PREDICCIÓN DE BROTES DE GRIPE AVIAR
AVIAN INFLUENZA OUTBREAKS PREDICTION

TRABAJO FIN DE GRADO
CURSO 2020-2021

AUTOR
CHENG JUN LIU ZHENG
EMILIO JOSÉ VALENCIA CALVOPIÑA

DIRECTORES
JOSÉ IGNACIO GÓMEZ PÉREZ
CHRISTIAN TOMÁS TENLLADO VAN DER REIJDEN

GRADO INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

PREDICCIÓN DE BROTES DE GRIPE AVIAR
AVIAN INFLUENZA OUTBREAKS PREDICTION

TRABAJO DE FIN DE GRADO EN INGENIERÍA INFORMÁTICA
DEPARTAMENTO DE ARQUITECTURA DE COMPUTADORES Y AUTOMÁTICA

AUTOR
CHENG JUN LIU ZHENG
EMILIO JOSÉ VALENCIA CALVOPIÑA

DIRECTORES
JOSÉ IGNACIO GÓMEZ PÉREZ
CHRISTIAN TOMÁS TENLLADO VAN DER REIJDEN

CONVOCATORIA: JUNIO 2021
CALIFICACIÓN: NOTA

GRADO EN INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

DÍA DE MES DE AÑO

DEDICATORIA

A mi familia, que me ha soportado durante todo este proceso y más tiempo aún. – Cheng Jun Liu Zheng

A mi familia por haberme brindado la oportunidad de estudiar gracias a su gran esfuerzo. A mis vecinos por su constante apoyo y a mi pareja por alentarme a ser mejor, por su cariño y apoyo en todo momento. -Emilio José Valencia Calvopiña

AGRADECIMIENTOS

A Nacho y a Christian por guiarnos y ayudarnos tanto hacia el buen camino por el que seguir el proyecto. A Irene Iglesias por apoyarnos tanto y hacer posible el desarrollo del modelo y a Carlos Eduardo por la interfaz web y por tener tanta paciencia con nuestros ficheros GeoJSON.

RESUMEN

Predicción de brotes de Gripe Aviar

Los brotes por Gripe Aviar, concretamente las variaciones de alta patogenicidad de este virus son un problema que afecta frecuentemente a toda Europa, especialmente en países ubicados al norte donde las temperaturas son más bajas lo que les convierte en entornos ideales para la supervivencia del virus.

Durante este trabajo se ha desarrollado una herramienta para poder predecir en qué comarcas ganaderas de España hay un mayor riesgo de que ocurra un brote, teniendo en cuenta brotes de Europa, tomados de la plataforma de Empres-i y proporcionados por la organización de WAHIS, rutas migratorias de diferentes especies de aves que suelen realizar trayectos entre partes de Europa y España, proporcionadas por la Sociedad Española de Ornitología (SEO) y previsiones de la temperatura para cada comarca ganadera.

La herramienta además generará reportes con información pertinente a las posibles alertas de cada comarca ganadera en formato PDF y CSV, que se enviarán a una unidad en Google Drive para su almacenamiento.

Todas estas tareas se realizarán con una frecuencia semanal y además se actualizarán los brotes que vayan reportándose en Empres-i, con la misma frecuencia.

Palabras clave

Gripe Aviar, Python, Neo4j, MongoDB, Markdown, SEO Birdlife, Geohash, Web-scraping, Empres-I, Brotes

ABSTRACT

Avian Influenza Outbreaks Prediction

Outbreaks due to Avian Influenza, specifically the highly pathogenic variations of this virus, are a problem that frequently affects all of Europe, especially in countries located to the north where temperatures are lower, which makes these environments ideal for the survival of the virus.

During this project, a tool has been developed to be able to predict in which livestock regions of Spain there is a greater risk of an outbreak occurring, considering outbreaks in Europe, downloaded from the Empres-i platform, and provided by the WAHIS organization, migratory routes of different species of birds that usually travel between parts of Europe and Spain, provided by the Spanish Ornithological Society (SEO) and temperature forecasts for each livestock region.

The tool will also generate reports with information relevant to the possible alerts of each livestock region in .pdf and .csv format, which will be sent to a Google drive unit for storage.

All these tasks will be carried out on a weekly basis and the outbreaks that are reported in Empres-i will also be updated with the same frequency.

Keywords

Avian Influenza, Python, Neo4j, MongoDB, Markdown, SEO Birdlife, Geohash, Web-scraping, Empres-I, Outbreaks

ÍNDICE DE CONTENIDOS

Dedicatoria V

Agradecimientos VII

Resumen IX

Abstract XI

Índice de contenidos XIII

Índice de figuras XVII

Índice de tablas XIX

Índice de códigos XXI

Capítulo 1 - Introducción 1

1.1 Motivación 1

1.2 Objetivos 3

1.3 Plan de trabajo 3

Capítulo 2 - Estado de la cuestión 5

2.1 Obtención de las temperaturas mínimas 6

2.2 Obtención de datos sobre brotes 7

2.3 Obtención de datos sobre rutas migratorias 8

2.4 Tecnologías para la implementación de la herramienta 8

2.4.1 Bases de datos 9

2.4.2 Lenguajes de programación 9

Capítulo 3 - Extracción y almacenamiento de los datos 11

3.1 Geohash 11

3.2 Extracción y almacenamiento de brotes 13

3.2.1 WAHIS-OIE y Empres-i 13
3.2.2 Almacenamiento de los brotes en MongoDB 15
3.3 Extracción y almacenamiento de rutas migratorias 16
3.3.1 SEO Birdlife 17
3.3.2 Almacenamiento de rutas en MongoDB 17
3.3.3 Almacenamiento de rutas en Neo4j 18
3.4 Extracción y almacenamiento de la temperatura mínima 23
3.4.1 AEMET OpenData 23
3.4.2 TuTiempo.net 25
3.4.3 Almacenamiento de temperaturas en MongoDB 27
Capítulo 4 - Aplicación del modelo y generación de alertas 31
4.1 Actualización de datos sobre brotes y temperaturas 31
4.2 Implementación del modelo 31
4.3 Archivos generados por la herramienta 35
4.3.1 Reportes 35
4.4 Interfaz web mediante ArcGIS 38
4.4.1 GeoJSON sobre alertas, rutas y brotes 39
Capítulo 5 - Conclusiones y trabajo futuro 43
5.1 Conclusiones 43
5.2 Trabajo futuro 43
Chapter 6 - Introduction 45
6.1 Motivation 45
6.2 Objectives 46
6.3 Workplan 47

Chapter 7 - Conclusions and Future Work 49

7.1 Conclusions 49

7.2 Future work 49

Capítulo 8 - Contribuciones 51

8.1 Emilio José Valencia Calvopiña 51

8.2 Cheng Jun Liu Zheng 53

Bibliografía 57

Apéndices 59

ÍNDICE DE FIGURAS

Figura 2.1 Mapa con los brotes de gripe aviar en Europa en el año 2021 5

Figura 2.2 Lenguajes más usados según Stack Overflow en 2020 10

Figura 3.1 Mapa mundial con cuadrículas de geohash 12

Figura 3.2 Reporte con información de un brote en el antiguo WAHIS-OIE 14

Figura 3.3 Página con la información de un brote en Empres-i 15

Figura 3.4 Primera versión de estructura en Neo4j 19

Figura 3.5 Solapamiento entre cuadrículas de comarca y geohash 20

Figura 3.6 Segunda versión de estructura en Neo4j 21

Figura 3.7 Tercera versión de estructura en Neo4j 22

Figura 3.8 Operaciones de valores-climatológicos 23

Figura 3.9 Ejemplo de consulta 24

Figura 3.10 URL datos 24

Figura 3.11 URL de metadatos 25

Figura 3.12 Consulta con latitud y longitud 26

Figura 3.13 Ejemplo salida JSON (TuTiempo.net) 26

Figura 4.1 Estudio con curva ROC del modelo 34

Figura 4.2 Mapa con los niveles de alerta de enero 2021 y comarcas donde hubo brotes 35

Figura 4.3 Cabecera y sumario PDF 36

Figura 4.4 Tabla de alertas PDF 37

Figura 4.5 Tabla de brotes PDF 37

Figura 4.6 Alertas CSV 37

Figura 4.7 Brotes (1) CSV 37

Figura 4.8 Brotes (2) CSV 37

Figura 4.9 Aplicación web para la visualización de alertas, rutas y brotes 39

Figura 4.10 Aplicación web mostrando varias rutas migratorias asociadas a brotes..... 41

ÍNDICE DE TABLAS

Tabla 3.1 Dimensiones de una cuadrícula geohash según la longitud 13

ÍNDICE DE CÓDIGOS

Código 1 Documento de un brote en MongoDB 16

Código 2 Documento de una ruta en MongoDB 18

Código 3 Documento de una estación en mongo 28

Código 4 Documento del histórico de una estación en mongo 29

Código 5 Documento del histórico final y la predicción de una comarca en Mongo .. 30

Código 6 Ejemplo de una alerta en GeoJSON 40

Código 7 Ejemplo de una ruta en GeoJSON 40

Código 8 Ejemplo de brote en GeoJSON 41

Capítulo 1 - Introducción

La Gripe Aviar es una enfermedad que afecta a las aves silvestres y domésticas, normalmente de forma asintomática debido a que en la mayoría de los casos se trata de gripe aviar de baja patogenicidad. Las aves silvestres acuáticas (gaviotas, cigüeñas, patos, etc.) son los principales huéspedes de este virus y también son las que más pueden transmitir el virus en términos de distancia.

Por otro lado, la variante altamente patógena puede provocar graves síntomas, pudiendo provocar la muerte del ave con facilidad y además es altamente contagiosa. Un contagio con este tipo de virus por parte de un ave en una granja puede extenderse muy fácilmente a todas las demás aves del recinto, al no haber mucho espacio entre unas y otras por lo general. Es esta la variante con más posibilidad de contagiar a un ser humano, en concreto los subtipos H5N1 Y H7N9, de nuevo pudiendo provocar graves síntomas en humanos.

Las epidemias generadas por este virus aparecen en la mayoría de los casos en el continente asiático y europeo. Dentro de Europa ocurren con mayor frecuencia en países localizados al norte, debido a que el virus es capaz de sobrevivir más tiempo en el medio ambiente con temperaturas bajas. Sin embargo y por desgracia, España no está exenta de este virus con ejemplos de brotes en 2017, y en 2020 y 2021 dentro de una nueva ola de este virus.

1.1 Motivación

La avicultura tiene una gran importancia para la economía española. De hecho, España se consolidó en 2019 como el cuarto país de Europa que más cantidad de carne produce en las más de 18.000 explotaciones que existen en el país[1]. Con el comienzo de la pandemia del COVID-19 todos los sectores se vieron perjudicados. A pesar de esta crisis provocada por la pandemia, los sectores agrícolas, ganaderas y pesqueras

[1] lamoncloa.gob.es

tuvieron un repunte del PIB de un 4,7% en 2020[2]. Y con respecto a la industria avícola española, esta genera alrededor de 2.300 millones de euros al PIB nacional y aproximadamente genera alrededor de unos 40.000 puestos de empleo[3].

Además, España es una gran exportadora de carne de ave. En estos años, se produjo un aumento del 2,5% del número de ventas en comparación a la década anterior. Los principales países a los que se les vende esta carne de ave son los países de la Unión Europea, de estos destacan los países vecinos (Francia y Portugal)[4].

Hay que destacar que a lo largo de todo el año se producen migraciones de diferentes especies de aves provenientes de países de Europa elevando la probabilidad de contagio en España y poniendo en peligro el sector avícola.

Las consecuencias de un brote de gripe aviar en España no son solo sanitarias sino también económicas pudiendo ser peligrosamente catastróficas. Los primeros afectados serían los avicultores y agricultores, puesto que esta enfermedad es altamente contagiosa entre las aves, obligando a sacrificar gran parte de las aves del recinto por razones de salud, suponiendo una gran pérdida económica para ellos. Además, de lo mencionado anteriormente, podría tener grandes repercusiones en la economía del país e incluso en Europa.

En cuanto a las consecuencias sanitarias, la variante de alta patogenicidad es muy letal en las aves, por otro lado, las infecciones en humanos son poco probables, pero siguen pudiendo causar graves síntomas.

Por estos motivos, es necesario tener un mecanismo para predecir y tener el tiempo suficiente para tomar las medidas necesarias y evitar pérdidas económicas sobre todo en la situación actual donde todo el país está mermado tras la crisis provocada por el COVID-19.

---

[2] efeagro.com

[3] avicultura.com

[4] https://avicultura.com/espana-las-exportaciones-de-carne-de-ave-cuadriplican-a-las-importaciones-en-cinco-anos/

1.2 Objetivos

El objetivo de este proyecto consiste en la elaboración de una herramienta capaz de predecir posibles brotes de gripe aviar en las comarcas ganaderas de España a partir de datos de brotes recientes en Europa, rutas migratorias de distintas especies y temperaturas para cada comarca dentro de un período semanal. Para esto se necesita:

- Una base de datos con brotes de Europa capaz de actualizarse periódicamente, así como una fuente de datos para extraer dicha información.
- Una base de datos con rutas migratorias de especies de aves importantes con respecto a la transmisión del virus en España.
- Una fuente de datos de donde recabar previsiones de temperaturas en las diferentes comarcas ganaderas y de donde tomar las temperaturas reales para fechas pasadas con el fin de almacenarlas en la base de datos, todo de forma periódica.
- Un modelo capaz de dar un nivel de alerta a cada comarca con los datos anteriores, además de poder generar un reporte con los datos tenidos en cuenta para calcular este nivel en cada alerta.

El código resultante se almacenará en este repositorio público de Github.

1.3 Plan de trabajo

Este es el plan de trabajo a seguir para el cumplimiento de los objetivos anteriormente detallados:

- Búsqueda y elección de las fuentes de datos: se analizarán posibles fuentes de datos para extraer cada tipo de dato necesario para el modelo. Una vez encontrados se procederá a elegir uno para cada tipo de dato.
- Diseño de los módulos software: se detallarán los módulos necesarios para la realización de los objetivos de este proyecto.
- Desarrollo de la herramienta: se analizarán tecnologías adecuadas para implementar los distintos módulos.

