A NOVEL COMPUTATIONAL ETHOLOGY FRAMEWORK FOR STUDYING ANIMAL BEHAVIOUR UNDER CLIMATE CHANGE (CEFABC2): A CASE STUDY OF LITTLE PENGUINS ON PHILLIP ISLAND

ABRAHAM ISAAC CAMELO GUERRERO

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF ARTS

GRADUATE PROGRAM IN INFORMATION SYSTEMS AND TECHNOLOGY
YORK UNIVERSITY
TORONTO, ONTARIO

JUNE 2026

Abstract

Animal behaviour is a biological indicator that provides insight into a species' interaction with its environment. Long-term records of animal behaviour form time series containing rhythms, anomalies, and trends in the activity of a species. Numerical and machine learning (ML) methods provide tools to automate the investigation of large amounts of data to detect patterns in complex behavioural time series. This thesis uses the little penguin colony on Phillip Island, Victoria, Australia as a case study, where the behaviour of this species has been recorded and analyzed over decades. To further support these studies, this thesis introduces a novel computational ethology framework on the study of animal behaviour under climate change (CEFABC2). This framework extracts and forecasts the dynamics of nightly penguin count time series, breeding success, and mean egg-laying date (MLD). CEFABC2 is built on four modules: the first module uses singular spectrum analysis (SSA) to process and reconstruct the nightly penguin count time series. From the SSA-reconstructed time series, the peaks are extracted and Gaussian Mixture Model (GMM) is implemented to estimate the peak width. The second module examines the correlations between the extracted peaks, the MLD, and breeding success. Gaussian process (GP) is applied to forecast the peaks of the nightly penguin count time series. Additionally, the performance of structural time series (STS), ARIMA, LSTM, GP, LightGBM, and Ridge models is evaluated for forecasting nightly penguin counts. The third module describes the climate of Phillip Island, using principal component analysis (PCA), K-means, and GMM to identify meteorological seasons. The fourth module integrates biological and climate variables, analyzing their correlations and forecasting breeding variables using ElasticNet, Ridge, Random Forest, and Gradient Boosting ML models. Finally, the CEFABC2 management system incorporates these components into a single tool aiming to support monitoring, forecasting, and future conservation management efforts towards the Phillip Island little penguin colony.

Acknowledgments

I would like to thank my supervisor, Professor Peter Khaiter, for his support and guidance throughout my master's journey. His experience in academia allowed me to understand and acquire the skills necessary for research. Through his expertise and interest in the applications of machine learning (ML) to ecology, I was able to appreciate how these two areas are connected.

I would also like to thank the examining committee, Professor Jairo Diaz-Rodriguez and Professor Arik Senderovich, for their time and feedback, which helped refine the work presented in this thesis.

I am grateful to God for this process of growth.

I would like to thank York University for providing me with the tools to advance in my professional career and contribute to the student community.

I would like to thank Phillip Island Nature Parks for supporting this project.

Finally, I would like to thank my family and friends, who were present throughout my master's journey. I am deeply grateful for their support, motivation, and words of encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Animal behaviour is the coordinated actions that an animal performs as a result of an external and internal stimulus (Levitis et al., 2009). As a biological indicator, animal behaviour not only informs about a species' state, but also about the environment it inhabits (Berger-Tal & Saltz, 2016). Since behavioural changes are one of the earliest signals of a colony about ecological shifts, studying animal behaviour allows the evaluation and organization of conservation programs (Berger-Tal et al., 2011).

Ethology is the discipline that studies animal behaviour, and the progress in machine learning (ML) and computer vision has provided this discipline with tools to record and analyze animal data implementing automated methods. This brings a new discipline denominated computational ethology, which implements numerical techniques to analyze animal behaviour (Anderson & Perona, 2014). For example, implementing ML methods, pose prediction of different species (Pereira et al., 2019; Hayden et al., 2022) and tracing animal movement (Romero-Ferrero et al., 2019; Hounslow et al., 2023) are two common applications in computational ethology. Nevertheless, such analytical techniques are generally applied to popular and easy-to-observe species (Couzin et al., 2023). Hence, the extension of this discipline to other species is still a research task.

Little penguins (*Eudyptula minor*) are an endemic species from Australia and New Zealand whose behaviour has been registered and analyzed for decades (Reilly & Cullen, 1981). This species is characterized by having an annual pattern in its behaviour, describing the breeding cycle and foraging strategies (Saraux et al., 2011; Salton et al., 2015). Additionally, this behaviour has also been associated with climate variables, observing the impact of changes in climate on biological variables such as mean egg-laying date (MLD) and breeding success (Johnson & Colombelli-Négrel, 2021). Even though quantitative models and some ML techniques have been applied to selected aspects of little penguin behaviour (Cullen et al., 2009), a research gap remains in the broader application of computational techniques to understand and forecast the behaviour of this species.

To address this gap, this thesis proposed a novel computational ethology framework for the study of animal behaviour under climate change (CEFABC2), that analyzes biological variables of the

little penguins, on one hand, and climatic variables of the region that the species inhabits, on the other hand. This framework is based on four modules that initially analyze the mean egg-laying date (MLD), breeding success and nightly penguin count. It then studies the climate variables separately, and it finally analyzes the biological and climate variables together in the last module. These modules conclude with a CEFABC2 management system that integrates all performed forecasts into a single tool. The goal of this thesis is to propose a set of computational units to analyze and forecast the biological variables, using the little penguin colony on Phillip Island, Australia as a case study.

The key research questions addressed in this study are:

[RQ1] How can Singular Spectrum Analysis (SSA) reconstruct the dominant temporal structure of the nightly penguin count time series?

[RQ2] How can seasonal activity peaks be extracted from the SSA-reconstructed nightly penguin count?

[RQ3] How can the temporal span of seasonal activity peaks extracted from the SSA-reconstructed nightly penguin count be estimated using Gaussian Mixture Models?

[RQ4] What are the temporal trends and relationships among nocturnal activity peaks, MLD, and breeding success in little penguins?

[RQ5] How accurately can Gaussian Process models forecast the day-of-year (DOY) of the next seasonal nocturnal activity peak in the nightly penguin count time series using partial-year observations?

[RQ6] Which model from among ARIMA, Ridge regression, LSTM, LightGBM, structural time series, and Gaussian Process can most accurately forecast the nightly penguin count across daily, weekly, and monthly resolutions?

[RQ7] What are the statistical and temporal characteristics of meteorological variables in the Phillip Island region?

[RQ8] What bioclimatic conditions characterize the Phillip Island region from 1981 to 2024?

[RQ9] How can PCA, DTW-based K-means, and Gaussian Mixture Models be used to identify the onset, duration, and midpoint of meteorological seasons in the Phillip Island region?

[RQ10] What relationships exist between climate variables and the MLD, breeding success, and nocturnal activity peak characteristics of little penguins?

[RQ11] How accurately can climate variables predict breeding characteristics in little penguins using Ridge regression?

[RQ12] How can CEFABC2 provide information about the state of the little penguin colony on Phillip Island using the forecasts of the biological variables?

The next chapters of this thesis are organized in the following style: Chapter 2 introduces the necessary concepts to understand the little penguin behaviour and, consequently, comprehend the nature of the biological variables used. It explains what animal behaviour is and how little penguins behave, including annual cycles, impact of the climate, and quantitative studies performed over this species in the literature. Chapter 3 presents an overview of the literature related to computational ethology and little penguins. It shows the main themes, productivity trends, and gaps in those knowledge areas.

Chapter 4 describes the region of study, data sources, and computational methods used in this thesis. It introduces CEFABC2, the proposed framework that implements computational techniques to study the little penguin colony. It describes the four modules of the framework, examining different aspects of animal behaviour and regional climate. Chapter 5 summarizes the results obtained by each module. It details the type of analysis performed and evaluates the performance of the predictive models. Finally, Chapter 6 contains the conclusions, limitations, and future work. Together, these chapters provide an extended instrument for applications of computational ethology in little penguin research.

The resulting research contributions are as follows:

RC1: Extraction of the main seasonal patterns in the nightly penguin count time series.

RC2: Evaluation of the effect of the quantity of SSA components on the reconstruction of the nightly penguin count time series.

RC3: Construction of new biological descriptors from the nightly penguin count, including peak timing and peak span.

RC4: Analysis of the temporal changes in reproductive and nocturnal activity variables across the study period.

RC5: Construction of the mismatch variable to describe the temporal difference between MLD and the main nocturnal activity peak.

RC6: Assessment of whether the variables extracted from the nightly penguin count provide complementary information about the reproductive dynamics of the colony.

RC7: Evaluation of the capacity of Gaussian Processes to predict the timing of seasonal activity peaks several months in advance.

RC8: Forecasting of the nightly penguin count time series using statistical, machine learning, deep learning, and probabilistic models.

RC9: Evaluation of the effect of SSA-based reconstruction on the forecasting performance of the nightly penguin count.

RC10: Analysis of the effect of temporal granularity on the forecasting of the penguin count, considering daily, weekly, and monthly representations.

RC11: Comparison of different model families for predicting the nightly penguin count under a multi-horizon forecasting strategy.

RC12: Analysis of the temporal variability of the climatic variables, considering their monthly distributions and long-term trends. This allows the description of how the regional climate changes across the year and across the study period.

RC13: Application of unsupervised machine learning models to characterize the meteorological seasons of the Phillip Island region. PCA, K-means, and GMMs are used to classify daily climatic observations into seasons and to extract seasonal descriptors, including season onset, season midpoint, and season span.

RC14: Construction of a comprehensive climate variable set required for the study of association between climate variables and biological variables.

RC15: Integration of the biological variables and climate variables into a single analytical module. This contribution relates the descriptors extracted from the nightly penguin count, MLD, and breeding success with the climate variables of the Phillip Island region.

RC16: Development of predictive models for biological variables using selected climate variables. This evaluates the capacity of climatic information to forecast MLD, breeding success, and peak span.

RC17: Construction of an integrated climate–behaviour modelling workflow within CEFABC2. This workflow combines correlation analysis, variable selection, trend analysis, ARDL analysis, and predictive modelling to study the colony under regional climatic conditions.

