



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE IBN KHALDOUN - TIARET



MEMOIRE

Présenté à :

Faculté des sciences appliquées
Département de Génie Electrique

Pour l'obtention du diplôme de **MASTER**
Spécialité : Commande Electrique

En vue de créer une startup



Par :



ABDELLAH Ismail
ALLAOUI Abdelghani

Sur le thème

Développement d'un Système Intelligent d'Incubation d'Œufs à l'aide de l'Apprentissage Profond

Soutenu publiquement le 22 / 06 / 2024 à Tiaret devant le jury composé de :

M. MIHOUB Youcef	MCA	Université de Tiaret	Président
M. BELFEDAL Cheikh	Pr	Université de Tiaret	Encadrant
M. ALLAOUI Tayeb	Pr	Université de Tiaret	Co-encadrant
M. BELABBAS Belkacem	MCA	Université de Tiaret	Examinateur
M. GUEMMOUR Mohamed	MCA	Université de Tiaret	Représentant de l'incubateur
M. BOUDALI Abdelkader		Directeur de CC Tiaret	Représentant du partenaire économique

2023-2024

REMERCIEMENTS

Tout d'abord, notre gratitude va à ALLAH, Seigneur du monde, pour nous avoir donné la santé et l'énergie nécessaires à l'accomplissement de ce projet.

Nous tenons particulièrement à remercier nos superviseurs, le Pr. BELFEDHAL Cheikh et ALLAOUI Tayeb, pour leur encadrement, leur aide tant financière que morale, et leurs conseils durant la réalisation de ce projet.

Nous remercions sincèrement les membres du jury d'avoir accepté de faire partie du comité d'examen. Nous tenons également à exprimer notre gratitude à tous nos professeurs, qui ont contribué à notre formation tout au long de nos années d'études.

Nos remerciements s'adressent également aux membres du laboratoire de génie énergétique et génie informatique (L2GEGI) de l'Université de Tiaret.

Nous remercions les examinateurs, M. MIHOUB Youcef, d'avoir accepté de présider la cérémonie de remise des diplômes, ainsi que M. BELABBAS Belkacem, pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Nous n'oublions pas notre gratitude envers tous nos anciens professeurs.

Nous voudrions également saluer chaleureusement toutes les nombreuses personnes qui, de près ou de loin, ont contribué à notre travail : enseignants, collègues participants à nos recherches, et collègues de travail que nous remercions pour leur formation et leur soutien.

Sommaire

Introduction générale	1
Chapitre I Généralités sur les couveuses	5
Introduction	6
I.1 Définition	7
I.2 Historique	7
I.3 Principes de fonctionnement	10
I.4 Types de couveuses	11
I.4.1 La couveuse ventilée	11
I.4.2 La couveuse statique	11
I.5 Paramètres essentiels pour une incubation réussie des œufs	12
I.5.1 Température	12
I.5.2 Ventilation	13
I.5.3 Humidité	13
I.5.4 Retournement	13
I.6 Limitations et défis	14
I.6.1 Limitations technologiques	14
I.6.2 Variabilité des résultats	14
I.6.3 Gestion de l'énergie	14
I.6.4 Adaptabilité aux différentes espèces	15
I.6.5 Coût d'acquisition et d'entretien	15
Conclusion	15
Chapitre II Les réseaux neuronaux et l'industrie avicole	16
Introduction	17
II.1 Définition	18
II.2 Processus d'intelligence artificielle	18
II.3 Les Réseaux Neuronaux	19
II.3.1 Définition	19
II.3.2 Le Perceptron	20
II.3.3 Réseau de neurones à propagation directe (Feed-forward)	21
II.3.4 Réseaux neuronaux récurrents (RNN)	22
II.4 L'Apprentissage d'un réseau de neurones	23
II.4.1 Fondamentaux de l'Entraînement des Modèles	23
II.4.2 Fonction de Perte (Loss Function)	24

II.4.3 Algorithmes d'Optimisation	24
II.4.4 Défis dans l'Entraînement des Modèles.....	24
II.5 Réseaux de mémoire à long terme (LSTM)	25
II.5.1 Pourquoi les LSTM ?	25
II.5.2 Architecture des LSTM.....	25
II.5.3 Composants d'une cellule LSTM	26
II.5.4 Applications des LSTM.....	27
II.6 Utilisation des réseaux neuronaux et de l'intelligence artificielle dans les couveuses d'œufs..	27
Conclusion.....	29
Chapitre III Conception de la couveuse	31
Introduction	32
III.1 Partie matériel.....	32
I.1.1 Enclos de couveuse.....	32
III.1.2 Capteurs.....	33
III.1.2.1 Capteur de température	33
III.1.2.2 Capteur d'humidité.....	34
III.1.2.3 Circuit de conditionnement :	34
III.1.3 Actionneurs.....	35
III.1.3.1 Élément chauffant.....	35
III.1.3.2 Humidificateur	36
III.1.3.3 Ventilateur de distribution	37
III.1.3.4 Ventilateurs de régulation et de renouvellement d'air.....	37
III.1.3.5 Système de retournement et portage d'œufs :	38
III.1.4 Relais.....	39
III.1.5 Energie solaire.....	39
III.1.5.1 Panneau solaire	40
III.1.5.2 Onduleur.....	40
III.1.5.3 Régulateur de Tension	41
III.1.5.4 Batterie.....	42
III.1.6 Disjoncteur de protection	43
III.2 Partie programme.....	43
III.2.1 Raspberry Pi.....	43
III.2.1.1 Types de Raspberry Pi.....	44
III.2.1.2 Fonctionnement de Raspberry Pi.....	45

III.2.2 Langage de programmation	45
Conclusion	46
Chapitre IV Mise en œuvre et analyse des résultats	47
Introduction	48
IV.1 Réalisation et mise en œuvre de la couveuse	48
IV.1.1 Construction de la couveuse	48
IV.1.2 Programmation	52
IV.2 Apprentissage et implémentation d'un modèle d'IA	58
IV.2.1 Collecte des données d'apprentissage	59
IV.2.2 Apprentissage du modèle	61
IV.2.2.1 Limitation :	61
IV.2.2.2 Résultat d'apprentissage	61
IV.2.3 L'implémentation du modèle dans la couveuse	65
IV.3 Résultat	65
Conclusion	67
Conclusion générale	68
Bibliographie	71

Liste des figures

Figure I.1: Illustration d'un four à œufs égyptien. [9]	8
Figure I.2: L'incubateur de Réaumur. [9]	9
Figure I.3: La première couveuse électrique (Petersime N°1). [13].....	10
Figure I.4: Incubateur statique. [15].....	12
Figure II.1: La structure Générale d'un perceptron.	20
Figure II.2: Un réseau neuronal multicouche de type feed-forward avec une couche d'entrée, deux couches cachées et une couche de sortie.	22
Figure II.3: Réseau neuronal récurrent (RNN) avec auto-feedback.....	23
Figure II.4: Une cellule de mémoire LSTM standard.	26
Figure III.1: Réfrigérateur.....	33
Figure III.2: Capteur de température DS18B20.....	34
Figure III.3: Capteur BME280.....	34
Figure III.4: Chaîne de mesure.	35
Figure III.5: Élément chauffant.....	36
Figure III.6: La dispositif d'humidification.....	36
Figure III.7: Ventilateur de distribution.	37
Figure III.8: Ventilateur d'extraction/admission.....	37
Figure III.9: Système de retournement.	38
Figure III.10: Vérin électrique.	39
Figure III.11: Module de relais à 6 canaux.	39
Figure III.12: Panneau solaire.....	40
Figure III.13: Onduleur.	41
Figure III.14: Régulateur de tension.	42
Figure III.15: Batterie au plomb.	42
Figure III.16: Disjoncteur magnétothermique bipolaire.	43
Figure III.17: Raspberry Pi 4 model B.....	44
Figure III.18: Popularité des langages de programmation en 2024. [48].....	46
Figure IV.1: La photo montre l'intérieur de la couveuse, y compris les composants de contrôle environnemental et le mécanisme de retournement.	49
Figure IV.2: La photo montre le circuit de contrôle comprenant le microcontrôleur et les relais reliés par des fils.	50
Figure IV.3: La photo montre l'arrière de la couveuse où se trouve le circuit d'alimentation et le réservoir d'eau secondaire.....	51
Figure IV.4: Système de Contrôle d'Incubateur Automatisé avec Raspberry Pi.....	52
Figure IV.5: Système de Contrôle Intelligent pour la couveuse.	59
Figure IV.6: Echantillon des données collectées sauvegardé dans un fichier csv (Excel).....	61
Figure IV.7: Le graphe montre la perte au cours de l'apprentissage et de la validation sur des époques pour le modèle LSTM.....	64
Figure IV.8: Les graphes représentent les changements de température et d'humidité pour le contrôle avec prédiction et sans prédiction.	66

Liste d'abréviation

LSTM: Long Short-Term Memory

IA: Intelligence Artificielle (Artificial Intelligence)

RNN: Réseaux Neuronaux Récurrents (Recurrent Neural Networks)

MPPT: Maximum Power Point Tracking

DC: Direct Current

AC: Alternating Current

Introduction générale

Introduction générale

L'industrie avicole joue un rôle important dans la satisfaction de la demande alimentaire mondiale, en particulier dans la production de produits avicoles tels que les œufs et la viande.

L'incubation des œufs est un processus essentiel dans ce secteur, car elle détermine la qualité des poussins éclos, ce qui a un impact sur la rentabilité des exploitations avicoles. Les méthodes traditionnelles d'incubation des œufs sont confrontées à des problèmes tels que des taux d'éclosion sous-optimaux, une surveillance laborieuse et un manque de contrôle précis des paramètres environnementaux. Cependant, avec l'avènement des technologies de pointe, une nouvelle ère de l'incubation des œufs a vu le jour, le Smart Egg Incubation System (système intelligent d'incubation des œufs) [1].

Le succès des incubateurs avicoles repose sur la maîtrise parfaite de quatre paramètres essentiels : la température, l'humidité, l'échange d'air et le retournement des œufs. La précision de leur régulation et de leur ajustement explique la réussite de ces dispositifs d'incubation.

L'apprentissage profond (Deep Learning), un sous-ensemble de l'intelligence artificielle (IA), implique l'utilisation de réseaux neuronaux à plusieurs couches (d'où le terme « profond ») pour analyser et interpréter des données complexes. Ces réseaux neuronaux sont conçus pour imiter la capacité du cerveau humain à apprendre par l'expérience, ce qui en fait des outils exceptionnellement puissants pour reconnaître des modèles et faire des prédictions sur la base de vastes ensembles de données. Dans le contexte de l'incubation des œufs, les algorithmes d'apprentissage profond peuvent être formés pour surveiller et contrôler divers facteurs environnementaux qui influencent le développement des embryons, tels que la température, l'humidité et la ventilation [2].

L'application de l'apprentissage profond aux incubateurs d'œufs peut révolutionner l'industrie avicole de plusieurs façons. Tout d'abord, elle peut améliorer la précision des contrôles environnementaux au sein de l'incubateur. Les couveuses traditionnelles s'appuient sur des réglages statiques ou de simples boucles de rétroaction pour maintenir des conditions optimales. En revanche, un incubateur alimenté par l'apprentissage profond peut apprendre en continu à partir de données en temps réel et ajuster les conditions de manière dynamique pour s'assurer que les

embryons se développent dans un environnement optimal. Cela se traduit par des taux d'éclosion plus élevés et des poussins en meilleure santé [3].

Deuxièmement, les réseaux neuronaux peuvent être utilisés pour surveiller le développement des embryons avec une précision sans précédent. En analysant des images et d'autres données de capteurs, ces systèmes peuvent détecter des signes subtils de détresse ou de développement anormal à un stade précoce du processus d'incubation. Cette détection précoce permet d'intervenir à temps, ce qui peut améliorer considérablement les chances de réussite de l'éclosion. Par exemple, les modèles d'apprentissage profond peuvent identifier les œufs qui ont peu de chances d'éclore, ce qui permet aux agriculteurs de concentrer leurs ressources sur ceux qui ont le plus de chances de réussir [4].

En outre, l'intégration de l'apprentissage profond dans les incubateurs d'œufs facilite l'automatisation du processus d'incubation, réduisant ainsi la nécessité d'une surveillance et d'une intervention manuelles. Cette automatisation permet non seulement de réduire les coûts de main-d'œuvre, mais aussi de minimiser les erreurs humaines, ce qui garantit un processus d'incubation plus cohérent et plus fiable. Pour les exploitations avicoles à grande échelle, cela peut conduire à des améliorations significatives de la productivité et de la rentabilité [5].

Le système développé dans notre projet consiste essentiellement à construire l'incubateur pour assurer une bonne isolation thermique, à automatiser le retournement des œufs, à mettre en œuvre les meilleurs capteurs de données et à utiliser des techniques d'apprentissage automatique et de réseaux neuronaux pour accroître la robustesse du contrôle de la température et de l'humidité de l'environnement.

En intégrant des réseaux neuronaux dans notre système, nous pouvons analyser les données en temps réel provenant des capteurs pour ajuster de manière dynamique et précise les conditions à l'intérieur de l'incubateur. Les réseaux neuronaux, grâce à leur capacité d'apprentissage continu, peuvent identifier des schémas subtils et des tendances qui échappent aux systèmes de contrôle traditionnels. Par exemple, ils peuvent anticiper les fluctuations de température dues à des facteurs externes et ajuster les systèmes de chauffage ou de refroidissement avant que ces variations n'affectent les œufs. De même, pour l'humidité, le réseau neuronal peut réguler les systèmes d'humidification et de déshumidification en temps réel, assurant un environnement stable et optimal.

En fin de compte, nous visons à maximiser les taux de réussite des éclosions et à améliorer la santé globale des poussins en créant un environnement d'incubation qui s'adapte de manière intelligente et proactive aux besoins des embryons.

Ce travail est dans le cadre de notre projet de fin d'études en Master, qui est structuré en quatre chapitres :

Chapitre 1 fournit un aperçu des incubateurs d'œufs, en discutant de leur développement historique et des paramètres essentiels nécessaires pour assurer des taux d'éclosion élevés et des poussins de qualité. Ce chapitre examine les aspects critiques tels que le contrôle de la température, les niveaux d'humidité, la ventilation et les mécanismes de retournement des œufs, offrant une compréhension complète des facteurs qui influencent l'incubation réussie.

Chapitre 2 se penche sur l'intelligence artificielle (IA) et les réseaux de neurones, en se concentrant spécifiquement sur les réseaux de neurones récurrents (RNN). Il explore l'application des RNN dans le domaine de l'aviculture, en particulier dans l'optimisation des processus d'incubation. Le chapitre explique les principes de l'IA et des réseaux de neurones, détaillant comment les RNN peuvent être utilisés pour prédire et contrôler efficacement les paramètres d'incubation.

Chapitre 3 illustre la conception et le développement du système d'incubation d'œufs. Nous détaillons les matériaux utilisés, les choix de conception, et la programmation mise en œuvre.

Chapitre 4 explique la phase de réalisation concrète du projet. Nous détaillons la mise en œuvre du système développé, en mettant en lumière les étapes clés du processus. Ensuite, nous passons à l'analyse des résultats obtenus à travers cette mise en œuvre. Cette section vise à présenter de manière claire et précise les résultats du travail accompli, tout en fournissant une évaluation critique de ces résultats.

Chapitre I Généralités sur les couveuses

Introduction

Les couveuses d'œufs représentent une pièce essentielle de l'agriculture moderne, offrant un contrôle précis et une atmosphère idéale pour l'incubation des œufs. Leur évolution au fil du temps reflète à la fois les avancées technologiques et notre compréhension croissante des besoins des embryons aviaires.

L'histoire des couveuses remonte à des siècles, mais leur développement significatif a commencé au XIXe siècle avec l'avènement de la révolution industrielle. Les premières versions étaient souvent rudimentaires, basées sur des principes simples de chaleur et d'humidité contrôlées. Cependant, au fur et à mesure que la science de l'embryologie avançait, les exigences précises pour une incubation réussie devenaient de plus en plus évidentes.

Au début du XXe siècle, des progrès dans les domaines de la thermodynamique et de la biologie ont conduit à des designs de couveuses plus sophistiqués. Des innovations telles que les thermostats automatiques et les systèmes de ventilation ont permis un contrôle plus fin de l'environnement interne, favorisant ainsi un taux de réussite d'incubation plus élevé.

L'avènement de l'électronique et de la technologie numérique a révolutionné le domaine des couveuses d'œufs. Les capteurs de température et d'humidité précis, associés à des microprocesseurs intelligents, ont permis un contrôle encore plus précis de l'environnement d'incubation. Les couveuses modernes sont capables de maintenir des conditions stables et optimales sur de longues périodes, ce qui améliore considérablement les taux d'éclosion et la santé des poussins.

Parallèlement aux progrès technologiques, la recherche scientifique continue à approfondir notre compréhension des besoins spécifiques des différents types d'œufs. Des études sur la physiologie embryonnaire, le développement des organes et les exigences en matière de nutrition ont alimenté l'innovation dans la conception des couveuses, permettant aux fabricants de personnaliser les paramètres d'incubation en fonction des espèces et des souches.

En outre, l'essor de l'agriculture durable et biologique a stimulé la demande de couveuses respectueuses de l'environnement, utilisant des sources d'énergie renouvelables et des matériaux écologiques.

I.1 Définition

Une couveuse (ou incubateur) est un appareil qui simule l'incubation aviaire en maintenant les œufs au chaud dans une plage de température donnée et dans un taux d'humidité correct, avec un mécanisme de rotation qui permet de les faire éclore.

L'incubateur est souvent utilisé pour la culture de bactéries, l'éclosion artificielle d'œufs ou la mise en place de conditions propices à une réaction chimique ou biologique. L'incubateur est utilisé pour faire éclore les œufs d'oiseaux et de reptiles. Il permet au fœtus à l'intérieur de l'œuf de se développer sans que la mère ait besoin d'être présente pour fournir de la chaleur. Les œufs de poule éclosent au bout d'environ 21 jours, mais d'autres espèces d'oiseaux peuvent prendre plus ou moins de temps [6]. Les couveuses sont également utilisées pour élever des oiseaux [7].

I.2 Historique

Les premiers à avoir découvert l'incubation artificielle des œufs sont les Égyptiens, qui disposaient d'une méthode d'incubation en 400 av. J.-C., en utilisant un four cylindrique dont le fond était chauffé par un feu. Les œufs à couver étaient placés sur un cône inversé partiellement recouvert de cendres. Les œufs étaient placés dans un panier tressé qui reposait sur les cendres. Le système était également doté d'un toit qui permettait à la fumée de s'échapper, mais qui empêchait la pluie de pénétrer dans le four. Les fours à œufs égyptiens sont généralement des structures en briques de forme pyramidale, dotées de deux chambres internes [8].

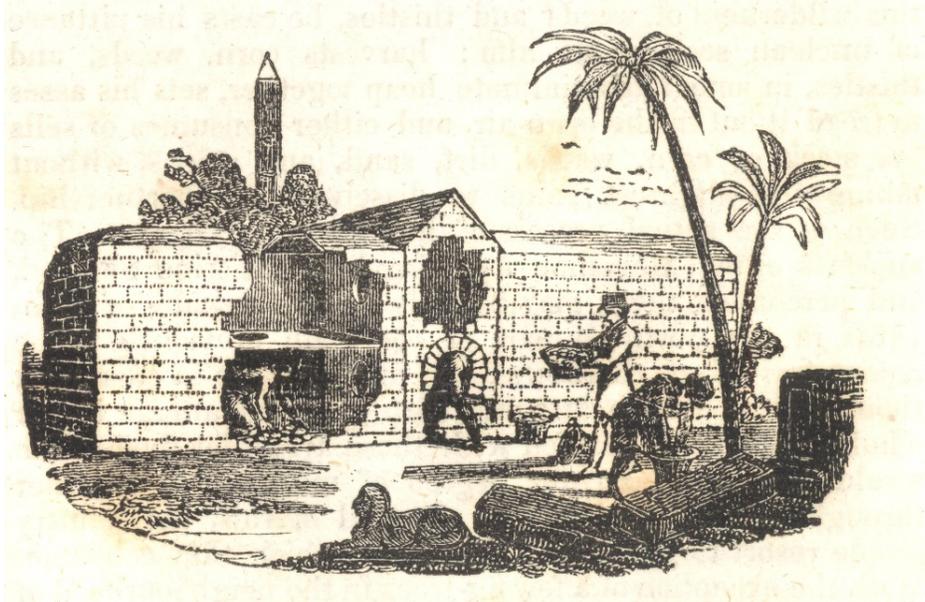


Figure I.1: Illustration d'un four à œufs égyptien. [9]

L'incubation scientifique contrôlée nécessitait une mesure précise et reproductible de la température, comme le thermomètre à alcool proposé par le naturaliste et scientifique français René Antoine Ferchault de Réaumur en 1730, et l'échelle de température qui porte son nom. Réaumur a utilisé le thermomètre dans son projet d'incubateur artificiel, présenté à l'Académie des sciences en 1747 et publié en 1749 sous le titre *l'Art de faire éclore et d'élever en toute saison des oiseaux domestiques de toutes espèces, soit par le moyen de la chaleur du fumier, soit par le moyen de celle du feu ordinaire* [10].



Figure I.2: L'incubateur de Réaumur. [9]

Lyman Byce a créé une couveuse à lampe à charbon en 1879 [11]. La première machine commerciale a été fabriquée par Hearson en 1881 [12].

Ira M. Petersime, de Gettysburg, Ohio, États-Unis, aurait inventé la couveuse électrique en 1922. Peu de temps après, lui et son fils, Ray, ont commencé à fabriquer des couveuses. Ils ont obtenu de nombreux brevets sur des caractéristiques de conception et des améliorations ultérieures. L'innovation de la couveuse électrique de grande capacité (Petersime a très tôt proposé un modèle pouvant contenir 15 000 œufs) a encouragé la croissance des couvoirs commerciaux à grande échelle.[13]



Figure I.3: La première couveuse électrique (Petersime N°1). [13]

Les couveuses modernes sont chauffées électriquement et contrôlées par un thermostat, tandis que certaines sont alimentées par l'énergie solaire et contrôlées par des microcontrôleurs afin d'assurer un contrôle précis des paramètres et d'automatiser le retournement des œufs.

I.3 Principes de fonctionnement

Le principe de fonctionnement d'une couveuse repose sur la création d'un environnement contrôlé qui reproduit les conditions nécessaires à l'incubation des œufs de manière artificielle. La couveuse maintient une température constante et précise, afin de simuler la chaleur corporelle d'une poule couveuse. Cette température constante est essentielle pour le développement embryonnaire optimal.