- Evaluación del modelo: una vez se termine el desarrollo se procederá a evaluarlo, comparando alertas del modelo con brotes que ocurrieron en España.
- Elaboración de la memoria del proyecto: por último, se redactarán los procedimientos y los resultados obtenidos durante el proceso de desarrollo de este proyecto.

Capítulo 2 - Estado de la cuestión

Dentro de Europa, como ya se ha mencionado, los países más afectados están en el norte, algunos ejemplos son Dinamarca, Alemania, Países Bajos y Polonia. Muchas de las aves acuáticas en Europa provienen de Rusia y Asia, donde pueden contagiarse del virus durante su estancia en los grandes lagos. Estas aves siguen rutas migratorias que cruzan el norte de Europa y finalmente acaban en España. En 2020 se ha producido una nueva ola de gripe aviar que sigue afectando a todo el continente europeo y asiático hasta el día de hoy, en el año 2021.

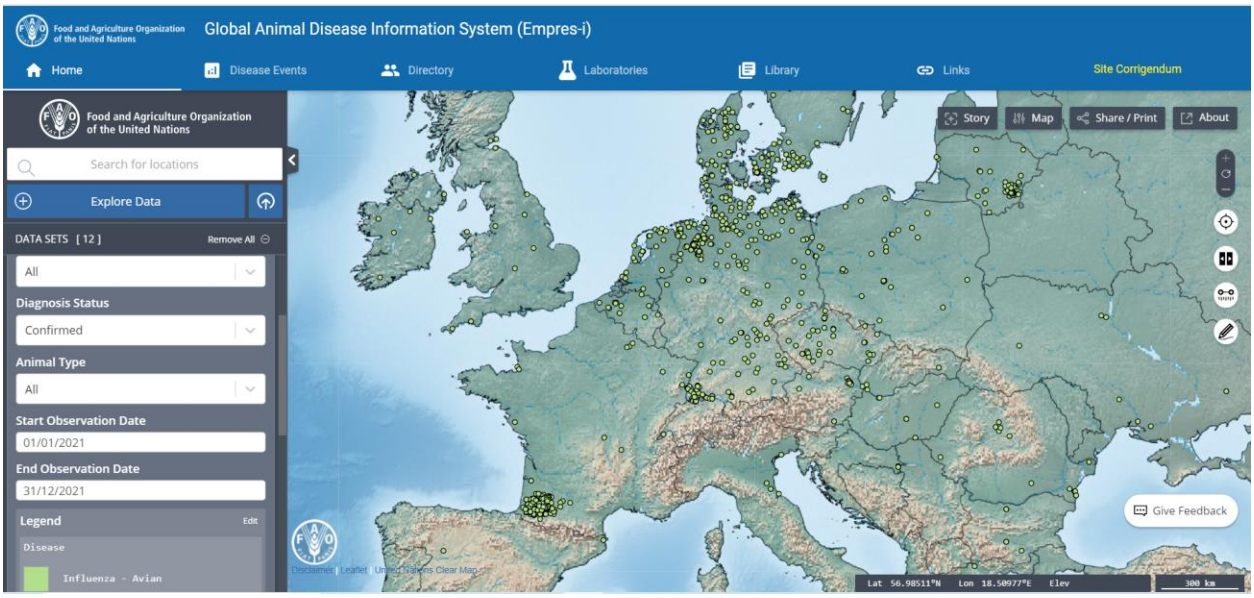


Figura 2.1 Mapa con los brotes de gripe aviar en Europa en el año 2021

Debido a esta ola, se han producido varios casos de contagio de gripe aviar de aves a humanos con la variante H5N8, principal variante en Europa como causante de varios de ellos. La pandemia del COVID-19 ha causado que gran parte de la atención internacional se concentre en ella, desviando la mirada de brotes de gripe aviar entre otras muchas. En este artículo[5] de la revista Science se alarma sobre el peligro de una nueva pandemia que siga a la ya tan dura que empezó a principios de 2020.

[5] Emerging H5N8 avian influenza viruses | Science ((Shi & F. Gao, 2021)sciencemag.org)

Este proyecto se basa en el TFM del curso 2018-19 realizado por Carlos Ballesteros de Andrés y titulado "*Conectografía de la gripe aviar: herramientas para predecir la difusión de la enfermedad*". Este proyecto de TFG sigue varias de las líneas de trabajo futuro enumeradas por Carlos en la memoria de dicho TFM.

A continuación, se detalla el estado actual del problema desde la perspectiva del proyecto.

2.1 Obtención de las temperaturas mínimas

La temperatura mínima es fundamental como parámetro para evaluar el riesgo de un posible brote en alguna comarca ganadera española debido a la supervivencia del virus en entornos con bajas temperaturas anteriormente mencionada. Para la obtención de dichos datos, hemos hecho uso de dos APIs:

- **AEMET OpenData**[6]: una API Rest proporcionada por AEMET, la Agencia Estatal de Meteorología de España. Esta API será usada con el fin de guardar las temperaturas mínimas de las estaciones meteorológicas desde el 2 de enero de 2017 hasta el día actual. Estos datos serán empleados con el fin de realizar estudios acerca de la efectividad del modelo en ese rango de tiempo. Esta API no proporciona predicciones de la temperatura para una semana, solo predicciones de las próximas 48 horas. Para cada comarca ganadera se le asigna la temperatura de la semana X en función de una lista de estaciones de la API de AEMET asociadas con dicha comarca ganadera según la cercanía. Lista proporcionada por la investigadora, Irene Iglesias.
- **API TuTiempo**[7]: el problema de las predicciones se resuelve con esta API, que nos proporciona la predicción de la temperatura mínima de los próximos 7 días. En este caso la predicción se realiza por coordenadas, al contrario de la API de AEMET, que presentaba un id único para cada estación meteorológica.

[6] https://opendata.aemet.es/dist/index.html?#/valores-climatologicos

[7] https://api.tutiempo.net/

Ambas APIs requieren de una "API Key" generada automáticamente al registrarse. Para la extracción de los datos, la propia AEMET ofrece un código de ayuda en Python usando la librería *requests*.

2.2 Obtención de datos sobre brotes

Es necesario disponer de una fuente de datos sobre brotes que sea confiable y en la medida de lo posible fácil de tratar y almacenar en nuestra base de datos. Existen varias plataformas de organizaciones oficiales con datos sobre brotes, pero la gran mayoría no ofrecen información a nivel internacional ni al menos a nivel europeo y tampoco lo ofrecen en un formato sencillo de procesar, tratándose normalmente de reportes en formato PDF. Al final, las opciones que se han encontrado son dos:

- **WAHIS-OIE**[8]: se trata del sistema de información zoosanitaria de la OIE (Organización Mundial Sanidad Animal) que recopila en una base de datos la información de brotes de enfermedades de origen animal recibida principalmente por los Servicios Veterinarios de todos los países del mundo ya sean pertenecientes o no a la OIE.
- **Empres-i**[9]: se trata del sistema global de información sobre enfermedades de origen animal, dirigido por la Organización de las Naciones Unidas para la Alimentación y la Agricultura. Ofrece una útil interfaz para buscar y representar datos sobre brotes, entre ellas las de gripe aviar. Los datos para esta enfermedad provienen justamente de la OIE.

Desde el inicio del proyecto y gracias al anterior trabajo de Carlos Ballesteros, WAHIS-OIE fue el proveedor de la información para este proyecto, pero en diciembre de 2020 cerró temporalmente su página web con el fin de mejorar la interfaz de esta. Ante este cierre y el desconocimiento sobre cuándo se volvería a reanudar dicha

[8] OIE-WAHIS

[9] Empres-Plus (fao.org)

página, se decidió buscar otra fuente de información relacionada con los brotes. Empres-i se convirtió en el remplazo hasta el final del proyecto.

2.3 Obtención de datos sobre rutas migratorias

La búsqueda de una fuente de datos sobre rutas migratorias resultó ser complicada. Existen portales web que permiten la visualización del movimiento de distintas especies de aves como **EuroBirdPortal**[10], proyecto en colaboración con varias organizaciones de ornitología, que permite ver un mapa con avistamientos a lo largo del tiempo en Europa. Sin embargo, ninguno de estos ofrece una forma de extraer estos datos ni de visualizarlos de otra manera aparte del mapa.

Gracias a la ayuda de Irene Iglesias, investigadora del INIA-CSIC[11], se consiguieron datos sobre posibles rutas migratorias provenientes de **SEO Birdlife**[12] (Sociedad Española de Ornitología) en formato CSV. Estos datos se consiguieron mediante la técnica de anillamiento, que consiste en colocar anillas, normalmente de aluminio, en aves llevando estas impreso información sobre dónde se anillaron y el centro de anillamiento que realizó el proceso entre otros datos. De esta forma si se encuentra a un individuo con una anilla, se puede extraer una ruta entre el punto donde se anilló y el punto en el que se ha capturado. Además, Irene proporcionó un fichero con las probabilidades semanales de migración de cada una de las aves a estudiar.

2.4 Tecnologías para la implementación de la herramienta

Las partes principales en la implementación de la herramienta se pueden enumerar en dos: almacenamiento de los datos y cálculo del modelo con estos datos.

---

[10] https://www.eurobirdportal.org/ebp/en/#home/

[11] identificación de la sede - csic.es

[12] SEO Birdlife - Sociedad Española de Ornitología

2.4.1 Bases de datos

Actualmente las bases de datos pueden clasificarse en dos grandes grupos:

- SQL o relacional: el tipo más popular de bases de datos durante muchos años. La información se estructura en tablas con un número determinado de columnas, representando la información que se puede guardar para cada fila. Las filas de una tabla se relacionan con otras mediante referencias a las demás tablas. Algunos ejemplos de este tipo son MySQL, PostgreSQL o SQL Server.
- NoSQL o no relacional: este tipo de base de datos ha tenido un gran auge de popularidad en los últimos años. Existen varios que almacenan la información en colecciones, en vez de tablas, donde la información se almacena en formato JSON. Sin embargo, hay muchos otros que guardan la información de distinta forma. Todas proporcionan mayor flexibilidad a la hora de estructurar los datos y escalabilidad con respecto a las SQL. Algunos ejemplos son MongoDB, Redis o Cassandra.

Debido a los constantes cambios que podrían surgir en la estructura de los datos, se optó por elegir una base de datos de tipo NoSQL basada en documentos, MongoDB. Gracias a su flexibilidad, lo cambios en los datos almacenados no supusieron un coste en términos de tiempo muy alto. Para la representación de rutas migratorias entre diferentes zonas de Europa, se hizo uso de otra base datos NoSQL llamada Neo4j, la cual se basa en grafos para almacenar información y hace de esta una estructura ideal para el almacenamiento y representación de rutas migratorias.

2.4.2 Lenguajes de programación

Existen muchos lenguajes de programación diferentes que se usan según el tipo de problema o tipo de plataforma en el que se va a desplegar. La popular plataforma

de **Stack Overflow** ofrece en una web[13] resultados de sus encuestas durante diferentes años.

Una gran parte de los encuestados en 2020 usa el lenguaje de programación **Python**, cuya popularidad ha aumentado sin parar a lo largo de los últimos años. Se trata de un lenguaje con una sintaxis sencilla, muy habitualmente usado en el campo del estudio de datos y que ofrece una amplia gama de librerías hechas por la comunidad, las cuales lo convierten en un lenguaje muy versátil y relativamente fácil de aprender, lo cual nos impulsó para usarlo como nuestro lenguaje de programación.

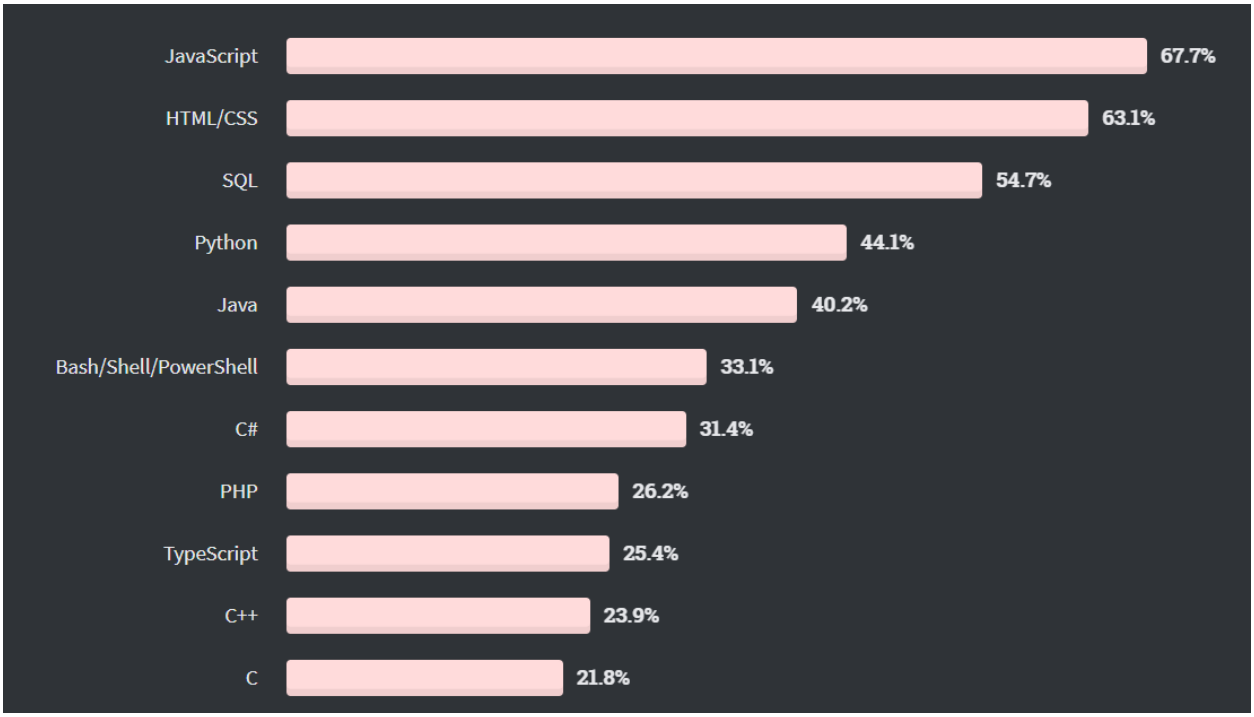


Figura 2.2 Lenguajes más usados según Stack Overflow en 2020

[13] Stack Overflow Developer Survey 2020

Capítulo 3 - Extracción y almacenamiento de los datos

En este capítulo se explicará con más detalle y profundidad el procedimiento de extracción de datos sobre brotes, rutas migratorias y temperaturas para las comarcas para su posterior almacenamiento.