RC18: Development of the CEFABC2 management system as an integrated tool that uses the biological and climatic variables to provide forecasts of the state of the little penguin colony on Phillip Island. This system unifies the forecasts of nightly penguin count, MLD, breeding success, and breeding peak in a single dashboard to inform park management.

This thesis offers practical and methodological contributions for analyzing biological and climate data, forecasting key animal behavioural variables, and supporting future conservation and management strategies for the Phillip Island little penguin colony.

Chapter 2: Background

This chapter provides the context necessary for the thesis. It introduces the key biological and environmental concepts and terms in connection with the related quantitative techniques.

2.1 Animal behaviour

Multiple definitions of animal behaviour can be found in the literature. After surveying 178 biologists, Levitis et al. (2009) described animal behaviour as "the internally coordinated responses (actions or inactions) of whole living organisms (individuals or groups) to internal and/or external stimuli, excluding responses more easily understood as developmental changes".

Multiple phenomena fall under this umbrella including foraging, cognition, social interactions, communication, breeding, and responses to risk (Alcock, 2009). A separate discipline, *ethology*, studies animal behaviour from four perspectives: causation, which examines the immediate mechanisms generating behaviour (hormones or environmental stimuli); development, which studies how the behaviour is learned or acquired through genetics; adaptive function analyzing the practical value of the behaviour for survival; and phylogeny, which studies the evolutionary history of the behaviour (Davies et al., 2012).

Ethology allows researchers to identify how species respond to environmental changes (e.g., weather, urbanization, or habitat loss), observe population dynamics, detect endangered species, and generate valuable information for conservation strategies (Wong & Candolin, 2015).

2.2 Little penguins

Behaviour of little penguins (*Eudyptula minor*) is studied in this thesis. They inhabit the southern coasts of Australia (Sievwright et al., 2019) and New Zealand (Stahel & Nicol, 1988). This species is also known as blue penguin and is the smallest type of penguins. One of their largest colonies is located on Phillip Island, Victoria, Australia (Dupuis et al., 2023). They nest in burrows dug into sand, under rocks, vegetation, or caves (Chilvers et al., 2015). Little penguins are central-place foragers and, consequently, they return to their colony to feed chicks. Their activity on land is strictly nocturnal, spending daylight hours in the water. Their annual breeding cycle is variable and they undergo a catastrophic moult where they replace all their feathers simultaneously (Rodríguez et al., 2018).

On Phillip Island, these penguins are the main touristic attraction. Hundreds of thousands of visitors come to see them in the evening parade, representing a significant financial contribution destined for the species protection (Rodríguez et al., 2018). This species has captured the interest of researchers for a long time, with records traced back to 1918 (Nicholls, 1918). The Phillip Island Nature Parks is a unique organisation regularly monitoring and evaluating the status of the colony in order to inform and implement conservation strategies that support population growth and recovery (https://www.australiasnaturehub.gov.au/action-inventory/population-recovery-little-penguins-phillip-island).

2.2.1 Annual lifecycle

Little penguins exhibit seasonal patterns in their behaviour, including distinct breeding and non-breeding periods (Fig. 1). Their breeding season occurs each year during the austral spring and summer, while the non-breeding periods take place around fall and winter (Reilly & Cullen, 1981). Throughout the phases of the annual cycle, clear variations in their behaviour are observed. For instance, in the Phillip Island colony, foraging activities are performed near the island during the breeding season, while longer trips occur during the non-breeding season (Sutton et al., 2017). These variations in behaviour during the annual cycle are reflected in the observations of individuals after sunset on the beach of the island (Salton et al., 2015). This section describes the annual lifecycle of the species from the perspective of the breeding season, foraging behaviour, and nest attendance.

2.2.1.1 The breeding cycle

The breeding season of the colony has been recorded to happen in the austral spring and summer (Dupuis et al., 2023). The timing varies from year to year and is delimited by the egg-laying event and the fledging of the chicks. In the case of Phillip Island, the egg-laying period peaks between September and October (Dann et al., 2000), yet it can still happen earlier in May or later in December (Reilly & Cullen, 1981). Geographical location is also another relevant factor of variance in the timing of this season. For instance, egg-laying activity has been recorded in Western Australia from April to November while in New South Wales, Australia - from July to November (Klomp et al., 1991). After the eggs are laid, the incubation period starts. It lasts about 35 days. Chicks then hatch, leading to the guard stage. This stage takes about three weeks, and one of the parents stays with the chick(s) while the other forages at sea. After this stage, the post-guard stage

starts, where the chick is able to thermoregulate itself and both parents can resume foraging at sea. This stage takes about 9 weeks (Chilvers et al., 2015). After this, chicks fledge and leave the burrows. Corresponding timelines are shown in **Fig. 1.**

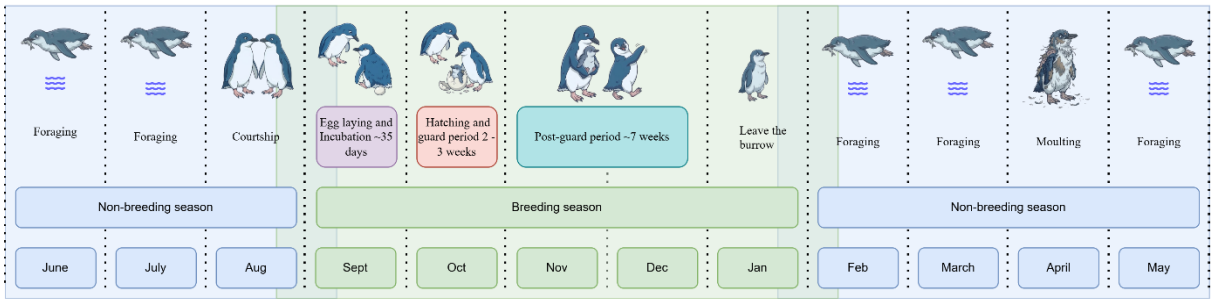

Fig. 1. Annual lifecycle and timeline of little penguins.

2.2.1.2 Foraging behaviour

Little penguins forage at sea, searching for food for their chicks and themselves. As visual hunters, they forage during daylight, returning ashore after sunset (Saraux et al., 2011). They optimize the daylight availability by leaving their burrows before sunrise and, hence, are known for their nocturnal activity on land. Regarding the foraging range, little penguins exhibit central-place foraging behaviour during the breeding season. Still, they show high plasticity in the range after the breeding season is over (Sutton et al., 2017). During the incubation or brooding period, the parents take turns and forage within a restricted range of 20–30 km from the burrows (Chilvers et al., 2015). During the post-guard period, long-term trips are more frequent, staying away from the burrows for longer periods and traveling longer distances. During the winter (non-breeding) period, the species has no chick-feeding-related restrictions and longer trips are common. Long-term trips have been recorded up to 710 km (Weavers, 1991), and they can take weeks.

2.2.1.3 Nest attendance

Leveraging the fact of the nocturnal ashore nature of little penguins, the number of individuals attending their nests at night is recorded. This count is governed by complex interactions with the environment and intrinsic factors of the colony. Some of these factors are the already mentioned breeding season and circadian rhythm, since they are related to the frequency with which little penguins leave/return to their burrows and the distance/time they travel at sea. During the winter/non-breeding season, attendance registers the lowest values of the year (Salton et al., 2015).

In contrast, more penguins can be observed during the breeding season due to their duties with the chicks. Attendance is high during the incubation and guard stages, while slightly decreasing during the post-guard stage. An additional attendance peak has been recorded on Phillip Island during autumn (April to May). This peak is led by older penguins who intend to breed more. Additionally, little penguins undergo a moulting season, when penguins stay in their burrows for about 3 weeks, renewing their feathers (Reilly & Cullen, 1981). This occurs between February and April. Other specific phenomena have been observed to have an effect on the time of the nest attendance, such as the moon phase (Rodríguez et al., 2016) or fog (Chiaradia et al., 2007).

2.2.3 The effect of climate

Species behaviour is strongly influenced by climate conditions. Sea surface temperature (SST) is a primary factor that affects the behaviour of little penguins. In Phillip Island studies, SST has been associated with earlier egg-laying dates and higher breeding success (Cullen et al., 2009). On the other hand, strong wind speeds during the breeding season have been linked to lower breeding success (Johnson & Colombelli-Négrel, 2021). Additionally, high humidity increases the survival of adults during the moulting period, while strong precipitation increases their mortality rates (Dann & Chambers, 2013). Ultimately, extremely high air temperatures provoke heat stress, causing hyperthermia (Ganendran et al., 2016).

2.2.4 Computational analysis /Quantitative characteristics

The section describes the most common quantitative characteristics applied in the studies on little penguins.

Breeding parameters are amongst the most frequent indicators, including mean egg-laying date (MLD) (Cannell et al., 2012), hatching success, breeding (fledging) success (Saraux & Chiaradia, 2022), and clutch size (Reilly & Cullen, 1981).

Foraging and diving metrics are largely studied due to the progress in bio-logging technology, with the focus on trip duration, dive depth, and swimming speed (Weavers, 1991).

Body condition is mostly characterized by body mass (Salton et al., 2015), though studies on bill length and depth also exist.

Species-related environmental variables include the most frequently studied SST followed by terrestrial wind speed, rainfall, humidity, and ambient air temperatures (Dann & Chambers, 2013).

Quantitative and statistical techniques. Multiple linear regression (MLR) and correlation analysis are widely used techniques to predict phenology. In particular, they have been applied to explain the interplay of breeding indicators with monthly environmental parameters. In particular, Cannell et al. (2012) hypothesised that breeding success and mean chick body mass variables could be explained through SST over the period April to May. MLR was used to explain poorer breeding success as SST increases. MLR was also used to link earlier MLD with high SST (Cullen et al., 2009). Other approaches involve generalized additive mixed models extracting non-linear relationships from the variables, such as in the study by Saraux & Chiaradia (2022), who related breeding performance of penguins to their age.