En parallèle, la couveuse contrôle également l'humidité de l'air à l'intérieur de la chambre d'incubation. Un niveau d'humidité adéquat est crucial pour empêcher les œufs de se dessécher et favoriser le développement correct de l'embryon

L'échangeur air-air assure la circulation de l'air à l'intérieur de la couveuse, garantissant ainsi une répartition uniforme de la chaleur et de l'humidité. Cela évite les zones de température ou d'humidité extrêmes qui pourraient compromettre le processus d'incubation.

Enfin, le retournement des œufs est une fonction importante des couveuses. Dans la nature, une poule couveuse retourne régulièrement ses œufs pour éviter que l'embryon ne colle à la coquille. Les couveuses reproduisent ce mouvement en retournant automatiquement les œufs à intervalles réguliers, assurant ainsi un développement embryonnaire uniforme.

I.4 Types de couveuses

Il existe deux catégories principales de couveuses [14] :

I.4.1 La couveuse ventilée

- Équipée d'un ou plusieurs ventilateurs d'extraction, elle réduit le gaz CO₂ et régule la température à l'intérieur de l'enceinte.
- Les ventilateurs assurent le renouvellement de l'air et maintiennent une température homogène dans la couveuse.
-

I.4.2 La couveuse statique

- Cet appareil ne dispose pas de système de ventilation. La résistance chauffante est placée au-dessus des œufs et le thermomètre est situé presque au même niveau que les œufs sur le support inférieur, qui simule un nid.
- La température est plus élevée dans la partie supérieure de la couveuse par rapport à la partie inférieure.
- Cette couveuse est dotée de deux événements : l'un pour l'entrée d'air et l'autre pour l'évacuation.

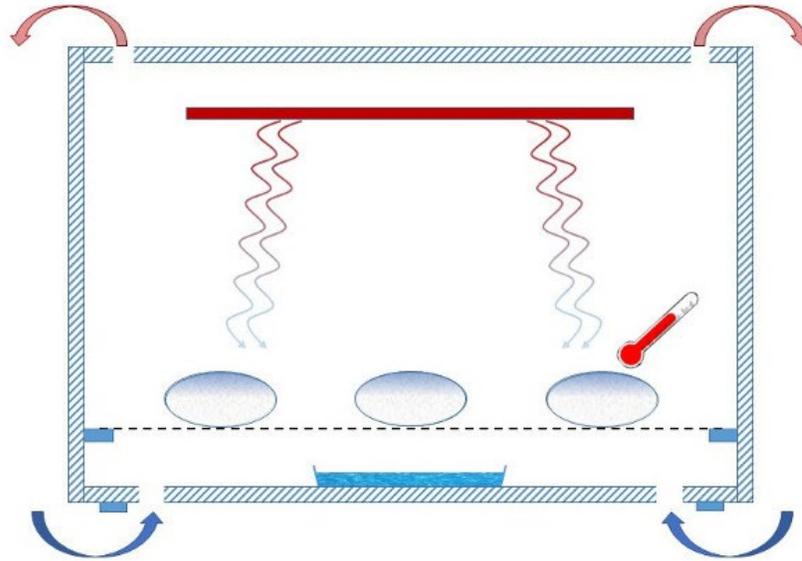


Figure I.4: Incubateur statique. [15]

Pour que l'incubation des œufs aboutisse à des éclosions réussies, certains paramètres-clés doivent être respectés, quel que soit le modèle d'incubateur. Cependant, il peut y avoir des variations dans la manière d'obtenir ces résultats.

I.5 Paramètres essentiels pour une incubation réussie des œufs

La réussite de l'incubation des œufs dépend de plusieurs facteurs cruciaux qui doivent être soigneusement pris en compte. Ces paramètres essentiels forment la base sur laquelle repose toute opération d'incubation efficace, et leur gestion adéquate est indispensable pour assurer le développement sain des embryons aviaires.

I.5.1 Température

La température est le facteur environnemental le plus critique pendant l'incubation, car l'embryon en développement ne peut tolérer que de très faibles fluctuations de température pendant la période d'incubation [16]. Les plages de température recommandées pour les incubateurs

(température de réglage des incubateurs) sont de 37,2 à 38,2°C pendant les 18 premiers jours (phase d'incubation). Pendant les trois derniers jours (phase d'éclosion), la température est abaissée entre 36,0 et 36,5°C [17].

I.5.2 Ventilation

Pendant l'incubation, les œufs produisent du dioxyde de carbone (CO₂). Si le taux de CO₂ dépasse 1%, l'embryon risque de mourir. Il est donc essentiel que l'incubateur soit équipé de trous d'aération permettant une bonne circulation de l'air. Il faut au moins deux trous : une entrée d'air (souvent près du ventilateur, s'il y en a un) et une sortie d'air. Certains incubateurs commerciaux n'ont qu'un seul trou, ce qui peut être insuffisant. Peu avant l'éclosion, chaque œuf de poule de 60g nécessite environ 1,7 litre d'air par heure. Pour d'autres types d'œufs, on peut ajuster cette valeur en fonction du poids de l'œuf. Dans un incubateur statique, l'air vicié doit être remplacé par de l'air frais, suivant un flux de bas en haut. Ajouter un ventilateur ou le placer incorrectement peut perturber ce flux et nuire aux embryons [18].

I.5.3 Humidité

L'humidité relative détermine le taux de perte d'humidité des œufs pendant l'incubation. Une faible humidité relative peut entraîner un dessèchement trop rapide du contenu des œufs, ce qui réduit le taux d'éclosion et produit des poussins plus petits que la normale (poussins de qualité médiocre) qui ne seront pas performants pendant la période de couvaie[19]. Pendant les 18 premiers jours, la plupart des fabricants d'incubateurs recommandent une humidité relative comprise entre 55 et 65 %. Ensuite, les exigences en matière d'humidité relative augmentent jusqu'à environ 70 %. Au moment de la ponte et de l'éclosion, l'humidité relative augmente (le dernier jour d'incubation) jusqu'à environ 75-80 % [17], [20].

I.5.4 Retournement

Le retournement des œufs est une autre opération importante de l'incubation artificielle qui affecte le développement embryonnaire et l'éclosion. Le retournement des œufs plusieurs fois par jour permet d'éviter que l'embryon ne colle à la membrane de la coquille, ce qui peut entraîner des

malformations ou la mort [21], [22]. La fréquence optimale de retournement est toutes les 1 à 2 heures pendant les 18 premiers jours d'incubation. Après le 18ème jour, les œufs ne doivent pas être dérangés jusqu'à l'éclosion [17], [23].

I.6 Limitations et défis

Bien que les couveuses d'œufs aient considérablement évolué grâce aux avancées technologiques et scientifiques, il reste encore des défis à surmonter. Ces défis reflètent à la fois la complexité de l'incubation des œufs et les exigences changeantes de l'industrie avicole moderne.

I.6.1 Limitations technologiques

Malgré les progrès réalisés, certaines technologies de couveuses peuvent encore présenter des limitations en termes de précision de la régulation de la température et de l'humidité, ce qui peut influencer la réussite de l'incubation.

I.6.2 Variabilité des résultats

Même avec des couveuses modernes, il existe une certaine variabilité dans les taux d'éclosion et la santé des poussins éclos. Comprendre et réduire cette variabilité reste un défi majeur pour les éleveurs.

I.6.3 Gestion de l'énergie

Les couveuses nécessitent souvent une consommation d'énergie importante pour maintenir les conditions optimales d'incubation. Des efforts sont nécessaires pour développer des technologies plus écoénergétiques et durables.

I.6.4 Adaptabilité aux différentes espèces

Les couveuses sont principalement conçues pour les œufs de poulet, mais leur adaptation à d'autres espèces avicoles peut poser des défis supplémentaires en raison des variations dans les besoins d'incubation.

I.6.5 Coût d'acquisition et d'entretien

Les couveuses de haute qualité peuvent représenter un investissement important pour les éleveurs, et leur entretien régulier peut également entraîner des coûts supplémentaires. Trouver un équilibre entre la performance et le coût reste un enjeu.

Pour relever ces défis, la recherche et le développement continu dans le domaine de l'incubation des œufs sont essentiels. Des avancées dans la technologie des capteurs, l'intelligence artificielle et la modélisation numérique peuvent contribuer à améliorer la précision, l'efficacité énergétique et l'adaptabilité des couveuses aux besoins spécifiques des éleveurs. De plus, une collaboration étroite entre les fabricants de couveuses, les chercheurs et les éleveurs peut conduire à des solutions innovantes pour surmonter ces défis et améliorer la productivité et la durabilité de l'industrie avicole.

Conclusion

Dans ce chapitre, nous avons abordé le point de vue général sur les incubateurs, leur histoire, les méthodes d'incubation et la gestion des couvoirs dans l'industrie avicole, et nous avons identifié les meilleures pratiques qui peuvent être mises en œuvre pour améliorer les taux d'éclosion et la qualité des poussins. Une bonne gestion des œufs à couvrir, l'environnement d'incubation et les opérations d'incubation artificielle sont des facteurs importants qui affectent le taux d'éclosio

Chapitre II Les réseaux neuronaux et l'industrie avicole

Introduction

L'intelligence artificielle (IA) a émergé comme une technologie transformatrice qui redéfinit de nombreux secteurs, y compris la santé, les finances, le transport et le divertissement. Au cœur de l'IA se trouve le développement d'algorithmes et de modèles computationnels capables d'accomplir des tâches nécessitant généralement une intelligence humaine, telles que l'apprentissage, le raisonnement, la résolution de problèmes et la perception. L'évolution de l'IA peut être retracée à ses débuts avec les systèmes symboliques et les systèmes experts, qui reposaient fortement sur des règles et de la logique prédéfinies. Cependant, ces premières approches ont rencontré des limitations significatives, en particulier lorsqu'il s'agissait de traiter de grandes quantités de données non structurées. Cela a conduit au développement d'apprentissage automatique (Machine Learning) et d'apprentissage profond (Deep Learning), qui sont désormais à la pointe de la recherche et des applications en IA [24].

Un aspect fondamental de l'IA moderne est le réseau neuronal, un modèle computationnel inspiré de la structure et du fonctionnement du cerveau humain. La forme la plus simple des réseaux neuronaux est le réseau feedforward, où les données circulent dans une seule direction, de la couche d'entrée à la couche de sortie. Bien qu'efficaces pour de nombreuses tâches, les réseaux feedforward rencontrent des difficultés avec les données séquentielles où le contexte et les dépendances temporelles sont cruciaux [25].

Pour répondre aux limitations des réseaux feedforward dans le traitement des données séquentielles, les réseaux neuronaux récurrents (RNN) ont été introduits. Contrairement aux réseaux feedforward, les RNN sont conçus avec des boucles dans leur architecture, permettant aux informations d'être retenues et utilisées à travers différentes étapes temporelles. Cette caractéristique rend les RNN particulièrement efficaces pour des tâches telles que la prévision de séries temporelles, la modélisation linguistique et la reconnaissance vocale [26]. Malgré leur potentiel, les RNN rencontrent souvent des défis avec les dépendances à long terme en raison de problèmes tels que la disparition et l'explosion des gradients lors de l'apprentissage [27].

Pour surmonter les lacunes des RNN traditionnels, les réseaux de mémoire à long terme (LSTM) ont été développés. Introduits par Hochreiter et Schmidhuber en 1997, les LSTM incluent des unités spéciales appelées cellules mémoire qui peuvent stocker des informations sur de longues périodes. L'architecture unique des LSTM les rend très efficaces pour les tâches impliquant des

dépendances à long terme, telles que la traduction automatique, l'analyse de sentiments et le traitement vidéo. Les LSTM ont été largement adoptés dans les milieux académiques et industriels en raison de leur robustesse et de leur capacité à apprendre des modèles temporels complexes [28].

L'implémentation de l'IA dans l'incubation des œufs, notamment pour la prédiction, révolutionne l'efficacité et les taux de réussite des processus d'éclosion. Les algorithmes d'IA analysent une multitude de points de données tels que la température et l'humidité, en prédisant les conditions optimales pour chaque étape de l'incubation. Cette capacité prédictive garantit que toute déviation par rapport à l'environnement idéal est rapidement corrigée, minimisant ainsi le risque de mortalité embryonnaire.

II.1 Définition

L'intelligence artificielle (IA) peut être définie comme la capacité d'une machine ou d'un programme informatique à montrer des comportements ou des actions qui seraient considérés comme intelligents s'ils étaient réalisés par un humain. L'IA englobe l'acquisition de l'intelligence par des moyens artificiels. Elle simule les processus cognitifs humains par des machines, leur permettant d'accomplir des tâches qui nécessitent généralement l'intelligence humaine, telles que l'apprentissage, le raisonnement, la résolution de problèmes et la compréhension du langage [29].

II.2 Processus d'intelligence artificielle

L'intelligence artificielle (IA) est un vaste domaine qui englobe divers processus et techniques permettant de construire des machines intelligentes capables d'effectuer des tâches qui requièrent généralement l'intelligence humaine. Voici le processus général de l'IA :

1. Définir le problème : la première étape de la construction d'un système d'IA consiste à définir le problème à résoudre. Il s'agit de comprendre les exigences du problème et de déterminer la faisabilité de la construction d'une solution d'IA [30].
2. Recueillir des données : Les systèmes d'IA ont besoin de grandes quantités de données pour apprendre et prendre des décisions. Vous devrez recueillir des données relatives au problème que vous essayez de résoudre. Ces données peuvent provenir de diverses sources, telles que des capteurs, des bases de données et des API externes.

3. Prétraiter les données : Une fois les données collectées, vous devez les prétraiter pour les rendre utilisables dans un système d'IA. Il peut s'agir de nettoyer, de transformer et de normaliser les données [30].
4. Choisissez les algorithmes : Ensuite, vous devez choisir les algorithmes que vous utiliserez pour former votre modèle d'IA. Le choix des algorithmes dépend du type de problème que vous essayez de résoudre et des données dont vous disposez [30].
5. Entraîner le modèle : Vous utiliserez les données et les algorithmes pour entraîner votre modèle d'IA. Il s'agit d'introduire les données dans le modèle et d'ajuster les paramètres des algorithmes jusqu'à ce que le modèle prédise avec précision les résultats [30].
6. Évaluation du modèle : Une fois notre modèle formé, nous devons évaluer ses performances sur de nouvelles données. Cela implique de tester le modèle sur un ensemble distinct de données qu'il n'a jamais vu auparavant [30].
7. Déployer le modèle : Si le modèle fonctionne bien sur les données de test, vous pouvez le déployer en production. Il s'agit de l'intégrer dans vos systèmes existants et de le rendre utilisable.

II.3 Les Réseaux Neuronaux

II.3.1 Définition

Un réseau de neurones est un modèle computationnel inspiré par la manière dont les réseaux neuronaux biologiques du cerveau humain traitent l'information. Il se compose de nœuds interconnectés, ou "neurones", qui travaillent ensemble pour résoudre des problèmes spécifiques ou reconnaître des motifs. Chaque connexion, ou "synapse", peut transmettre un signal d'un neurone à un autre, et ces signaux sont généralement ajustés en fonction de leur force grâce à un processus d'apprentissage. Les réseaux de neurones sont largement utilisés en apprentissage automatique et en intelligence artificielle pour des tâches telles que la classification, la régression et le regroupement [25].

II.3.2 Le Perceptron

Le type le plus élémentaire de neurone artificiel est appelé perceptron. Les perceptrons sont constitués d'un certain nombre de liens d'entrée externes, d'un seuil et d'un seul lien de sortie externe. En outre, les perceptrons ont une entrée interne, b , appelée biais. Le perceptron prend un vecteur de valeurs d'entrée réelles, toutes pondérées par un multiplicateur. Au cours d'une phase d'apprentissage précédente, le perceptron apprend ces poids sur la base de données d'apprentissage. Il additionne toutes les valeurs d'entrée pondérées et « déclenche » si la valeur résultante est supérieure à un seuil prédéfini [31].

$$\begin{cases} 1 & \text{si } \sum_{i=1}^n W_i x_i + b > \text{la seuil} \\ -1 & \text{sinon.} \end{cases}$$

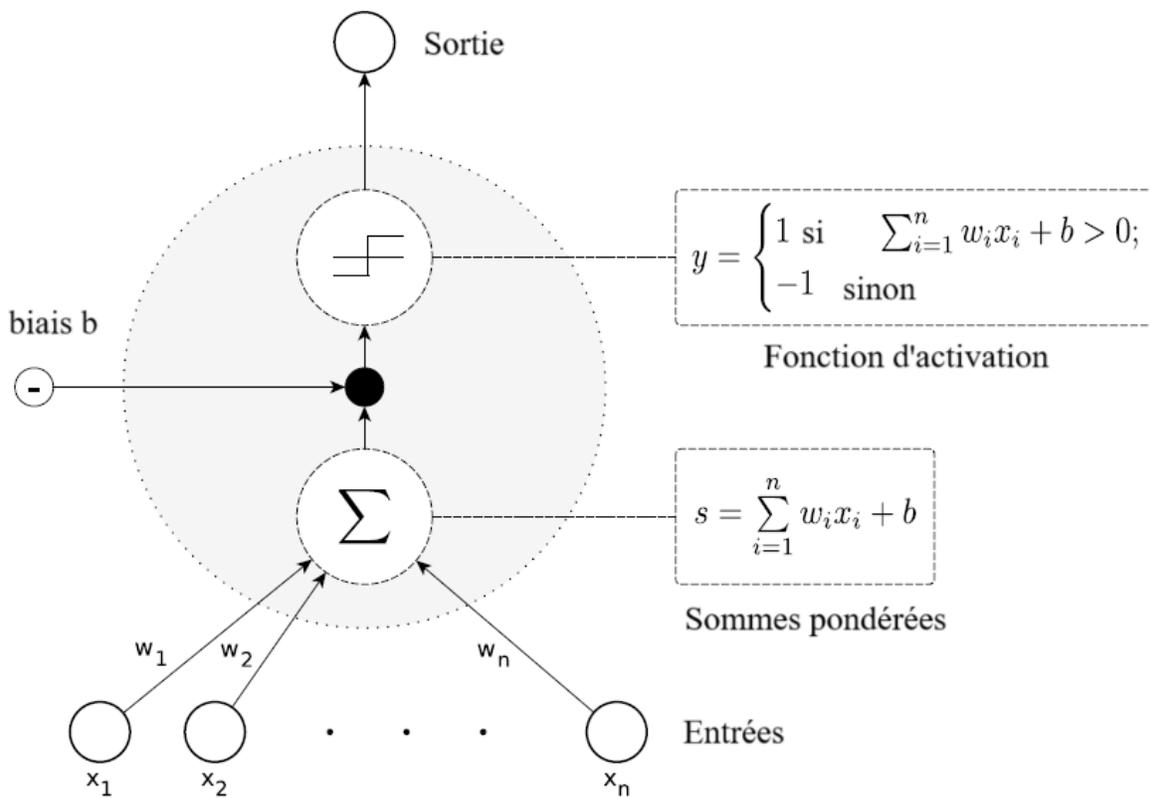


Figure II.1: La structure Générale d'un perceptron.

II.3.3 Réseau de neurones à propagation directe (Feed-forward)

Dans les réseaux neuronaux à propagation directe (FFNN), des ensembles de neurones sont organisés en couches, où chaque neurone calcule une somme pondérée de ses entrées. Les neurones d'entrée reçoivent des signaux de l'environnement et les neurones de sortie présentent des signaux à l'environnement. Les neurones qui ne sont pas directement connectés à l'environnement, mais qui sont connectés à d'autres neurones, sont appelés neurones cachés. Les réseaux neuronaux à progression directe ne comportent pas de boucle et sont entièrement connectés. Cela signifie que chaque neurone fournit une entrée à chaque neurone de la couche suivante et qu'aucun des poids ne fournit une entrée à un neurone d'une couche précédente. Le type le plus simple de réseaux neuronaux de type feed-forward est le réseau de perceptron à une couche. Les réseaux neuronaux à une couche se composent d'un ensemble de neurones d'entrée, appelé couche d'entrée, et d'un ensemble de neurones de sortie, appelé couche de sortie. Les sorties des neurones de la couche d'entrée sont directement connectées aux neurones de la couche de sortie. Les poids sont appliqués aux connexions entre la couche d'entrée et la couche de sortie [31].

Des ensembles de neurones organisés en plusieurs couches peuvent former des réseaux multicouches à connexion directe. Les couches d'entrée et de sortie sont connectées par l'intermédiaire d'au moins une couche cachée, construite à partir d'un ou plusieurs ensembles de neurones cachés.

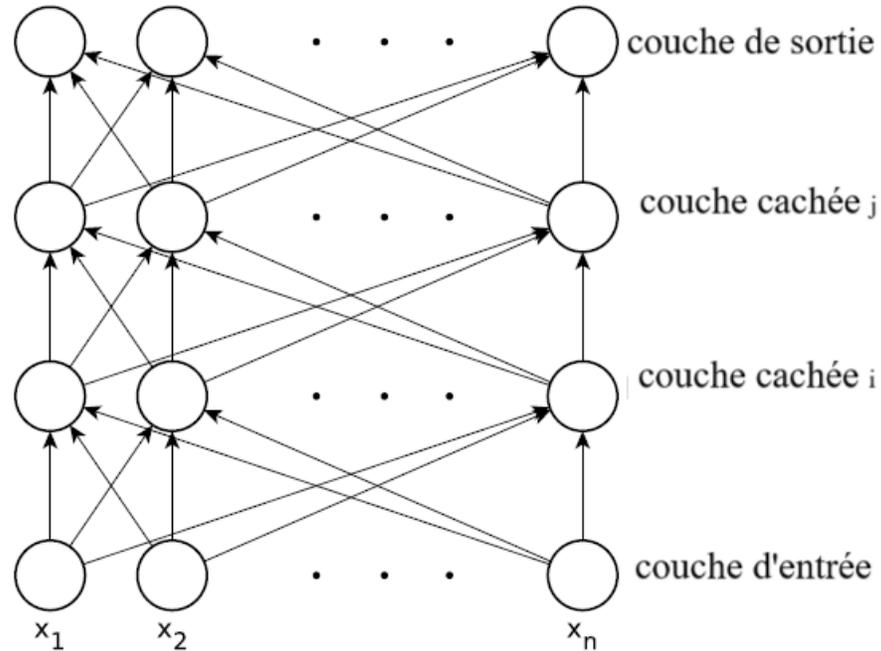


Figure II.2: Un réseau neuronal multicouche de type feed-forward avec une couche d'entrée, deux couches cachées et une couche de sortie.