Antes de esto se explicará la técnica de **geohashing**, la cual ha simplificado las operaciones para asociar zonas geográficas y rutas en el desarrollo del proyecto.

3.1 Geohash

Para poder realizar las asociaciones entre brote, ruta y comarca se necesitaba primero encontrar rutas en una zona alrededor de un brote y posteriormente verificar si el otro extremo de la ruta está en España.

La técnica del **geohash** consiste en dividir el mapa mundial en rectángulos, cada uno con un código para identificarlos, como se ve en la Figura 3.1, este proceso sigue de forma recursiva dentro de cada uno de los recuadros añadiendo un carácter al código geohash correspondiente cada vez que se baja un nivel. De esta forma podemos enmarcar varias coordenadas dentro de una zona geohash, es decir, podemos asociar brotes y rutas comparando los códigos de geohash que se calculan a partir de sus coordenadas.

Además, existen varias librerías de Python destinadas al uso de estos geohashes. Las dos librerías usadas en el proyecto fueron *Geolib*[14] y *Geopy*[15], las cuales permiten calcular el geohash asociado a un punto geográfico representado por los valores de latitud y longitud entre otras funciones.

[14] geolib · PyPI

[15] geopy · PyPI

Figura 3.1 Mapa mundial con cuadrículas de geohash

Al inicio del proyecto, se especificó que un brote se asociaría con las rutas en las que alguno de sus extremos estuviera localizado dentro de un radio de 25 km alrededor de dicho brote, para emular este radio se optó por usar geohash con longitud de 4 caracteres cuya cuadrícula tiene unas dimensiones de 39 km x 19.5 km.

Longitud en caracteres	Dimensiones de la cuadrícula
1	5004km x 5004km
2	1251km x 625km
3	156km x 156km
4	39km x 19.5km
5	4.9km x 4.9km

Tabla 3.1 Dimensiones de una cuadrícula geohash según la longitud

3.2 Extracción y almacenamiento de brotes

Como se explicó anteriormente, la página de WAHIS-OIE fue la fuente de datos de brotes al principio del proyecto. Su portal para la búsqueda de brotes contiene la información más completa, pues es la fuente de datos de la otra opción que estudiamos, el portal de Empres-i.

3.2.1 WAHIS-OIE y Empres-i

Hoy en día, una práctica muy habitual en este tipo de portales web es la de ofrecer una API para el acceso a sus datos, pero este no fue el caso. El buscador sobre brotes tampoco ofrecía ninguna forma de descargar los datos, por lo que se optó por aplicar la técnica del ***Web scraping***.

Esta técnica consiste en emular las peticiones que haría una personal real a una web y recoger los datos de la página que devolviese esta petición, todo de forma automática. El lenguaje de Python permite trabajar con la librería *requests* para realizar estas peticiones y la información se puede extraer mediante expresiones regulares, las cuales nos permiten buscar por patrones en la página del brote.

Como ya se mencionó en el anterior capítulo, WAHIS-OIE sería la fuente de datos inicial del proyecto. Se estudió el proceso con el que se notificaban casos, con el fin de entender dónde se encontraba la información que se necesitaba extraer. Cuando un brote es detectado, el servicio veterinario que lo haya descubierto notifica a WAHIS-OIE generando un **reporte**, el cual contiene información sobre dicho brote o brotes. A cada brote se le asigna un identificador único y se añaden a la base de datos de la

organización, ofreciendo acceso a esta información mediante su portal web. Esta información incluye campos como el país y localidad donde se produjo el brote, la fecha en la que se hizo el reporte, el serotipo del virus o el número de casos avistados.

Highly pathogenic avian influenza, Libya

Print Close

Information received on 12/03/2014 from Dr Abdunaser Dayhum, Director General, National Centre of Animal Health, Ministry of Agriculture, Animal & Marine Wealth , Zawia, Libya

Summary

Report type	Immediate notification
Date of start of the event	04/03/2014
Date of pre-confirmation of the event	09/03/2014
Report date	12/03/2014
Date submitted to OIE	12/03/2014
Reason for notification	First occurrence of a listed disease
Manifestation of disease	Clinical disease
Causal agent	Highly pathogenic avian influenza virus
Serotype	H5N1
Nature of diagnosis	Clinical, Laboratory (basic), Laboratory (advanced), Necropsy
This event pertains to	the whole country

New outbreaks (1)

Outbreak 1	Kroum Alkhail, TUBRUQ
Date of start of the outbreak	04/03/2014
Outbreak status	Resolved (06/03/2014)
Epidemiological unit	Farm
Affected animals	
Affected population	45 local backyard chickens and 5 ducks

Species	Susceptible	Cases	Deaths	Destroyed	Slaughtered
Birds	50	19	15	35	0

Figura 3.2 Reporte con información de un brote en el antiguo WAHIS-OIE

Sin embargo, en diciembre de 2020 el portal de WAHIS-OIE **cerró** temporalmente para mejorar la interfaz de la web y usar tecnologías más modernas en ella, tras lo cual se publicó el nuevo portal[16] 2 meses después.

La técnica *Web scraping* requiere del estudio previo de la estructura de una página para poder extraer la información de ella. Estimando el tiempo que se necesitaría para volver a realizar el proceso con la nueva estructura, se decidió usar el portal de Empres-i el cual permite descargar los datos desde 2017 en un fichero CSV. En este fichero aparecen muchos de los datos que ofrecía WAHIS-OIE como el país, el

[16] OIE-WAHIS

serotipo del virus o el número de casos, pero carecían de la unidad epidemiológica asociada al brote la cual indicaba si este se había producido en un entorno doméstico o más bien entre aves silvestres. Empres-i también ofrece acceso a la información sobre brotes en su web de forma similar a como lo hacía WAHIS-OIE y con una página sencilla de comprender, lo cual no era de extrañar pues WAHIS-OIE es su fuente de información en materia de brotes de gripe aviar. Por lo que se volvió a usar *Web scraping* para recolectar esta información, apoyándonos en el campo *Disease Event ID,* el identificador único para cada brote, que aparecía en el fichero CSV para realizar las peticiones a su web.

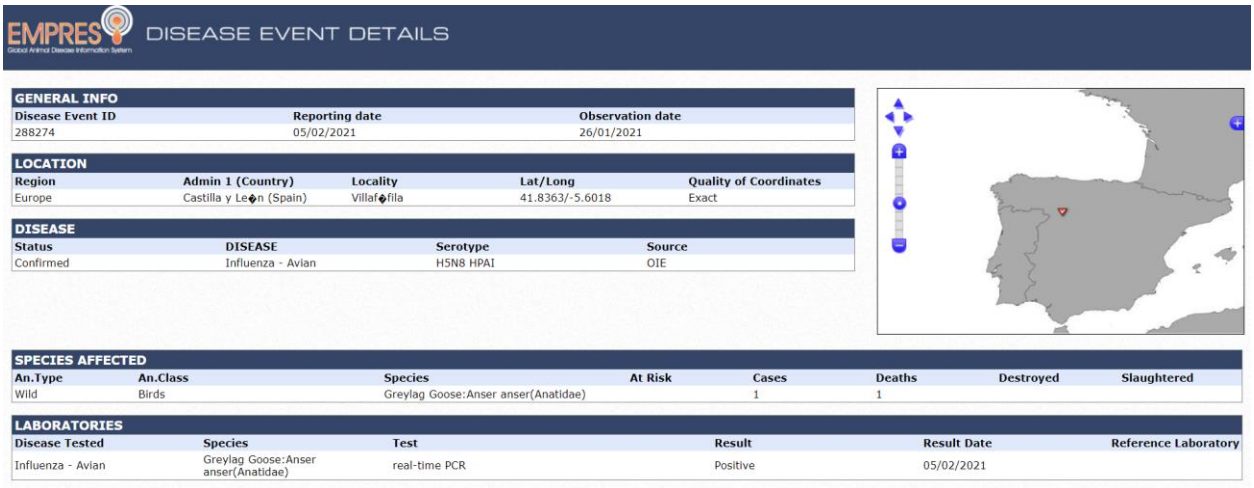


Figura 3.3 Página con la información de un brote en Empres-i

3.2.2 Almacenamiento de los brotes en MongoDB

Los brotes se almacenan en una colección llamada **outbreaks** dentro de MongoDB, donde cada documento sigue la estructura del Código 1. Los campos más importantes en cada documento son:

- Latitud y Longitud: las coordenadas donde se produjo el brote. Es necesario saber esta información para poder asociar un brote con una posible ruta migratoria con destino en España.
- Geohash: calculado a partir de las coordenadas del brote, necesario para simplificar la asociación entre rutas y brotes.
- Fecha de observación y reporte: necesario para delimitar en el tiempo los brotes que se tienen en cuenta para la aplicación del modelo.

- Epiunit: también llamado *Animal Type* en Empres-i, se trata del tipo de unidad epidemiológica que permite saber si el brote sucedió en un entorno salvaje, por ejemplo. Este puede ser *Wild*, *Backyard* o *Domestic*. Necesario en la aplicación del modelo.


```
_id: ObjectId("60858129a3ab48c04c678876")
oieid: 288291
disease: "Influenza - Avian"
serotype: "H5N8 HPAI"
lat: 53.4
long: 0.22
city: "Nr Louth"
country: "England"
region: "Europe"
location: "Nr Louth, U.K. of Great Britain and Northern Ireland, Europe"
observation_date: 2020-11-23T00:00:00.000+00:00
report_date: 2021-02-05T00:00:00.000+00:00
date: 2020-11-23T00:00:00.000+00:00
species: "Wild,Mute Swan:Cygnus olor(Anatidae)"
Diagnosis Source: "OIE"
Humans Affected: 0
Human Deaths: 0
cases: 1
deaths: 1
epiunit: "Wild"
geohash: "u12pv10jw37z"
urlFR: "http://empres-i.fao.org/empres-i/2/obd?idOutbreak=288291"
```


Código 1 Documento de un brote en MongoDB

3.3 Extracción y almacenamiento de rutas migratorias

Las aves acuáticas silvestres son los principales propagadores de la gripe aviar, transmitiendo el virus a través del agua de los lagos o humedales donde suelen habitar. Estas, según la época del año, migran hacia otras regiones del planeta siguiendo lo que se conocen como **rutas migratorias**. Las rutas migratorias son los desplazamientos que en el momento de la migración las aves realizan. Normalmente suelen seguir un patrón según la zona y las características geográficas de dicha zona.

Las aves que contraen la variante de alta patogenicidad del virus y posteriormente siguen estas rutas hasta España son la principal causa de brotes en España y esto es lo que la herramienta del proyecto busca predecir.

3.3.1 SEO Birdlife

Gracias a la Sociedad Española de Ornitología[17], a sus esfuerzos y a la investigadora Irene Iglesias, disponemos de un archivo CSV con datos sobre las rutas de más de 40 especies y más de 28 mil rutas conseguidas mediante la técnica de anillamiento, explicada anteriormente en la sección Obtención de datos sobre rutas migratorias del Capítulo 2 - .

3.3.2 Almacenamiento de rutas en MongoDB

Las rutas se almacenan en la colección **migrations** dentro de MongoDB, cada ruta se sigue la estructura del Código 2. Los campos más importantes en la estructura son:

- Latitud y Longitud: las coordenadas del extremo de la ruta que está fuera de España, necesario para la asociación entre rutas y brotes.
- Geohash: calculado a partir de las coordenadas anteriores, necesario para simplificar la asociación entre rutas y brotes.
- Especie: código de la especie de la ruta migratoria.
- Comarca_SG: tomado directamente del CSV, gracias a este campo se puede asociar una ruta con una comarca ganadera en España de forma directa.

[17] SEO Birdlife - Sociedad Española de Ornitología

```
_id: ObjectId("608bded13866ca5ed7edbbb4")
Id: 56356
Anilla: 0
Centro: "GBT"
InfoAnilla: 2
Especie: 5910
ESPECIE1: "Larus fuscus"
Sexo: "U"
FechaAnillamiento: 28666
Provincia: "GBSG10"
PAIS: "GB"
Lat: 56.16666667
Long: -4.166666667
CentroR: "GBT"
AnillaR: 0
EspecieR: 5910
especieR_n: "Larus fuscus"
ProvinciaR: "ES07509"
PAIS1: "ES"
Provincia2: "CANTABRIA"
Municipio1: "SANTANDER"
LatR: 43.45
LongR: -3.816666667
codigo_asu: 5910
COMARCA_SG: "SP39075"
COM_SGSA_N: "SANTANDER"
geohash: "gcvp90vft9s1"
geohashR: "eztr1juf95p1"
```


Código 2 Documento de una ruta en MongoDB

3.3.3 Almacenamiento de rutas en Neo4j

Neo4j se trata de otra base de datos NoSQL como MongoDB. Sin embargo, Neo4j tiene la particularidad de almacenar datos, no en documentos, sino en grafos.

Para este proyecto, Neo4j se ha utilizado para almacenar una ruta como una relación entre dos nodos, ambos representando zonas geográficas. Esta base de datos se ha reestructurado varias veces para intentar mejorar el rendimiento y precisión al consultar rutas asociadas a un brote.

3.3.3.1 Zonas de geohash como nodos

Esta primera versión constaba de un solo tipo de nodo, los nodos geohash. Estos contenían solamente el geohash calculado a partir de las coordenadas de cada extremo de la ruta. Las relaciones entre estos nodos representaban las rutas que asocian una cuadrícula geohash con otra.