Support vector machines (SVM) have been implemented to classify the type of movement individual animals performed at sea. Based on accelerometer data, Carroll et al. (2018) classified whether the penguin was swimming or handling prey. K-means clustering was also applied to distinguish between the into bottom time, dive duration, and maximum depth in penguin's movement at sea (Phillips et al., 2021). Hidden Markov models (HMM) have been used to reconstruct foraging paths (Dupuis et al., 2023). Conditional inference trees (CIT) were employed to model foraging characteristics, such as trip distance or success of catching prey, using environmental variables as predictors (Cavallo et al., 2020). The latter model also helped to quantify the feature importance of the environmental variables. Principal component analysis (PCA) has been carried out on breeding performance variables (Zimmer et al., 2011). From our observations, applications of machine learning (ML) methods in the little penguin studies remain rather limited.

2.3 Computational ethology

Investigation of animal behavioural patterns requires extensive monitoring datasets and advanced quantitative methods. Such an integration of the disciplines has been achieved within the field of *computational ethology*. The term was coined by Anderson and Perona (2014) describing how the computational tools help overcome limitations in classical ethology, such as the time-consuming efforts in the collection of behavioural observations, the subjectivity in the observations, the limited recording capacity, and the lack of granular information for complex behaviours. In

essence, computational ethology records animal behaviour using high resolution sensors, then analyzes the records using quantitative techniques, and builds models that explain the observed phenomena (Berman, 2018).

Essentially, c*omputational ethology* is the research field that studies animal behaviour through automated computational processes. This domain has grown due to two aspects: the development of sensors/tracking devices and ML advances (Arablouei et al., 2021). The study of animal behaviour requires both the collection of data (observation) and its interpretation (Immelmann, 2012). Historically, data collection has been performed manually (Goodwin et al., 2020), being time consuming. Nowadays, the research community has multiple instruments to automatically collect data, such as GPS (Global Positioning System) used in collars attached to animals to track their location (Northrup et al., 2019), or tags placed on sea animals to record their displacement underwater (Takagi et al., 2025). It should be noted that, though there are publications on ML methods in ethology, ranging from disease detection (Riaz et al., 2025) to identifying a specific animal activity (Kleanthous et al., 2018), their corpus on little penguins remains rather scarce.

This thesis is focused on analyzing animal behaviour using quantitative methods. Hence, to understand more about this scientific field and how the present study fits into and contributes to it, a domain-specific bibliometric analysis and literature review are performed in the next chapter.

Chapter 3: Bibliometric Analysis and Literature Review

In this chapter, we start by performing a bibliometric analysis of publications on computational ethology and little penguins. First, we describe the workflow of bibliometric analysis and formulate multiple research questions, all related to the topic of the thesis. Gaps and growth opportunities within the domain are identified. Finally, a review of the most important publications is presented.

3.1 Bibliometric analysis

Bibliometric analysis is a standard and well-known method to understand the structure of a research area. It can be conducted in two ways: performance analysis and science mapping analysis. Performance analysis measures the contribution of research constituents (Cucari et al., 2023). This analysis is performed at multiple levels: authors, institutions, countries, or publication outlets (Iqbal et al., 2023). It studies the impact of each category through the number of citations they have and their productivity through the number of publications. This method answers questions such as who, where, and how much scientific knowledge has produced. Some of the common metrics are number of publications, most productive actors, and citation counts.

Complementarily, science mapping focuses on the relationships between the constituents. It uses network analysis and examines three types of structures: *conceptual structure*, which identifies the main topics and their connections; *intellectual structure*, where similar articles are clustered; and *social structure*, which displays collaborations between authors (Iqbal et al., 2023). Dedicated software is used for these studies, of which the most well-known are VOSviewer (Van Eck & Waltman, 2016), CiteSpace (Chen, 2014), Bibliometrix (Derviş, 2019), and SciMAT (Cobo Martín, 2012). The workflow of a bibliometric analysis is shown in **Fig. 2**.

Defining the scope is the initial step in bibliometric analysis (Donthu et al., 2021) where the research questions are determined and the bibliometric data is collected. For the purposes of the present study, seven research questions have been formulated:

Q1: What are the publication volume and trends in the domain of computational ethology and research on little penguins?

Q2: Who are the most productive contributors in the domain?

Q3: What are the most influential publications in the domain?

Q4: What are the major themes in the domain?

Q5: How do the themes evolve through time?

Q6: What are the research streams in the domain?

Q7: What are the characteristics of the existing studies on behavioural time series in the domain?

The questions *Q1-Q3* are answered through performance analysis, the questions *Q4-Q6* are addressed with science mapping, and the question *Q7* is responded based on observations from the bibliometric analysis.

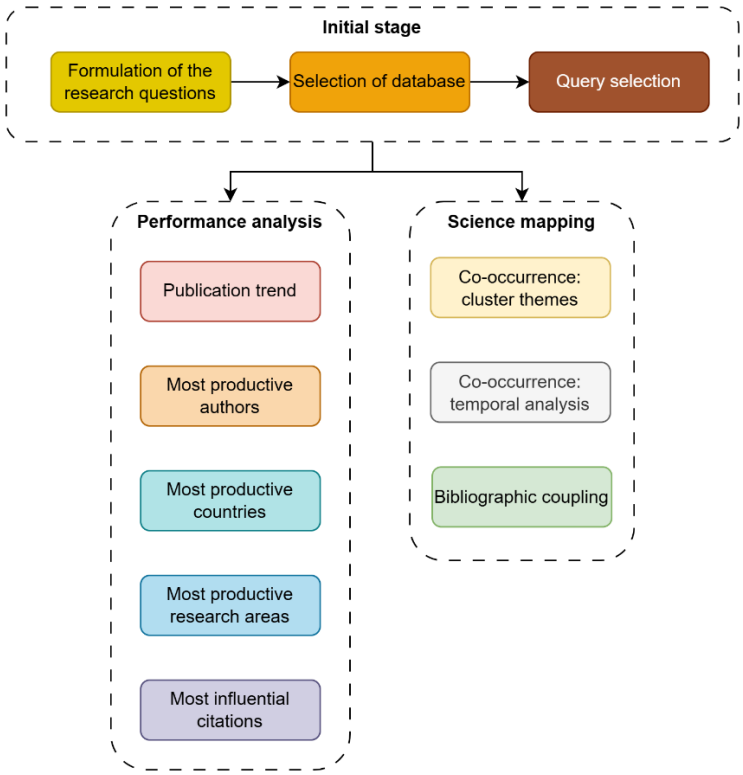

Fig. 2. Bibliometric analysis workflow.

Additionally, we extracted available publications on little penguins in connection with the computational ethology domain. Accordingly, the following three research questions have been added:

Q8: What are the major themes of the literature on little penguins?

Q9: Which papers in the little penguin literature fall within the domain of computational ethology?

Q10: What gaps can be detected in the literature on computational ethology on in relation to little penguins?

Web of Science core collection (https://clarivate.com/academia-government/scientific-and-academic-research/research-discovery-and-referencing/web-of-science/) was used for data retrieval , with the query *TS = ("machine learning" OR "computational ethology" OR "time series" OR "deep learning" OR "statistical model*") AND ("animal movement" OR "movement ecology" OR "behavioural ecology" OR "animal behavior" OR "animal behaviour") AND LA = (English) AND DT = (Article OR Review).* The query returned 887 publications.

3.1.1 Q1: What are the publication volume and trends in the domain of computational ethology and research on little penguins?

To answer this question, we analyzed the productivity trend in each domain. **Fig. 3a** shows the evolution of the number of publications over the years on both computational ethology and little penguins. Regarding the computational ethology domain, there were few publications from 1980 to 2007, with large gaps in the 1980s and 1990s. After 2007, this field started to grow rapidly, reaching 167 publications in 2025. In contrast, the little penguin literature showed higher publication activity from 1985 to around 2015. After that, the productivity of the computational ethology field increased significantly faster than that of the little penguins. Fig. **3b** shows the intersection of these two fields which share just eight publications in common, having the earliest published in 2005. Some of them are mentioned later in this chapter.

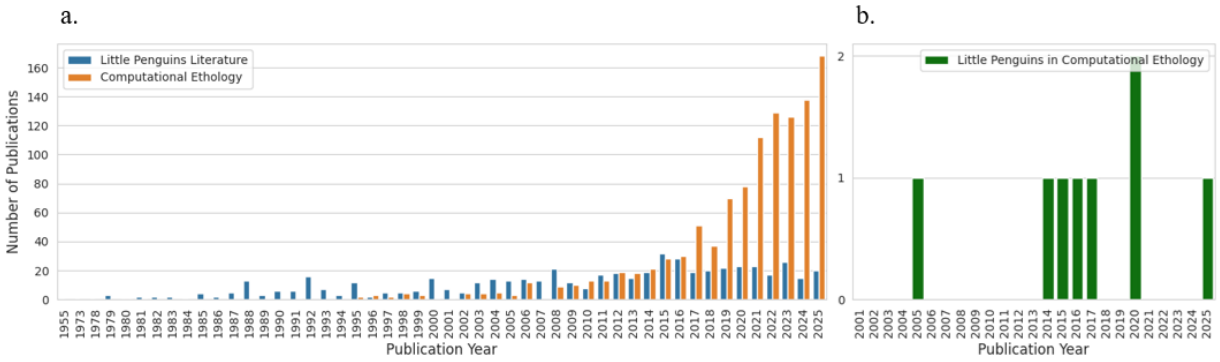

Fig. 3. Publication trend on: a. computational ethology and little penguins, and b. little penguins in computational ethology.

3.1.2 Q2: Who are the most productive contributors in the domain?

This question was addressed at three levels: authors, countries, and research areas. **Fig. 4** shows the most productive authors within retrieved publications. R. Langrock is the most productive author, with 20 publications. This author has a strong focus on Hidden Markov Models (HMMs) and their application to animal movement modelling (Leos-Barajas et al., 2017; Adam et al., 2019; Stoye et al., 2025). K. Yoda is the second most prominent author, whose publications are centered on deep learning applications to analyze movement trajectories captured through animal-borne sensors (e.g., accelerometers) (Otsuka et al., 2024). The next two authors, M. Auger-Méthé and T. A. Patterson, have a similar to R. Langrock focus, leveraging HMMs and State-Space Model (SSM) methods (Dupont et al., 2025; McClintock et al., 2020). R. Nathan deals with ML implementations using GPS (Yu et al., 2021).