II.3.4 Réseaux neuronaux récurrents (RNN)

Les réseaux neuronaux récurrents (RNN) sont un type de réseau neuronal artificiel conçu pour traiter des données séquentielles. Contrairement aux réseaux neuronaux traditionnels, les RNN ont des boucles dans leur architecture, ce qui permet aux informations d'être transmises d'une étape à l'autre, ce qui les rend particulièrement efficaces pour les tâches impliquant des données séquentielles telles que la modélisation du langage et la prédiction de séries temporelles [32]. Ils maintiennent un état caché qui capture les informations des étapes temporelles précédentes, mettant à jour cet état caché avec chaque nouvelle entrée grâce à des poids partagés entre les étapes temporelles. Malgré leurs avantages, les RNN sont confrontés à des défis tels que l'évanouissement et l'explosion des gradients, qui rendent l'apprentissage difficile sur de longues séquences, ce qui peut être essentiel pour des prédictions de température précises [33].

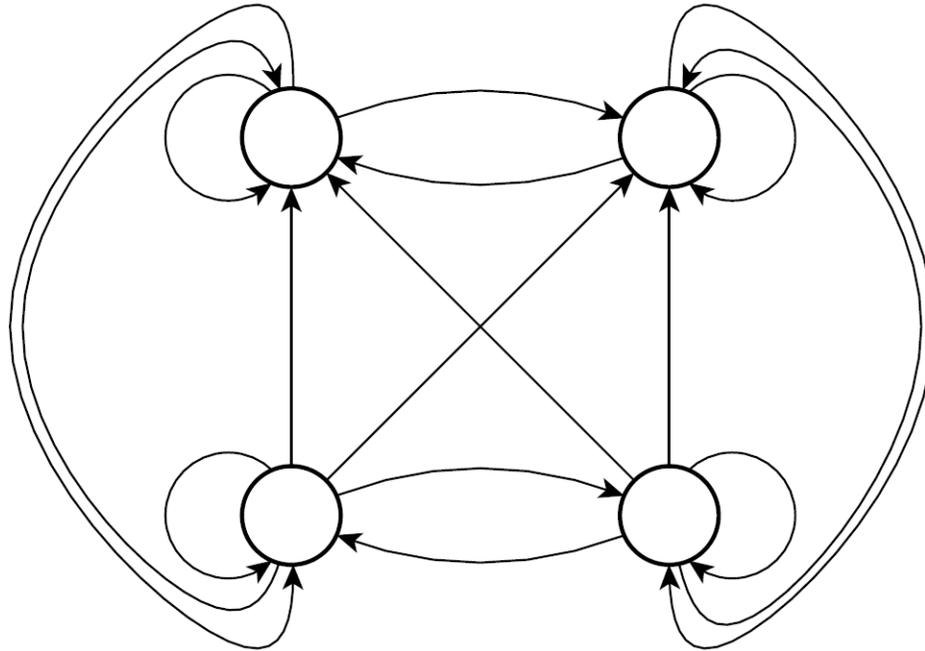


Figure II.3: Réseau neuronal récurrent (RNN) avec auto-feedback.

II.4 L'Apprentissage d'un réseau de neurones

L'apprentissage d'un réseau de neurones est un processus fondamental dans le domaine de l'apprentissage automatique et de l'intelligence artificielle. Il consiste à ajuster les poids du modèle en fonction des données traitées, dans le but de minimiser les erreurs de prédiction et d'améliorer la précision.

II.4.1 Fondamentaux de l'Entraînement des Modèles

L'entraînement d'un modèle commence par l'initialisation des poids dans le réseau de neurones. L'objectif principal est de trouver l'ensemble optimal de poids qui minimise l'erreur entre les sorties prédites et les valeurs cibles réelles. Cela est généralement réalisé par un processus itératif appelé rétropropagation, qui utilise l'algorithme du gradient pour ajuster les poids en fonction du gradient d'erreur. L'entraînement d'un réseau de neurones implique l'utilisation de l'algorithme de rétropropagation pour rendre l'algorithme du gradient réalisable pour l'optimisation des poids [34].

La rétropropagation calcule le gradient de la fonction de perte par rapport à chaque poids par la règle de chaîne, en mettant à jour itérativement les poids pour réduire la perte.

II.4.2 Fonction de Perte (Loss Function)

La fonction de perte mesure la différence entre les sorties prédites et les valeurs réelles. Les fonctions de perte courantes incluent l'erreur quadratique moyenne (MSE) pour les tâches de régression et l'entropie croisée pour les tâches de classification.

II.4.3 Algorithmes d'Optimisation

Les algorithmes d'optimisation tels que l'algorithme du gradient stochastique (SGD), Adam et RMSprop sont utilisés pour minimiser la fonction de perte en ajustant les poids du réseau.

L'algorithme du gradient est l'un des algorithmes d'optimisation les plus fondamentaux et les plus couramment utilisés. Son objectif principal est de minimiser une fonction de perte en ajustant itérativement les poids du modèle, en calculant le gradient (ou la dérivée) de la fonction de perte par rapport aux poids du modèle, en utilisant ces gradients, les poids sont mis à jour dans la direction opposée au gradient pour réduire la fonction de perte. Cette mise à jour se fait selon la formule suivante :

$$\omega = \omega - \eta \cdot \nabla L(\omega)$$

Où ω représente les poids, η est le taux d'apprentissage et $\nabla L(\omega)$ est le gradient de la fonction de perte L par rapport aux poids.

II.4.4 Défis dans l'Entraînement des Modèles

L'entraînement des réseaux de neurones présente plusieurs défis :

- Surapprentissage (Overfitting) : Lorsqu'un modèle performe bien sur les données d'entraînement mais mal sur les données non vues, on parle de surapprentissage. Des techniques telles que la régularisation, le dropout et l'arrêt précoce peuvent atténuer le surapprentissage.

- Gradients Évanescents/Explosifs : Dans les réseaux profonds bouclés, les gradients peuvent devenir extrêmement petits ou grands, ce qui entrave l'entraînement efficace. L'utilisation de fonctions d'activation comme ReLU peuvent résoudre ces problèmes.
- Ressources Computationnelles : L'entraînement des réseaux de neurones profonds peut être intensif en ressources computationnelles. L'utilisation des GPU et du calcul distribué peut accélérer le processus d'entraînement.

L'entraînement d'un réseau de neurones est difficile en raison de la complexité de trouver l'ensemble optimal de poids [35].

II.5 Réseaux de mémoire à long terme (LSTM)

Les réseaux à mémoire à long terme (LSTM) sont un type de réseau neuronal récurrent (RNN) capable d'apprendre des dépendances à long terme. Ils ont été introduits par Hochreiter et Schmidhuber en 1997 pour remédier aux limites des RNN traditionnels, en particulier leur incapacité à mémoriser efficacement des informations sur de longues séquences [33].

II.5.1 Pourquoi les LSTM ?

Les RNN traditionnels souffrent du problème de la disparition du gradient, ce qui les rend inefficaces pour capturer les dépendances à long terme dans les séquences, ce qui les rend idéaux pour prédire les changements environnementaux tels que la température et l'humidité. Les LSTM atténuent ce problème en incorporant une architecture spéciale qui leur permet de conserver des informations sur des périodes prolongées [33].

II.5.2 Architecture des LSTM

L'idée centrale des LSTM est l'état de la cellule, qui agit comme une courroie de transmission, transportant des informations à travers de nombreux pas de temps. Les LSTM ont une structure en chaîne, mais contrairement aux RNN standard, chaque module répétitif comporte quatre couches en interaction au lieu d'une seule.

II.5.3 Composants d'une cellule LSTM

1. État de la cellule (C_t) : Représente la mémoire du réseau.
2. État caché (h_t) : Sortie de la cellule LSTM à un pas de temps donné.
3. Porte d'entrée (i_t) : Détermine la quantité de nouvelles informations à ajouter à l'état de la cellule.
4. Porte de l'oubli (f_t) : Décide quelles informations doivent être éliminées de l'état de la cellule.
5. Porte de sortie (o_t) : Contrôle la partie de l'état de la cellule qui doit être affichée.

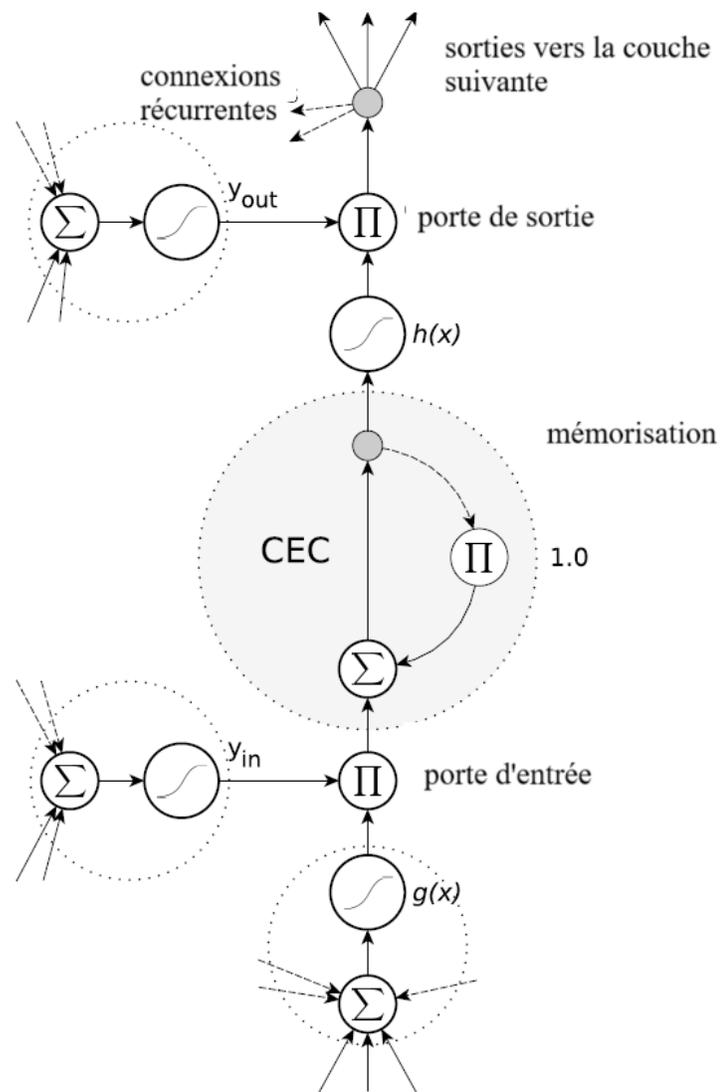


Figure II.4: Une cellule de mémoire LSTM standard.

II.5.4 Applications des LSTM

Les LSTM sont largement utilisées dans divers domaines, notamment :

1. *Traitement du langage naturel (NLP) :*

- Modélisation du langage : Prédiction du mot suivant dans une séquence.
- Traduction automatique : Traduction de textes d'une langue à une autre.
- Génération de texte : Création d'un texte de type humain à partir de séquences d'entrée [36].

2. *Prédiction de séries temporelles :*

- Prédiction du marché boursier : Préviation des prix des actions sur la base de données historiques
- Préviation météorologique : Préviation des conditions météorologiques à l'aide des données météorologiques passées [37].

3. *Reconnaissance de la parole :*

- Assistants vocaux : Reconnaissance et traitement du langage parlé pour aider les utilisateurs [38].

II.6 Utilisation des réseaux neuronaux et de l'intelligence artificielle dans les couveuses d'œufs

1. *Taux d'éclosion améliorés :*

- Conditions environnementales optimisées : Les systèmes d'IA veillent à ce que l'incubateur conserve en permanence une température, une humidité et une ventilation optimales. Ce contrôle précis crée un environnement stable qui est crucial pour le développement des embryons, réduisant le risque de problèmes de développement qui peuvent survenir en raison des fluctuations.
- Apprentissage adaptatif : L'IA peut tirer des enseignements des cycles d'incubation précédents et ajuster les paramètres pour les lots suivants afin d'améliorer les résultats. Ce processus d'apprentissage continu permet à l'IA d'affiner les conditions pour différents types d'œufs, ce qui améliore les taux d'éclosion.

- Détection précoce des problèmes : Les systèmes d'IA peuvent détecter et corriger les problèmes potentiels (p. ex. les hausses ou les baisses de température) avant qu'ils n'aient un impact négatif sur les œufs, ce qui augmente les chances de réussite de l'éclosion.

2. *Efficacité énergétique :*

- Gestion intelligente de l'énergie : L'IA peut optimiser l'utilisation des systèmes de chauffage, de refroidissement et de ventilation, en veillant à ce qu'ils ne fonctionnent que lorsque c'est nécessaire. Cette gestion intelligente réduit le gaspillage d'énergie, ce qui permet de réaliser d'importantes économies d'énergie.
- Prévion de la charge : L'IA peut prédire les besoins en énergie sur la base des données historiques et des conditions actuelles, ce qui permet une utilisation plus efficace de l'énergie. En anticipant le moment où la consommation d'énergie atteindra son maximum, le système peut ajuster les paramètres pour lisser la demande.
- Ajustements dynamiques : L'IA peut procéder à des ajustements en temps réel de l'environnement de l'incubateur, en utilisant la quantité minimale d'énergie nécessaire pour maintenir des conditions optimales. Il s'agit notamment d'ajuster le fonctionnement des ventilateurs, des chauffages et des humidificateurs en fonction des besoins en temps réel.

3. *Collecte et analyse des données :*

- Surveillance complète : Les systèmes d'IA recueillent de nombreuses données sur la température, l'humidité, le retournement des œufs et d'autres paramètres critiques tout au long de la période d'incubation. Ces données fournissent un aperçu détaillé de l'environnement d'incubation et peuvent être utilisées pour identifier les tendances et les anomalies.
- Perspectives prédictives : L'analyse des données collectées permet d'identifier les modèles qui sont en corrélation avec les éclosions réussies. Ces informations peuvent être utilisées pour prédire les résultats et optimiser le processus d'incubation pour les prochains lots.
- Amélioration continue : En analysant les données historiques, l'IA peut améliorer en permanence les protocoles d'incubation. Elle peut suggérer des changements de

température et d'humidité, des horaires de retournement et d'autres paramètres pour améliorer les taux d'éclosion.

4. *Surveillance des œufs :*

- Automatisation du mirage : Les méthodes traditionnelles de mirage consistent à inspecter manuellement les œufs à l'aide d'une lampe pour vérifier leur développement. L'analyse d'images alimentée par l'IA peut automatiser ce processus, en utilisant des caméras et des algorithmes sophistiqués pour surveiller le développement des embryons à l'intérieur des œufs.
- Suivi du développement : L'IA peut suivre et analyser des images d'ovules pour surveiller leurs stades de développement. Cela permet de détecter rapidement des problèmes tels que des œufs non viables ou des problèmes de développement, ce qui permet d'intervenir à temps.
- Prédiction de l'éclosion : En analysant les données visuelles, l'IA peut prédire le moment de l'éclosion de chaque œuf, ce qui permet une meilleure préparation et une meilleure allocation des ressources. Cela permet de gérer plus efficacement le processus d'éclosion.
- Évaluation de la santé : L'analyse d'images par l'IA permet d'évaluer la santé des embryons en détectant les signes de détresse ou de développement anormal. Cela peut conduire à des interventions précoces qui peuvent sauver les embryons à risque.

Conclusion

Dans ce chapitre, nous avons offert une vue d'ensemble approfondie de l'intelligence artificielle (IA) et des réseaux de neurones, en mettant particulièrement l'accent sur les réseaux de neurones à mémoire à long court terme (LSTM). Nous avons démontré comment ces technologies avancées peuvent transformer le domaine de l'aviculture, notamment par l'optimisation des processus d'incubation.

En exposant les principes fondamentaux de l'IA et des réseaux de neurones, ce chapitre met en lumière la capacité des LSTM à prédire et contrôler avec précision les paramètres critiques de l'incubation. Contrairement aux RNN classiques, les LSTM sont spécialement conçus pour retenir des informations sur des périodes plus longues, ce qui les rend particulièrement aptes à traiter les

séquences de données complexes et les dépendances à long terme caractéristiques des cycles d'incubation.

Ces avancées promettent non seulement d'améliorer les taux d'éclosion, mais aussi d'augmenter l'efficacité énergétique et de faciliter une collecte et une analyse de données plus sophistiquées. En outre, autre type de l'IA permettent une surveillance innovante des œufs via l'analyse d'images, fournissant des évaluations précises du développement embryonnaire et détectant les anomalies à un stade précoce.

En somme, l'intégration des LSTM dans l'incubation avicole représente une évolution significative vers des pratiques plus intelligentes et optimisées, ouvrant la voie à des progrès considérables dans l'industrie.

Chapitre III Conception de la couveuse

Introduction

La conception et la fabrication d'un incubateur d'œufs représentent une intersection fascinante entre la biologie, la technologie et l'innovation agricole, visant à améliorer l'efficacité et le taux de réussite de l'éclosion des œufs. Cette entreprise nécessite une compréhension approfondie de la température, de l'humidité et de la ventilation précises. Le processus de conception et de construction d'une couveuse efficace implique un mélange d'ingénierie mécanique, de science des matériaux et de principes biologiques. Les éléments clés à prendre en compte sont la sélection de matériaux appropriés pour assurer l'isolation thermique, la mise en œuvre de systèmes fiables de chauffage et de contrôle de l'humidité, et l'incorporation de mécanismes permettant de retourner régulièrement les œufs pour éviter que les embryons ne collent à la coquille.

En nous lançant dans notre projet de conception et de construction d'une couveuse, nous visons à intégrer ces concepts fondamentaux aux avancées technologiques contemporaines afin de créer un système hautement efficace et fiable. Notre projet s'appuiera sur une planification et une recherche détaillée pour sélectionner les matériaux et les composants optimaux qui garantissent une bonne isolation thermique et un contrôle précis de l'environnement. En incorporant la technologie numérique moderne, nous mettrons en place des capteurs avancés et des contrôles automatisés qui permettront de surveiller et d'ajuster en temps réel la température, l'humidité et la rotation des œufs.

III.1 Partie matériel

I.1.1 Enclos de couveuse

Nous avons recyclé un réfrigérateur comme enclos pour la couveuse, en tirant parti de ses propriétés d'isolation thermique inhérentes. L'isolation robuste du réfrigérateur réduit les fluctuations de température et d'humidité. Le réfrigérateur est similaire à celui de la figure III.1



Figure III.1: Réfrigérateur.

III.1.2 Capteurs

III.1.2.1 Capteur de température

Le DS18B20 est un capteur de température numérique réputé pour sa précision et sa facilité d'utilisation dans diverses applications de détection de température. Une caractéristique notable est sa capacité à fonctionner sur une large plage de températures tout en maintenant une haute précision. De plus, il communique via une interface 1-Wire simple, ce qui permet de connecter plusieurs capteurs à une seule broche de microcontrôleur. Ces qualités font du DS18B20 un choix privilégié pour la surveillance de la température dans divers domaines, de l'automatisation industrielle aux projets de bricolage. Le capteur est représenté dans la Figure III.2.



Figure III.2: Capteur de température DS18B20.

III.1.2.2 Capteur d'humidité

Le BME280 est un capteur environnemental très polyvalent. Il combine les fonctionnalités de mesure de la pression atmosphérique, de la température et de l'humidité dans un seul petit boîtier. Une des caractéristiques les plus remarquables du BME280 est sa haute précision et sa stabilité sur une large plage de températures et d'humidités. Il offre également une résolution élevée pour chacune des mesures qu'il effectue. Le capteur est représenté dans la Figure III.3.

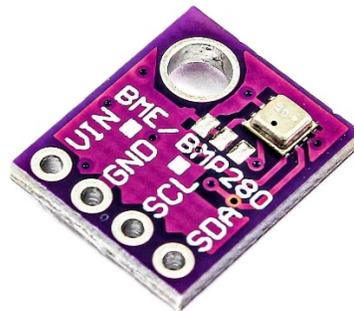


Figure III.3: Capteur BME280.

III.1.2.3 Circuit de conditionnement :

Nous avons rencontré un problème important où les lectures des capteurs n'étaient pas fiables en raison des interférences des lignes électriques des composants voisins. Ces interférences introduisaient du bruit dans les signaux des capteurs, ce qui rendait difficile l'obtention de mesures

précises. Le bruit électrique des lignes d'alimentation déformait les sorties des capteurs, ce qui se traduisait par des données erratiques et incohérentes. Ce problème a mis en évidence la nécessité d'un conditionnement efficace des signaux pour filtrer le bruit et stabiliser les relevés des capteurs.

Les circuits de conditionnement sont essentiels dans les systèmes de capteurs car ils garantissent la précision, la fiabilité et la facilité d'utilisation des données des capteurs en effectuant des tâches telles que l'amplification, le filtrage du bruit et la conversion analogique-numérique. Ces circuits améliorent les signaux bruts des capteurs, les rendant robustes et interprétables pour un traitement et une analyse ultérieure. Sans un conditionnement adéquat du signal, les données acquises par les capteurs peuvent être faussées par le bruit, la non-linéarité ou les décalages, ce qui entraîne des mesures erronées et des décisions potentiellement erronées basées sur ces données [39].

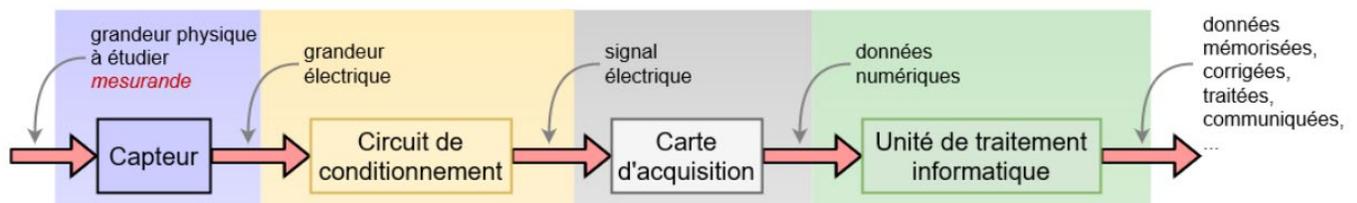


Figure III.4: Chaîne de mesure.

III.1.3 Actionneurs

III.1.3.1 Élément chauffant

Pour le réglage de la température, nous avons utilisé deux résistances chauffantes de 150W pour un contrôle efficace et suffisant. La résistance utilisée est représentée dans la figure III.5



Figure III.5: Elément chauffant.

III.1.3.2 Humidificateur

Afin de maintenir l'humidité idéale dans l'environnement, nous avons utilisé un dispositif submersible (figure III.6) qui libère l'humidité dans l'air à partir d'un réservoir d'eau, pour éviter d'ouvrir la porte pour le remplissage d'eau, nous avons ajouté un réservoir secondaire à l'extérieur de la couveuse qui se connecte au réservoir intérieur.