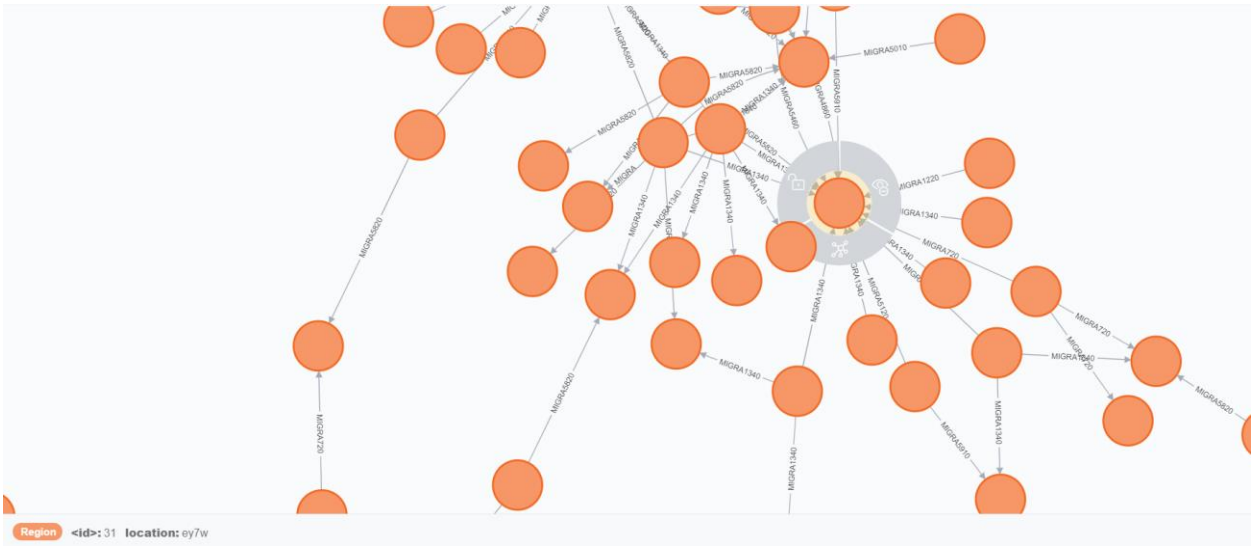


Figura 3.4 Primera versión de estructura en Neo4j

Esta estructura presentaba dos problemas principales:

- Precisión de la asociación entre brotes y rutas: pueden existir casos en los que un brote esté situado muy cerca del límite que separa su zona geohash de otra. En estos casos puede haber rutas cuyo punto esté situado justamente en la zona geohash vecina, provocando que no se asocien brotes y rutas aun estando muy cerca entre sí.
- Complejidad en la asociación ruta-comarca: la asociación también se debe hacer entre rutas y comarcas. En este caso no hay un destino fijo para cada ruta sino una zona geohash en España. Debido a esto se decidió formar una cuadrícula alrededor de cada comarca y calcular para cada cuadrícula geohash de una ruta, el porcentaje del área que solapaba con una cuadricula de una comarca ganadera. En la Figura 3.5, se puede apreciar de forma visual el proceso, la cuadrícula *eyy0* correspondiente al punto de la ruta está completamente solapada por la cuadrícula de la comarca (porcentaje de solapamiento del 100%), por lo que se aceptaría la asociación ruta-comarca.

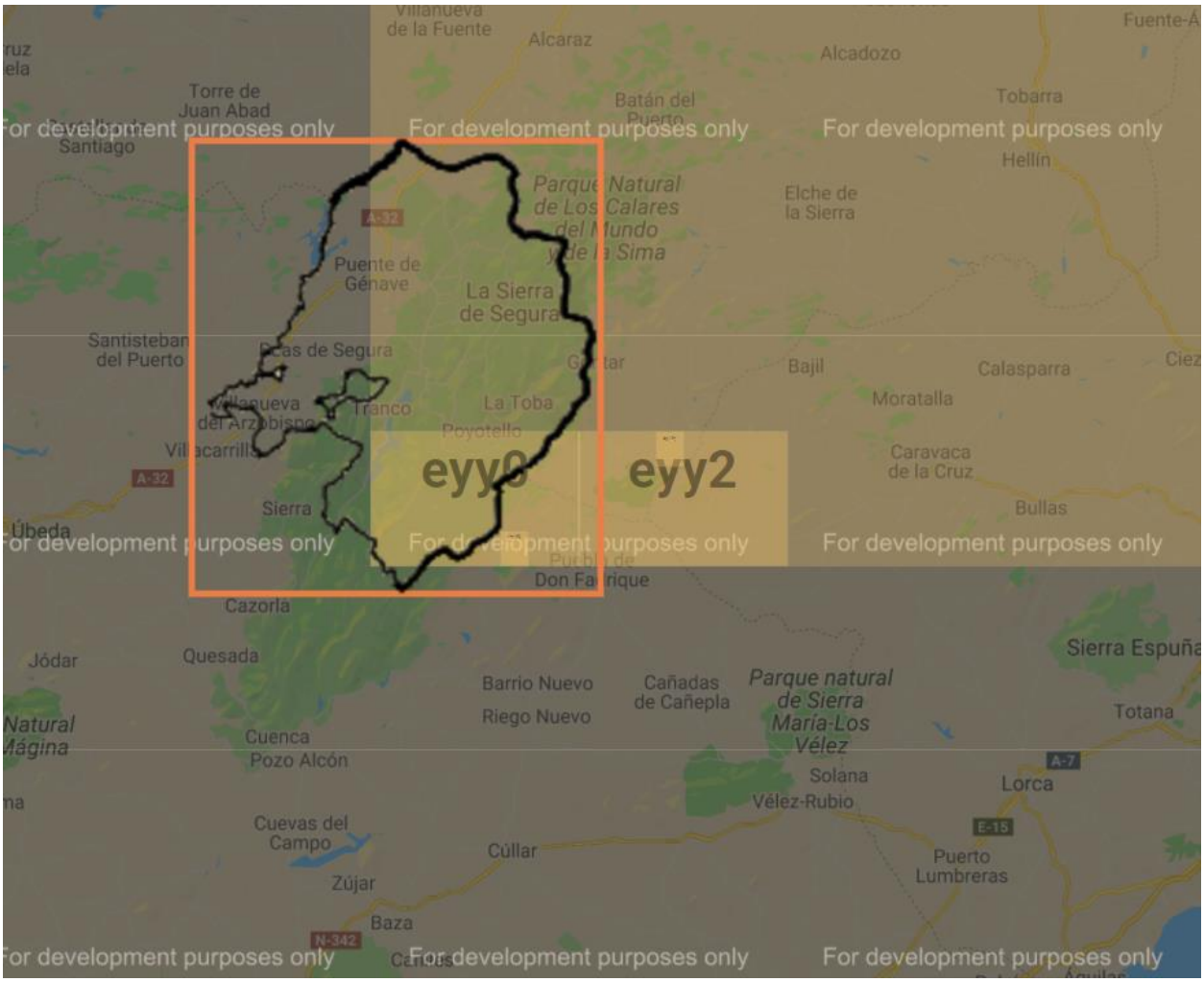


Figura 3.5 Solapamiento entre cuadrículas de comarca y geohash

3.3.3.2 Nodos de brote y comarca

Debido a los problemas encontrados en la estructura anterior se pensó en una versión, esta vez con nodos representando brotes y comarcas directamente. Un nodo brote sería relacionado mediante una ruta hasta un nodo comarca, como se muestra en la Figura 3.6, siendo el nodo azul el brote y el nodo naranja la comarca.

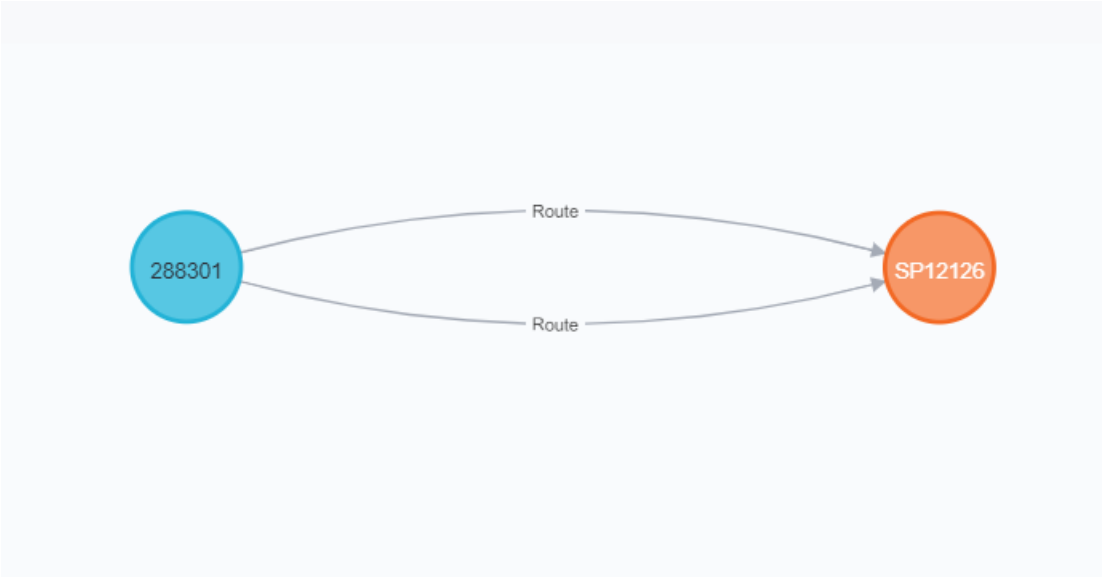


Figura 3.6 Segunda versión de estructura en Neo4j

Esta estructura se asemejaba mucho al modelo ideal para consultar conexiones entre brotes de Europa y comarcas en España. No obstante, esta nueva idea también conllevaba grandes problemas:

- Gran coste computacional: para cada brote había que revisar todas las rutas con el fin de verificar la cercanía de cada uno con el brote y añadir esta ruta a la base de datos.
- Necesidad de actualización frecuente: cada semana aparecen nuevos brotes de gripe aviar en Europa. La base de datos en Neo4j ahora tenía que actualizarse con estos nuevos brotes, requerimiento no necesario anteriormente.

3.3.3.3 Nodos de geohash y de comarcas

Se pensó en una nueva forma de asociar rutas con brotes que arreglase los problemas anteriores. Esta consistía en comparar los cuatro primeros dígitos de los geohashes correspondientes a los puntos de las rutas y de los brotes para detectar si se encontraban en la misma zona geohash, en el caso de lo estuvieran se aceptaba la asociación. Con esto también se asociaban brotes y comarcas, ya que cada ruta consta de un punto en una zona de Europa y otro que lo asocia a una comarca española. De esta forma no haría falta actualizar los grafos de Neo4j e incluso se podría

prescindir de la base de datos ya que no sería necesaria para realizar las comparaciones, pues con los datos almacenados en MongoDB serían suficientes. Sin embargo, para disminuir el coste computacional que requería este proceso, se optó por rediseñar la estructura Neo4j para reemplazar los nodos brotes por nodos geohash, vistos anteriormente en la primera versión, pero manteniendo los nodos comarca. Las relaciones (las rutas) ahora unirían zonas geohash de Europa, donde puede haber brotes, con comarcas ganaderas en España directamente, como se ve en la Figura 3.7. El nodo rojo representa la zona geohash, la cual contiene varias rutas a diferentes comarcas (nodos naranjas).

Figura 3.7 Tercera versión de estructura en Neo4j

Esta nueva estructura no soluciona el problema de la precisión de asociaciones brote-ruta del principio, por lo que se hace un posterior tratamiento de los datos con Python. Este tratamiento consiste en, no solo consultar en la base de datos por la zona geohash de un brote, sino también por las zonas que rodean a este geohash, lo cual es trivial ya que cada zona geohash está rodeada siempre por las mismas zonas vecinas. De esta forma se garantiza seleccionar todas las posibles rutas asociadas a un brote.

Posteriormente se calcula la distancia entre los puntos de uno de los extremos de cada ruta y el punto de un brote, si el punto de una ruta esta a una distancia menor de 25km, es decir la ruta está dentro de un radio de 25km alrededor del brote, se acepta la asociación ruta-brote.

3.4 Extracción y almacenamiento de la temperatura mínima

A continuación, se procederá a la explicación detallada de la extracción y almacenamiento de la temperatura mínima. Como ya se explicó en el capítulo 2.1de esta memoria, se hará uso de dos APIs: AEMET OpenData y la API de TuTiempo.net. Ambas con distintas finalidades.

3.4.1 AEMET OpenData

La API de AEMET, con respecto a los valores climatológicos que ofrece, principalmente proporciona históricos de temperatura en las ubicaciones de sus estaciones meteorológicas, pero no proporciona previsiones semanales. Las únicas predicciones que proporciona son las predicciones de los dos próximos días, lo cual no es útil, puesto que la herramienta usará la predicción de la temperatura mínima de la siguiente semana como parámetro para el modelo que calcula el nivel de alerta para las comarcas ganaderas.

Se hará uso de los valores-climatológicos haciendo uso de las operaciones que la API proporciona. La búsqueda se realiza según los identificadores de las estaciones como se puede ver en la Figura 3.8.

Figura 3.8 Operaciones de valores-climatológicos

Para solucionar este inconveniente, en cada URL que se vaya a utilizar se debe añadir al final lo siguiente: /?api_key=<API key obtenida>.

Tras realizar la consulta aparece una descripción del estado de la respuesta. Cuando la consulta es exitosa aparecerán cuatro campos: "descripción", "estado", "datos" y "metadatos". A continuación, se expone un ejemplo.


```
{
  "descripcion" : "exito",
  "estado" : 200,
  "datos" : "https://opendata.aemet.es/opendata/sh/75e3e4da",
  "metadatos" : "https://opendata.aemet.es/opendata/sh/0556af7a"
}
```


Figura 3.9 Ejemplo de consulta

En caso de error, solo aparecerán los campos de "descripción" y "estado" describiendo el porqué del fracaso de la consulta.