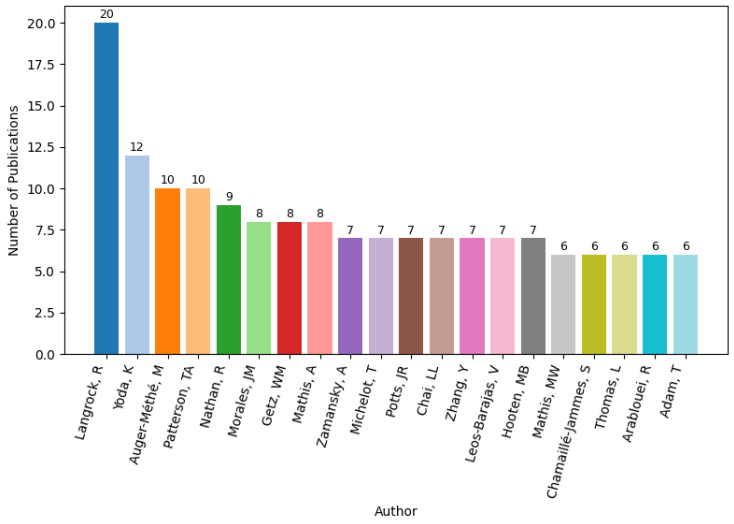

Fig. 4. Most productive authors.

Table 1 shows the contribution of the top 10 most productive countries in the domain. The USA, England, and Germany are the top three countries, whose researchers produced the majority (~61%) of all publications. There are 91 countries in total that have contributed to this area, with 27 countries contributing one publication each.

The collection of retrieved publications represents 72 research areas. **Fig. 5** shows the top 10 research areas. Environmental sciences and ecology dominate the area studying the dynamics and relationships between organisms and their surroundings. Since computational ethology involves the use of technology to automate the study of animal behaviour, it is not surprizing that computer

science, behavioral sciences, zoology, agriculture, science and technology, veterinary science, and engineering are among the most prominent research areas.

Table 1. Top 10 of country-wise article production

Country	Publications
USA	403
ENGLAND	152
GERMANY	138
AUSTRALIA	108
CHINA	90
FRANCE	86
CANADA	80
SCOTLAND	79
JAPAN	72
SPAIN	49

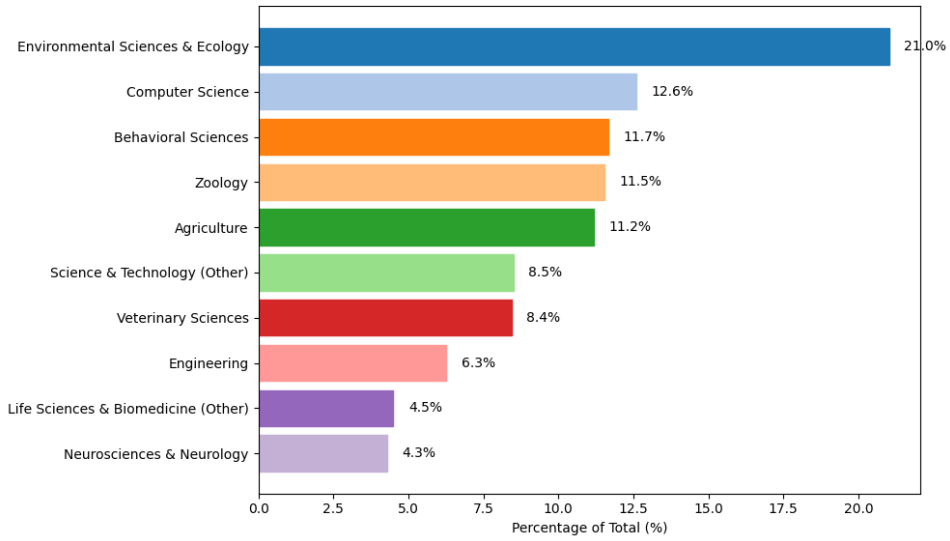

Fig. 5. Most productive research areas.

3.1.3 Q3: What are the most influential publications in the domain?

Table 2 shows the top 10 most cited publications. These publications connect machine learning and deep learning with the study of animal behaviour. Computer vision stands out as a useful

Table 2. Top 10 most cited publications

Paper Title	Authors	Year Published	Journal / Conference	Total Citations
DeepLabCut: markerless pose estimation of user-defined body parts with deep learning	Mathis et al.	2018	Nature Neuroscience	2897
Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning	Norouzzadeh et al.	2018	Proceedings of the National Academy of Sciences	695
Use of resistance surfaces for landscape genetic studies: considerations for parameterization and analysis	Spear et al.	2010	Molecular Ecology	536
ctmm: an R package for analyzing animal relocation data as a continuous-time stochastic process	Calabrese et al.	2016	Methods in Ecology and Evolution	472
A connectome and analysis of the adult Drosophila central brain	Scheffer et al.	2020	eLife	452
Applications for deep learning in ecology	Christin et al.	2019	Methods in Ecology and Evolution	418
Perspectives in machine learning for wildlife conservation	Tuia et al.	2022	Nature Communications	412
JAABA: interactive machine learning for automatic annotation of animal behavior	Kabra et al.	2013	Nature Methods	394
SLEAP: A deep learning system for multi-animal pose tracking	Pereira et al.	2022	Nature Methods	386
Applications of machine learning in animal behaviour studies	Valletta et al.	2017	Animal Behaviour	370

technology for analyzing animal pose and species (Mathis et al., 2018; Norouzzadeh et al., 2018; Christin et al., 2019; Kabra et al., 2013; Valletta et al., 2017). At least four of these publications provide new tools to the research community on implementing ML/DL techniques for animal data. In these publications, the importance of ML usage for ecology is demonstrated.

3.1.4 Q4: What are the major themes in the domain?

To explore the main themes, keyword co-occurrence analysis was applied. Co-occurrence analysis connects related keywords, whose link is defined by the number of times the two keywords have appeared together in publications. A set of keywords that are commonly linked to each other are clustered. **Fig. 6** shows the keywords of the publication collection and their links. Twelve clusters were identified.

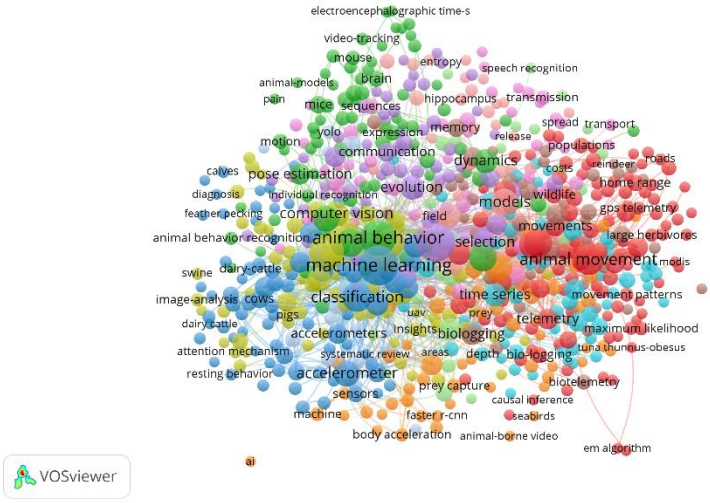

Fig. 6. Co-occurence analysis.

The first (red) cluster includes keywords related to animal movement and habitat. Additionally, it contains keywords related to bio-logging, general computational methods, and different species. In the second (green) cluster, many keywords are related to ML methods, technologies for behaviour recording (some of them focused on video format), and behavioural variables. The third (blue) cluster contains keywords related to foraging behaviour, physiology, and the marine environment. This cluster focuses on quantifying animal energetic expenditure and physiological states, and relating them to foraging behaviour, prey availability, and environmental variability, particularly in marine systems and seabird communities. The fourth (yellow) cluster embraces

several concepts related to population dynamics and long-term analysis. This cluster focuses on modeling demographic variables in relation to climate.

The fifth (purple) cluster involves breeding-related and climate variables, associating breeding performance with environmental conditions. Cluster six (cyan) contains human-interaction terms and wildlife responses. The seventh (orange) cluster is related to probabilistic modelling techniques to describe animal behaviour. The theme of the eighth (brown) cluster is around social structures of species and collective behaviour, incorporating words related to interactions among individuals. The ninth cluster (pink) introduces keywords about data management, devices for data collection, and data visualization. The tenth (light red) cluster is also method-focused, yet its topic is around time series, temporal patterns, and mathematical modelling. The eleventh (light green) cluster introduces concepts related to optimization, such as parameter estimation, model calibration, and accuracy. Ultimately, the twelfth (light blue) cluster addresses the theoretical foundations of animal behaviour, focusing on how individuals make adaptive decisions.

In this set of clusters, we found themes related to animal movement, foraging behaviour, population dynamics, breeding, and social structures. These subjects are studied with probabilistic modelling, time series structures, and ML methods.

3.1.5 Q5: How do the themes evolve through time?

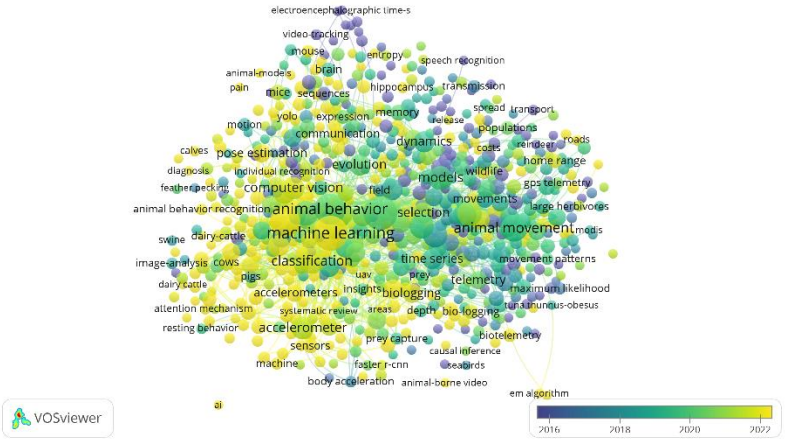

Fig. 7. Average year of keywords.