Figure III.6: Le dispositif d'humidification.

III.1.3.3 Ventilateur de distribution

Pour assurer une répartition uniforme de la température et de l'humidité et éviter les points chauds, nous avons mis en place un ventilateur qui assure un flux d'air adéquat à l'intérieur.



Figure III.7: Ventilateur de distribution.

III.1.3.4 Ventilateurs de régulation et de renouvellement d'air

Pour assurer des conditions optimales de faible taux de CO₂ et de taux élevé d'oxygène, ainsi que pour réduire la chaleur et l'humidité excessives, nous avons installé deux ventilateurs d'extraction et d'admission d'air qui sont représentés dans la figure III.8.

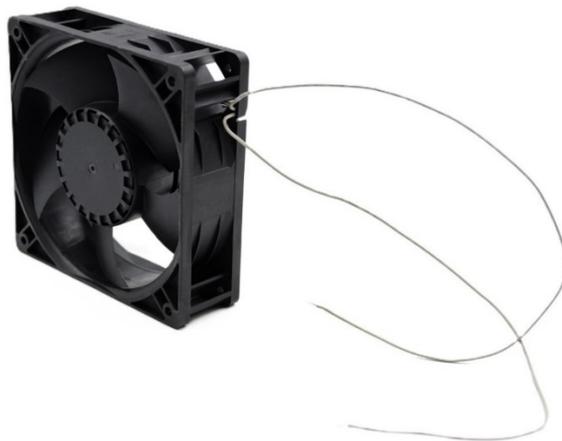


Figure III.8: Ventilateur d'extraction/admission.

III.1.3.5 Système de retournement et portage d'œufs :

Afin d'assurer une rotation constante et régulière des œufs. Le système comprend trois plateaux reliés à une tige commune, fixée sur les bords des plateaux. Cette tige permet aux plateaux de se déplacer ensemble, en tournant autour d'un axe central situé au milieu de chaque plateau (figure III.9). Un vérin électrique (figure III.10) est monté solidement et aligné pour pousser ou tirer la tige, ce qui fait basculer les plateaux de 45 degrés dans l'une ou l'autre direction.



Figure III.9: Système de retournement.



Figure III.10: Vérin électrique.

III.1.4 Relais

Les relais agissent comme des pré actionneurs, en gérant des composants de grande puissance et en automatisant des processus essentiels. Ils agissent comme des interrupteurs contrôlés par des signaux de faible puissance provenant du microcontrôleur.



Figure III.11: Module de relais à 6 canaux.

III.1.5 Energie solaire

L'énergie solaire est un excellent choix pour alimenter les couveuses en raison de sa durabilité, de sa rentabilité et de sa fiabilité. L'utilisation de l'énergie solaire réduit la dépendance à l'égard

des sources d'énergie non renouvelables, ce qui diminue les émissions de carbone et favorise la préservation de l'environnement. En outre, dans les zones rurales ou isolées où l'accès à l'électricité est limité, les couveuses solaires offrent une alternative fiable, garantissant une alimentation électrique constante et des conditions d'incubation optimales. Cette approche permet également de minimiser les coûts opérationnels au fil du temps, car l'énergie solaire est gratuite une fois l'installation initiale terminée [40].

III.1.5.1 Panneau solaire

Un panneau solaire est un dispositif composé de nombreuses cellules photovoltaïques (PV), conçues pour convertir directement la lumière du soleil en électricité. Ces cellules sont généralement constituées de silicium et fonctionnent en absorbant les photons de la lumière du soleil, qui déplacent ensuite les électrons dans le silicium, créant ainsi un courant électrique.



Figure III.12: Panneau solaire.

III.1.5.2 Onduleur

Un onduleur est un composant essentiel d'un système d'énergie solaire. Il convertit le courant continu (DC) généré par les panneaux solaires en courant alternatif (AC), qui est la norme utilisée par la plupart des appareils ménagers et le réseau électrique. Les onduleurs jouent un rôle crucial en garantissant la compatibilité et l'utilisation de l'énergie solaire dans les applications quotidiennes. Outre la conversion, les onduleurs remplissent également des fonctions importantes telles que la maximisation de la puissance de sortie grâce au suivi du point de puissance maximale

(MPPT), la surveillance des performances du système et la garantie de la sécurité en arrêtant le système en cas de défaillance [41].



Figure III.13: Onduleur.

III.1.5.3 Régulateur de Tension

Un régulateur de tension est un dispositif essentiel dans les systèmes électriques et électroniques, conçu pour maintenir une tension de sortie constante malgré les variations de la tension d'entrée ou des conditions de charge. Dans les systèmes d'énergie solaire, les régulateurs de tension garantissent que l'électricité générée par les panneaux solaires reste stable et dans la plage requise pour un fonctionnement sûr et efficace des appareils connectés. Ils protègent contre les fluctuations de tension qui pourraient potentiellement endommager les batteries, les onduleurs et d'autres composants. En fournissant des niveaux de tension constants, les régulateurs de tension améliorent la fiabilité et la longévité des systèmes d'énergie solaire et d'autres appareils électroniques [42].



Figure III.14: Régulateur de tension.

III.1.5.4 Batterie

Les batteries sont des dispositifs fondamentaux de stockage de l'énergie utilisés dans diverses applications, y compris les systèmes d'énergie solaire. Elles stockent l'énergie électrique sous forme chimique et la restituent en fonction des besoins. Dans les installations d'énergie solaire, les batteries jouent un rôle crucial en stockant l'électricité excédentaire produite pendant les périodes d'ensoleillement maximal pour l'utiliser lorsque la lumière du soleil n'est pas disponible, par exemple pendant la nuit ou par temps nuageux. Cela permet une alimentation électrique continue et améliore la fiabilité des systèmes d'énergie solaire, en particulier dans les installations hors réseau ou hybrides [43].



Figure III.15: Batterie au plomb.

III.1.6 Disjoncteur de protection

Un disjoncteur magnétothermique est un composant électrique essentiel utilisé dans divers systèmes pour se prémunir contre les situations de surintensité. Combinant les mécanismes de protection thermique et magnétique, il réagit aux surintensités prolongées en déclenchant une interruption du circuit par la flexion d'un bilame sous l'effet de la chaleur générée. En outre, il réagit rapidement aux augmentations massives et soudaines du courant, caractéristiques des scénarios de court-circuit, en activant un mécanisme de déclenchement magnétique qui ouvre rapidement l'interrupteur. Largement utilisés dans les applications électriques, les interrupteurs magnétiques thermiques offrent une protection solide contre les surcharges soutenues et les courts-circuits, garantissant ainsi la sécurité et l'intégrité des systèmes électrique [44].



Figure III.16: Disjoncteur magnétothermique bipolaire.

III.2 Partie programme

III.2.1 Raspberry Pi

La Raspberry Pi est une série de petits ordinateurs à carte unique abordables développés par la Fondation Raspberry Pi, dans le but de promouvoir l'éducation en informatique et de faciliter les projets DIY (Do It Yourself) en électronique et en programmation (Upton & Halfacree, 2012). Ces ordinateurs de la taille d'une carte de crédit sont dotés de différents modèles avec des spécifications variées, incluant des processeurs, de la RAM, des options de stockage et des interfaces de connectivité différentes. Les appareils Raspberry Pi fonctionnent sous des systèmes d'exploitation

basés sur Linux et prennent en charge un large éventail de langages de programmation, ce qui en fait des plates-formes polyvalentes pour des applications éducatives, de loisirs et commerciales [45]. Ces ordinateurs de la taille d'une petite carte sont dotés de différents modèles avec des spécifications variées, incluant des processeurs, de la RAM, des options de stockage et des interfaces de connectivité différentes. Les appareils Raspberry Pi fonctionnent sous des systèmes d'exploitation basés sur Linux et prennent en charge un large éventail de langages de programmation, ce qui en fait des plates-formes polyvalentes pour des applications éducatives, de loisirs et commerciales.

III.2.1.1 Types de Raspberry Pi

Depuis son lancement en 2012, la fondation Raspberry Pi a introduit plusieurs itérations de la carte, chacune répondant à des besoins utilisateur spécifiques. Ces itérations comprennent :

Raspberry Pi 1 Modèles A/B/A+/B+ : Les modèles inauguraux ayant lancé la révolution Raspberry Pi.

Raspberry Pi 2 Modèle B : Introduit une puissance de traitement et une RAM améliorées.

Raspberry Pi 3 Modèle B/B+ : Intègre les fonctionnalités Wi-Fi et Bluetooth.

Raspberry Pi 4 Modèle B : Le modèle utilisé dans notre projet, offrant des améliorations significatives de performance, des configurations de RAM multiples et la prise en charge de la sortie double écran.



Figure III.17: Raspberry Pi 4 model B.

III.2.1.2 Fonctionnement de Raspberry Pi

Le Raspberry Pi, en tant qu'ordinateur à faible coût et à faible consommation d'énergie, possède des composants tels qu'un processeur ARM, de la mémoire RAM, et des interfaces d'entrée/sortie variées. Les utilisateurs peuvent connecter des périphériques comme des claviers et des souris, et utiliser les broches GPIO pour interagir avec des capteurs et des dispositifs externes, facilitant ainsi la réalisation de projets de calcul physique et de prototypage [46].

III.2.2 Langage de programmation

Python est largement considéré comme la meilleure option pour la programmation et l'intelligence artificielle (IA), et ce pour plusieurs raisons convaincantes. Il dispose d'un écosystème complet de bibliothèques, y compris des bibliothèques populaires telles que TensorFlow, Keras et PyTorch, qui facilitent le développement efficace de modèles d'IA. La syntaxe de Python est claire et lisible, ce qui réduit la barrière d'entrée pour les nouveaux programmeurs et permet des cycles de développement plus rapides. En outre, Python offre d'excellents outils de visualisation tels que Matplotlib et Seaborn, qui sont essentiels pour présenter les données de manière compréhensible. Le soutien solide de la communauté garantit une amélioration continue et une assistance aux développeurs. En outre, Python est indépendant de la plateforme et flexible, ce qui permet une intégration transparente avec d'autres langages de programmation et systèmes, ce qui en fait un choix idéal pour la création d'applications d'IA robustes [47].

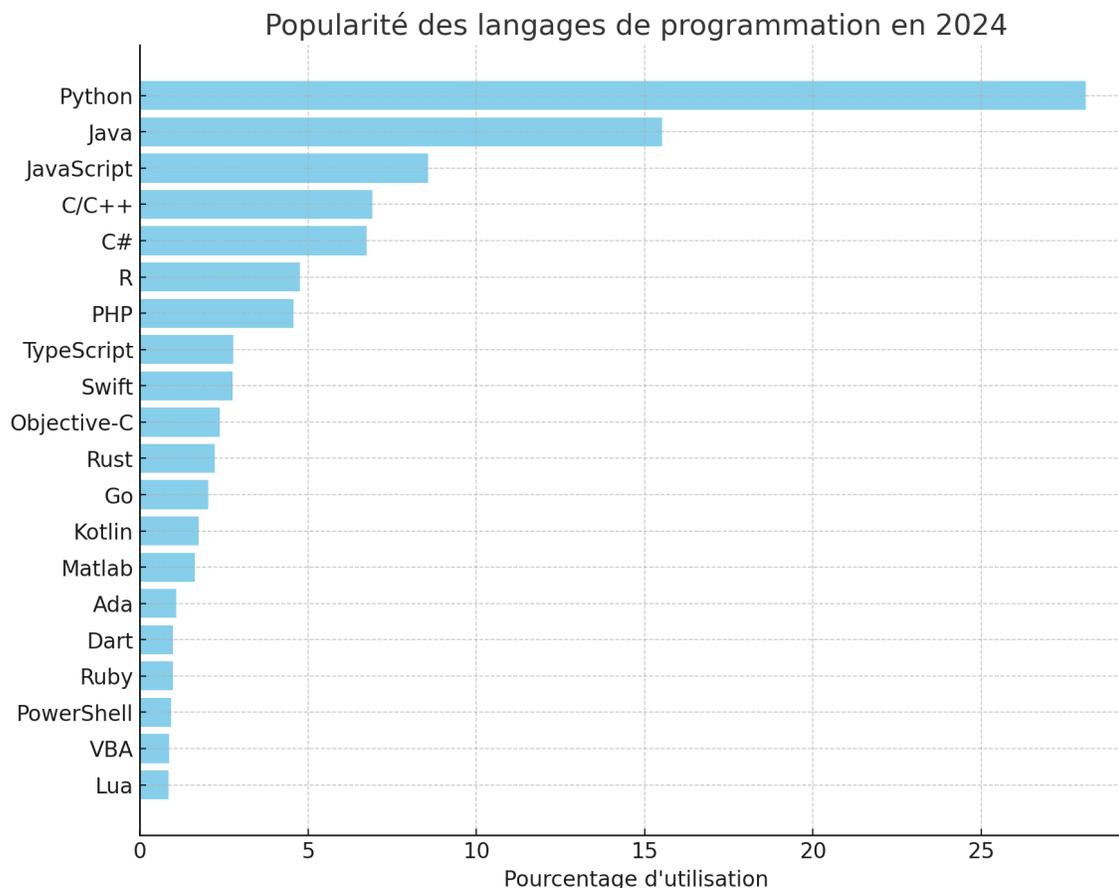


Figure III.18: Popularité des langages de programmation en 2024. [48]

Conclusion

Dans ce chapitre, nous avons détaillé la construction d'un incubateur d'œufs utilisant un réfrigérateur pour l'isolation thermique et des capteurs de haute qualité pour un contrôle robuste. Un Raspberry Pi a été choisi comme unité de contrôle pour sa versatilité et sa compacité, s'intégrant de manière fluide avec des capteurs et un vérin électrique pour le retournement automatisé des œufs. Le langage Python a été utilisé pour la programmation en raison de sa simplicité et de l'étendue de sa bibliothèque. Cette combinaison a permis de créer un système d'incubation des œufs efficace, fiable et adaptable.

Chapitre IV Mise en œuvre et analyse des résultats

Introduction

Ce chapitre explore le développement et la mise en œuvre de la couveuse, en comparant les performances d'un système traditionnel à celles d'un système amélioré par l'intelligence artificielle (IA). Le projet consiste à construire un prototype d'incubateur physique capable de fournir des conditions environnementales optimales pour l'incubation des œufs.

IV.1 Réalisation et mise en œuvre de la couveuse

IV.1.1 Construction de la couveuse

La construction de l'incubateur d'œufs a impliqué une intégration méticuleuse des composants mentionnés précédemment afin de créer un environnement stable propice à l'éclosion des œufs, le tout géré par un microcontrôleur Raspberry Pi. Nous avons commencé par concevoir un système permettant de maintenir des niveaux de température et d'humidité précis en incorporant l'insolation thermique du réfrigérateur et la haute précision des deux capteurs (température et humidité). En utilisant le Raspberry Pi, nous avons contrôlé un élément chauffant et un humidificateur immergé dans un réservoir d'eau, tous deux régulés par les données en temps réel des capteurs de température et d'humidité. Pour assurer des conditions uniformes dans toute la couveuse, nous avons mis en place un système de circulation d'air avec un ventilateur qui fonctionne en permanence et deux ventilateurs responsables de l'admission et de l'évacuation de l'air. Un mécanisme de rotation des œufs, entraîné par le vérin électrique, a été incorporé pour faire tourner les œufs à intervalles réguliers avec l'aide de fin de course intégrés à l'intérieur du vérin.



Figure IV.1: La photo montre l'intérieur de la couveuse, y compris les composants de contrôle environnemental et le mécanisme de retournement.

En connectant ces composants au Raspberry Pi via des broches GPIO et en utilisant des relais pour les éléments à haute puissance, nous avons pu automatiser l'ensemble du processus.

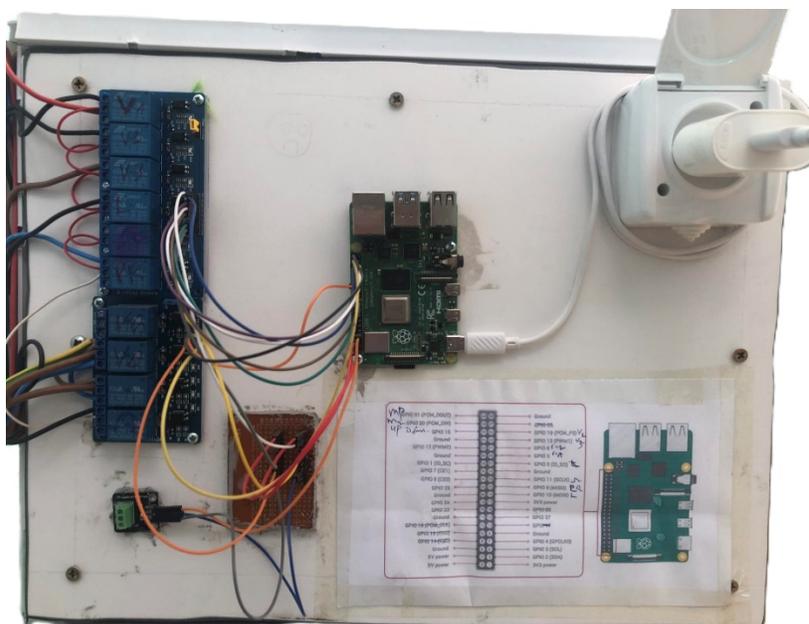


Figure IV.2:La photo montre le circuit de contrôle comprenant le microcontrôleur et les relais reliés par des fils.

A l'arrière, nous avons connecté la ligne électrique qui passe par le disjoncteur bipolaire et qui est ensuite reliée aux différents composants et aux relais par des câbles d'alimentation provenant de la boîte de jonction,



Figure IV.3:La photo montre l'arrière de la couveuse où se trouve le circuit d'alimentation et le réservoir d'eau secondaire.

Une fois toutes ces composantes connectées ensemble, nous avons le prototype fini où tous les composants communiquent entre eux via le raspberry pi en tant qu'unité de contrôle.

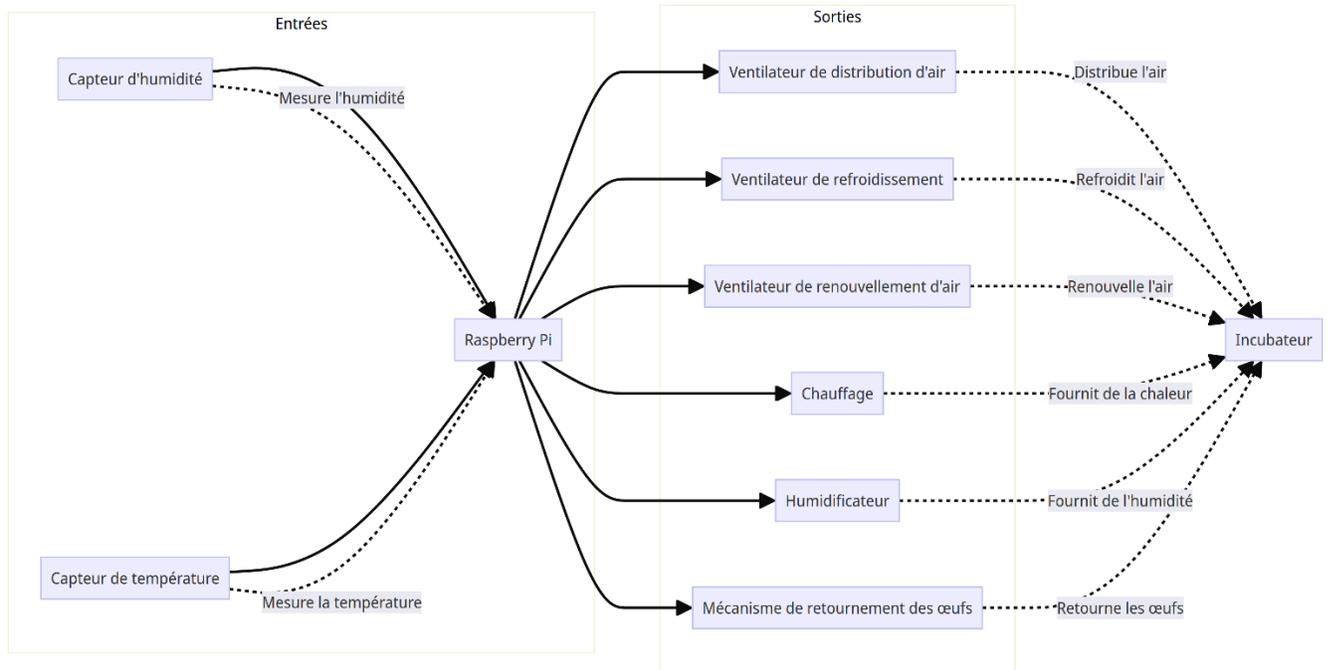


Figure IV.4: Système de Contrôle d'Incubateur Automatisé avec Raspberry Pi.