El campo "datos" es una URL que contiene toda la información solicitada (Figura 3.10) y el campo "metadatos" describe cada campo que aparece en la URL de datos (Figura 3.11)


```
[ {
  "latitud" : "413515N",
  "provincia" : "BARCELONA",
  "altitud" : "74",
  "indicativo" : "0252D",
  "nombre" : "ARENYS DE MAR",
  "indsinop" : "08186",
  "longitud" : "023224E"
}, {
  "latitud" : "411734N",
  "provincia" : "BARCELONA",
  "altitud" : "4",
  "indicativo" : "0076",
  "nombre" : "BARCELONA AEROPUERTO",
  "indsinop" : "08181",
  "longitud" : "020412E"
}, {
```


Figura 3.10 URL datos

```
{
  "unidad_generadora": "Servicio del Banco de Datos Nacional de Climatología",
  "periodicidad": "1 vez al día",
  "descripcion": "Inventario de estaciones para el apartado Valores Climatología",
  "formato": "application/json",
  "copyright": "© AEMET. Autorizado el uso de la información y su reproducción citando a AEMET como autora de la misma.",
  "notaLegal": "http://www.aemet.es/es/nota_legal",
  "campos": [
        {"id":"latitud",
        "descripcion": "latitud de la estación",
        "tipo_datos": "string",
        "requerido": true
    },
        {"id":"provincia",
        "descripcion": "provincia donde reside la estación",
        "tipo_datos": "string",
        "requerido": true

    },
```


Figura 3.11 URL de metadatos

Tras completar los pasos previos, desde Python usando la librería *requests*, se extraen los datos de la URL que están en formato JSON devueltos tras la consulta a la API.

Otro de los inconvenientes de esta API, es que varias de las estaciones dejaban de proporcionar datos a partir de una determinada fecha e incluso podría haber semanas sin temperatura mínima. Estos vacíos de información suponían grandes problemas para el correcto cálculo del modelo, ya que lo dejaba sin uno de los parámetros clave y sobre todo el tiempo y esfuerzo para asegurar que todas las comarcas no tengan ninguna semana sin temperatura mínima.

3.4.2 TuTiempo.net

Para las consultas, se debe acceder al apartado de *Documentación - JSON*. Una vez dentro, aparecerán las distintas operaciones. La escogida fue la búsqueda por latitud y longitud (Figura 3.12).

Con latitud y longitud

También puedes realizar la petición indicando la latitud y la longitud, en este caso, se dará la información de la localidad más cercana de la que dispongamos datos. Hay que tener en cuenta que si no tenemos datos a menos de 30 km la salida dará un error.

```
https://api.tutiempo.net/json/?lan={lan}&apid={Clave API}&ll={ll}
```

{lan}: Idioma de salida, puede ser

- **es** - Español
- **en** - Inglés
- **fr** - Francés
- **pt** - Portugués
- **de** - Alemán
- **it** - Italiano

{Clave API}: Clave API

{ll}: latitud,longitud. Se añade latitud y longitud separados por una coma (,). En los datos de latitud y longitud el separador decimal tiene que ser un punto(.).

Ejemplo

```
https://api.tutiempo.net/json/?lan=es&apid=zwDX4azaz4X4Xqs&ll=40.4178,-3.7022
```

Figura 3.12 Consulta con latitud y longitud

Las predicciones son de los próximos 7 días (Figura 3.13). Por lo que, para conseguir la temperatura semanal es suficiente con hacer la media de las temperaturas de esos 7 días.

Ejemplo salida JSON


```
{
    "copyright": "Tutiempo Network, S.L.",
    "use": "Se autoriza el uso de estos datos citando a https://www.tutiempo.net como fuente de los mismos.",
    "information": {
        "temperature": "ºC",
        "wind": "km/h",
        "humidity": "%",
        "pressure": "hPa"
    },
    "web": "https://www.tutiempo.net",
    "language": "es",
    "locality": {
        "name": "Madrid",
        "url_weather_forecast_15_days": "https://www.tutiempo.net/madrid.html",
        "url_hourly_forecast": "https://www.tutiempo.net/madrid.html?datos=por-horas",
        "country": "España",
        "url_country": "https://www.tutiempo.net/espana.html"
    },
    "day1": {
        "date": "2019-11-6",
        "temperature_max": 14,
        "temperature_min": 8,
        "icon": "7",
        "text": "Cubierto",
        "humidity": 56,
        "wind": 21,
        "wind_direction": "Oeste",
        "icon_wind": "O",
        "sunrise": "7:47",
        "sunset": "18:09",
```


Figura 3.13 Ejemplo salida JSON (TuTiempo.net)

Para asignar una predicción a una comarca ganadera se usa el centroide de dicha comarca para la consulta de la API y se almacena en la base de datos. Una limitación de la API es que la búsqueda se realiza en un radio de 30 km según la latitud y longitudes especificadas y, si no hay datos en ese rango, la salida será nula. Utilizando únicamente el centroide como punto de referencia, muchas de las comarcas no disponen de predicción. Para solucionar este problema, se han creado cuatro niveles

de búsqueda donde acceden aquellas comarcas que no han tenido predicción en el anterior nivel:

- Primer nivel: búsqueda en la API según el centroide de las comarcas ganaderas.
- Segundo nivel: búsqueda en la API según las coordenadas de la comarca que aparece en la colección **estaciones.** La diferencia de estas coordenadas con las coordenadas que aparecen en la colección **comarcas** es que estas tienen menor precisión, es decir, menos decimales. Se usan puesto que, en algunos casos, la búsqueda con menos decimales si arrojan resultados, en cambio con las coordenadas de las comarcas de la colección estaciones, no.
- Tercer nivel: búsqueda en la API según las coordenadas de las estaciones más cercanas. Se recorre la lista de todas las estaciones cercanas correspondientes con la API de AEMET, se extraen dichas coordenadas y se realiza la consulta.
- Cuarto nivel: como último recurso se busca todas las comarcas de la provincia de la comarca que no tiene predicción y se le asigna la predicción de las comarcas de esa provincia que si tienen. En caso de tampoco obtener predicción se realiza el mismo procedimiento según la Comunidad Autónoma en la que se encuentre.

3.4.3 Almacenamiento de temperaturas en MongoDB

Se hace uso de tres colecciones distintas: estaciones, histórico y temperatura.

3.4.3.1 Estaciones

En esta colección se guarda la relación entre una comarca ganadera y las estaciones meteorológicas más cercanas. Por ello presenta 496 documentos, mismo número de documentos que la colección **comarcas**. Cada relación sigue la estructura del Código 3. Los campos más relevantes son:

- comarca_sg: es el identificador único de cada comarca.

- indicativo: es el identificador único de la estación principal asociada a dicha comarca.
- estacionesAdd: un array que contiene por orden de cercanía los indicativos de las estaciones más cercanas. Usada en caso de que la estación principal no pueda proveer datos.


```
_id: ObjectId("605108606fc0d8d08c458b87")
FID_1: 56
objectid: 4456
comarca_sg: "SP08096"
com_sgsa_n: "VALLES ORIENTAL (GRANOLLERS)"
FID_2: 0
Field1: 0
nombre: "ARENYS DE MAR"
indicativo: "0252D"
latitud: "413515N"
longitud: "023224E"
latitud_D: 41.5875
longitud_D: 2.54
Distance: 0.04126398
estacionesAdd: Array
    0: "0252D"
    1: "0229I"
    2: "0200E"
    3: "0076"
    4: "0367"
    5: "0149X"
    6: "0324A"
    7: "0370B"
    8: "0372C"
    9: "9585"
    10: "9619"
    11: "9720X"
    12: "0016A"
```


Código 3 Documento de una estación en mongo

3.4.3.2 Histórico

Esta colección se usa para albergar las temperaturas recogidas por AEMET OpenData, según cada estación. Por este motivo, hay un total de 210 documentos. Todos los documentos se organizan siguiendo la estructura del Código 4, cuyos campos son:

- idEstacion: identificador único de la estación.
- histórico(semanal): un diccionario cuya clave es el año y el valor es un array con la temperatura mínima de todas las semanas de ese mismo año.

- boolCompleto: un diccionario cuya clave es el año y el valor es un array con el número de semana que no tiene temperatura mínima. Es decir, es un diccionario auxiliar, que se utiliza para saber por cada año las semanas que no tienen temperatura mínima. Haciendo que completar esa información con información de las estaciones más cercanas sea más fácil.

```
  _id: ObjectId("6094013a7a065d26a849c36b")
  idEstacion: "0002I"
v historico(semanal): Object
  > 2017: Array
  > 2018: Array
  > 2019: Array
  > 2020: Array
  > 2021: Array
v boolCompleto: Object
  > 2017: Array
  > 2018: Array
  > 2019: Array
  > 2020: Array
  > 2021: Array
```

Código 4 Documento del histórico de una estación en mongo

3.4.3.3 Temperatura

Colección que contiene, para cada comarca ganadera, la temperatura por semanas desde el 2017 hasta la fecha actual, así como la predicción para la siguiente semana. La colección sigue la estructura del Código 5. Los campos que lo conforman son:

- comarca_sg: identificador de la comarca ganadera.
- históricoFinal: un diccionario cuya clave es el año y el valor es un array con la temperatura mínima de todas las semanas de ese mismo año. No hay valores nulos.
- predicción: es la predicción proporcionada por la API de TuTiempo.net

```
_id: ObjectId("609401627a065d26a849c43d")
comarca_sg: "SP08096"
historicoFinal: Object
  > 2017: Array
  > 2018: Array
  > 2019: Array
  > 2020: Array
  > 2021: Array
prediccion: 9.571428571428571
```

Código 5 Documento del histórico final y la predicción de una comarca en Mongo

Capítulo 4 - Aplicación del modelo y generación de alertas

Tras haber asentado la obtención de los datos y su almacenamiento, se pasó a la fase de diseño de la herramienta. Revisando de nuevo los objetivos, aún son necesarios ciertos módulos en nuestra herramienta para poder aplicar el modelo usando los datos adquiridos. En las próximas secciones se va a detallar el proceso de implementación para cada módulo.

4.1 Actualización de datos sobre brotes y temperaturas

Los datos sobre brotes y temperaturas en cada comarca deben estar actualizados para la correcta aplicación del modelo y funcionamiento de la herramienta. Para ambas colecciones se han utilizado scripts para actualizar los datos, lo cual se hace de forma semanal mediante el uso del servicio ***cron***, añadiendo una regla en el fichero de ***crontab*** para que cada lunes a las 1:00 se ejecuten los scripts.

El **script de actualización de brotes** manda una petición a la web de Empres-i que devuelve un archivo en formato CSV con los brotes de 1 año actualizados. Se almacena en la carpeta *data,* se abre dicho fichero y se buscan los brotes de la última semana. Si el brote ya existe se actualiza en la base de datos, si no se añade.

El **script de actualización de temperaturas** realiza una consulta a AEMET OpenData obteniendo las temperaturas mínimas de la semana pasada y se guardan en la base de datos. Una vez actualizado el histórico se procede a realizar otra consulta, pero a la API de TuTiempo para guardar en la base de datos la predicción de la semana actual para la que se va a ejecutar el modelo.

4.2 Implementación del modelo

La implementación del modelo consta de tres partes principales:

- **Factorías de datos**: las factorías principales son la de los brotes y temperaturas, que hacen consultas a las bases de datos y transforman la información obtenida a un formato más conveniente para el modelo. Básicamente crean

diccionarios con las comarcas como clave y datos asociados a cada comarca como valor. Con esto asociamos cada comarca con los brotes en Europa mediante las rutas almacenadas en Neo4j y también con su temperatura mínima.

- **Controlador**: le encarga de pasar los datos al modelo y de controlar el número de ejecuciones se han de realizar. Según una fecha y un número de semanas, reitera el proceso de ejecución del modelo empezando por la semana de la fecha especificada y para las semanas anteriores.
- **Modelo**: es la parte que usa los datos proporcionados por el controlador para calcular un nivel de riesgo para cada comarca. Para calcular el riesgo de una comarca *i*, para una semana *t* se tienen en cuenta los brotes surgidos en una ventana temporal de 3 meses (12 semanas) en el pasado. Para ello se sigue esta fórmula, proporcionada por la investigadora Irene Iglesias:

  $Riesgo_{i,t} = supervivencia\ del\ virus_{i,t} * \sum (riesgo_ruta_i)$

  - $Riesgo_ruta_i$: Es el riesgo que genera una ruta migratoria que asocia una comarca *i* con un brote. El sumatorio se hace con todas las rutas provenientes de brotes que se produjeron dentro de la ventana temporal de 3 meses mencionada anteriormente. Este riesgo a su vez se calcula con la fórmula:

    Riesgo_ruta = ProbMigración * Epiunit

    - ProbMigración: con un valor entre 0 y 1, es la probabilidad de que la especie asociada a la ruta se desplace hacia la comarca ganadera durante la semana actual.
    - Epiunit: la constante asociada a la unidad epidemiológica del brote de la ruta. Un brote de aves domésticas apenas presenta un peligro pues normalmente se encuentran en cautiverio, mientras que las silvestres son las más peligrosas. Los valores son los siguientes:
      - Domésticas: 0.1

- Cautivas: 0.3
- Silvestres: 1

- Supervivencia del virus $_{i,t}$: número de días que el virus de la gripe puede sobrevivir según el pronóstico de la temperatura mínima de la comarca *i* durante la semana actual *t*. Esta a su vez se calcula con la fórmula:

  Supervivencia = -7.82 * ln(temperatura en grados Celsius (C°)) + 29.94

  Este valor se saturará en 66. También, en caso de que la temperatura sea igual o esté por debajo de los 0°.