Fig. 7 shows the average year when the keyword appeared in the collected literature. Until 2018, themes were around classical ecology, having keywords such as population dynamics and

breeding. In 2019, concepts related to statistical modelling and decision theory were integrated. Then in 2020–2021, there was an explosive ML integration to analyze GPS data or implement computer vision. In 2022–2023, big data terms appeared, fostering the expansion of data infrastructure. In the last two years, state-of-the-art models are being referenced, such as transformers. These trends show the evolution of the field and the transition from classical analytics to computer-based approaches.

3.1.6 Q6: What are the research streams in the domain?

Bibliographic coupling was implemented to analyze the research streams. It connects papers that share references. The more references they share, the stronger the connection. Similar publications are then clustered under the assumption that similarity can be measured based on these shared references. **Fig. 8** shows the six clusters found in the literature on computational ethology.

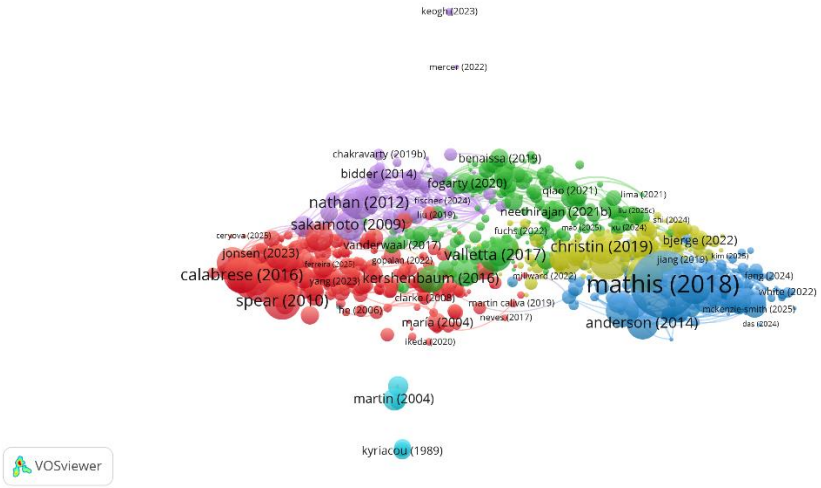

Fig. 8. Bibliographic coupling for computational ethology.

Cluster 1 (red) focuses primarily on the statistical modeling of animal movement and ecology. Hidden Markov Models (HMMs) and State-Space Models (SSMs) are the main probabilistic methods implemented in this section to classify animal behaviour captured with telemetry (Hu et al., 2021; Li & Bolker, 2017). In this cluster, several papers introduce explanatory models for species habitat selection using classical ecological models, such as Resource Selection Functions (RSFs) and Step Selection Functions (SSFs) (Fukasawa & Higashide, 2025; Forester et al., 2009). Theoretical movement models are also implemented to describe dispersal patterns of species (Scharf & Buderman, 2020).

Cluster 2 (green) is focused on the automated monitoring of farm animal behaviour and welfare using sensors and ML. Modelling activity patterns of livestock to optimize processes, such as feeding time or early disease detection, is a recurrent topic in this cluster (Hashimoto et al., 2025). Wearable sensors, such as accelerometers and gyroscopes, are frequently used to capture data. This data is commonly used to classify different aspects of cattle behaviour, such as grazing, ruminating, lying, or standing (Arablouei et al., 2021). ML and DL algorithms are implemented and compared in the classification tasks that such processes entail (Kleanthous et al., 2018). Hence, this cluster represents the agricultural applications of ML in computational ethology.

Cluster 3 (blue) integrates the use of DL and computer vision with the study of animal behaviour. Pose estimation is a recurrent topic, where computer vision is implemented to track specific moving parts of an animal (e.g., joints) (Gillis & Datta, 2017; Odo et al., 2023). There is also great interest in investigating the connection between the brain and animal behaviour (Urbaniak et al., 2024).

Cluster 4 (yellow) also integrates the use of DL with computer vision. Here, advanced DL models, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers are widely applied (Fazzari et al., 2025). Wildlife monitoring is performed through trap cameras, used for species identification and counting (Bilodeau et al., 2022). Farm animal welfare and management are also studied in this cluster, for example, computer vision for process optimization, such as tracking chicken eggs laid on the floor (Bist et al., 2023).

Cluster 5 (purple) applies ML to analyze bio-logging data. Movement classification is performed to detect whether animals are resting, traveling, mating, or performing other activities (Aulsebrook et al., 2024). Random Forests (RF), Support Vector Machines (SVM), k-Nearest Neighbors (kNN), and XGBoost are among widely used supervised ML algorithms. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are amongst the frequently used DL algorithms (Otsuka et al., 2024). Unsupervised ML models are also used to classify different behavioural states without attributing prior labels (Sur et al., 2023). This cluster, likewise cluster 2, contains the applications of ML models to analyze behavioural data collected from on-animal sensors, yet broader applications are found.

Cluster 6 (cyan) emphasizes the study of fine-scale behavioural patterns. One common theme is the study of Drosophila (fruit flies) using cameras (Martin, 2004). Studies analyze courtship

sounds and movement patterns of this species. This cluster is quite isolated in the image, sharing just a few connections with other clusters.

3.1.7 Q7: What are the characteristics of the existing studies on behavioural time series in the domain?

The research datasets on animal behaviour have multiple time spans, ranging from a few days (McKenzie-Smith et al., 2025) to several decades (Piper et al., 2023). The data sampling rate also varies from hundreds of records per second to just one record per season (Arndt et al., 2018). One of the largest datasets found in this pile of papers involves 30 years of observations on the sleeping locations of primates on the every day basis (Jacobson et al., 2024). Yet, few datasets with these long-term and high-sampling rate are found in the literature. Larger datasets are also present with spans of 37 years, but their sampling rate is coarser (Thorson, 2022).

3.1.8 Q8: What are the major themes of the literature on little penguins?

To observe the major themes in little penguin literature, a keyword co-occurrence analysis was performed and generated ten clusters (**Fig. 9**).

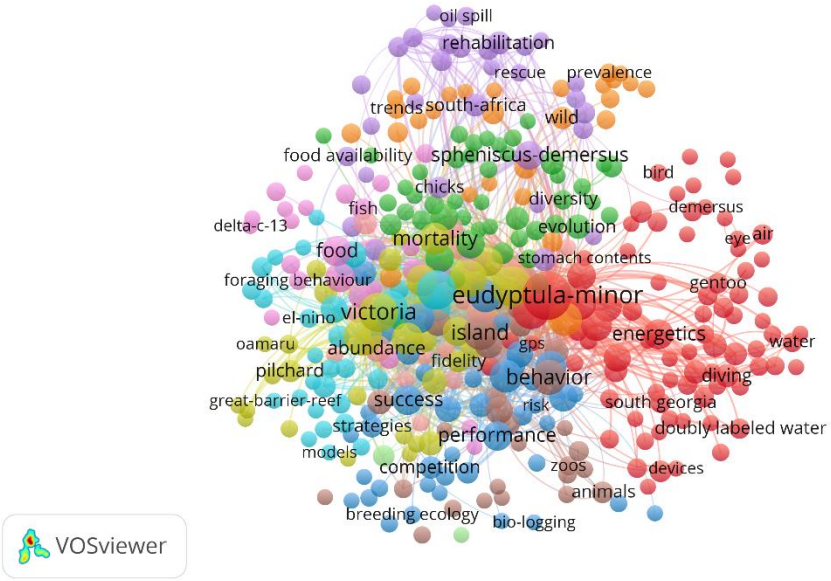

Fig. 9. Co-occurrence analysis in the little penguin literature.

Cluster one (red) contains keywords related to energetics, metabolism, and foraging of little penguins and other animal species. Hence, this cluster characterizes the balance between energy

acquisition through underwater foraging and energy consumption caused by diving behaviour, metabolism, and thermoregulatory processes. **Cluster two (green)** contains keywords related to population dynamics, genetics, and geographical locations. This suggests a general topic of how little penguins are distributed geographically and what their genetics is. **Cluster three (blue)** has keywords about breeding phenology, social interactions of the colony, and anthropogenic disturbances. This cluster depicts reproductive behaviour and characteristics that affect it. **Cluster four (yellow)** is related to population modeling and survival.

Cluster five (purple) combines keywords on oil spills, restoration, breeding performance, and other marine species. Hence, the themes of this cluster are environmental disturbances and their impact on marine species. **Cluster six (cyan)** introduces climate-related topics and their relationship to marine ecosystems and foraging performance of the little penguins. **Cluster seven (orange)** explores health issues and pathogens in the species. **Cluster eight (brown)** connects human–penguin interactions and conservation efforts. **Cluster nine (pink)** includes keywords mostly concerning penguin diet. Finally, **Cluster ten (light green)** contains keywords related to movement ecology.

As can be concluded from the co-occurrence analysis on little penguins, this research field is purely ecological, where most of the keywords are related to classical ecological statistics and quantitative inference. Here, no machine learning or deep learning terminology is found. Although from previous observations, we know that there are a few studies applying SVM, HMM, K-means, and PCA, but they are not captured by the VOSviewer tool.

3.1.9 Q9: Which papers in the little penguin literature fall within the domain of computational ethology?

There are some papers on little penguins that can be linked to the field of computational ethology. The diving behaviour of little penguins have been analyzed using biologging data and correlated to environmental factors using generalized linear models (Meyer et al., 2020). Additionally, accelerometer time-series data has been analyzed using support vector machine (SVM) to predict successful prey (Carroll et al., 2014). The prediction of the mean egg-laying date using linear models and sea surface temperature (SST) as the predictor was conducted by (Cullen et al., 2009).