IV.1.2 Programmation

Ce code permet à l'incubateur de maintenir les valeurs de température et d'humidité souhaitées en lisant les valeurs des capteurs et en contrôlant les éléments, et de retourner les œufs à la fréquence voulue.

```

1. import RPi.GPIO as GPIO
2. import time
3. import multiprocessing
4. import smbus2
5. import bme280
6. from w1thermsensor import W1ThermSensor
7.
8. # Définir les numéros de broche GPIO
9. HEATER_PIN = 1
10. DISTRIBUTION_FAN_PIN = 16
11. HUMIDIFIER_PIN = 9
12. COOLING_FAN_PIN = 19
13. LAMP_PIN = 10
14. AIRFLOW_FAN_PIN = 12
15. MOTOR_UP_PIN = 5
16. MOTOR_DOWN_PIN = 6
17.
18. # Initialiser la configuration GPIO
  
```

```

19. def setup_gpio():
20.     GPIO.setmode(GPIO.BCM) # Utiliser le mode de numérotation BCM
21.     GPIO.setup(HEATER_PIN, GPIO.OUT) # Configurer la broche du chauffage comme sortie
22.     GPIO.setup(DISTRIBUTION_FAN_PIN, GPIO.OUT) # Configurer la broche du ventilateur de
distribution comme sortie
23.     GPIO.setup(HUMIDIFIER_PIN, GPIO.OUT) # Configurer la broche de l'humidificateur comme
sortie
24.     GPIO.setup(COOLING_FAN_PIN, GPIO.OUT) # Configurer la broche du ventilateur de
refroidissement comme sortie
25.     GPIO.setup(LAMP_PIN, GPIO.OUT) # Configurer la broche de la lampe comme sortie
26.     GPIO.setup(AIRFLOW_FAN_PIN, GPIO.OUT) # Configurer la broche du ventilateur de flux d'air
comme sortie
27.     GPIO.setup(MOTOR_UP_PIN, GPIO.OUT) # Configurer la broche de montée du moteur comme sortie
28.     GPIO.setup(MOTOR_DOWN_PIN, GPIO.OUT) # Configurer la broche de descente du moteur comme
sortie
29.
30.     # Définir l'état initial
31.     GPIO.output(HEATER_PIN, GPIO.HIGH) # Chauffage éteint initialement
32.     GPIO.output(DISTRIBUTION_FAN_PIN, GPIO.LOW) # Ventilateur de distribution toujours allumé
33.     GPIO.output(HUMIDIFIER_PIN, GPIO.HIGH) # Humidificateur éteint initialement
34.     GPIO.output(COOLING_FAN_PIN, GPIO.HIGH) # Ventilateur de refroidissement éteint
initialement
35.     GPIO.output(LAMP_PIN, GPIO.LOW) # Lampe éteinte initialement
36.     GPIO.output(AIRFLOW_FAN_PIN, GPIO.HIGH) # Ventilateur de flux d'air éteint initialement
37.     GPIO.output(MOTOR_UP_PIN, GPIO.HIGH) # Moteur éteint initialement
38.     GPIO.output(MOTOR_DOWN_PIN, GPIO.HIGH) # Moteur éteint initialement
39.
40. # Fonction pour lire la température
41. def read_temperature(sensor):
42.     return sensor.get_temperature() # Lire la température du capteur
43.
44. # Fonction pour lire l'humidité
45. def read_humidity(bus, address, calibration_params):
46.     data = bme280.sample(bus, address, calibration_params) # Lire les données du capteur
BME280
47.     return data.humidity # Retourner l'humidité
48.
49. # Fonction pour imprimer la température et l'humidité
50. def print_data(temperature, humidity):
51.     try:
52.         while True:
53.             print(f"Temperature: {temperature.value} C, Humidity: {humidity.value} %")
54.             time.sleep(10) # Attendre 10 seconds
55.     except Exception as e:
56.         print("Error occurred in print_data:", e) # Afficher une erreur si elle se produit
57.
58. # Contrôler le chauffage en fonction de la température avec hystérésis
59. def control_heater(heater_status, temperature):
60.     target_temp = 37.5 # Température cible
61.     hysteresis = 0.2 # Valeur d'hystérésis
62.     try:
63.         while True:
64.             temp_value = temperature.value
65.             if temp_value < target_temp - hysteresis:
66.                 GPIO.output(HEATER_PIN, GPIO.LOW) # Allumer le chauffage
67.                 heater_status.value = 1 # Mettre à jour l'état du chauffage
68.             elif temp_value > target_temp + hysteresis:
69.                 GPIO.output(HEATER_PIN, GPIO.HIGH) # Éteindre le chauffage
70.                 heater_status.value = 0 # Mettre à jour l'état du chauffage
71.                 time.sleep(1) # Attendre 1 seconde
72.     except Exception as e:
73.         print("Error occurred in control_heater:", e) # Afficher une erreur si elle se produit
74.     finally:
75.         GPIO.output(HEATER_PIN, GPIO.HIGH) # Assurer que le chauffage est éteint
76.         heater_status.value = 0 # Mettre à jour l'état du chauffage

```

```

77.
78. # Contrôler l'humidificateur en fonction du niveau d'humidité avec hystérésis
79. def control_steamer(humidifier_status, humidity):
80.     target_humidity = 55 # Humidité cible
81.     hysteresis = 2 # Valeur d'hystérésis
82.     try:
83.         while True:
84.             humidity_value = humidity.value
85.             if humidity_value < target_humidity - hysteresis:
86.                 GPIO.output(HUMIDIFIER_PIN, GPIO.LOW) # Allumer l'humidificateur
87.                 humidifier_status.value = 1 # Mettre à jour l'état de l'humidificateur
88.             elif humidity_value > target_humidity + hysteresis:
89.                 GPIO.output(HUMIDIFIER_PIN, GPIO.HIGH) # Éteindre l'humidificateur
90.                 humidifier_status.value = 0 # Mettre à jour l'état de l'humidificateur
91.                 time.sleep(1) # Attendre 1 seconde
92.     except Exception as e:
93.         print("Error occurred in control_steamer:", e) # Afficher une erreur si elle se
produit
94.     finally:
95.         GPIO.output(HUMIDIFIER_PIN, GPIO.HIGH) # Assurer que l'humidificateur est éteint
96.         humidifier_status.value = 0 # Mettre à jour l'état de l'humidificateur
97.
98. # Contrôler le ventilateur de refroidissement en fonction de la température et de l'humidité
99. def control_cooling_fan(fan_status, temperature, humidity):
100.    try:
101.        while True:
102.            temp_value = temperature.value
103.            humidity_value = humidity.value
104.            # Contrôler le ventilateur en fonction de la température et de l'humidité
105.            if temp_value > 38 ou humidity_value > 58:
106.                GPIO.output(COOLING_FAN_PIN, GPIO.LOW) # Allumer le ventilateur de
refroidissement
107.                fan_status.value = 1 # Mettre à jour l'état du ventilateur
108.            else:
109.                GPIO.output(COOLING_FAN_PIN, GPIO.HIGH) # Éteindre le ventilateur de
refroidissement
110.                fan_status.value = 0 # Mettre à jour l'état du ventilateur
111.                time.sleep(1) # Attendre 1 seconde
112.    except Exception as e:
113.        print("Error occurred in control_cooling_fan:", e) # Afficher une erreur si elle se
produit
114.    finally:
115.        GPIO.output(COOLING_FAN_PIN, GPIO.LOW) # Éteindre le ventilateur de refroidissement
116.        fan_status.value = 0 # Mettre à jour l'état du ventilateur
117.
118. # Contrôler le ventilateur de flux d'air
119. def control_airflow_fan(airflow_fan_status):
120.    try:
121.        while True:
122.            GPIO.output(AIRFLOW_FAN_PIN, GPIO.LOW) # Allumer le ventilateur de flux d'air
123.            airflow_fan_status.value = 1 # Mettre à jour l'état du ventilateur
124.            time.sleep(180) # Ventilateur fonctionne pendant 3 minutes
125.            GPIO.output(AIRFLOW_FAN_PIN, GPIO.HIGH) # Éteindre le ventilateur de flux d'air
126.            airflow_fan_status.value = 0 # Mettre à jour l'état du ventilateur
127.            time.sleep(7020) # Attendre les 117 minutes restantes
128.
129.    except Exception as e:
130.        print("Error occurred in control_airflow_fan:", e) # Afficher une erreur si elle se
produit
131.    finally:
132.        GPIO.output(AIRFLOW_FAN_PIN, GPIO.LOW) # Éteindre le ventilateur de flux d'air
133.        airflow_fan_status.value = 0 # Mettre à jour l'état du ventilateur
134.
135. # Fonction pour faire monter le moteur
136. def turn_motor_up():

```

```

137.     GPIO.output(MOTOR_UP_PIN, GPIO.LOW) # Allumer le moteur vers le haut
138.     GPIO.output(MOTOR_DOWN_PIN, GPIO.HIGH) # Éteindre le moteur vers le bas
139.
140. # Fonction pour faire descendre le moteur
141. def turn_motor_down():
142.     GPIO.output(MOTOR_UP_PIN, GPIO.HIGH) # Éteindre le moteur vers le haut
143.     GPIO.output(MOTOR_DOWN_PIN, GPIO.LOW) # Allumer le moteur vers le bas
144.
145. # Fonction pour arrêter le moteur
146. def stop_motor():
147.     GPIO.output(MOTOR_UP_PIN, GPIO.HIGH) # Éteindre le moteur vers le haut
148.     GPIO.output(MOTOR_DOWN_PIN, GPIO.HIGH) # Éteindre le moteur vers le bas
149.
150. # Contrôler la rotation du moteur
151. def control_motor_rotation():
152.     rotation = 0 # Variable de rotation initiale
153.     try:
154.         while True:
155.             if rotation == 0:
156.                 turn_motor_down() # Faire descendre le moteur
157.                 time.sleep(20) # Attendre 20 secondes
158.                 stop_motor() # Arrêter le moteur
159.                 rotation = 1 # Changer la direction de rotation
160.                 time.sleep(3600) # Attendre 1 heure
161.             elif rotation == 1:
162.                 turn_motor_up() # Faire monter le moteur
163.                 time.sleep(20) # Attendre 20 secondes
164.                 stop_motor() # Arrêter le moteur
165.                 rotation = 0 # Changer la direction de rotation
166.                 time.sleep(3600) # Attendre 1 heure
167.         except Exception as e:
168.             print("Error occurred in control_motor_rotation:", e) # Afficher une erreur si elle se
produit
169.         finally:
170.             stop_motor() # Assurer que le moteur est arrêté
171.
172. # Fonction pour lire les capteurs et mettre à jour les variables partagées
173. def read_sensors(temp_sensor, bus, address, calibration_params, temperature, humidity):
174.     try:
175.         while True:
176.             temperature.value = read_temperature(temp_sensor) # Mettre à jour la température
177.             humidity.value = read_humidity(bus, address, calibration_params) # Mettre à jour
l'humidité
178.             time.sleep(1) # Attendre 1 seconde
179.         except Exception as e:
180.             print("Error occurred in read_sensors:", e) # Afficher une erreur si elle se produit
181.         finally:
182.             bus.close() # Fermer le bus I2C pour éviter les fuites de ressources
183.
184. if __name__ == "__main__":
185.     try:
186.         setup_gpio() # Initialiser GPIO
187.
188.         # Initialiser les capteurs
189.         temp_sensor = W1ThermSensor() # Capteur de température
190.         i2c_bus = smbus2.SMBus(1) # Bus I2C
191.         bme280_address = 0x76 # Adresse du capteur BME280
192.         bme280_params = bme280.load_calibration_params(i2c_bus, bme280_address) # Charger les
paramètres de calibration
193.
194.         # Variables partagées
195.         heater_status = multiprocessing.Value('i', 0) # État du chauffage
196.         humidifier_status = multiprocessing.Value('i', 0) # État de l'humidificateur
197.         fan_status = multiprocessing.Value('i', 0) # État du ventilateur
198.         airflow_fan_status = multiprocessing.Value('i', 0) # État du ventilateur de flux d'air

```

```
199.     motor_rotation_status = multiprocessing.Value('i', 0) # État de la rotation du moteur
200.     temperature = multiprocessing.Value('d', 0.0) # Température
201.     humidity = multiprocessing.Value('d', 0.0) # Humidité
202.
203.     # Créer des processus pour chaque contrôle de composant
204.     sensor_process = multiprocessing.Process(target=read_sensors, args=(temp_sensor,
i2c_bus, bme280_address, bme280_params, temperature, humidity))
205.     heater_process = multiprocessing.Process(target=control_heater, args=(heater_status,
temperature))
206.     steamer_process = multiprocessing.Process(target=control_steamer,
args=(humidifier_status, humidity))
207.     cooling_fan_process = multiprocessing.Process(target=control_cooling_fan,
args=(fan_status, temperature, humidity))
208.     airflow_fan_process = multiprocessing.Process(target=control_airflow_fan,
args=(airflow_fan_status,))
209.     motor_process = multiprocessing.Process(target=control_motor_rotation)
210.     print_process = multiprocessing.Process(target=print_data, args=(temperature,
humidity))
211.
212.     # Démarrer les processus
213.     sensor_process.start()
214.     heater_process.start()
215.     steamer_process.start()
216.     cooling_fan_process.start()
217.     airflow_fan_process.start()
218.     motor_process.start()
219.     print_process.start()
220.
221.     # Joindre les processus pour attendre qu'ils se terminent
222.     sensor_process.join()
223.     heater_process.join()
224.     steamer_process.join()
225.     cooling_fan_process.join()
226.     airflow_fan_process.join()
227.     motor_process.join()
228.     print_process.join()
229.
230.     except KeyboardInterrupt:
231.         print("Interruption clavier détectée.")
232.     finally:
233.         # Nettoyer GPIO
234.         GPIO.cleanup()
235.         i2c_bus.close() # Fermer correctement le bus I2C
```

Explication de programme :

`'import RPi.GPIO as GPIO'`: Importe la bibliothèque RPi.GPIO pour contrôler les broches GPIO du Raspberry Pi.

`'import time'`: Importe le module time pour utiliser les fonctions liées au temps, comme les délais.

`'import multiprocessing'`: Importe le module multiprocessing pour permettre l'exécution parallèle de plusieurs processus.

`'import smbus2'`: Importe la bibliothèque smbus2 pour la communication avec les périphériques I2C.

``import bme280``: Importe le module `bme280` pour interagir avec le capteur BME280.

``from w1thermsensor import W1ThermSensor``: Importe la classe `W1ThermSensor` pour lire les données des capteurs de température 1-Wire.

``HEATER_PIN = 1`` (et autres définitions de broches): Définit les numéros des broches GPIO associées aux différents composants.

``def setup_gpio()``: Définit une fonction pour initialiser les configurations GPIO, y compris la définition des broches en mode sortie et l'état initial des composants.

``GPIO.setmode(GPIO.BCM)``: Configure le mode de numérotation des broches GPIO en mode BCM.

``GPIO.setup(HEATER_PIN, GPIO.OUT)``: Configure la broche du chauffage comme sortie.

``GPIO.output(HEATER_PIN, GPIO.HIGH)``: Définit l'état initial du chauffage comme éteint.

``def read_temperature(sensor)``: Déclare une fonction pour lire la température à partir d'un capteur de température.

``def read_humidity(bus, address, calibration_params)``: Déclare une fonction pour lire l'humidité à partir du capteur BME280.

``def control_heater(heater_status, temperature)``: Déclare une fonction pour contrôler le chauffage en fonction de la température avec une hystérésis.

``target_temp = 37.5``: Définit la température cible pour le chauffage.

``if temp_value < target_temp - hysteresis``: Allume le chauffage si la température est inférieure à la cible moins l'hystérésis.

``if temp_value > target_temp + hysteresis``: Éteint le chauffage si la température est supérieure à la cible plus l'hystérésis.

``def control_steamer(humidifier_status, humidity)``: Déclare une fonction pour contrôler l'humidificateur en fonction du niveau d'humidité avec une hystérésis.

``def control_cooling_fan(fan_status, temperature, humidity)`:` Déclare une fonction pour contrôler le ventilateur de refroidissement en fonction de la température et de l'humidité.

``if temp_value > 38 or humidity_value > 58`:` Allume le ventilateur de refroidissement si la température dépasse 38°C ou si l'humidité dépasse 58%.

``def control_airflow_fan(airflow_fan_status)`:` Déclare une fonction pour contrôler le ventilateur de flux d'air.

``time.sleep(180)`:` Fait fonctionner le ventilateur pendant 3 minutes.

``time.sleep(7020)`:` Éteint le ventilateur pendant les 117 minutes restantes.

``def control_motor_rotation()`:` Déclare une fonction pour contrôler la rotation du moteur.

``def read_sensors(temp_sensor, bus, address, calibration_params, temperature, humidity)`:` Déclare une fonction pour lire les capteurs et mettre à jour les variables partagées.

``if __name__ == "__main__":`:` Point d'entrée principal du script pour initialiser et lancer les processus.

``setup_gpio()`:` Appelle la fonction pour initialiser les configurations GPIO.

``multiprocessing.Process(target=read_sensors, ...)`` (et autres) : Crée des processus distincts pour lire les capteurs et contrôler les différents composants.

``process.start()`:` Démarre chaque processus créé.

``process.join()`:` Attend la fin de chaque processus.

IV.2 Apprentissage et implémentation d'un modèle d'IA

L'apprentissage du modèle IA nécessite une large collection de données qui affectent directement ou indirectement les niveaux de température et d'humidité de l'incubateur. Ces données peuvent être collectées pendant la période d'incubation, ce qui signifie que le modèle peut être amélioré pendant qu'il est en cours d'exécution.

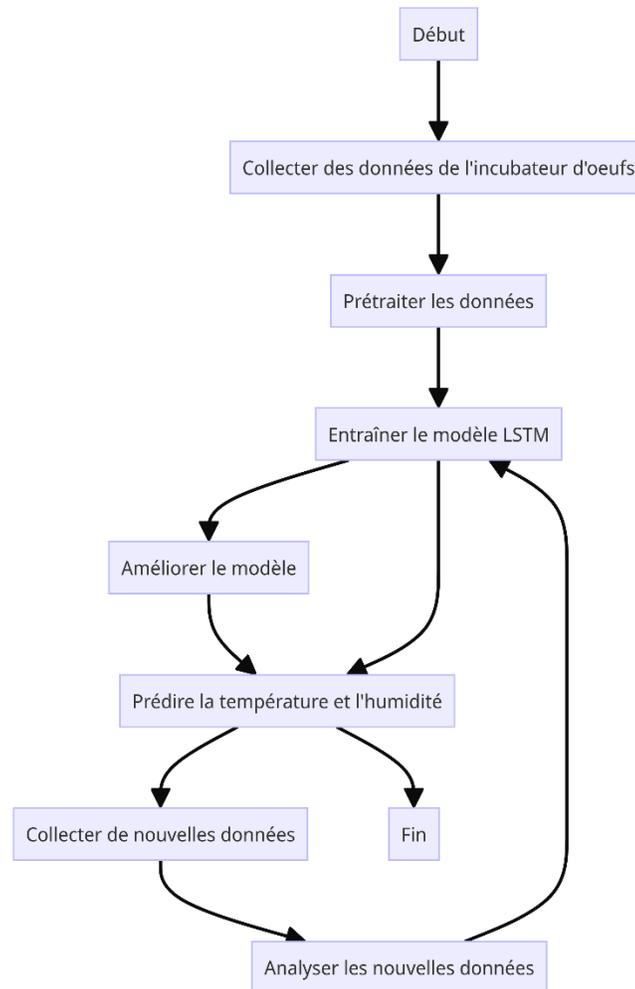


Figure IV.5: Système de Contrôle Intelligent pour la couveuse.

IV.2.1 Collecte des données d'apprentissage

Afin de collecter des données d'apprentissage pour notre modèle d'IA de la couveuse, nous devons faire fonctionner la couveuse et enregistrer en permanence des paramètres critiques tels que la température, l'humidité et l'état de tous les composants (chauffages, ventilateurs, unités de contrôle de l'humidité, etc.) Ce processus de collecte de données implique la lecture de capteurs et de systèmes d'enregistrement de données à l'intérieur de l'incubateur afin de capturer des informations en temps réel. Il est essentiel de collecter des données dans l'incubateur spécifique sur lequel nous avons l'intention d'utiliser le modèle, car chaque incubateur peut avoir des caractéristiques uniques, telles que des propriétés thermiques différentes, des schémas de

circulation de l'air et des comportements des composants. Ces facteurs uniques influencent l'environnement d'incubation et l'efficacité du modèle d'IA. En recueillant des données directement auprès de notre incubateur cible, nous nous assurons que le modèle est formé sur les données les plus pertinentes et les plus précises, ce qui permet d'obtenir de meilleures prédictions et des stratégies de contrôle adaptées aux nuances opérationnelles de l'incubateur en question. Cette spécificité renforce la fiabilité et les performances du modèle, améliorant ainsi le taux de réussite de notre processus d'incubation.

Pour enregistrer les données dans un fichier Excel, nous avons ajouté cette partie au programme principal :

```
1. import csv # Importer le module csv
2. from datetime import datetime # Importer datetime pour obtenir l'heure actuelle
3. import time # Importer time pour gérer les délais
4.
5. # Fonction pour enregistrer les données de l'incubateur
6. def log_data(heater_status, humidifier_status, fan_status, airflow_fan_status, temperature,
humidity):
7.     with open('incubator_data.csv', 'a', newline='') as csvfile: # Ouvrir le fichier CSV en
mode append
8.         fieldnames = ['Timestamp', 'Temperature', 'Humidity', 'Heater_Status',
'Humidifier_Status', 'Fan_Status', 'Airflow_Fan_Status'] # Définir les noms des colonnes
9.         writer = csv.DictWriter(csvfile, fieldnames=fieldnames) # Créer un objet DictWriter
avec les noms de colonnes
10.
11.         writer.writeheader() # Écrire l'en-tête dans le fichier CSV
12.         try:
13.             while True: # Boucle infinie pour l'enregistrement continu des données
14.                 writer.writerow({'Timestamp': datetime.now(), 'Temperature': temperature.value,
'Humidity': humidity.value,
15.                                 'Heater_Status': heater_status.value, 'Humidifier_Status':
humidifier_status.value,
16.                                 'Fan_Status': fan_status.value, 'Airflow_Fan_Status':
airflow_fan_status.value}) # Écrire une ligne de données dans le fichier CSV
17.                 csvfile.flush() # S'assurer que les données sont écrites immédiatement dans le
fichier
18.                 time.sleep(1) # Attendre une seconde avant de capturer les prochaines données
19.             except Exception as e: # Gérer les exceptions
20.                 print("Error occurred in logging data:", e) # Afficher un message d'erreur en cas
d'exception
```

Après avoir fait fonctionner le système pendant quelques jours, nous avons pu collecter environ 235 025 points de données pour chaque variable.

	A	B	C	D	E	F
1	Temperature	Humidity	Heater_Status	Humidifier_Status	Fan_Status	Airflow_Fan_Status
2	24.625	23.56983555	1	1	0	1
3	24.625	23.56983555	1	1	0	1
4	24.625	23.56248151	1	1	0	1
5	24.625	23.56248151	1	1	0	1
6	24.625	23.78043941	1	1	0	1
7	24.625	23.78043941	1	1	0	1
8	24.625	23.88616085	1	1	0	1
9	24.625	23.88616085	1	1	0	1
10	24.6875	23.95025805	1	1	0	1
11	24.6875	24.10886205	1	1	0	1
12	24.6875	24.10886205	1	1	0	1
13	24.6875	24.52159298	1	1	0	1
14	24.6875	24.52159298	1	1	0	1
15	24.75	25.0173942	1	1	0	1
16	24.75	25.0173942	1	1	0	1
17	24.875	25.50081596	1	1	0	1
18	24.875	25.50081596	1	1	0	1
19	24.875	25.96096381	1	1	0	1
20	24.875	25.96096381	1	1	0	1
21	24.9375	26.41521543	1	1	0	1
22	24.9375	26.94557558	1	1	0	1
23	24.9375	26.94557558	1	1	0	1
24	25	27.56525047	1	1	0	1
25	25	27.56525047	1	1	0	1
26	25.0625	28.13170249	1	1	0	1
27	25.0625	28.13170249	1	1	0	1

Figure IV.6: Echantillon des données collectées sauvegardé dans un fichier csv (Excel).