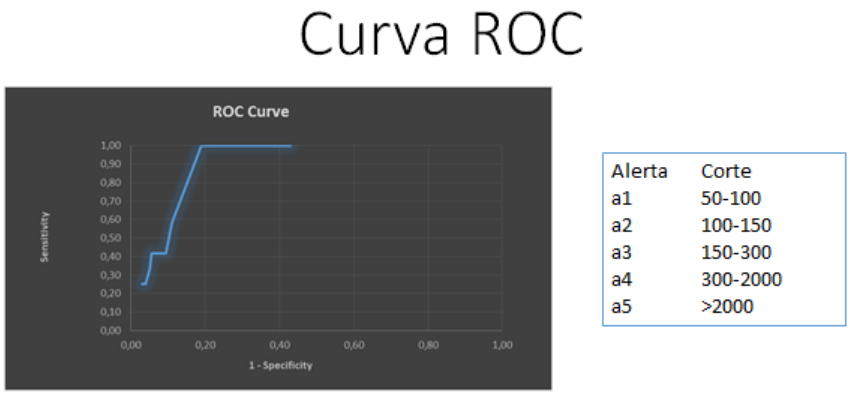
Una vez calculado el riesgo este debe traducirse a un nivel de alerta del 1 al 5 para poder ser visualizado en la aplicación web. Por ello se sigue el siguiente esquema:

- Alerta nivel 1: 50-100
- Alerta nivel 2: 100-150
- Alerta nivel 3: 150-300
- Alerta nivel 4: 300-2000
- Alerta nivel 5: >2000

Para ajustar el riesgo a los niveles de alerta necesarios y probar la precisión de las distintas versiones del tratamiento de los datos, la investigadora Irene Iglesias hizo varios estudios con la ejecución del modelo durante el periodo de un año (52 semanas) entre los años 2020 y 2021, pues en estos dos años se produjo una ola de gripe aviar por toda Europa, provocando también casos en España.

Estos estudios consistían, principalmente, en la generación de una gráfica mostrando una curva ROC. Esta gráfica consta de dos ejes, el vertical representando **sensibilidad**, probabilidad de que un caso positivo (hubo verdaderamente un brote en una comarca) sea predicho como positivo, y **1 - especificidad** en el eje horizontal, donde la especificidad es probabilidad de que un caso negativo (no hubo brote) sea predicho como negativo. En la Figura 4.1 se muestra la gráfica resultante junto con una tabla que muestra diferentes umbrales de riesgo y la cantidad de Falsos Positivos y

Negativos (alertas mal predichas), la cantidad los verdaderos positivos y negativos (número de comarcas en los hubo brote y en los que no hubo), además de la sensibilidad y especificidad correspondientes a cada umbral.



Alerta	Corte
a1	50-100
a2	100-150
a3	150-300
a4	300-2000
a5	>2000

Cut-off	Realidad		Alertas mal diagnosticadas				
OD Level	Positivo	Negativo	Falso positivo	Falso negativo	Sensitivity	Specificity	1-Spec
0	12	11264	0	0	1,00	1,00	0,00
50	12	6416	4848	0	1,00	0,57	0,43
150	12	9132	2132	0	1,00	0,81	0,19
300	7	10018	1246	5	0,58	0,89	0,11
600	5	10202	1062	7	0,42	0,91	0,09
1000	5	10643	621	7	0,42	0,94	0,06
2000	4	10703	561	8	0,33	0,95	0,05
5000	3	10828	436	9	0,25	0,96	0,04
>5000	3	10945	319	9	0,25	0,97	0,03

Figura 4.1 Estudio con curva ROC del modelo

Con el estudio se intentó averiguar qué umbral ofrecía mayor especificidad, con el objetivo de levantar el menor número de falsas alarmas posible. Así se optó por fijar el umbral con un valor de 2000, adquiriendo así una especificidad del 95%. Si bien este umbral es el más adecuado para el modelo, también conllevaba una sensibilidad de solamente el 33%, lo que significaba 2 de cada 3 brotes no serían correctamente predichos.

Los pocos casos que ha habido en España hacen que sea muy difícil mejorar la precisión con la que se predicen posibles brotes. En la Figura 4.2 se muestran los niveles generados por el modelo en enero de 2021, con las dos comarcas donde se produjeron brotes rodeados en azul. Este mes junto con diciembre de 2020 fueron los meses con mayor número de brotes en Europa durante esta ola de gripe aviar. Como bien muestra la imagen gran parte de las comarcas mostraban un nivel de alerta elevado, pero solo hubo brotes en dos comarcas con niveles de alerta 3 y 5.

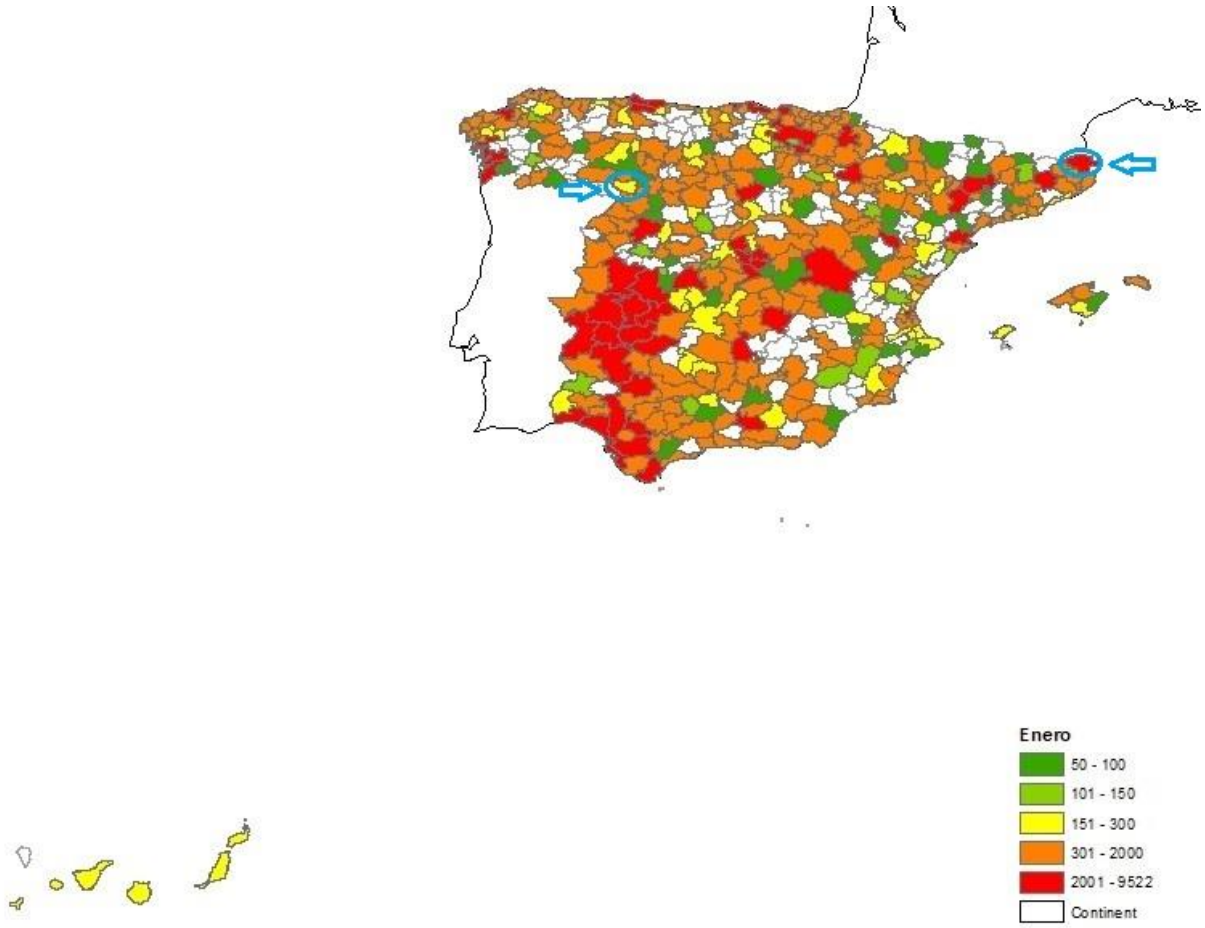


Figura 4.2 Mapa con los niveles de alerta de enero 2021 y comarcas donde hubo brotes

4.3 Archivos generados por la herramienta

Una vez aplicado el modelo para el número de semanas especificado, se debe guardar esta información de forma persistente. Para cubrir las distintas necesidades del proyecto, se generan los siguientes archivos:

4.3.1 Reportes

La herramienta genera reportes de las alertas con información importante en formato PDF y CSV. Todos los reportes irán destinados a una carpeta Drive.

4.3.1.1 Reportes PDF

Se genera un fichero PDF, por cada semana donde se haya ejecutado la herramienta. Haciendo uso del lenguaje de marcado ligero Markdown.

El formato del reporte lo compone una cabecera y un sumario como en la Figura 4.3, y varias tablas.

- Cabecera: formado por la fecha y el periodo de inicio a fin (lunes de la siguiente semana).
- Sumario: formado por el número de comarcas ganaderas en alerta y el número de brotes totales que han intervenido en el cálculo del nivel de alerta.
- Tablas:
    - Tabla de alertas: es una única tabla que contiene todas las alertas generadas para esa semana, con información de la comarca (Comarca, ID-CG), el número de brotes y los valores empleados para calcular la alerta como pueden ser el número de movimientos de riesgo que debe ser mayor o igual al número de brotes, el grado de alerta sin cuantificar del 1 al 5, la temperatura de esa semana y la supervivencia del virus en días (Figura 4.4).
    - Tablas de brotes relacionados con una alerta: por cada alerta habrá un determinado número de tablas de brotes según el número de brotes que hayan intervenido en el modelo. Esta tabla presenta información de dicho brote (Identificador, fechas, localización) e información de las especies de aves, entre otros (Figura 4.5).

DiFLUsion: Informe de Alerta Semanal

- *Fecha*: 21-12-2020
- *Periodo de*: 21-12-2020 a 28-12-2020

Sumario del informe

- *Número de comarcas ganaderas en alerta*: 395
- *Número de brotes en Europa asociados con movimientos de riesgo a España*: 587

Figura 4.3 Cabecera y sumario PDF

Tabla de alertas

Nº	Fecha	Comarca	ID CG	Nº brotes	Nº mov. Riesgo	Grado alerta	Temperatura estimada	Supervivencia del virus en días
1	21-12-2020	SANLUCAR	SP41087	272	1425	13297.2793	9.07	12.6959

Figura 4.4 Tabla de alertas PDF

Alerta 273

- *Id comarca*: SP11016
- *Localización comarca*: LITORAL

ID	Event ID	Reporting date	Observational date	Country	Location	Latitud	Longitud	An. Type	Species	Cases	Deaths	Especie movimiento	Cód. Especie	Prob mov semanal
0	282530	17-11-2020	12-11-2020	Netherlands	Terwolde	52.26	6.06	Domestic	Domestic Unspecified bird	5840	3840	Ciconia ciconia	1340	0.5404

Figura 4.5 Tabla de brotes PDF

4.3.1.2 Reportes CSV

Son solo dos ficheros, uno destinado a guardar todas las alertas producidas en el periodo de un año y el otro a guardar todos los brotes, al igual que el CSV de alertas también será de un periodo de un año.

El fichero de alertas presenta los mismos campos que la tabla alerta del PDF (Figura 4.6). Por otro lado, el fichero de brotes tiene más campos (Figura 4.7 y Figura 4.8).

A	B	C	D	E	F	G	H	I	J
Nº	Fecha	Comarca	ID CG	Nº brotes	Nº mov. Riesgo	Grado alerta	Temperatura estimada	Supervivencia del virus en días	
1	18-05-2020	CALZADA DE CALATI	SP13027	4	39	3.788.261	5.73	162.905	
2	18-05-2020	ZARAGOZA	SP50297	2	2	3.1051	11.74	10.6774	

Figura 4.6 Alertas CSV

ID	Nº Alerta	Comarca	ID CG	Grado alerta	Fecha alerta	Event ID	Temperatura es	Supervivencia d	Reporting date	Observational da	Country	Location	Latitud
0	1	CALZADA DE C	SP13027	2.105.957	01-06-2020	267566	13.11	9.8136	27-03-2020	27-03-2020	Germany	Bülstringen	52.36
1	1	CALZADA DE C	SP13027	2.105.957	01-06-2020	267399	13.11	9.8136	24-03-2020	24-03-2020	Germany	Limbach-Oberfro	50.85

Figura 4.7 Brotes (1) CSV

Latitud	Longitud	Ponderacion bro	Riesgo brote	An. Type	Species	Cases	Deaths	Especie movimie	Cód. Especie	Prob mov semanal
52.36	11.32	0.1	7.442	Domestic	Domestic Turkey	380	130	Ciconia ciconia	1340	7.583
50.85	12.75	1	7.4415	Wild	Wild Anatidae (u	1	1	Ciconia ciconia	1340	7.583

Figura 4.8 Brotes (2) CSV

Todos estos ficheros se subirán a una unidad de Google Drive[18], plataforma de almacenamiento en la nube muy popular y extendido. Esto es realizado mediante

[18] Almacenamiento en la nube para casa y el trabajo - Google Drive

Python, en concreto con la librería de Pydrive2[19] que recoge las posibilidades que ofrece la API de Google Drive y lo hace mediante clases dedicadas a la autenticación y manejo de archivos principalmente.

Debido al límite de tamaño de almacenamiento en Drive y para una mejor organización, este generador de reportes está programado para que, una vez finalizado el año natural, se almacene los CSV y todos los PDF en un archivo comprimido cuyo nombre será <año>.csv. Los CSV serán borrados de la carpeta principal, tras finalizar el año, y se generarán unos nuevos CSV. Por otro lado, los PDF se irán borrando según avanzan las semanas, es decir, pasada una semana se borra el PDF más antiguo.

4.4 Interfaz web mediante ArcGIS

De forma paralela al proyecto, también se desarrolló una interfaz gráfica con el objetivo de ofrecer una forma de visualizar en un mapa las distintas alertas que el modelo generaba sobre comarcas en España, los brotes de gripe aviar que aparecían por Europa y rutas migratorias potencialmente peligrosas que conectasen comarcas y brotes.