3.1.10 Q10: What gaps can be detected in the literature on computational ethology on in relation to little penguins?

Table 3. Relevant studies on computational ethology and little penguins

No.	Study	Author	Relevance to This Thesis	Identified Gap Addressed in This Work
1	Predicting onset and success of breeding in little penguins Eudyptula minor from	Cullen et al. (2009)	This is one of the few papers that aim to forecast a biological variable of Little Penguins. It finds strong correlations with sea surface temperature	Other climate variables should be analyzed. Also their relationship with other biological variables
2	Daily nest attendance and breeding performance in the little penguin *Eudyptula minor* at Phillip Island, Australia	Chiaradia and Kerry (1999)	These two papers are among the few that analyzes the daily attendance of penguins deeply. One analyzed the attendance patterns during breeding season. The other analyzed how the attendance is affected based on an external factor	The papers disregards attendance patterns during non-breeding season. Additionally, no long-term analysis of the attendance is performed
3	Penguin colony attendance under artificial lights for ecotourism	Rodríguez et al. (2018)		
4	Change detection in animal movement using discrete wavelet analysis	Sur et al. (2014)	They analyzed the time series by decomposing them with discrete wavelets. They show the practical applications of non-parametric decomposition models	Wavelet models still has a predefined basis. Using a full-adaptive data-driven decomposition technique can provide additional insights

Throughout the literature on the species, multiple variables are analyzed to characterize their behaviour. Nevertheless, most of the variables either provide a long-term perspective with wide temporal granularity, as is the case for the mean egg-laying date (Cullen et al., 2009), or analyze variables with fine granularity for a short period, as is the case with daily nest attendance (Chiaradia & Kerry, 1999).

In the literature on little penguins related to computational analysis, some applications of machine learning have been used to address specific tasks, such as behavioural classification or event detection. However, the implementation of these methods to understand long-term temporal patterns in biological variables remains limited (Carroll et al., 2014).

Additionally, although the impact of climate on little penguins has been extensively studied, most analyses consider climatic variables individually or within small sets of predictors. This approach overlooks the potential combined effects of climatic variables on the biological variables under study (Meyer et al., 2020).

Table 3 shows some of the relevant studies on computational ethology and little penguins, highlighting methodological approaches, relevance to this thesis, and the gaps that the present work addresses.

3.2 Conclusion

Computational ethology has emerged as a rapidly expanding interdisciplinary field integrating machine learning, signal processing, and quantitative modelling to study animal behaviour. Yet it still has room for contributions to the study of complex long-term biological time series with fine granularity.

Literature on little penguin demonstrates a substantial opportunity for applications of computational analytical techniques. This species has been widely studied from ecological and biological perspectives, and modern rapid development of ML algorithms and availability of time-series provide effective resources for additional insights.

The next chapter outlines methodology underlying this thesis.

Chapter 4: Methodology

The chapter introduces the Phillip Island as the study area, data sources used in this research and a novel computational ethology framework on animal behaviour study under climate change (CEFABC2) in application to the little penguin colony.

4.1 Study region

Phillip Island (38°29′S 145°14′E / 38.483°S 145.233°E) is located in the south of Australia, south-southeast of Melbourne, Victoria (**Fig. 10**). It hosts one of the largest colonies of little penguins (Sutherland & Dann, 2012). This colony is famous for the spectacular little penguin Parade that takes place every evening, whereby they come onshore from the sea to rest for the night, serving as the main tourist attraction (https://penguins.org.au/). The island is characterized by a temperate climate, with mean temperatures around 10°C in winter and 20°C in summer. The wettest months and strongest winds are observed in winter.

Over the years, this region has been experiencing warming temperatures, and the trend is expected to continue (Chambers et al., 2014). Typically, studies of little penguins on Phillip Island also incorporate meteorological data from Port Phillip Bay and Bass Strait, given that both aquatic locations are foraging areas for individuals from the colony.

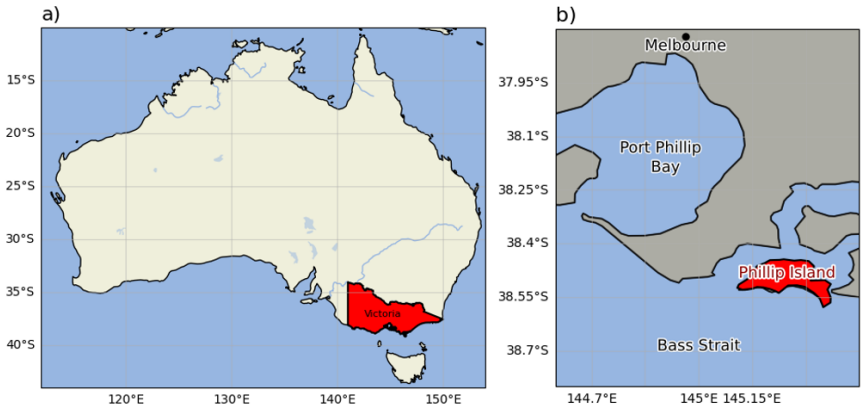

Fig. 10. Area of study.

4.2 Data sources

The NASA Prediction of Worldwide Energy Resources (POWER) project was used for the collection of climate variables. This project provides global meteorological data with a grid

resolution of 0.5° for research and practical applications (Sparks, 2018). These data are accessed through an API and have been used in applications related to agriculture, such as crop yield simulation and evapotranspiration modeling, and in hydrology, such as rainfall prediction and drought simulation. This dataset is particularly useful for climate change analyses, providing over 40 years of daily climate data.

In this study, the following daily variables were collected: mean air temperature, maximum air temperature, minimum air temperature, precipitation, relative humidity, and wind speed. The extracted data span from 1981 to the end of 2024, and the surface over which these data were collected is shown in **Fig. 11**. This region covers a grid tile delineated by 35°S to 45°S and 138°E to 152°E, due to its relevance to the breeding phenology of the colony as demonstrated by Cullen et al. (2009).

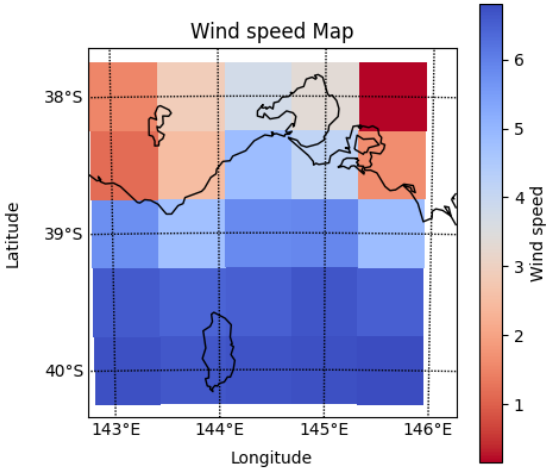

Fig. 11. Area of the climate data collection (the wind speed is shown as an example).

Additionally, we collected the sea surface temperature (SST) over the same region, covering the same range of years. The data were obtained from the NOAA National Centers for Environmental Information (NCEI) Optimum Interpolation Sea Surface Temperature (OISST) dataset, Version 2 (Reynolds et al., 2008), which provides the monthly average values of the SST.

Finally, the data related to the little penguin colony in Phillip Island were provided by Phillip Island Nature Parks (https://www.penguins.org.au/). These data include the nightly penguin count, MLD, and the breeding success. The nightly penguin count contains data from 1982 to the end of 2024,

and it consists of records of the number of counted penguins arriving at Summerland Peninsula on Phillip Island. The data of the nightly penguin count are shown in **Fig. 12**.

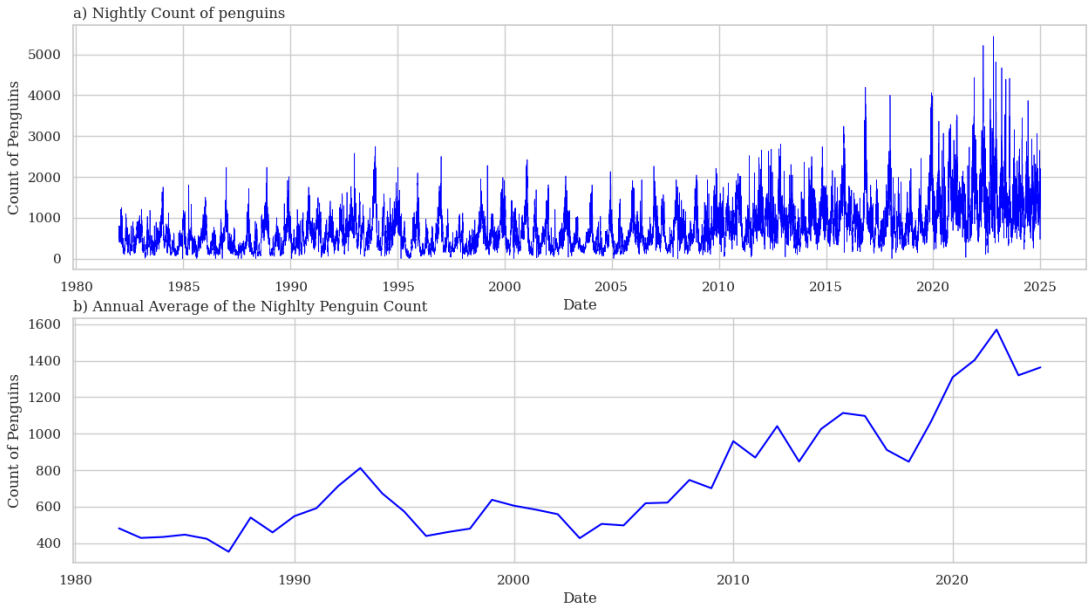

Fig. 12. Nightly penguin data: a) raw nightly count; b) annual average of the nightly count.

The MLD and breeding success have annual granularity. **Fig. 13** shows the raw time series of the MLD and breeding success from 1982 to 2023.

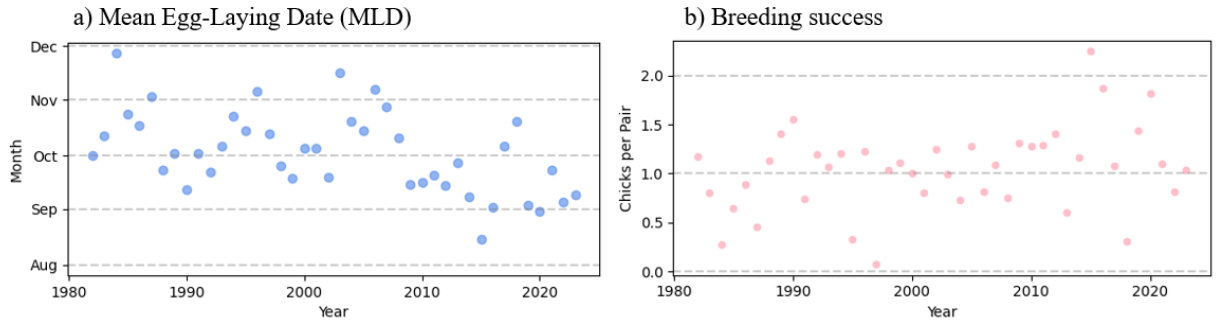

Fig. 13. Annual raw data on: a) MLD and b) breeding success.