IV.2.2 Apprentissage du modèle

IV.2.2.1 Limitation :

Nous avons d'abord tenté d'entraîner le modèle à l'aide d'un Raspberry Pi. Cependant, nous avons rencontré d'importantes limitations matérielles. La puissance de traitement et la mémoire limitées du Raspberry Pi n'étaient pas suffisantes pour répondre aux exigences de calcul liées à l'entraînement d'un modèle d'apprentissage automatique complexe. Par conséquent, nous avons décidé de passer à un ordinateur plus puissant, qui a fourni les ressources et la vitesse de traitement nécessaires pour entraîner efficacement le modèle, en garantissant de meilleures performances et une plus grande précision.

IV.2.2.2 Résultat d'apprentissage

Ce code est utilisé pour former un modèle LSTM afin de prédire les valeurs futures de température et d'humidité sur la base des données historiques de l'incubateur, en prenant la

température et l'humidité précédentes comme entrées ainsi que l'état de chaque composant (éléments chauffants, humidificateur et ventilateurs) :

```

1. import pandas as pd # Importation de pandas pour la manipulation des données
2. import numpy as np # Importation de numpy pour les opérations numériques
3. import torch # Importation de PyTorch pour les opérations de machine learning
4. import torch.nn as nn # Importation des modules de réseaux neuronaux de PyTorch
5. import torch.optim as optim # Importation des modules d'optimisation de PyTorch
6. from torch.utils.data import DataLoader, TensorDataset # Importation des DataLoader et
TensorDataset de PyTorch
7. from sklearn.metrics import mean_squared_error # Importation de mean_squared_error de scikit-
learn
8. import matplotlib.pyplot as plt # Importation de matplotlib pour la visualisation des données
9. from torch.utils.tensorboard import SummaryWriter # Importation de SummaryWriter pour
l'utilisation de TensorBoard
10.
11. # Charger les données
12. data = pd.read_csv('incubator_data.csv') # Lire le fichier CSV contenant les données
13.
14. # Créer des séquences pour le LSTM
15. def create_sequences(data, seq_length, pred_step):
16.     xs, ys = [], [] # Initialiser les listes pour les entrées (xs) et les cibles (ys)
17.     for i in range(len(data) - seq_length - pred_step + 1):
18.         x = data[i:i+seq_length] # Sélectionner Température, Humidité, État du chauffage, État
de l'humidificateur, État du ventilateur, État du ventilateur d'aération
19.         y = data[i+seq_length+pred_step-1, [0, 1]] # Sélectionner Température et Humidité pour
les cibles
20.         xs.append(x) # Ajouter la séquence d'entrée à la liste xs
21.         ys.append(y) # Ajouter la cible correspondante à la liste ys
22.     return np.array(xs), np.array(ys) # Convertir les listes en tableaux numpy et les
retourner
23.
24. SEQ_LENGTH = 10 # Longueur de la séquence pour l'entrée
25. PRED_STEP = 10 # Prédire la valeur au 20ème pas
26. X, y = create_sequences(data.values, SEQ_LENGTH, PRED_STEP) # Créer les séquences d'entrée et
les cibles
27.
28. # Convertir en tenseurs PyTorch
29. X_tensor = torch.tensor(X, dtype=torch.float32) # Convertir X en tenseur PyTorch de type
float32
30. y_tensor = torch.tensor(y, dtype=torch.float32) # Convertir y en tenseur PyTorch de type
float32
31.
32. # Diviser en ensembles d'entraînement et de test
33. TRAIN_SPLIT = int(0.8 * len(X)) # Définir la proportion de données pour l'entraînement
34. train_dataset = TensorDataset(X_tensor[:TRAIN_SPLIT], y_tensor[:TRAIN_SPLIT]) # Créer le
dataset d'entraînement
35. test_dataset = TensorDataset(X_tensor[TRAIN_SPLIT:], y_tensor[TRAIN_SPLIT:]) # Créer le
dataset de test
36.
37. # Définir le modèle LSTM
38. class LSTMModel(nn.Module): # Définir la classe du modèle LSTM héritant de nn.Module
39.     def __init__(self, input_size, hidden_size, output_size, num_layers=2): # Initialisation
du modèle
40.         super(LSTMModel, self).__init__() # Appeler le constructeur de la classe parente
41.         self.lstm1 = nn.LSTM(input_size, hidden_size, num_layers=num_layers, batch_first=True)
# Définir la couche LSTM
42.         self.fc = nn.Linear(hidden_size, output_size) # Définir la couche fully connected
43.
44.     def forward(self, x): # Définir la propagation avant
45.         out, _ = self.lstm1(x) # Passer l'entrée à travers la couche LSTM
46.         out = self.fc(out[:, -1, :]) # Obtenir la sortie du dernier pas de temps

```

```

47.         return out # Retourner la sortie
48.
49. model = LSTMModel(input_size=X.shape[2], hidden_size=64, output_size=2, num_layers=2) #
Initialiser le modèle avec les paramètres spécifiés
50.
51. # Définir l'optimiseur et la fonction de perte
52. optimizer = optim.Adam(model.parameters(), lr=0.001) # Définir l'optimiseur Adam avec un taux
d'apprentissage de 0.001
53. criterion = nn.MSELoss() # Définir la fonction de perte comme l'erreur quadratique moyenne
(MSE)
54.
55. # Définir les DataLoader
56. train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True) # Créer le DataLoader
pour l'entraînement avec un batch size de 32
57. test_loader = DataLoader(test_dataset, batch_size=32, shuffle=False) # Créer le DataLoader
pour le test avec un batch size de 32
58.
59. # Évaluer le modèle sur l'ensemble de test
60. def evaluate(model, test_loader, criterion):
61.     model.eval() # Mettre le modèle en mode évaluation
62.     test_loss = 0.0 # Initialiser la perte de test à 0
63.     with torch.no_grad(): # Désactiver la calcul des gradients
64.         for inputs, targets in test_loader: # Itérer sur le DataLoader de test
65.             outputs = model(inputs) # Obtenir les prédictions du modèle
66.             test_loss += criterion(outputs, targets).item() # Calculer la perte et l'ajouter à
la perte totale
67.     return test_loss / len(test_loader) # Retourner la perte moyenne
68.
69. # Fonction d'entraînement
70. def train(model, train_loader, test_loader, optimizer, criterion, epochs, writer):
71.     model.train() # Mettre le modèle en mode entraînement
72.     train_losses = [] # Initialiser la liste pour les pertes d'entraînement
73.     val_losses = [] # Initialiser la liste pour les pertes de validation
74.     for epoch in range(epochs): # Itérer sur le nombre d'époques
75.         epoch_loss = 0.0 # Initialiser la perte d'époque à 0
76.         for inputs, targets in train_loader: # Itérer sur le DataLoader d'entraînement
77.             optimizer.zero_grad() # Réinitialiser les gradients
78.             outputs = model(inputs) # Obtenir les prédictions du modèle
79.             loss = criterion(outputs, targets) # Calculer la perte
80.             loss.backward() # Faire la rétropropagation
81.             optimizer.step() # Mettre à jour les poids
82.             epoch_loss += loss.item() # Ajouter la perte au total de l'époque
83.         train_losses.append(epoch_loss / len(train_loader)) # Ajouter la perte moyenne de
l'époque à la liste
84.
85.         # Calculer la perte de validation
86.         val_loss = evaluate(model, test_loader, criterion) # Évaluer le modèle sur l'ensemble
de test
87.         val_losses.append(val_loss) # Ajouter la perte de validation à la liste
88.
89.         writer.add_scalar('Training Loss', epoch_loss / len(train_loader), epoch) #
Enregistrer la perte d'entraînement dans TensorBoard
90.         writer.add_scalar('Validation Loss', val_loss, epoch) # Enregistrer la perte de
validation dans TensorBoard
91.
92.         print(f'Epoch [{epoch+1}/{epochs}], Train Loss: {epoch_loss / len(train_loader)}, Val
Loss: {val_loss}') # Afficher les pertes pour chaque époque
93.
94.     return train_losses, val_losses # Retourner les pertes d'entraînement et de validation
95.
96. # Écrivain TensorBoard
97. writer = SummaryWriter('logs') # Initialiser le SummaryWriter pour TensorBoard
98.
99. # Entraîner le modèle

```

```

100. train_losses, val_losses = train(model, train_loader, test_loader, optimizer, criterion,
epochs=50, writer=writer) # Entraîner le modèle avec les paramètres spécifiés
101.
102. # Sauvegarder le modèle final
103. torch.save(model.state_dict(), 'final_model.pth') # Sauvegarder les poids du modèle
104.
105. # Évaluer le modèle sur l'ensemble de test
106. test_loss = evaluate(model, test_loader, criterion) # Évaluer le modèle sur le test set
107. print(f'Test Loss: {test_loss}') # Afficher la perte de test
108.
109. # Fermer le SummaryWriter
110. writer.close() # Fermer l'écrivain TensorBoard
111.
112. # Tracer l'historique de l'entraînement
113. plt.figure(figsize=(14, 5)) # Définir la taille de la figure
114.
115. # Tracer la perte d'entraînement et de validation
116. plt.plot(train_losses, label='Perte d apprentissage') # Tracer la perte d'entraînement
117. plt.plot(val_losses, label='Perte de validation') # Tracer la perte de validation
118. plt.title('Perte') # Définir le titre du graphique
119. plt.xlabel('Époque') # Définir l'étiquette de l'axe x
120. plt.ylabel('Perte') # Définir l'étiquette de l'axe y
121. plt.legend() # Afficher la légende
122. plt.show() # Afficher le graphe
123.

```

Une fois l'apprentissage terminé, nous obtenons le graphe de la figure IV.7 :

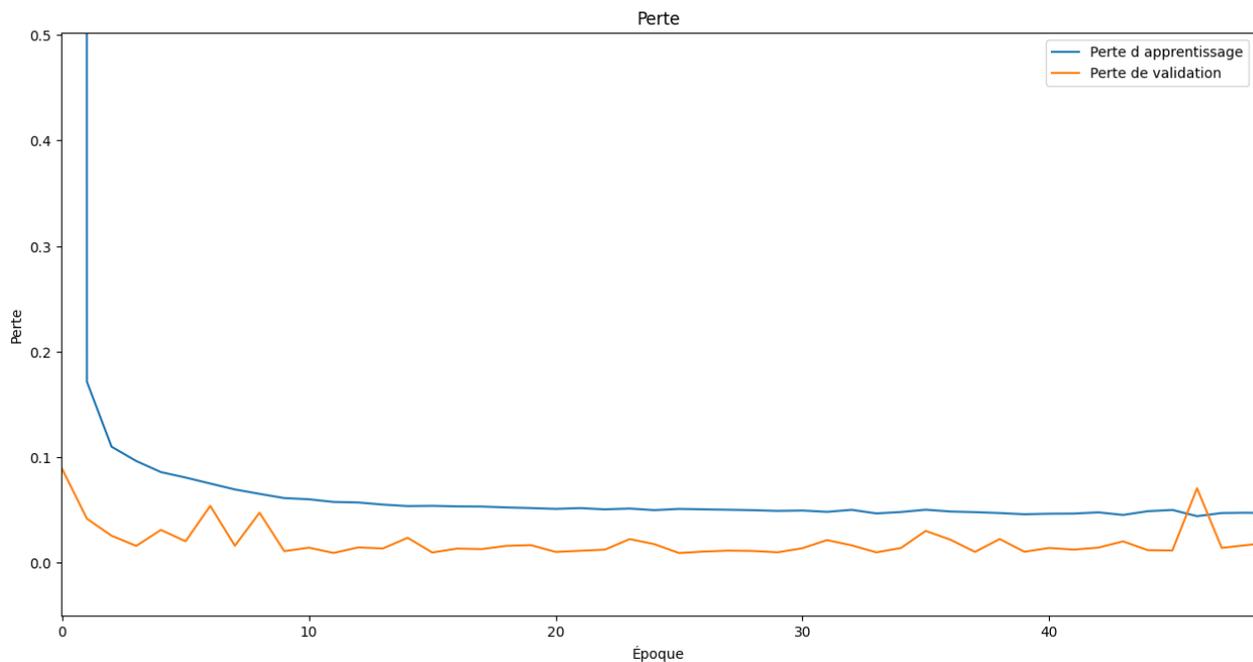


Figure IV.7: Le graphe montre la perte au cours de l'apprentissage et de la validation sur des époques pour le modèle LSTM.

Interprétation du graphe :

Le graphe montre que la perte d'apprentissage commence à environ 0,5, diminue rapidement au cours des premières époques, puis se stabilise progressivement à une valeur faible avec des fluctuations mineures. De même, la perte de validation commence à une valeur faible, diminue initialement et reste relativement stable avec de légères fluctuations. Cela indique que le modèle apprend efficacement à partir des données, les deux pertes diminuant et se stabilisant à des valeurs faibles, ce qui suggère un bon ajustement sans surajustement. La convergence des deux pertes signifie que le modèle a appris avec succès les modèles des données d'apprentissage et qu'il se généralise bien aux données de validation, ce qui indique une performance robuste.

IV.2.3 L'implémentation du modèle dans la couveuse

Afin d'utiliser le modèle appris, nous modifions le programme principal pour charger le fichier sauvegardé contenant les poids. Les données provenant de capteurs de température, d'humidité et les états de chaque élément sont lues périodiquement et stockées dans un mémoire tampon. Cette mémoire tampon est ensuite utilisée comme entrée pour le modèle LSTM, qui prédit la température et l'humidité futures en fonction des lectures actuelles et précédentes. Ces prédictions sont ensuite utilisées pour contrôler divers composants du système, tels que le chauffage, l'humidificateur et les ventilateurs, afin de maintenir les conditions environnementales souhaitées. Les prédictions du modèle guident les décisions de contrôle en temps réel, permettant ainsi de maintenir un environnement stable et confortable.

IV.3 Résultat

Nous avons saisi les changements de température et d'humidité dans les deux systèmes, avec et sans prédiction par le modèle LSTM.

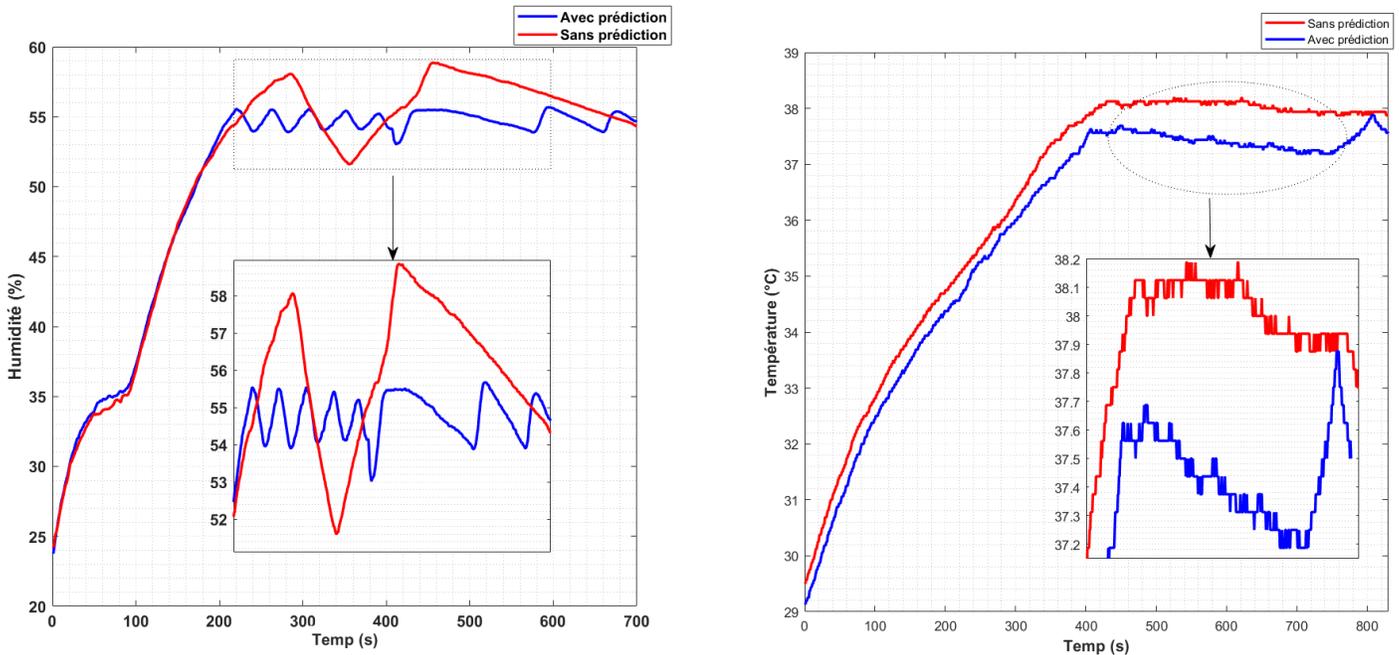


Figure IV.8: Les graphes représentent les changements de température et d'humidité pour le contrôle avec prédiction et sans prédiction.

Interprétation du résultat :

Les résultats indiquent une amélioration significative de la régulation de la température et de l'humidité lors de l'utilisation du modèle d'IA pour la prédiction par rapport au système sans ce modèle. Sans le modèle d'IA, la température dans l'incubateur dépasse les 38°C et l'humidité atteint 58%, ce qui prend un temps considérable pour revenir aux niveaux souhaités. Ce dépassement et cet ajustement tardif peuvent être attribués à la nature réactive du système sans IA, qui ne réagit aux changements qu'une fois qu'ils se sont produits, ce qui entraîne un retard dans les actions correctives.

En revanche, avec le modèle d'IA, la température et l'humidité restent dans les fourchettes cibles de 37,3°C à 37,7°C et de 54% à 56%, respectivement, et les ajustements sont effectués rapidement et efficacement. Le modèle d'IA utilise l'analyse prédictive pour prévoir les changements de température et d'humidité en fonction de diverses données et tendances. En anticipant les fluctuations avant qu'elles ne se produisent, le système d'IA peut procéder de manière proactive à des ajustements plus petits et plus fréquents pour maintenir des conditions optimales. Cette capacité prédictive réduit la probabilité de dépassement et garantit que l'environnement reste

dans la plage souhaitée de manière plus cohérente et plus efficace. La capacité du modèle d'IA à apprendre et à s'adapter à la dynamique spécifique de l'incubateur améliore encore sa réactivité et sa stabilité, ce qui se traduit par des réactions plus rapides et un meilleur contrôle des conditions internes.

Conclusion

En conclusion, ce chapitre a détaillé la réalisation concrète du projet, en décrivant en profondeur la mise en place du système développé. Les étapes essentielles du processus ont été expliquées, clarifiant les techniques et méthodes utilisées. L'analyse des résultats obtenus a démontré de manière convaincante l'efficacité supérieure du modèle AI. Les données montrent que l'IA permet de maintenir les conditions de température et d'humidité dans des plages optimales plus rapidement et de manière plus stable par rapport au système non-AI. Cette évaluation critique souligne les avantages significatifs du modèle prédictif, prouvant ainsi que l'intégration de l'IA améliore considérablement la performance et la réactivité du système d'incubation.

Conclusion générale

Dans cette étude approfondie sur l'incubation des œufs, nous avons plongé dans les mécanismes et les défis de cette pratique cruciale dans l'industrie avicole. En explorant les tenants et les aboutissants de l'incubation, nous avons souligné l'importance capitale du contrôle minutieux de divers paramètres tels que la température, l'humidité, la ventilation et les cycles de retournement des œufs. Ces éléments sont essentiels pour garantir des taux d'éclosion élevés et des poussins en bonne santé. À mesure que nous progressions dans notre exploration, nous avons constaté que l'intelligence artificielle, et en particulier les réseaux neuronaux LSTM, offraient une solution prometteuse pour optimiser ces processus. Ces algorithmes d'apprentissage profond ont le potentiel de prédire avec précision les tendances et de contrôler les conditions d'incubation de manière dynamique, permettant ainsi une gestion plus efficace et proactive de l'environnement d'incubation. En concrétisant ces concepts théoriques, nous avons développé un système d'incubation d'œufs innovant, intégrant pleinement les avantages de l'IA pour une gestion automatisée et optimisée. L'analyse approfondie des résultats obtenus a confirmé l'efficacité de notre approche, mettant en lumière les améliorations significatives dans les taux d'éclosion et la qualité des poussins. En fin de compte, cette étude offre un aperçu précieux sur la convergence entre l'incubation traditionnelle et les technologies de pointe, ouvrant la voie à de nouvelles perspectives pour une aviculture plus efficace et durable.

1. Compréhension des Fondamentaux de l'Incubation :
 - Exploration des paramètres critiques de l'incubation, y compris la température, l'humidité, la ventilation et les cycles de retournement des œufs.
2. Intégration de l'Intelligence Artificielle :
 - Utilisation des réseaux neuronaux LSTM pour prédire et contrôler efficacement les conditions d'incubation, offrant ainsi une approche proactive et adaptative.
3. Conception et Développement du Système d'Incubation :
 - Mise en œuvre des concepts théoriques dans la conception et le développement d'un système d'incubation innovant, intégrant pleinement les avantages de l'IA.
4. Résultats et Analyse :
 - Évaluation approfondie des performances du système développé, mettant en lumière les améliorations significatives dans les taux d'éclosion et la qualité des poussins.

5. Perspectives Futures :

- Ouverture de nouvelles perspectives pour une aviculture plus efficace et durable, grâce à la convergence entre l'incubation traditionnelle et les technologies de pointe.