Esta interfaz fue desarrollada por Carlos Eduardo Blanco Urbina, empleado del INIA-CSIC durante este proyecto, como una aplicación web[20] alojada en Github. Esta fue desarrollada usando ArcGIS[21], un conjunto herramientas software muy popular especializada en la creación y manejo de mapas, además del análisis y tratamiento de información geográfica.

Por último, para alimentar esta aplicación se generan semanalmente ficheros con información sobre alertas, rutas y brotes. Estos contienen los datos suficientes para representarse sobre el mapa y poder ser filtrados para que aparezcan solo los objetos dentro una ventana de tiempo, el cual puede modificarse con una barra deslizante.

---

[19] PyDrive2 · PyPI

[20] App Gripe Aviar 3D (influenzaaviar.github.io)

[21] ArcGIS Online | Web GIS Mapping Software for Everyone (esri.com)

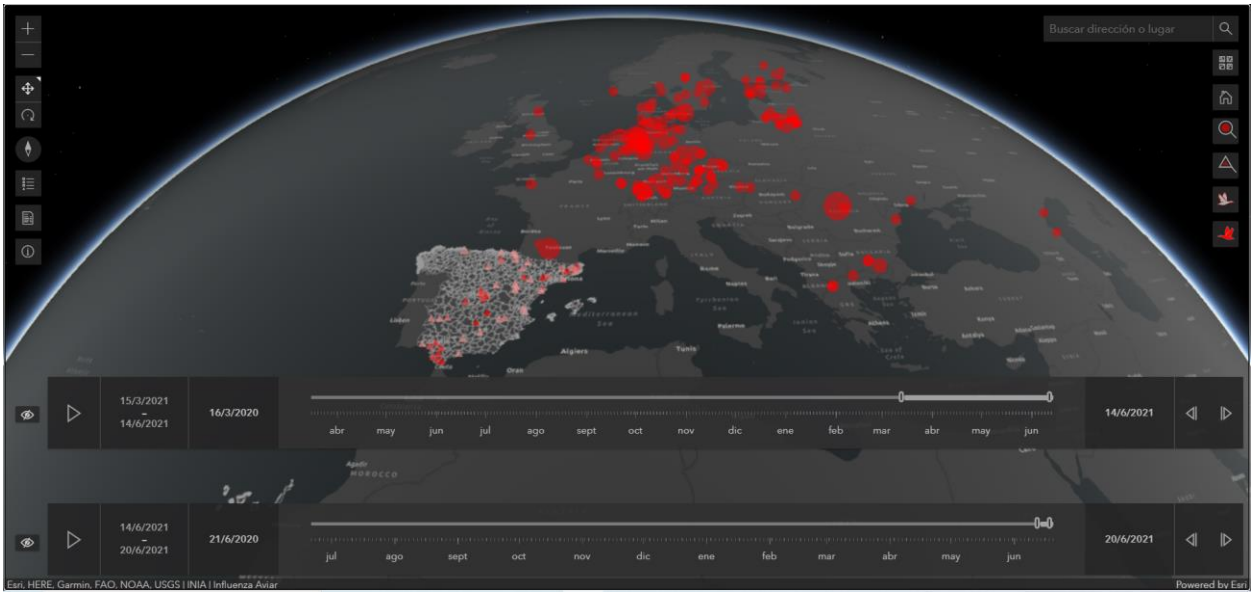


Figura 4.9 Aplicación web para la visualización de alertas, rutas y brotes

4.4.1 GeoJSON sobre alertas, rutas y brotes

Los archivos en formato GeoJSON[22] se usan ampliamente para representar elementos geográficos en aplicaciones web. Son archivos JSON con una estructura específica para poder representar puntos, líneas o polígonos.

La herramienta genera los siguientes ficheros GeoJSON:

- Alertas: lista de objetos de tipo punto, cuyas coordenadas representan el centroide de la comarca a la que está asociada dicha alerta. Cada objeto contiene un identificador de alerta (idAlerta) conformado por el identificador de la comarca ganadera (comarca_sg) al reportDate unidos por '_', el nivel de riesgo, el reportDate (hora 1:00 del lunes de la semana para la que se predice la alerta en milisegundos desde epoch) y una URL al reporte en PDF alojado en Google drive entre otras propiedades.

[22] GeoJSON

```
{"type": "FeatureCollection",
 "features": [{
    "type": "Feature",
    "geometry": {"type": "Point", "coordinates": [-3.71444455552, 38.6549177036]},
    "properties": {
        "idAlerta": "SP13027_1589756400000.0",
        "Riesgo": 5,
        "reportDate": 1589756400000.0,
        "comarca": "CALZADA DE CALATRAVA",
        "informe": "https://drive.google.com/file/d/1_KgPsKlwr9t1pUGYcOi9A0tuSewGEPUt/view?usp=drivesdk"
        }
    }]
}
```


Código 6 Ejemplo de una alerta en GeoJSON

- Rutas: lista de objetos de tipo línea, cuyas coordenadas representan el centroide de una comarca y el punto de un brote. Se usan para unir comarcas y brotes que estén asociados mediante rutas migratorias.


```
{"type": "FeatureCollection",
 "features": [{
   "type": "Feature",
   "geometry": {"type": "LineString", "coordinates": [[-3.71444455552, 38.6549177036], [11.32, 52.36]]},
   "properties": {
     "idBrote": 267566,
     "idAlerta": "SP13027_1589752800000.0",
     "idComarca": "SP13027"}
 }]
}
```


Código 7 Ejemplo de una ruta en GeoJSON

- Brotes: lista de objetos de tipo punto, cuyas coordenadas representan el punto de un brote. Contienen información destacable del brote como número de casos, serotipo del virus o las especies de ave principalmente afectadas.

```
{
    "type": "FeatureCollection",
    "features": [{
        "type": "Feature",
        "geometry": {"type": "Point", "coordinates": [24.9466, 42.29193]},
        "properties": {
            "id": 265136,
            "country": "Bulgaria",
            "observationDate": 1581894000000,
            "city": "Bulgaria",
            "species": "Domestic,Duck",
            "cases": 9142,
            "serotipo": "H5N8 HPAI"
        }}
    ]
}
```


Código 8 Ejemplo de brote en GeoJSON

La aplicación web representa las alertas como iconos en forma de triángulo de color rojo y con distintas intensidades para denotar niveles de riesgo. También permite la visualización de las rutas migratorias desde una comarca hacia brotes en Europa al hacer clic sobre una alerta. Los brotes por otro lado son representados como círculos cuyo radio aumenta según el número de casos asociado, igualmente se mostrarán las rutas migratorias asociadas a un brote si se hace clic sobre él.

Figura 4.10 Aplicación web mostrando varias rutas migratorias asociadas a brotes

Capítulo 5 - Conclusiones y trabajo futuro

En este capítulo aparecerán las conclusiones obtenidas de este proyecto, además de posibles líneas de trabajo futuro para mejorar la herramienta.

5.1 Conclusiones

Durante este proyecto, se ha querido desarrollar una herramienta capaz de calcular un nivel de riesgo de brote de Gripe Aviar para cada comarca ganadera española. Para esto se han usado datos sobre brotes en Europa, rutas migratorias de varias especies de aves acuáticas y las temperaturas mínimas de las comarcas ganaderas.

Los datos sobre brotes fueron adquiridos inicialmente del portal de WAHIS-OIE y finalmente de la web de Empres-i. Esto se hizo mediante descarga directa y la técnica de *Web scraping* para mandar peticiones y extraer estos datos de su web. Los datos sobre temperaturas se obtuvieron de AEMET y tutiempo.net, ambas mediante el uso de sus APIs y finalmente las rutas migratorias fueron proporcionadas por la organización SEO-Birdlife.

Tras esto, se pudo aplicar el modelo para la obtención de niveles de riesgo y mejorar la precisión de estos. La herramienta también es capaz de generar archivos con información sobre los niveles en cada comarca y los datos usados para calcularlos, además de poder subir estos ficheros a su respectiva plataforma de almacenamiento en la nube.

5.2 Trabajo futuro

Al inicio del proyecto, se especificaron las variables que se iban a usar en el modelo para el cálculo del riesgo de posibles brotes. Aparte de las que actualmente se están utilizando, se debatió sobre el posible uso de dos variables más:

- La estacionalidad de las aves: datos sobre el movimiento de las aves según la estación del año podrían ayudar a mejorar el uso de rutas migratorias en el cálculo de riesgo.

- La cercanía entre puntos de llegada de aves migratorias y humedales del país: estos lugares son ideales para la transmisión del virus entre aves silvestres.

Por ello, uno de los trabajos a futuro sería el empleo de estas variables en el modelo, con el fin de obtener una mayor precisión al calcular el riesgo de un posible brote.

Otro de los trabajos a futuro, consiste en el desarrollo de una herramienta "*offline*". Esta permitiría explorar distintos parámetros usados en el modelo como las constantes asociadas al tipo de ave de una ruta migratoria o la ventana de tiempo usada en la búsqueda de brotes, y además aplicar este nuevo modelo en semanas anteriores a la actual. Esta herramienta tendría el objetivo principal de comparar los niveles de riesgo resultantes con brotes reales que ocurrieron en comarcas de España para así establecer un umbral de riesgo que se podría usar para predecir si una comarca está en riesgo o no. La herramienta generaría una matriz de confusión mostrando los números de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos, además de una gráfica con una curva ROC utilizando diferentes umbrales.

Las colecciones en MongoDB pueden crecer de forma indefinida si no se piensa en un mecanismo para limitar su tamaño. Una forma de conseguir esto sería la de borrar las entradas más antiguas intentando que las colecciones solo almacenaran la información dentro de una ventana de tiempo. Esto se haría de forma periódica y para no perder estas entradas antiguas, se podrían comprimir en un fichero y ser guardadas en otro lugar. Aunque esto solo mitigaría el problema, podría ser suficiente para evitar tamaños de datos excesivos.

Chapter 6 - Introduction

Avian Influenza is a disease that affects wild and domestic birds, usually asymptomatically because most of the cases are caused by low pathogenic avian influenza. Wild aquatic birds (seagulls, storks, ducks…) are the main hosts of this virus and the ones that can transmit the virus the most in terms of distance.

On the other hand, the highly pathogenic variant can cause dangerous symptoms, which can easily kill a bird and is also highly contagious. An infection of this type of virus in a poultry farm can easily spread to all the other birds as generally there isn´t much space between them. This is the variant that is also the most probable to infect a human being, specifically the H5N1 and H7N9 subtypes, which can also cause serious symptoms in humans.

The epidemics generated by this virus appear in most cases on the Asian and European continents. Within Europe, they occur more frequently in countries located to the north, because the virus can survive longer out in the environment with low temperatures. However, and unfortunately, Spain is not exempt from this virus with examples of outbreaks in 2017, and in 2020 and 2021 within a new wave of this virus.

6.1 Motivation

Poultry farming is of great importance to the Spanish economy. In fact, Spain was consolidated in 2019 as the fourth country in Europe that produces the most meat in the more than 18,000 farms that exist within the[23]country. With the beginning of the COVID-19 pandemic, all sectors were harmed. Despite this crisis caused by the pandemic, the agricultural, livestock and fishing sectors had a GDP rebound of 4.7% in 2020.[24] And

[23] lamoncloa.gob.es

[24] feagro.com

regarding the Spanish poultry industry, it generates around 2,300 million euros to the national GDP and approximately generates 40,000 jobs.[25]

In addition, Spain is a major exporter of poultry meat. In these years, there was an increase of 2.5% in the number of sales compared to the previous decade. The main countries to which this poultry meat is sold are in the European Union, of which the neighboring countries (France and Portugal) stand out. [26]

It should be noted that throughout the year there are migrations of different species of birds from European countries raising the probability of infection in Spain and endangering the poultry sector.

The consequences of an outbreak of avian influenza in Spain are not only sanitary but also economic and can be dangerously catastrophic. The first to be affected would be poultry farmers, since this disease is highly contagious among birds, forcing the slaughter of a large part of the birds in the farm for health reasons, assuming a great economic loss for them. In addition, the above could have a major impact on the country's economy and even on Europe.

As for the health consequences, the high pathogenic variant is very lethal in birds, on the other hand, infections in humans are less probable, but can still cause serious symptoms.

For these reasons, it´s necessary to have a mechanism to predict and buy enough time to take the necessary measures and avoid economic losses especially in the current situation where the whole country is weakened after the crisis caused by COVID-19.

6.2 Objectives

The objective of this project is the development of a tool capable of predicting possible outbreaks of avian influenza in the livestock regions of Spain using data from

[25] avicultura.com

[26] https://avicultura.com/espana-las-exportaciones-de-carne-de-ave-cuadriplican-a-las-importaciones-en-cinco-anos/

recent outbreaks in Europe, migratory routes of different species and temperatures for each region within a weekly period. For this the following would be needed:

- A database with outbreaks in Europe capable of being updated periodically, as well as a data source to extract such information.
- A database with migratory routes of important bird species that are related to the transmission of the virus in Spain.
- A data source to collect temperature forecasts in the different livestock regions and from where to take the real temperatures for past dates to store them in the database, all on a regular basis.
- A model capable of giving an alert level to each region with the previous data, in addition to being able to generate a report with the data considered to calculate this level in each alert.

The resulting code will be stored in this public Github repository.

6.3 Workplan

This is the work plan which will be followed to meet the objectives detailed above:

- Search and choice of data sources: possible data sources will be analyzed to extract each type of data necessary for the model. Once found, we will proceed to choose one for each type of data.
- Design of the software modules: the modules necessary to complete the objectives of this project will be detailed.
- Development of the tool: appropriate technologies will be analyzed to implement the different modules.
- Evaluation of the model: once the development is finished, it will be evaluated, comparing alerts of the model with outbreaks that occurred in Spain.
- Preparation of the project report: finally, the procedures and results obtained during the development process of this project will be drafted.

Chapter 7 - Conclusions and Future Work

In this chapter the conclusions obtained from this project will appear, as well as possible future lines of work to improve the tool.

7.1 Conclusions

During this project, an attempt has been made to develop a tool capable of calculating a risk level of an outbreak of Avian Flu for each Spanish livestock region. For this, data of outbreaks in Europe, migratory routes of several species of aquatic birds and minimum temperatures for livestock regions have been used.

