit la valeur du capteur de son, connecté à l'entrée analogique A0, pour surveiller les niveaux sonores en temps réel. Cette valeur est affichée dans la console, permettant ainsi un diagnostic et un suivi visuel des données enregistrées par le capteur. Lorsque la valeur sonore mesurée dépasse le seuil prédéfini, une alerte est automatiquement déclenchée pour signaler une activité potentiellement suspecte.

Lecture des données GPS et Construction de l'URL et envoi des données :

```
if (gps.location.isValid()) {
    double latitude = gps.location.lat();
    double longitude = gps.location.lng();

    Serial.print("Latitude : ");
    Serial.println(latitude, 6);
    Serial.print("Longitude : ");
    Serial.println(longitude, 6);

    // Construire l'URL avec les données GPS
    String url = String(server) + "/update?api_key=" + apiKey +
        "&field1=" + String(latitude, 6) +
        "&field2=" + String(longitude, 6);

    HTTPClient http;
    http.begin(client, url); // Initialise la requête HTTP
    int httpCode = http.GET(); // Envoie la requête GET

    if (httpCode > 0) {
        Serial.println("Données GPS envoyées à ThingSpeak");
    } else {
        Serial.println("Erreur lors de l'envoi des données");
    }

    http.end(); // Termine la requête HTTP
}
```

Figure 55 : Construction de l'URL et envoi des données

Les données GPS sont lues et validées afin de s'assurer de leur exactitude. Si les données sont jugées valides, les coordonnées de latitude et de longitude sont récupérées. Ensuite, une URL est construite pour l'envoi des données, intégrant la clé API ainsi que les informations de localisation. Une requête GET est alors envoyée via HTTP pour mettre à jour les données sur la plateforme ThingSpeak.

Affichage des Données dans le Moniteur Série



```
COM4
Valeur du son : 879
Seuil atteint, envoi des coordonnées GPS...
Données GPS envoyées à ThingSpeak
Valeur du son : 878
Seuil atteint, envoi des coordonnées GPS...
Données GPS envoyées à ThingSpeak
Valeur du son : 877
Seuil atteint, envoi des coordonnées GPS...
Données GPS envoyées à ThingSpeak
Valeur du son : 878
Seuil atteint, envoi des coordonnées GPS...
Données GPS envoyées à ThingSpeak
Valeur du son : 876
Seuil atteint, envoi des coordonnées GPS...
```

Figure 56 : Affichage des Données dans le Moniteur Série

Annexe 2 : Code avec Lora

Inclusions des bibliothèques

```
#include <SPI.h>
#include <LoRa.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
```

Les bibliothèques utilisées dans ce code incluent SPI.h, qui est essentielle pour la communication SPI et est utilisée par le module LoRa. La bibliothèque LoRa.h est spécifique à la gestion du module LoRa, tandis que TinyGPS++.h facilite la gestion des données GPS. Enfin, SoftwareSerial.h permet la communication série sur d'autres broches numériques, offrant ainsi une flexibilité dans la configuration des connexions.

Définition des constantes et paramètres

```
// Paramètres LoRa
#define LORA_CS_PIN 10 // Pin CS pour le LoRa
#define LORA_RESET_PIN 9 // Pin Reset pour le LoRa
#define LORA_IRQ_PIN 2 // Pin IRQ pour le LoRa

// Paramètre du capteur ADMP
#define ADMP_PIN A0 // Le capteur ADMP est connecté à la broche analogique A0

// Paramètres GPS
SoftwareSerial gpsSerial(D1, D2); // RX, TX pour le module GPS
TinyGPSPlus gps;

const int seuil = 20; // Seuil du capteur de son
```

LORA_CS_PIN, LORA_RESET_PIN et LORA_IRQ_PIN sont des pins spécifiques utilisées pour le module LoRa, tandis que ADMP_PIN désigne le pin analogique à laquelle le capteur de son est connecté. gpsSerial est une instance créée pour établir la communication avec le module

GPS sur les broches D1 (RX) et D2 (TX). L'instance `gps` de la classe `TinyGPSPlus` est utilisée pour gérer les données GPS, et la variable `seuil` représente le seuil sonore qui déclenche l'envoi de données lorsque celui-ci est dépassé.

Fonction `setup`

```
void setup() {
  Serial.begin(9600);           // Communication avec le moniteur série
  gpsSerial.begin(9600);       // Initialisation de la communication série avec le GPS

  // Initialisation du module LoRa
  LoRa.setPins(LORA_CS_PIN, LORA_RESET_PIN, LORA_IRQ_PIN);
  if (!LoRa.begin(868000000)) { // Fréquence de 868 MHz pour l'Europe
    Serial.println("Échec de l'initialisation du LoRa");
    while (1);
  }
  Serial.println("LoRa initialisé");
}
}
```

moniteur série. La fonction `gpsSerial.begin(9600)` initialise la communication série avec le module GPS. Ensuite, `LoRa.setPins(...)` définit les pins spécifiques pour le module LoRa. La méthode `LoRa.begin(868000000)` démarre le module LoRa à la fréquence de 868 MHz, qui est couramment utilisée en Europe. En cas d'échec de l'initialisation, un message d'erreur est affiché et le programme se bloque pour éviter toute opération incorrecte.

Fonction `loop`

```
void loop() {
  // Lire la donnée du capteur ADMP
  int soundValue = analogRead(ADMP_PIN); // Lire la valeur analogique du capteur ADMP
  Serial.print("Valeur du capteur ADMP : ");
  Serial.println(soundValue);           // Afficher la valeur lue sur le moniteur série

  // Si le seuil est atteint, lire les données GPS
  if (soundValue > seuil) {
    Serial.println("Seuil atteint, lecture des données GPS...");

    // Attendre jusqu'à ce que les données GPS soient prêtes
    while (gpsSerial.available() > 0) {
      gps.encode(gpsSerial.read());
    }

    // Vérification des données GPS
    if (gps.location.isValid()) {
      double latitude = gps.location.lat();
      double longitude = gps.location.lng();

      // Construire le message à envoyer via LoRa
      String message = String("Lat: ") + String(latitude, 6) +
        " Lon: " + String(longitude, 6);

      // Envoi du message via LoRa
      LoRa.beginPacket();
      LoRa.print(message); // Envoyer les données GPS
    }
  }
}
```

La fonction `analogRead(ADMP_PIN)` lit la valeur du capteur de son connecté à la pin A0 et affiche cette valeur sur le moniteur série pour un diagnostic. Ensuite, le programme vérifie si la valeur sonore dépasse le seuil défini ; si c'est le cas, il tente de lire les données GPS. Pour cela, il utilise une boucle pour lire les données GPS disponibles et les encode avec `gps.encode()`. Les coordonnées GPS sont ensuite validées ; si elles sont valides, elles sont récupérées et formatées

dans un message. Ce message, contenant les coordonnées, est ensuite envoyé via le module LoRa. Un message de confirmation de l'envoi des données est affiché sur le moniteur série. Enfin, le programme attend 10 secondes avant de procéder à la prochaine vérification des niveaux sonores.