Bibliographie

- [1] M. Ohemu, U. Umar, A. Edmund, et Z. Ohunene, « Design and Implementation of a Remotely Monitored Smart Egg Incubator », *Int. J. Sci. Eng. Res.*, vol. 12, p. 1009-1017, nov. 2021.
- [2] K. K. Çevik, H. E. Koçer, et M. Boğa, « Deep Learning Based Egg Fertility Detection », *Vet. Sci.*, vol. 9, n° 10, Art. n° 10, oct. 2022, doi: 10.3390/vetsci9100574.
- [3] S. Cakic, T. Popovic, S. Krco, D. Nedic, D. Babic, et I. Jovovic, « Developing Edge AI Computer Vision for Smart Poultry Farms Using Deep Learning and HPC », *Sensors*, vol. 23, n° 6, Art. n° 6, janv. 2023, doi: 10.3390/s23063002.
- [4] V. Raudonis, A. Paulauskaite-Taraseviciene, et K. Sutiene, « Fast Multi-Focus Fusion Based on Deep Learning for Early-Stage Embryo Image Enhancement », *Sensors*, vol. 21, n° 3, Art. n° 3, janv. 2021, doi: 10.3390/s21030863.
- [5] J. Zhou, Y. Liu, S. Zhou, M. Chen, et D. Xiao, « Evaluation of Duck Egg Hatching Characteristics with a Lightweight Multi-Target Detection Method », *Animals*, vol. 13, n° 7, Art. n° 7, janv. 2023, doi: 10.3390/ani13071204.
- [6] « What is an Egg Incubator? », All Things Nature. Consulté le: 17 mai 2024. [En ligne]. Disponible sur: <https://www.allthingsnature.org/what-is-an-egg-incubator.htm>
- [7] « Incu-ba-tors: Better than any hen », mag | The Magazine by ebm-papst. Consulté le: 17 mai 2024. [En ligne]. Disponible sur: https://mag.ebmpapst.com/en/industries/machines/incubators-better-than-any-hen_14945/
- [8] V. Traverso, « The Egyptian Egg Ovens Considered More Wondrous Than the Pyramids », Atlas Obscura. Consulté le: 17 mai 2024. [En ligne]. Disponible sur: <http://www.atlasobscura.com/articles/egypt-egg-ovens>
- [9] « Incubator (egg) », *Wikipedia*. 7 septembre 2023. Consulté le: 31 mai 2024. [En ligne]. Disponible sur: [https://en.wikipedia.org/w/index.php?title=Incubator_\(egg\)&oldid=1174322267](https://en.wikipedia.org/w/index.php?title=Incubator_(egg)&oldid=1174322267)
- [10] *Art de Faire Eclorre de d'Elever en Toute Saison des Oiseaux Domestiques de Toutes Especes, Soit par le Moyen de la Chaleur du Fumier, Soit par le Moyen de Celle du Feu Ordinaire*. Consulté le: 17 mai 2024. [En ligne]. Disponible sur: <https://www.biomedrarebooks.com/product/1232/Art-de-Faire-Eclorre-de-dElever-en-Toute-Saison-des-Oiseaux-Domestiques-de-Toutes-Especes-Soit-par-le-Moyen-de-la-Chaleur-du-Fumier-Soit-par-le-Moyen-de-Celle-du-Feu-Ordinaire>
- [11] E. Rude, *Tastes Like Chicken: A History of America's Favorite Bird*. Pegasus Books, 2016.
- [12] « The History Of Incubation ». Consulté le: 17 mai 2024. [En ligne]. Disponible sur: https://web.archive.org/web/20131220175348/http://www.pleysierincubators.com/history_incubation.htm
- [13] « Company history | Petersime ». Consulté le: 17 mai 2024. [En ligne]. Disponible sur: <https://www.petersime.com/about-us/company-history/>
- [14] « Chapitre 8. Incubation ». Consulté le: 31 mai 2024. [En ligne]. Disponible sur: <https://www.fao.org/4/Y4359F/y4359f0b.htm>
- [15] Domi, « Modèles d'incubateurs », Poules et Cie. Consulté le: 31 mai 2024. [En ligne]. Disponible sur: <https://poulesetcie.com/modeles-incubateurs/>

- [16] D. Bell et W. Weaver, « Commercial Chicken Meat and Egg Production: 5th Edition », *J. Appl. Poult. Res.*, vol. 11, p. 224-225, juill. 2002, doi: 10.1093/japr/11.2.224.
- [17] R. B. Paguntalan et V. H. Oquino, « Design and Development of a Microcontroller based Egg Incubator for Small Scale Poultry Production », *Glob. J. Sci. Front. Res.*, avr. 2016, Consulté le: 1 juin 2024. [En ligne]. Disponible sur: <https://www.semanticscholar.org/paper/Design-and-Development-of-a-Microcontroller-based-Paguntalan-Oquino/26c1673d400b46d2cab4257754a8d0863d57cc69>
- [18] A. M. Fouad, D. Ruan, H. K. El-Senousey, W. Chen, S. Jiang, et C. Zheng, « Harmful Effects and Control Strategies of Aflatoxin B1 Produced by *Aspergillus flavus* and *Aspergillus parasiticus* Strains on Poultry: Review », *Toxins*, vol. 11, n° 3, p. 176, mars 2019, doi: 10.3390/toxins11030176.
- [19] W. Katekhong et S. Charoenrein, « Changes in physical and gelling properties of freeze-dried egg white as a result of temperature and relative humidity », *J. Sci. Food Agric.*, vol. 96, n° 13, p. 4423-4431, oct. 2016, doi: 10.1002/jsfa.7653.
- [20] M. Çam, Z. Kaya, S. Güler, H. Harman, et K. Kirikçi, « Influence of egg storage time, position and turning on egg weight loss, embryonic mortality and hatching traits in chukar partridge (*Alectoris chukar*) », *Ital. J. Anim. Sci.*, vol. 21, p. 1632-1641, déc. 2022, doi: 10.1080/1828051X.2022.2150095.
- [21] J. Markson et C. Brundage, « Cooling Periods Enhance Specific Pathogen Free (SPF) Poultry Egg Hatchability », vol. 2, p. 1-3, déc. 2019, doi: 10.24966/AZS-7779/100011.
- [22] H. Adriaensen *et al.*, « How Egg Storage Duration Prior to Incubation Impairs Egg Quality and Chicken Embryonic Development: Contribution of Imaging Technologies », *Front. Physiol.*, vol. 13, p. 902154, mai 2022, doi: 10.3389/fphys.2022.902154.
- [23] M. Ayalew, *Modern Poultry Production Text Book*. LAP LAMBERT Academic Publishing, 2016.
- [24] A. Sabharwal et B. Selman, « S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Third Edition. », *Artif Intell*, vol. 175, p. 935-937, avr. 2011, doi: 10.1016/j.artint.2011.01.005.
- [25] Y. LeCun, Y. Bengio, et G. Hinton, « Deep learning », *Nature*, vol. 521, n° 7553, p. 436-444, mai 2015, doi: 10.1038/nature14539.
- [26] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, et S. Khudanpur, « Recurrent neural network based language model », présenté à Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010, sept. 2010, p. 1045-1048. doi: 10.21437/Interspeech.2010-343.
- [27] Y. Bengio, P. Simard, et P. Frasconi, « Learning long-term dependencies with gradient descent is difficult », *IEEE Trans. Neural Netw.*, vol. 5, n° 2, p. 157-166, mars 1994, doi: 10.1109/72.279181.
- [28] A. Graves, A. Mohamed, et G. Hinton, « Speech recognition with deep recurrent neural networks », in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada: IEEE, mai 2013, p. 6645-6649. doi: 10.1109/ICASSP.2013.6638947.
- [29] P. Grewal, « A Critical Conceptual Analysis of Definitions of Artificial Intelligence as Applicable to Computer Engineering », *IOSR J. Comput. Eng.*, vol. 16, p. 09-13, janv. 2014, doi: 10.9790/0661-16210913.

- [30] M. A. Hamame, « Development and design of an intelligent system for poultry diseases prediction », Thesis, Ibn Khaldoun University, 2023. Consulté le: 5 juin 2024. [En ligne]. Disponible sur: <http://dspace.univ-tiaret.dz:80/handle/123456789/13597>
- [31] R. C. Staudemeyer et E. R. Morris, « Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks ». arXiv, 12 septembre 2019. doi: 10.48550/arXiv.1909.09586.
- [32] D. E. Rumelhart, G. E. Hinton, et R. J. Williams, « Learning representations by back-propagating errors », *Nature*, vol. 323, n° 6088, p. 533-536, oct. 1986, doi: 10.1038/323533a0.
- [33] S. Hochreiter et J. Schmidhuber, « Long Short-Term Memory », *Neural Comput.*, vol. 9, n° 8, p. 1735-1780, nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [34] « Training Neural Networks | Machine Learning », Google for Developers. Consulté le: 6 juin 2024. [En ligne]. Disponible sur: <https://developers.google.com/machine-learning/crash-course/training-neural-networks/video-lecture>
- [35] J. Brownlee, « Why Training a Neural Network Is Hard », MachineLearningMastery.com. Consulté le: 6 juin 2024. [En ligne]. Disponible sur: <https://machinelearningmastery.com/why-training-a-neural-network-is-hard/>
- [36] I. Sutskever, O. Vinyals, et Q. V. Le, « Sequence to Sequence Learning with Neural Networks ». arXiv, 14 décembre 2014. doi: 10.48550/arXiv.1409.3215.
- [37] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, et P.-A. Muller, « Deep learning for time series classification: a review », arXiv.org. Consulté le: 6 juin 2024. [En ligne]. Disponible sur: <https://arxiv.org/abs/1809.04356v4>
- [38] A. Y. Hannun, A. L. Maas, D. Jurafsky, et A. Y. Ng, « First-Pass Large Vocabulary Continuous Speech Recognition using Bi-Directional Recurrent DNNs ». arXiv, 8 décembre 2014. doi: 10.48550/arXiv.1408.2873.
- [39] R. Pallas-Areny et J. Webster, *Sensors and signal conditioning, Second Edition*. 2001.
- [40] Y. S. Mohammed, M. W. Mustafa, N. Bashir, et A. S. Mokhtar, « Renewable energy resources for distributed power generation in Nigeria: A review of the potential », *Renew. Sustain. Energy Rev.*, vol. 22, n° C, p. 257-268, 2013.
- [41] A. Kumar, D. Gupta, et V. Gupta, « A Comprehensive Review on Grid-Tied Solar Photovoltaic System », *J. Green Eng.*, vol. 7, p. 213-254, janv. 2017, doi: 10.13052/jge1904-4720.71210.
- [42] N. Mohan, T. M. Undeland, et W. P. Robbins, *Power Electronics: Converters, Applications, and Design*, 3rd edition. Hoboken, NJ: Wiley, 2002.
- [43] S. A. Kalogirou, *Solar Energy Engineering: Processes and Systems*, 2nd edition. Amsterdam: Academic Press, 2013.
- [44] P. Gevorkian, *Solar Power Generation Problems, Solutions and Monitoring*. 2016. doi: 10.1017/CBO9781316341414.
- [45] E. Upton et G. Halfacree, *Raspberry Pi user guide*. Chichester: Wiley, 2012.
- [46] J. W. Jolles, « Broad-scale applications of the Raspberry Pi: A review and guide for biologists », *Methods Ecol. Evol.*, vol. 12, n° 9, p. 1562-1579, 2021, doi: 10.1111/2041-210X.13652.
- [47] A. Ryabtsev, « 8 Reasons Why Python is Good for Artificial Intelligence and Machine Learning », Software Development Blog & IT Tech Insights | Django Stars. Consulté le: 10 juin 2024. [En ligne]. Disponible sur: <https://djangostars.com/blog/why-python-is-good-for-artificial-intelligence-and-machine-learning/>

[48] « Most Popular Programming Languages in 2024 & Beyond ». Consulté le: 10 juin 2024. [En ligne]. Disponible sur: <https://www.orientsoftware.com/blog/most-popular-programming-languages/>



البطاقة التقنية للمشروع

ABDELLAH Ismail ALLAOUI Abdelghani	الاسم و اللقب Votre prénom et nom Your first and last Name
SMART INC	الاسم التجاري للمشروع Intitulé de votre projet Title of your Project
	الصفة القانونية للمشروع Votre statut juridique Your legalstatus
0657 18 25 09	رقم الهاتف Votre numéro de téléphone Your phone number
ismail.hattori@gmail.com	البريد الالكتروني Votre adresse e-mail Your email address
Tiaret	مقر مزاولة النشاط (الولاية- البلدية) Votre ville ou commune d'activité Your city or municipality of activity





طبيعة المشروع

<p>Les objectifs principaux de ce projet sont :</p> <ul style="list-style-type: none"> • En automatisant le contrôle de la température et de l'humidité et en utilisant des algorithmes prédictifs, nous visons à améliorer les taux d'éclosion et à réduire la dépendance à la surveillance humaine. • Ce système devrait être capable de s'adapter aux variations des conditions environnementales et de prendre des décisions en temps réel pour maintenir des conditions optimales. 	<p>Le projet vise à Développer un système intelligent d'incubation d'œufs qui utilise l'intelligence artificielle pour contrôler les conditions d'incubation.</p>
---	---

Value Proposition القيمة المقترحة أو العرض المقدم

تحديد المشكل الذي يواجهه الزبون

<p>améliorer les taux d'éclosion et réduire la dépendance à la surveillance humaine</p>	<p>ما هي المشكلة التي تريد حلها؟</p>
<p>Les faibles taux d'éclosion dans les incubateurs d'œufs sont souvent dus à un mauvais contrôle de la température et de l'humidité, qui sont essentiels au développement de l'embryon. Des réglages de température incohérents ou des fluctuations peuvent perturber le processus d'incubation, tandis que des niveaux d'humidité incorrects peuvent assécher les membranes de l'œuf ou provoquer des problèmes respiratoires.</p> <p>En outre, un retournement inadéquat ou irrégulier</p>	<p>ما هي البيانات المتوفرة لديك التي تدل على وجود المشكلة المحددة؟</p>



des œufs peut entraîner un développement anormal, et une mauvaise ventilation peut entraîner l'accumulation de gaz nocifs. Enfin, la qualité des œufs eux-mêmes, y compris les problèmes de fertilité et un mauvais stockage avant l'incubation, a également un impact significatif sur les taux d'éclosio

)

1. Brinsea Ovation 28 EX

Le Brinsea Ovation 28 EX est connu pour son contrôle précis de la température et de l'humidité, avec un retournement automatique des œufs et un affichage numérique pour une surveillance facile. Il est également équipé d'une pompe à humidité pour maintenir des niveaux optimaux.

2. Couveuse numérique GQF Sportsman

Cette couveuse est conçue pour une grande capacité et est équipée de commandes numériques pour la température et l'humidité. Elle est également équipée d'un retournement automatique des œufs et d'une ventilation efficace pour assurer une bonne circulation de l'air.

3. Nurture Right 360

La couveuse Nurture Right 360 offre une vue à 360 degrés pour la surveillance et dispose d'un retournement automatique des œufs, d'un contrôle numérique de la température et d'un affichage de l'humidité. Il est convivial et conçu

ما هي المشاريع الأخرى التي استهدفت نفس المشكلة والتي جرى تنفيذها؟





pour les débutants comme pour les éleveurs expérimentés.

4. Mini-cuveuse numérique entièrement automatique Magicfly

Cette cuveuse compacte est idéale pour l'éclosion à petite échelle. Elle dispose d'un affichage LED de la température et de l'humidité, d'un retournement automatique des œufs et d'un couvercle transparent pour faciliter l'observation.

5. Rcom Pro 20

La Rcom Pro 20 est une cuveuse haut de gamme dotée de fonctions avancées telles que le contrôle automatique de la température et de l'humidité, le retournement automatique des œufs et un affichage numérique. Il dispose également d'une fonction de refroidissement progressif pour simuler les conditions naturelles d'incubation.

Ces cuveuses intelligentes sont conçues pour offrir des conditions optimales d'incubation des œufs, ce qui rend le processus d'éclosion plus efficace et augmente les chances de réussite des .éclosions

1. Amélioration des taux d'éclosion
Les incubateurs intelligents offrent un contrôle précis de la température, de l'humidité et de la ventilation, ce qui permet d'obtenir des taux d'éclosion plus élevés et plus réguliers que dans les incubateurs traditionnels.

2. Cohérence et fiabilité
Les systèmes automatisés garantissent que les conditions restent stables et cohérentes, réduisant ainsi la variabilité et l'erreur humaine associées aux ajustements manuels.

ماهي أهداف مشروعك و/أو نتائجه المتوقعة؟



3. Facilité d'utilisation

Les couveuses intelligentes sont souvent dotées d'interfaces conviviales, d'affichages numériques et d'un retournement automatique des œufs, ce qui les rend plus faciles à utiliser pour les éleveurs débutants ou expérimentés.

4. Surveillance améliorée

Les capteurs intégrés et les écrans numériques fournissent des données en temps réel sur la température, l'humidité et le développement des œufs, ce qui permet un meilleur suivi et des réponses plus rapides en cas de problème.

5. Collecte et analyse des données

Les modèles avancés peuvent inclure la capacité de collecter et d'analyser des données sur plusieurs cycles d'éclosion, ce qui aide les utilisateurs à affiner leurs processus d'incubation et à améliorer les résultats au fil du temps.

6. Bien-être des animaux

En fournissant des conditions optimales, les incubateurs intelligents peuvent améliorer la santé et le bien-être des embryons en développement, ce qui se traduit par des poussins plus forts et en meilleure santé.

7. Efficacité énergétique

Les incubateurs intelligents modernes sont souvent conçus pour être économes en énergie, ce qui réduit le coût global de fonctionnement et l'impact sur l'environnement.

Dans l'ensemble, les couveuses intelligentes devraient rendre le processus d'éclosion plus efficace, plus fiable et plus fructueux, ce qui profitera aussi bien aux amateurs qu'aux éleveurs commerciaux





القيمة المقترحة وفق المعايير التالية

<p>L'intégration de l'IA dans les incubateurs d'œufs permet une surveillance en temps réel et des ajustements automatiques pour garantir des conditions d'éclosion optimales. Les algorithmes d'IA peuvent analyser les données historiques et en temps réel pour prédire la température, l'humidité et les horaires de rotation les mieux adaptés aux différents types d'œufs, ce qui améliore les taux d'éclosion. En outre, l'IA peut détecter rapidement les anomalies ou les problèmes potentiels, en procédant à des ajustements précis pour maintenir des conditions stables et améliorer l'efficacité globale. Cette automatisation avancée réduit les erreurs humaines et augmente la fiabilité et la réussite du processus d'incubation.</p>	<p>القيمة المبتكرة أو الجديدة</p>
	<p>القيمة بالتخصيص</p>
<p>L'objectif est de concevoir un système abordable et accessible aux couturiers amateurs et professionnels. L'utilisation de composants électroniques et de techniques de fabrication économiques permet de réduire les coûts et de proposer une solution plus attractive en termes de prix.</p>	<p>القيمة بالسعر</p>



<p>Le système est conçu pour être</p> <ul style="list-style-type: none"> • Facile à utiliser, avec une interface utilisateur intuitive qui ne nécessite pas de compétences techniques avancées. Cette simplicité d'utilisation, combinée à une conception robuste et fiable, assure une adoption facile et une satisfaction utilisateur élevée. • Compact et ergonomique. 	<p>القيمة بالتصميم</p>
<p>L'utilisation d'un Raspberry Pi doté de capacités d'intelligence artificielle dans un incubateur d'œufs est très efficace, car elle permet d'assurer un contrôle et une surveillance précis à faible coût. Le Raspberry Pi peut exécuter des algorithmes d'intelligence artificielle qui analysent en temps réel les données fournies par les capteurs de température, d'humidité et de développement des œufs. Le système peut ainsi procéder à des ajustements immédiats pour maintenir des conditions optimales, ce qui améliore considérablement les taux d'éclosion. La nature compacte et abordable du Raspberry Pi rend accessible l'automatisation avancée, tandis que sa connectivité permet la surveillance et le contrôle à distance. Cette combinaison permet d'améliorer l'efficacité, de réduire les erreurs humaines et de garantir un taux de réussite plus élevé dans l'éclosion des œufs.</p>	<p>القيمة بالأداء العالي</p>
<p>Notre couveuse est idéale pour tous les types d'œufs grâce à ses réglages polyvalents et</p>	<p>القيمة بالخدمة الشاملة</p>



<p>personnalisables. Avec un contrôle précis de la température, de l'humidité et du retournement automatique des œufs, il répond aux besoins spécifiques des différentes espèces, des poulets aux oiseaux exotiques. Des fonctions avancées telles que l'intégration de l'IA et la surveillance en temps réel garantissent le maintien constant de conditions optimales, ce qui se traduit par des taux d'éclosion plus élevés et des poussins en meilleure santé. Son interface conviviale et sa capacité d'adaptation en font un outil adapté aux éleveurs amateurs et professionnels, offrant des performances fiables et une grande facilité d'utilisation pour une large gamme de types d'œufs.</p>	
<ul style="list-style-type: none">• Durabilité• Gain de temps et d'énergie• Amélioration de la qualité de poussins• Réduction des frustrations• Accroissement de la sécurité	قيم أخرى





Customer Segments شرائح العملاء أو الزبائن

Démographique (B2C)	Démographique (B2B)	Psychographiqu e العوامل النفسية و الشخصية	Comportementa l السلوكيات
Age العمر +16	Secteur القطاع تجاري صناعي	Classe sociale طبقة الاجتماعية متوسطة	Usage استخدام منتظم
Sexe الجنس ذكر و انثى	Nombre d'employés عدد العمال في القطاع 5	Niveau de vie المستوى المعيشي متوسط	Loyauté الوفاء
Revenus annuel متوسط الدخل	Maturité de l'entreprise نضج المؤسسة	Valeurs القيم	Intérêt اهتمام الامان و الاستقرارية



	Etat matrimonial الحالة الاجتماعية	Situation financière الحالة المالية للمؤسسة مستقرة		Passion الهواية و شغف
Ville الدائرة او البلدية	Niveau d'étude المستوى الدراسي متوسط	Détention/ actionnariat الملكية/المساهمة		Sensibilité حساسيات
Quartier الحي	Profession المهنة	Valorisation/ capitalisation boursière التقييم / القيمة السوقية	Présence digitale et sur les réseaux sociaux استعمال التكنولوجيا في التواصل	Habitude de consommation عادة الاستهلاك
Climat المناخ	Culture الثقافة	Business model نموذج الأعمال	Centres d'intérêts مراكز الاهتمام	Mode de paiement طرق الدفع يد بيد بطاقة الدفع



		Secteur servi القطاع الذي يخدمه صناعي تجاري		Connaissance المعرفة متعلم
	Langue اللغة لا تحتاج لغة مخصصة	Technologie utilisée التكنولوجيا المستعملة		Nature de la demande طبيعة الطلب
		Format du produit ou packaging شكل المنتج أو التعبئة والتغليف		Fréquence d'achat عدد مرات الطلب على السلعة

Channels قنوات التوزيع

<p>Site web de l'entreprise : Créer un site web où les clients peuvent acheter l'appareil directement. Le site web peut contenir des descriptions détaillées des produits, des vidéos de démonstration et des avis d'utilisateurs.</p> <p>Expositions et événements : Participez à des expositions de couture et à des événements connexes pour présenter le produit et le vendre directement aux clients. Des promotions et des</p>	المبيعات المباشرة
--	-------------------



<p>réductions peuvent être proposées pour attirer les acheteurs.</p> <p>Vente en ligne : Utilisez des plateformes de commerce électronique telles qu'Amazon, eBay et Souq.com pour vendre le produit directement à des clients situés dans différents endroits.</p>	
<p>Accord avec les grossistes locaux : Vendre le produit en gros à des grossistes qui peuvent le distribuer à des magasins de détail ou directement aux clients.</p> <p>Offrir des incitations et des remises : Inciter les grossistes à acheter de plus grandes quantités en leur proposant des remises et des incitations pour les achats importants.</p>	<p>تجار الجملة</p>
<p>Nommer des distributeurs agréés : Collaborer avec des distributeurs agréés qui peuvent distribuer le produit dans un large éventail de magasins et de réseaux commerciaux. Les distributeurs peuvent assurer une large couverture géographique et l'accès à de nouveaux clients. Fournir une assistance technique et une formation : Fournir une assistance technique et une formation aux distributeurs afin de s'assurer qu'ils comprennent parfaitement le produit et qu'ils sont en mesure de le promouvoir et de le vendre efficacement.</p>	<p>الموزعون</p>





Magasins de détail spécialisés : Distribuer le produit dans des magasins de détail spécialisés dans la vente d'outils et de fournitures de couture, où les clients intéressés par ce domaine peuvent facilement le trouver.