Data on outbreaks were initially acquired from the WAHIS-OIE portal and finally from the Empres-i website. This was done through direct download and the Web scraping technique to send requests and extract this data from the website. The data on temperatures were obtained from AEMET and tutiempo.net, both by using their APIs and finally the migratory routes were provided by the SEO-Birdlife organization.

After this, the model could be applied to obtain risk levels and improve their precision. The tool is also capable of generating files with information on the levels in each region and the data used to calculate them, as well as being able to upload these files to their respective cloud storage platform.

7.2 Future work

At the beginning of the project, the variables to be used in the model to calculate the risk of possible outbreaks were specified. Apart from those currently being used, the possible use of two more variables was discussed:

- The seasonality of birds: data on the movement of birds according to the season could help to improve the use of migratory routes in the calculation of risk.
- The proximity between arrival points of migratory birds and the country's wetlands: these places are ideal for the transmission of the virus between wild birds.

For these reasons, one of the future tasks would be the use of these variables in the model to obtain greater precision when calculating the risk of a possible outbreak.

Another future work consists in the development of an "offline" tool. This would allow modifying different parameters used in the model such as the constants associated with the type of bird on a migratory route, or the time window used in the search for outbreaks and apply this new model for weeks prior to the current one. This tool would have the main objective of comparing the resulting risk levels with real outbreaks that occurred in regions of Spain to establish a risk threshold that could be used to predict whether a region is at risk or not. The tool would generate a confusion matrix showing the numbers of true positives, false positives, true negatives, and false negatives, as well as a graph with a ROC curve using different thresholds.

Collections in MongoDB can grow indefinitely if a mechanism to limit their size is not thought of. One way to achieve this would be to delete older entries by trying to get collections to only store the information within a time window. This would be done periodically and in order not to lose these old entries, they could be compressed into a file and saved elsewhere. Although this would only mitigate the problem, it could be enough to avoid excessive data sizes.

Capítulo 8 - Contribuciones

8.1 Emilio José Valencia Calvopiña

Participación en la actualización del script de Web scraping del portal de WAHIS-OIE. Tras realizar la búsqueda de posibles fuentes de datos, se optó por seguir usando el portal de WAHIS-OIE como fuente de datos sobre brotes, pues este fue usado durante el proyecto en el que se basa este TFG. Dicho proyecto contaba con un script de Web scraping para este fin, sin embargo, este código ya era obsoleto en relación con la actual web de WAHIS-OIE debido a que el portal había cambiado la estructura de su interfaz, por lo que el script era inutilizable y necesitaba ser modificado para que la extracción fuese correcta.

Implementación del modeloV1, usado actualmente. Donde se implementa la función con la que se calcula el riesgo de alerta de una comarca en concreto. Teniendo en cuenta todos los brotes relacionados con dicha comarca, la probabilidad de migración de la especie de la semana pasada de cada brote y la temperatura mínima de la semana con la que se está ejecutando el modelo en la zona de la comarca. Corrigiendo el problema de las semanas de la probabilidad de migración en el modeloV1. La ejecución del código en general se hace de lunes a lunes, el fichero con los datos de la probabilidad de migración se distribuía por semanas, pero empezando del 1 de cada mes y por mes, suponiendo cuatro semanas (del 1 al 7, del 8 al 14, del 15 al 22 y del 22 a fin de mes). Puesto que el fichero que albergaba las probabilidades no seguía la misma distribución del proyecto en general, se debía de adaptar.

Encargado de la búsqueda de fuentes de datos que proporcionasen información meteorológica, en concreto de las temperaturas mínimas de las fechas pasadas como de la predicción de la semana siguiente. Pieza fundamental en el cálculo del modelo. La búsqueda de una fuente de datos que pudiese ofrecer datos de predicciones fue la más complicada ya que las fuentes halladas no presentaban una forma de acceso sencilla. En su mayoría, para tener acceso a esos datos se debía de hacer uso de la técnica de Web scraping pero el nivel de complejidad era alto y con el fin de evitar el problema que hubo con WAHIS-OIE, que tras el cambio de la interfaz, el script

encargado de extraer los datos se quedó inutilizado, obligando al responsable técnico a modificar el script según los cambios producidos. Estos cambios en la interfaz pueden suponer una mayor complejidad haciendo inviable el uso de esta técnica. Tras una búsqueda exhaustiva, se encontró la API de tuTiempo.net. Cuya extracción de datos es sumamente sencilla.

Implementación del código para la extracción de las temperaturas mínimas y encargado de la corrección de los problemas ocasionados por AEMET Opendata, la primera fuente fue proporcionada por los directores José Ignacio y Christian. Esta API presentaba múltiples problemas (expuestos con mayor detalle en el subcapítulo AEMET OpenData de este documento). Y los problemas ocasionados por la API de tutiempo.net. Asegurando que ninguna comarca ganadera se quede sin temperatura mínima. Además, el código está desarrollado para actualizar el histórico añadiendo la temperatura de la última semana y obtener la predicción de la semana actual, cuando se ejecute la herramienta cada lunes.

Implementación del nuevo script de extracción de brotes de Empress-i. Debido al problema con WAHIS-OIE que provoco el cambio de portal que surtía de brotes al proyecto, se tuvo que generar un nuevo script. Este script se ejecuta cada lunes, descargando un fichero CSV de la página de Empress-i con los brotes y tras un filtrado, seleccionando los brotes de la última se añaden estos nuevos brotes a la base de datos. Se asegura que los campos con valor NaN se sustituyan y no se añadan en la base de datos ya que supondría un problema a la hora de usarlo en los diferentes módulos del proyecto. Además, se puede dar el caso donde un brote ya exista en la base de datos, por lo que se buscará y se actualizarán los campos. Para completar los restos de campos necesarios se usa un sencillo código de Web scraping del reporte individual de cada brote, extrayendo datos de casos, muertes, URL entre otros.

Diseño e implementación del código encargado de generar ficheros CSV y PDF, (ReportBuilder). Se hace uso del lenguaje de marcado Markdown para generar ambos tipos de ficheros. Bastante sencillo y cómodo de usar. Para los ficheros PDF, se hace uso de Pandoc que convierte el fichero en formato Makdown en un archivo PDF. Se genera un archivo PDF por cada semana de ejecución. Para los archivos CSV se hace uso de la librería unicodecsv, actualizando los CSV de alertas y brotes que se encuentran

almacenados en Drive. Además de comprimir todos los archivos cuando se acaba el año, evitando problemas de almacenamiento que supondría fallos en la ejecución de la herramienta.

Participación en el resto de los módulos corrigiendo errores que surgiesen como cambios en los archivos GeoJSON usados en la página web del proyecto, en la depuración de errores del controlador, en las factorías como la de TempBuilder encargada de gestionar las temperaturas a usar en el modelo tanto si son temperaturas mínimas pasadas o las predicciones, en scripts encargado de almacenar las comarcas ganaderas procedentes de los archivos proporcionados por la investigadora Irene Iglesias entre otros.

8.2 Cheng Jun Liu Zheng

Participación en la actualización del script de Web scraping del portal de WAHIS-OIE. Tras realizar la búsqueda de posibles fuentes de datos, se optó por seguir usando el portal de WAHIS-OIE como fuente de datos sobre brotes, pues este fue usado durante el proyecto en el que se basa este TFG. Dicho proyecto contaba con un script de Web scraping para este fin, sin embargo, este código ya era obsoleto en relación con la actual web de WAHIS-OIE debido a que el portal había cambiado la estructura de su interfaz, por lo que el script era inutilizable y necesitaba ser modificado para que la extracción fuese correcta.

Actualización y mejora del script dedicado a la inserción de datos en Neo4j. El proyecto anterior también contaba con un script para almacenar datos sobre las rutas migratorias en Neo4j, este generaba las consultas necesarias para crear los nodos y relaciones correspondientes a la primera versión de la base datos en Neo4j. Sin embargo, el script generaba algunas veces consultas con errores menores como falta de comas o dobles comillas, las cuales provocaban errores al ser ejecutadas en la base de datos. Además, el código ignoraba algunas rutas migratorias. Si varias rutas partían de una misma zona y terminaban también en la misma zona geohash, el código solo generaba una relación entre estos dos nodos, ignorando todas las demás. Se corrigieron los errores menores y se añadió una propiedad en cada ruta (relación entre nodos en

Neo4j) a modo de contador que mostrara el número real de rutas que existían entre dos zonas geohash.

Diseño e implementación de la segunda y tercera versión de la estructura de datos en Neo4j, junto con sus correspondientes scripts para insertar nodos y relaciones en la base de datos. Estas versiones explicadas en este documento se hicieron con el objetivo principal de asociar rutas y brotes de una forma más precisa. La primera versión no contemplaba casos extremos en los que una ruta no era asociada a un brote aun estando a menos de 25km de este, por pertenecer a diferentes zonas geohash. La segunda versión asociaba brotes y rutas de una manera precisa, pero conllevaba la necesidad de actualizar periódicamente la base de datos de Neo4j junto con las de MongoDB. Finalmente, la tercera versión se apoyó en un tratamiento posterior de los datos obtenidos de Neo4j para arreglar estos principales problemas sin perder precisión.

Diseño general de la estructura del código (factorías de datos, modelo y controlador) e implementación del código, especialmente las clases de outbreakBuilder, modelV0 (primera versión del modelo) y controller. La factoría de datos se diseñó para poder añadir nuevas variables de forma sencilla y pasar datos al controller con un formato establecido para cada uno. El controller se encargaría de tomar estos datos y usarlos para ejecutar el modelo y generar los archivos de salida necesarios. OutbreakBuilder es la factoría de datos encargada de proporcionar brotes asociados a las distintas comarcas mediante rutas de aves, esta clase es la que necesita datos de Neo4j y un cambio en la estructura de los grafos afectaba a esta factoría. Por otro lado, modelV0 fue una versión de prueba del modelo que solo usaba el número de brotes asociados a una comarca como variable, las comarcas se ordenaban de mayor a menor según este número y se dividían en 5 partes iguales para asignar un nivel de alerta a cada parte. Por supuesto, esta versión del modelo daba resultados muy imprecisos.

Implementación de la clase generadora de GeoJSON, GeojsonGenerator, encargado de componer los archivos correspondientes a los brotes, rutas y alertas, para la interfaz de la aplicación web, además de la implementación del script de Python para la subida de archivos GeoJSON al repositorio de Github correspondiente a la aplicación web. Uno de los tipos de archivos principales de salida son los ficheros

GeoJSON que se utilizan para alimentar la aplicación web y esta clase se encarga de generarlos con datos provenientes de la ejecución del modelo. Estos ficheros son almacenados en una carpeta del sistema que posteriormente es enviada al repositorio de Github correspondiente a la aplicación web, mediante el script que se ha mencionado.

Implementación de la clase dedicada a la subida de ficheros a Google Drive, gdriveUploader, cuyo objetivo es el de subir, descargar y eliminar ficheros en Google Drive. Los ficheros PDF y CSV de salida debían de ser enviados a una unidad en Google Drive destinada a esto. Para ello se creó esta clase, que hace uso de una librería capaz de usar la API de Google Drive en Python, llamada Pydrive2. Esta clase permite subir ficheros (borrando los ficheros en Google Drive con el mismo nombre) a una unidad y borrar, descargar o extraer la URL de un fichero según su nombre.

BIBLIOGRAFÍA

Ballesteros de Andrés, C. (2019). *Conectografía de la gripe aviar: herramientas para predecir la difusión de la enfermedad.* Obtenido de E-Prints Complutense: https://eprints.ucm.es/id/eprint/57469/

Shi, W., & F. Gao, W. (21 de Mayo de 2021). *Emerging H5N8 avian influenza viruses.* Recuperado el 13 de Junio de 2021, de Science Magazine: https://doi.org/10.1126/science.abg6302

APÉNDICES

Apéndice A - Dependencias software del proyecto

A continuación, se enumeran las herramientas software y librerías de código que utiliza el proyecto. El código requiere de un servidor con un Sistema Operativo compatible con todas las herramientas principales, así como de una conexión a internet capaz de conectar con las APIs de AEMET y TuTiempo.net y con la web de Empres-i.

Principales dependencias

El siguiente listado recoge todo el software necesario para la ejecución del código:

- Python3: lenguaje de programación usado para hacer el código. Es también recomendable el uso de versiones >= 3.9, pues este fue el usado en el proyecto.
- Pip3: necesario para usar las librerías de Python. Usar la versión acorde con la versión de Python que se esté usando.
- MongoDB: para el manejo de la base de datos principal, la cual almacena datos sobre brotes, rutas y temperaturas entre otros.
- Neo4j: destinado al almacenamiento de rutas migratorias entre zonas de brote y comarcas mediante una estructura de grafos.
- Pandoc: usado para convertir los reportes de ficheros Markdown a PDF.

Librerías de Python usadas

Se puede obtener información de todas estas librerías en la web de Pypi.org

- Geolib, geopy y pygeohash: ofrecen operaciones relacionadas con el manejo de geohashes.

- GitPython: librería usada para interactuar con repositorios git. Usado para la subida de ficheros geoJSON al repositorio de la interfaz web.
- Pydrive2: ofrece clases pensadas para facilitar el uso de la API de Google Drive. Usado para la subida de ficheros CSV Y PDF a una unidad, entre otras funciones.
- Numpy: una popular librería especializada en operaciones sobre valores numéricos, aunque solo se usa su función de logaritmo neperiano en la fórmula del modelo.
- Pandas: otra famosa librería que se usa para manejar datos estructurados. En el proyecto se utiliza para realizar operaciones sobre matrices, principalmente con datos extraídos de ficheros CSV.
- Unicodecsv: permite el tratamiento de ficheros CSV con caracteres Unicode.
- Requests: usado para realizar peticiones a las APIs y a la web de Empres-i, extrayendo los datos que devolvían.
- Pymongo: librería oficial de MongoDB para interactuar con su base de datos.
- Neo4j: librearía oficial de Neo4j para interactuar con su base de datos.