The following section describes the CEFABC2 framework and its underlying methods.

4.3 CEFABC2 framework and methods

The methodological objective of this study is to identify and model the temporal patterns in the behaviour of the little penguin colony on Phillip Island, and to evaluate their relationship with regional climate dynamics. To achieve this objective, the CEFABC2 framework was developed, integrating time series analysis and ML techniques.

The CEFABC2 framework consists of four modules (**Fig. 14**). *Module 1* focuses on the data processing techniques used to detect the seasonal components of the time series and to engineer new behavioural features from those components. In this module we implemented singular spectrum analysis (SSA) to decompose and reconstruct the time series, and Gaussian Mixture Models (GMMs) to estimate the width of the identified peaks in the reconstructed time series.

Module 2 examines the relationship between the derived variables (i.e., activity peaks and peak width), the MLD, and breeding success. In this module, we built Gaussian processes (GP) to estimate the day of the year (DOY) of time series peaks. Furthermore, we assessed the performance of statistical, ML and DL models to forecast the nightly penguin count.

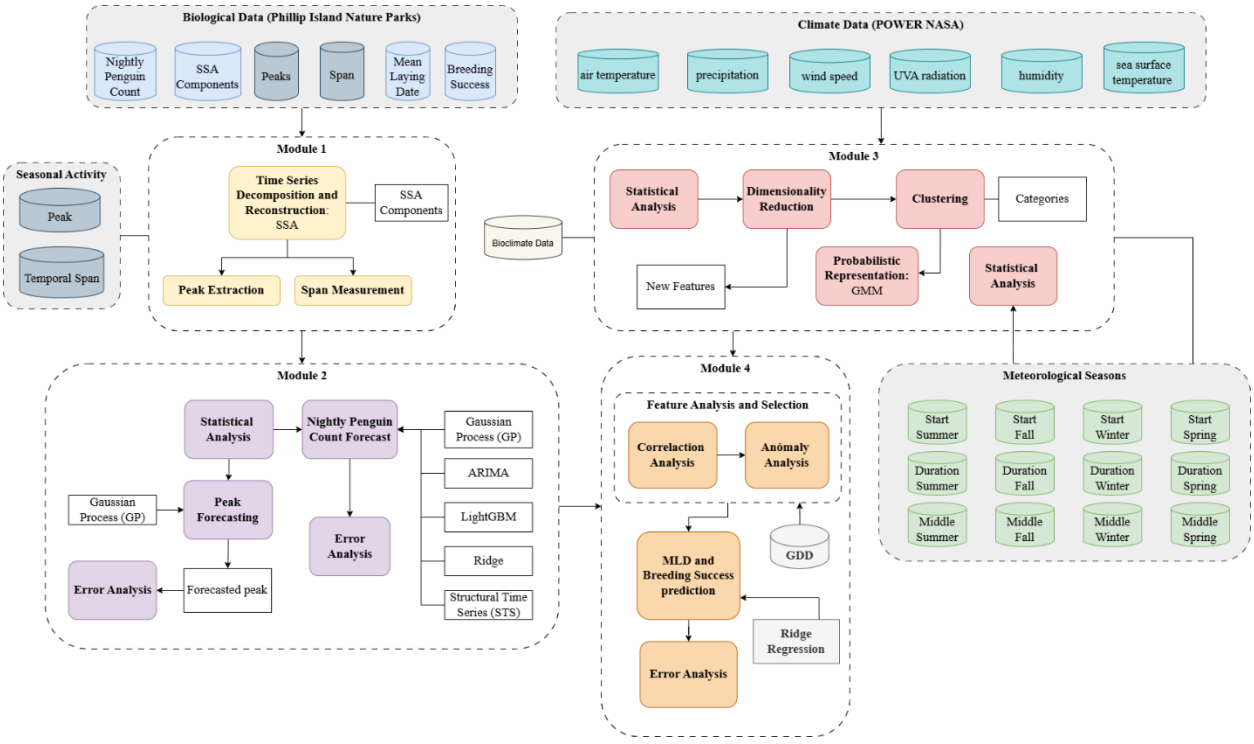

Fig. 14. CEFABC2 structure.

Module 3 presents the analysis of our climate data using unsupervised ML learning techniques to extract the duration, start, and middle DOY of the meteorological seasons. We used principal component analysis (PCA) for dimensionality reduction, and K-means and GMM to group days into meteorological seasons.

Module 4 integrates the climate and behavioural variables. In this module, anomalies (deltas) relative to the mean were calculated for each variable and compared using bar charts. Additionally, Pearson correlation analysis and ridge regression were used to develop forecasting models for MLD, breeding success, and peaks.

Overall, this framework links long-term ecological observations with computational analysis tools, providing an analytical framework to study the temporal dynamics of animal populations and their relationship with environmental factors.

4.4 Module 1: Time series data preprocessing

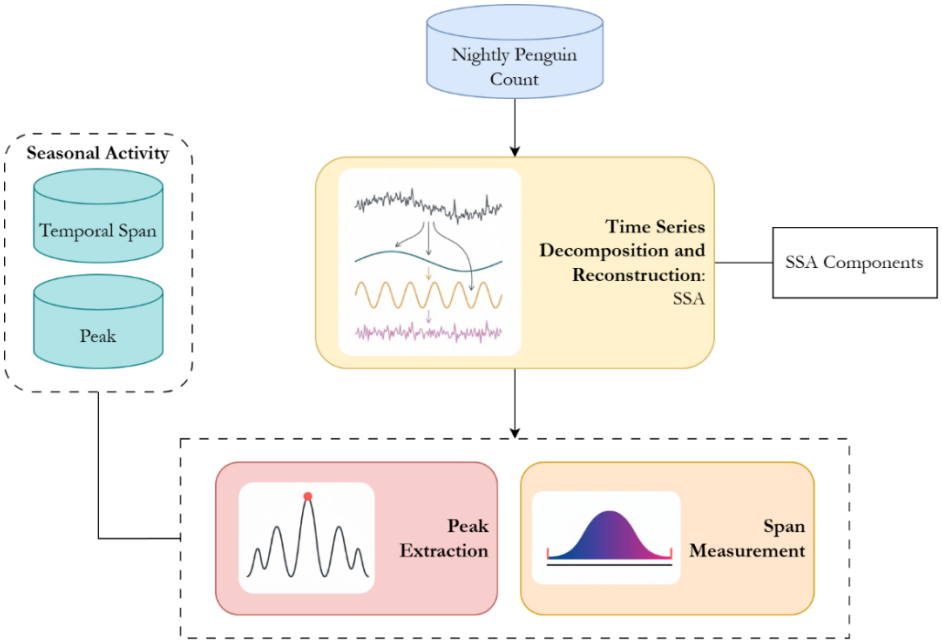

Fig. 15. Module 1 workflow.

Ecological time series reflect the interaction of multiple overlapping processes, making decomposition techniques necessary to isolate and study their underlying components (Ives et al., 2003). Biological systems also exhibit marked temporal and spatial variability, which makes ecological time series inherently non-stationary (Rollinson et al., 2021). This non-stationarity

results from changing environmental conditions, disturbances, and the interaction of multiple system drivers. In addition, ecological observations often contain substantial noise (Grove et al., 2020). Because these signals combine both structured and stochastic components, decomposition constitutes a necessary preprocessing step, preceding modeling.

At this step, the nightly penguin count series are preprocessed and two additional behavioural features are derived. **Fig. 15** presents a workflow of this module. The source code can be found in Appendix 1.

4.4.1 Decomposition and reconstruction of the time series

Singular Spectrum Analysis (SSA) was selected over classical decomposition techniques, such as Seasonal-Trend decomposition using Loess (STL) and Fourier-based approaches due to its non-parametric nature and its ability to handle non-stationary time series (Trendafilova, 2021). As will be demonstrated in the results chapter, the nightly penguin count has irregular patterns, shifting from year to year. This would be an obstacle for STL which requires predefined seasonal boundaries (Hyndman & Athanasopoulos, 2018), making SSA an adequate technique for the task.

SSA decomposes time series into interpretable components, including trend, oscillatory patterns, and noise. It can be used for complex time series exhibiting non-linear behavior (Trendafilova, 2021).

The components extracted by SSA are modes of variability within the time series. Each component is unique, carrying distinct information about the time series. In practice, the first components are used to reconstruct the time series, since they carry a large portion of the variance (Hassani, 2007).

The first component is the mode that captures the lowest frequency of the time series, hence it is associated with the trend of the time series, since it captures the long-term trajectory (Cui et al., 2021). The subsequent modes capture variability with higher frequencies and are associated with seasonal components. As we continue to inspect the higher modes, finer-scale variations are captured and represented by high-frequency modes. These tend to be considered noise in the time series.

The general process of SSA can be explained in four steps (Stratigakos et al., 2021): embedding, decomposition, grouping, and reconstruction (**Fig. 16**). The first step maps the time series into a multidimensional trajectory matrix, where the window length L is the main parameter which

defines the dimension of the lagged vectors $\boldsymbol{x}_t = [x_t, x_{t+1}, \ldots, x_{t+L-2}, x_{t+L-1}]$ that form the columns of the trajectory matrix **X**.

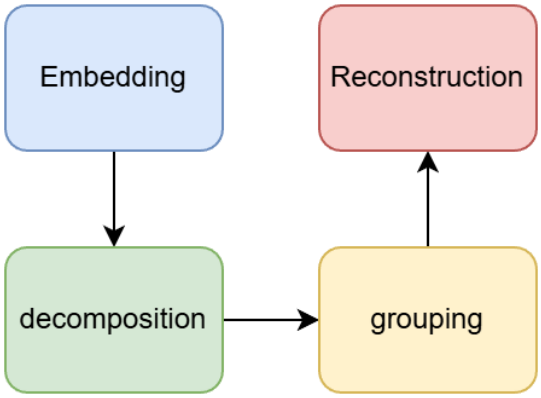

Fig. 16. SSA process.