Grands magasins : Collaborer avec les supermarchés pour placer le produit dans leurs rayons d'articles ménagers ou d'outils à main.

Promotions en magasin : Organiser des promotions et des ateliers en magasin pour familiariser les clients avec le produit et ses avantages et les inciter à l'acheter.

توزيع التجزئة

Customer Relationship العلاقة مع العملاء

1. Comprendre les besoins des clients

Étude de marché : Réaliser des études de marché pour mieux comprendre les besoins et les attentes des clients.

Enquêtes : Recueillir les réactions des clients au moyen d'enquêtes et de questionnaires afin d'identifier les points forts et les points faibles des produits.

2. Une communication constante

Présence numérique : utiliser les médias sociaux et le courrier électronique pour communiquer en permanence avec les clients et les tenir informés des dernières mises à jour et offres.

Bulletins d'information : Envoyez régulièrement des lettres d'information sur les améliorations apportées aux produits, des conseils et des offres

كيف تدير علاقاتك مع العملاء؟



spéciales.

3. Assistance technique

Service clientèle : Fournir une assistance technique par téléphone, par courrier électronique et par chat en direct pour aider les clients à résoudre leurs problèmes.

Matériel pédagogique : Créer des supports pédagogiques tels que des tutoriels vidéo et des manuels pour aider les clients à utiliser correctement le produit.

4. Offres et récompenses

Programmes de fidélisation : Créez des programmes de fidélisation qui offrent des réductions ou des offres spéciales aux clients réguliers.

Promotions : proposez des promotions et des remises spéciales aux nouveaux et aux anciens acheteurs.

5. Recueillir et analyser les données relatives aux clients

Systèmes de gestion de la relation client (CRM) : Utiliser les systèmes de gestion de la relation client (GRC) pour collecter et analyser les données relatives aux clients, telles que l'historique des achats, les préférences en matière de produits et les interactions avec le service d'assistance technique.

Analyse des données : Utiliser l'analyse des données pour comprendre le comportement des clients et identifier les possibilités d'amélioration des produits et des services.

6. Parvenir à une excellente expérience client

Simplifier le processus d'achat : Offrir un





processus d'achat facile et transparent en ligne et dans les magasins physiques.

Suivi après la vente : Communiquer avec les clients après l'achat pour s'assurer de leur satisfaction et résoudre les problèmes éventuels.

7. Recevoir un retour d'information et y répondre

Questionnaires de satisfaction : Envoyer des questionnaires pour mesurer la satisfaction des clients après l'utilisation du produit.

Réponse rapide : Traiter rapidement et efficacement les commentaires et les plaintes afin d'améliorer l'expérience du client.

8. Construire une communauté de marque

Forums et groupes : Créez des forums et des groupes en ligne où les clients peuvent partager leur expérience et échanger des conseils.

Événements et ateliers : Organisez des événements et des ateliers pour dialoguer avec les clients et leur apprendre à tirer le meilleur parti du produit.

9. Améliorer le produit en fonction du retour d'information des clients

Développement continu : Améliorer continuellement le produit en fonction du retour d'information et des attentes des clients.

Impliquer les clients dans le développement : Impliquer les clients dans les processus de développement et d'innovation par le biais d'enquêtes et de sessions de discussion.



<p>Salesforce CRM est l'un des programmes CRM les plus populaires et complets sur le marché. Il offre une large gamme de fonctionnalités adaptées aux besoins variés des entreprises.</p> <p>HubSpot CRM est connu pour sa facilité d'utilisation et son intégration transparente avec d'autres outils de marketing et de vente.</p> <p>Zoho CRM est une solution flexible et économique, adaptée aux petites et moyennes entreprises.</p> <p>Pipedrive est une solution CRM simple à utiliser, conçue principalement pour les équipes de vente.</p>	<p>ماهية أهم البرامج التي ستعتمد عليها في إدارة العلاقة مع الزبون</p> <p>Microsoft Dynamics</p> <p>Monday CRM</p> <p>Zoho CRM</p> <p>.....الخ</p>
--	---

الشركاء الأساسيون Key Partners

طبيعة الشراكة	معلومات حول الشركاء	الشركاء
Fournir le matériel technique et l'équipement nécessaires à la réalisation du projet, tels que les ordinateurs, les navigateurs, les outils de programmation et les contrôleurs	1. Fournisseurs de technologie et d'équipement	الشريك الأول
Distribuer des produits finis aux magasins et aux utilisateurs finaux, assurer la disponibilité des produits sur les différents marchés et gérer les réseaux de	2. Distributeurs et agents	الشريك الثاني



vente au détail.		
Fournir des services de transport et de fret pour les matières premières et les produits finis entre différents sites, en veillant à ce que les marchandises soient livrées à temps et en toute sécurité.	3. PartnersLogistics (Entreprise de transport express)	الشريك الثالث
Fournir des financements et des prêts pour le développement de projets, fournir des services financiers tels que la gestion de comptes et des conseils financiers.	4. Partenaires financiers (Banque Nationale d'Algérie (BNA))	الشريك الرابع

structure Costs هيكل التكاليف

10000 دج	تكاليف التعريف بالمنتج أو المؤسسة Frais d'établissement
تكاليف الماء 1500 دج تكاليف الكهرباء 4000 دج تكاليف الهاتف 4000 دج	تكاليف الحصول على العدادات (الماء - الكهرباء) (..... Frais d'ouverture de compteurs (eaux-gaz-....)



تكاليف (التكوين - برامج الاعلام الالي المختصة) شراء برامج مساعدة 2000 دج	Logiciels, formations
20000 دج	Dépôt marque, brevet, modèle تكاليف براءة الاختراع و الحماية الصناعية و التجارية
10000 دج	Droits d'entrée تكاليف الحصول على تكنولوجيا او ترخيص استعمالها
0 دج	Achat fonds de commerce ou parts شراء الأصول التجارية أو الأسهم
10000 دج في شهر واحد في ثلاث سنوات : 360000 دج	Droit au bail الحق في الإيجار
3000 دج	Caution ou dépôt de garantie وديعة أو وديعة تأمين
2000 دج	Frais de dossier رسوم إيداع الملفات
15000 دج	Frais de notaire ou d'avocat تكاليف الموثق - المحامي -
18000 دج	Enseigne et éléments de communication تكاليف التعريف بالعلامة و تكاليف قنوات الاتصال
0 دج	Achat immobilier شراء العقارات
20000 دج	Travaux et aménagements الأعمال والتحسينات الاماكن
100000 دج	Matériel الآلات - المركبات - الاجهزة
10000 دج	Matériel de bureau تجهيزات المكتب



10000 دج	Stock de matières et produits تكاليف التخزين
500000 دج	Trésorerie de départ التدفق النقدي (الصندوق) الذي تحتاجه في بداية المشروع.

المجموع = 1099500 دج

نفقاتك أو التكاليف الثابتة الخاصة بمشروعك

10000 دج	Assurances التأمينات
10000 دج	Téléphone, internet الهاتف و الانترنت
3000 دج	Autres abonnements اشتراكات أخرى
60000 دج	Carburant, transports الوقود و تكاليف النقل
0 دج	Frais de déplacement et hébergement تكاليف التنقل و المبيت
الاجمال: 60000 دج في السنة 180000 دج خلال 3 سنوات	Eau, électricité, gaz فواتير الماء - الكهرباء - الغاز
8000 دج	Mutuelle التعاضدية الاجتماعية
10000 دج	Fournitures diverses لوازم متنوعة
10000 دج	Entretien matériel et vêtements صيانة المعدات والملابس



5000 دج	Nettoyage des locaux تنظيف المباني
شريك فالمشروع	Budget publicité et communication ميزانية الإعلان والاتصالات

المجموع = 296000 دج

Revenue Stream مصادر الإيرادات

500000 دج	Apport personnel ou familial المساهمة الشخصية أو العائلية
300000 دج	Apports en nature (en valeur) التبرعات العينية
1000000 دج بنك البركة الإسلامي	Prêt n°1 (nom de la banque) قرض رقم 1 اسم البنك
	Prêt n°2 (nom de la banque) قرض رقم 2 اسم البنك
	Prêt n°3 (nom de la banque) قرض رقم 3 اسم البنك
	Subvention n°1 (libellé) منحة 1
	Subvention n°2 (libellé) منحة 2
	Autre financement (libellé) تمويل آخر

المجموع = 1800000 دج



رقم الأعمال

Votre chiffre d'affaires de la première année بيع المنتج في السنة الأولى

متوسط أيام العمل في الشهر	بيع المنتج في السنة الأولى
20	40% 1Mois الشهر 5
20	71,42% 2Mois الشهر 7
20	66,66% 3Mois الشهر 12
20	25% 4Mois الشهر 20
20	20% 5Mois الشهر 25
20	16% 6Mois الشهر 30
20	14% 7Mois الشهر 35
20	12,5% 8Mois الشهر 40
20	11% 9Mois الشهر 45
20	10% 10Mois الشهر 50
20	9,09% 11Mois الشهر 55
20	16,6% 12Mois الشهر 60

المجموع = 384 الة

النسبة المئوية للزيادة في حجم الأعمال بين كل شهر لسنة الأولى؟

Votre chiffre d'affaires de la deuxième année بيع المنتج في السنة الثانية

متوسط أيام العمل في الشهر	بيع المنتج في السنة الثانية
20	16,6% 1Mois الشهر 70
20	14,28% 2Mois الشهر 80
20	12,5% 3Mois الشهر 90
20	100 الشهر 4Mois
20	120 الشهر 5Mois



20	140 الشهر 6 Mois
20	160 الشهر 7 Mois
20	180 الشهر 8 Mois
20	200 الشهر 9 Mois
20	220 الشهر 10 Mois
20	250 الشهر 11 Mois
20	280 الشهر 12 Mois

المجموع = 1890

النسبة المئوية للزيادة في حجم الأعمال بين كل شهر لسنة الثانية؟

بيع المنتج في السنة الثالثة **Votre chiffre d'affaires de la troisième année**

متوسط أيام العمل في الشهر	بيع المنتج في السنة الثالثة
20	300 الشهر 1 Mois
20	320 الشهر 2 Mois
20	340 الشهر 3 Mois
20	360 الشهر 4 Mois
20	380 الشهر 5 Mois
20	400 الشهر 6 Mois
20	420 الشهر 7 Mois
20	440 الشهر 8 Mois
20	450 الشهر 9 Mois
20	460 الشهر 10 Mois
20	470 الشهر 11 Mois
20	500 الشهر 12 Mois



المجموع = 4840

النسبة المئوية للزيادة في حجم الأعمال بين كل شهر لسنة الثالثة ؟

تطور حجم رقم الأعمال في السنة

- النسبة المئوية للزيادة في حجم الأعمال بين السنة 1 والسنة 2؟
- النسبة المئوية للزيادة في حجم الأعمال بين السنة 2 والسنة 3؟

حاجتك لرأس المال العامل

30 يوم	متوسط مدة الاعتمادات الممنوحة للعملاء بالأيام Durée moyenne des crédits accordés aux clients en jours
30 يوم	متوسط مدة ديون الموردين بالأيام Durée moyenne des dettes fournisseurs en jours

رواتب الموظفين و مسؤولين الشركة

$120000 = 30000 * 4$ دج في الشهر 1440000 دج في السنة 4320000 دج	رواتب الموظفين Salaires employés
56000 دج 2016000 دج	صافي أجور المسؤولين Rémunération nette dirigeant

Business Model Canvas

Designed for:

Designed by:

Date:

Version:

Key Partners

Who are our Key Partners? Who are our key suppliers? Which Key Resources are we acquiring from partners? Which Key Activities do partners perform?

MOTIVATIONS FOR PARTNERS: Optimization and economy, Reduction of risk and uncertainty, Acquisition of particular resources and activities

Key Activities

What Key Activities do our Value Propositions require? Our Distribution Channels? Customer Relationships? Revenue streams?

CATEGORIES: Production, Problem Solving, Platform/Network

Key Resources

What Key Resources do our Value Propositions require? Our Distribution Channels? Customer Relationships? Revenue Streams?

TYPES OF RESOURCES: Physical, Intellectual (brand patents, copyrights, data), Human, Financial

Value Propositions

What value do we deliver to the customer? Which one of our customer's problems are we helping to solve? What bundle of products and services are we offering each Customer Segment? Which customer needs are we satisfying?

CHARACTERISTICS: Newness, Performance, Customization, "Get the Job Done", Design, Brand/Status, Cost Reduction, Risk Reduction, Accessibility, Convenience/Usability

Customer Relationships

What type of relationship does each Customer Segment expect us to establish and maintain with them? Which ones are we established? How are they interacting with the rest of our business model? How costly are they?

Channels

Through which Channels do our Customer Segments want to be reached? How are we reaching them now? How are our Channels integrated? Which ones are best? Which ones are most cost-effective? How are we integrating them with our routines?

Customer Segments

For whom are we creating value? Who are our most important customers? Is our customer base a Mass Market, Niche Market, Segmented, Diversified, Multisided Platform?

Cost Structure

What are the most important costs inherent in our business model? Which Key Resources are most expensive? Which Key Activities are most expensive?

BUSINESS MODELS: Cost Driven (leanest cost structure, low price value proposition, mass automation, extensive outsourcing), Value Driven (focused on value creation, premium proposition).

SAMPLE CHARACTERISTICS: Fixed Costs (salaries, rents, utilities), Variable costs, Economies of scale, Economies of scope

Revenue Streams

For what value are our customers really willing to pay? For what do they currently pay? How are they currently paying? How would they prefer to pay? How much does each Revenue Stream contribute to overall revenues?

TYPES: Transactional sale, Usage fee, Subscription Fees, Lending/Renting/Leasing, Licensing, Brokerage fees, Advertising

FIXED PRICING: List Price, Product feature dependent, Customer segment dependent

DYNAMIC PRICING: Negotiation (bargaining), Yield Management, Real-time-Market

Business Model Canvas

Designed for:

Designed by:

Date:

Version:

Key Partners

- Fournisseurs de technologie et d'équipement
- Distributeurs et agents
- PartnersLogistics (Entreprise de transport express)
- Partenaires financiers (Banque Nationale d'Algérie (BNA))
- Partenaires marketing et publicitaires

Key Activities

Le projet vise à Développer un système intelligent d'incubation d'œufs qui utilise l'intelligence artificielle pour contrôler les conditions d'incubation.

- Élaborer une stratégie marketing complète comprenant l'analyse du marché, l'identification du public cible et l'élaboration de messages marketing.
- Organiser les processus d'expédition et de livraison pour s'assurer que les produits parviennent aux clients à temps et en bon état.
- Fournir une assistance technique aux clients pour les aider à installer et à utiliser le système, et résoudre les problèmes techniques qu'ils peuvent rencontrer.
- S'adapter à l'évolution du marché et de la technologie grâce à l'innovation continue et à l'optimisation des produits.

Value Propositions

- Paramètres personnalisables : Les utilisateurs peuvent ajuster les paramètres d'incubation en fonction du type d'œuf, de la durée et de la température d'incubation.
- Coût abordable : En offrant un produit automatisé à un prix compétitif, le projet vise à rendre la technologie accessible aux couturiers amateurs et professionnels sans compromettre la qualité.
- Facilité d'utilisation : Interface utilisateur intuitive et design ergonomique qui rendent le système facile à utiliser, même pour les personnes ayant peu de compétences techniques.

Customer Relationships

Fournir un service d'assistance technique 24 heures sur 24 pour aider les clients à résoudre les problèmes techniques liés au système.

Fournir des lignes d'assistance par téléphone et par courrier électronique pour interagir directement avec les clients et répondre rapidement à leurs questions.

Organiser des ateliers et des cours de formation pour apprendre aux clients à utiliser efficacement le système et s'assurer qu'ils tirent le meilleur parti des fonctionnalités disponibles.

Envoyer des notifications et des alertes aux clients concernant les nouvelles mises à jour, les offres spéciales et la maintenance régulière.

Proposer des programmes de récompense pour les clients réguliers, comprenant des réductions, des cadeaux et des offres spéciales.

Créer un système de points permettant aux clients de collecter des points lors de l'achat de produits ou de l'utilisation de services, et de les échanger contre des récompenses.

Fournir des services après-vente tels que l'entretien régulier, le remplace-

Customer Segments

les amateurs d'élevage de poules et de production d'œufs

Éleveur professionnel de poulets

grands élevages de poules et d'œufs

ment des pièces endommagées et la mise à niveau du système

Key Resources

Ressources humaines (ingénieurs et développeurs, équipe de vente et de marketing, équipe dédiée à l'assistance technique et à la maintenance
Ressources humaines (ingénieurs et développeurs, équipe de vente et de marketing, équipe dédiée à l'assistance technique et à la maintenance des clients après la vente, etc.)

Ressources techniques (matériel de développement, logiciels de développement, infrastructure informatique)

Ressources financières (capital, financement durable)

Ressources physiques (composants électroniques, équipement de fabrication)

• Channels

- Fournir une plateforme de commerce électronique sur le site officiel de l'entreprise, où les clients peuvent acheter le produit directement en ligne.
- Travailler avec des grossistes pour distribuer le produit en grandes quantités, ce qui permet d'atteindre des marchés plus larges et d'augmenter le volume des ventes.
- Utiliser les plateformes de médias sociaux telles que Facebook, Instagram et Twitter pour promouvoir le produit et interagir avec les clients potentiels.
- Utiliser la publicité payante en ligne telle que Google AdWords et la publicité sur les médias sociaux pour atteindre un public plus large.
- Établir des centres de service

et de maintenance dans des endroits stratégiques pour fournir une assistance technique et des services après-vente aux clients.

- Collaborer avec les établissements d'enseignement qui proposent des cours de couture et de design afin de fournir le produit dans le cadre des outils pédagogiques.

Cost Structure

Revenue Streams

ملخص

تتعمق هذه الدراسة في آليات وتحديات حضانة البيض، وهي ممارسة حيوية في صناعة الدواجن. تؤكد الأبحاث على أهمية التحكم الدقيق في مختلف المعايير مثل درجة الحرارة والرطوبة لضمان ظروف مثالية لحضانة البيض. من خلال الاستكشاف، أصبح واضحاً أن الذكاء الاصطناعي، وخاصة شبكات الذاكرة طويلة وقصيرة المدى يقدم حلاً واعداً لتحسين هذه العمليات. يمكن لهذه الخوارزميات العميقة التنبؤ بالاتجاهات بدقة والتحكم بشكل ديناميكي في ظروف الحضانة، مما يؤدي إلى إدارة أكثر فعالية واستباقية لبيئة الحضانة. من خلال تطبيق هذه المفاهيم النظرية، تم تطوير نظام حضانة بيض مبتكر، يدمج بشكل كامل مزايا الذكاء الاصطناعي للإدارة الآلية والمُحسنة. أكدت التحليلات المتعمقة للنتائج فعالية هذا النهج. في النهاية، تقدم هذه الدراسة رؤى قيمة حول التقارب بين ممارسات الحضانة التقليدية والتكنولوجيا المتقدمة، مما يمهّد الطريق لمزارع دواجن أكثر فعالية واستدامة.

كلمات مفتاحية: حضانة البيض، التحكم في درجة الحرارة، التحكم في الرطوبة، الذكاء الاصطناعي، شبكات الذاكرة طويلة وقصيرة المدى، صناعة الدواجن

Abstract

This study delves into the mechanisms and challenges of egg incubation, a crucial practice in the poultry industry. The research emphasizes the importance of meticulously controlling various parameters such as temperature and humidity to ensure optimal conditions for egg incubation. Throughout the exploration, it becomes evident that artificial intelligence, particularly Long Short-Term Memory (LSTM) networks, offers a promising solution for optimizing these processes. These deep learning algorithms can accurately predict trends and dynamically control incubation conditions, leading to more efficient and proactive management of the incubation environment. By implementing these theoretical concepts, an innovative egg incubation system was developed, fully integrating the advantages of AI for automated and optimized management. An in-depth analysis of the results confirmed the effectiveness of this approach. Ultimately, this study provides valuable insights into the convergence of traditional incubation practices and cutting-edge technologies, paving the way for more efficient and sustainable poultry farming.

Key words: Egg Incubation, Temperature Control, Humidity Control, Artificial Intelligence, LSTM, Poultry Industry

Résumé

Cette étude approfondie sur l'incubation des œufs explore les mécanismes et les défis de cette pratique cruciale dans l'industrie avicole. En soulignant l'importance capitale du contrôle minutieux de divers paramètres tels que la température et l'humidité, nous visons à garantir des conditions optimales pour l'incubation des œufs. À mesure que nous progressions dans notre exploration, nous avons constaté que l'intelligence artificielle, en particulier les réseaux neuronaux LSTM, offraient une solution prometteuse pour optimiser ces processus. Ces algorithmes d'apprentissage profond ont le potentiel de prédire avec précision les tendances et de contrôler les conditions d'incubation de manière dynamique, permettant ainsi une gestion plus efficace et proactive de l'environnement d'incubation. En concrétisant ces concepts théoriques, nous avons développé un système d'incubation d'œufs innovant, intégrant pleinement les avantages de l'IA pour une gestion automatisée et optimisée. L'analyse approfondie des résultats obtenus a confirmé l'efficacité de notre approche. En fin de compte, cette étude offre un aperçu précieux sur la convergence entre l'incubation traditionnelle et les technologies de pointe, ouvrant la voie à une aviculture plus efficace et durable.

Mot clés : Incubation des œufs, Contrôle de la température, Contrôle de l'humidité, Intelligence artificielle, LSTM, Industrie avicole