To identify periodic patterns, such as breeding and non-breeding cycles, the window length L must be sufficiently large to capture the dominant seasonal components in the series (Golyandina & Korobeynikov, 2014). In practice, L is chosen to be greater than the periodic patterns to ensure separability between the trend and seasonal components. Conversely, when L is smaller than the periodic patterns, SSA only smooths the input time series (Golyandina & Korobeynikov, 2014).

Due to the strong annual seasonality in the nightly penguin count, we chose a window length of 400, which is above the annual cycle (365 steps), and the experiments with similar window lengths (between 390 and 410) revealed that the model is robust to moderate variations of L.

$$\boldsymbol{X} = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_{N-L-1} & x_{N-L} & x_{N-L+1} \\ x_2 & x_3 & x_4 & \cdots & x_{N-L} & x_{N-L+1} & x_{N-L+2} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{L-1} & x_L & x_{L+1} & \cdots & x_{N-3} & x_{N-2} & x_{N-1} \\ x_L & x_{L+1} & x_{L+2} & \cdots & x_{N-2} & x_{N-1} & x_N \end{bmatrix}$$

After matrix **X** is built, singular value decomposition (SVD) is applied. This technique decomposes the matrix into a sum of interpretable components. This decomposition is performed using the covariance matrix $S = XX^T$. From S, the eigenvectors U_i and eigenvalues λ_i are extracted to build the factor vectors $V_i = X^T U_i / \sqrt{\lambda_i}$ and build each matrix component X_i (i = 1,…, L) (**eq. 1**). The orthogonality of the eigenvalues λ_i, here known as *singular values*, determines the independence

of each component. Hence, each component captures a different characteristic of the original time series and λ_i is the feature that informs about the energy or variance that each component captures. Singular values are generally used for ranking, keeping the components of highest value for the reconstruction of the time series.

$$X = \sum_{i=1}^{d} \sqrt{\lambda_i} U_i V_t^T = \sum_{i=1}^{d} X_i \quad \textbf{eq. 1}$$

In the grouping step, a subset of components is chosen to reconstruct the time series based on the concept of separability, where distinct groups of components represent different properties of the time series, such as trend, seasonality, and noise. The selection of components is guided by the analysis of the singular values λ_i, which characterize the information captured by each component.

A spectral analysis of the singular values is performed to select the subset of components. In this analysis, large singular values indicate dominant signals, i.e., trends and seasonalities. Additionally, plateaus of similar values represent harmonic characteristics (Hassani, 2007). Slowly decreasing values often characterize noise. We also performed a spectral analysis of the eigenvectors, since they reflect the time series components they represent, describing the variability captured by each of them.

In this work, we plotted the eigenvalues of each component, identifying drops and plateaus that indicate dominant features. We kept the top 5 components for the reconstruction, since after these components there is a significant drop and four of these components lie on the same plateau. Additionally, these are the 5 components with the largest explained variance, i.e., information from the time series. Nevertheless, to assess the performance of forecasting models in Module 2, we reconstructed the nightly penguin count data with the top 10, 20, and 30 components. This allows us to improve performance as more components are included.

Finally, the diagonal averaging process, or Hankelization (Hyndman & Athanasopoulos, 2018), is performed to transform the group into a time series. Consequently, the reconstructed time series is built from the components established in the group. SSA provides a method to extract trends and seasonal components that are important in the analysis of time series. Additionally, SSA is also used as a smoothing technique, eliminating residual fluctuations (or noise).

4.4.2 Extraction of seasonal patterns

4.4.2.1 Peak extraction

Visual inspection (**Fig. 22**) showed that the reconstruction with five components preserved the major annual and sub-annual oscillations while suppressing short-term irregular fluctuations. Consequently, peak detection was performed on the SSA-reconstructed time series using the first five components, rather than on the raw nightly penguin count. Local maxima were then identified from the reconstructed count time series. No additional minimum prominence or minimum distance criteria were imposed, because the reconstruction with five SSA components already removed high-frequency fluctuations and retained only the dominant seasonal structures of the series. Therefore, the detected peaks correspond to the main seasonal activity patterns rather than to short-term noise in the original observations.

4.4.2.2 Extraction of span: Gaussian mixture models (GMM)

To estimate the span of the seasonal activity peaks, GMMs were fitted for each biological year from July 1 to June 30. Based on ecological knowledge of the species and inspection of the reconstructed time series, two Gaussian components ($K = 2$) were used to represent the dominant annual activity peaks observed: one associated with the breeding season and another corresponding to the moulting period.

The model was initialized using the K-means++ strategy (Du et al., 2025) to improve the stability of the optimization by placing initial component means in regions of high data density. The parameters of the mixture were then estimated using the Expectation-Maximization (EM) algorithm (McLachlan et al., 2019).

Given a set of observations $x_1, \ldots, x_N$, the likelihood of the GMM is defined as:

$$p(x_n) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(x_n \mid \mu_k, \sigma_k^2),$$

where π_k, μ_k, and σ_k^2 correspond to the mixing coefficient, mean, and variance of component k, respectively. The EM algorithm iteratively maximizes this likelihood through:

E-step:

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \sigma_k^2)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n \mid \mu_j, \sigma_j^2)}$$

M-step:

$$\mu_k = \frac{\sum_{n=1}^{N} \gamma_{nk}\, x_n}{\sum_{n=1}^{N} \gamma_{nk}}, \sigma_k^2 = \frac{\sum_{n=1}^{N} \gamma_{nk}\, (x_n - \mu_k)^2}{\sum_{n=1}^{N} \gamma_{nk}}, \pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk}$$

For each Gaussian component k, the mean $\mu_k$represents the timing of the activity peak, while the standard deviation $\sigma_k$characterizes its temporal dispersion. In this work, the span of each seasonal peak is defined as the interval $[\mu_k - 2\sigma_k, \mu_k + 2\sigma_k]$, which captures the core temporal window of the activity peak.

4.5 Module 2: Behavioural pattern analysis

The objective of this module is to model and analyze the temporal dynamics of the colony using the variables derived from the previous module. In particular, this module aims to establish relationships between the biological variables MLD and breeding success and the derived variables activity peaks and spans, and also to evaluate the modeling capabilities for the nightly penguin count time series.

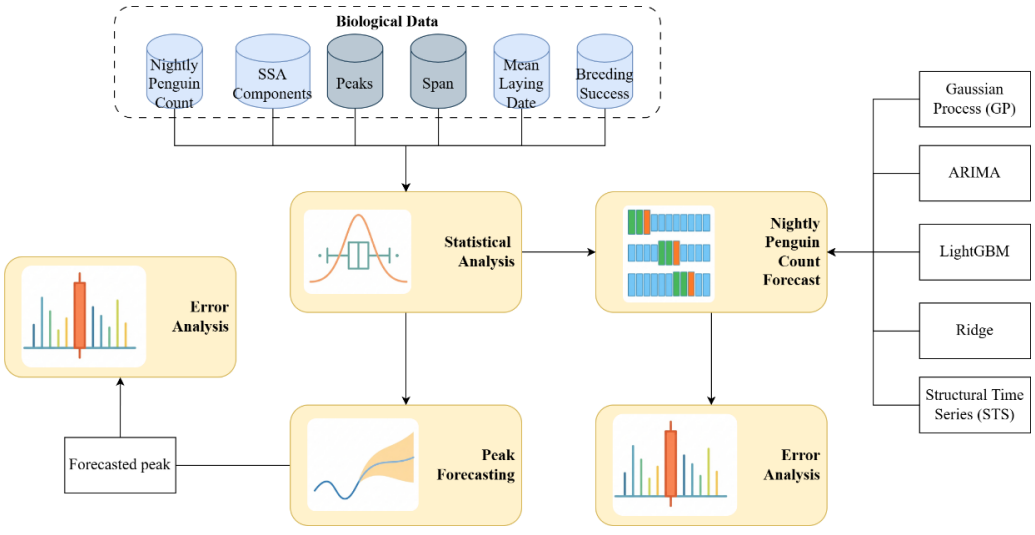

Fig. 17. Module 2 workflow.

In general, this module addresses the need to understand the behaviour of the colony from two perspectives: (1) statistical analysis of the biological variables, and (2) predictive modeling of the time series.

The general structure of this module is presented in **Fig. 17**, where the analytical and predictive processing of the variables is illustrated.

4.5.1 Statistical analysis of the behavioural variables

In order to understand the relationships and characteristics among the peaks of nocturnal activity, the span obtained from the peaks, MLD, and breeding success, this section aims to quantify the trends, variability, and relationships among the behavioural variables.

First, we analyzed the trends of these variables: peak timing (DOY), peak span, MLD, and breeding success. The trends were evaluated using the Mann-Kendall test, a non-parametric test that detects monotonic trends, which do not need to be linear but must be persistent (Wang et al., 2020). The significance level used was 0.05, and the slope was calculated for each variable.

Subsequently, we analyzed the distributions of these variables by extracting the means and standard deviations that characterize them. At this stage, we constructed a new variable, here called mismatch, which is the temporal difference between the peak that occurs during the breeding season and MLD (**Eq. 2**). This variable captures the synchrony between the beginning of the reproductive period and the nocturnal activity of the penguins on the beach.

$$Mismatch = Peak - MLD \qquad \textbf{Eq. 2}$$

Finally, we conducted a correlation analysis among MLD, peaks, spans, mismatch, and breeding success. Pearson correlation was used to measure the relationship, with the significance level set at 0.05.

4.5.2 Forecasting the peak

This section aims to forecast the peaks of nocturnal activity of little penguins on Phillip Island. This task is formulated as a season-onset prediction problem, whose objective is to estimate the timing of maximum activity using only partial observations from the current year and previous years. This forecast provides information about the temporal dependencies of the nightly penguin count series; that is, it evaluates the possibility of forecasting high-activity events using only the

penguin count time series. Additionally, this model allows the construction of a baseline to detect future deviations in the activity peak and to provide a reference point in the literature for peak forecasting, which has not been observed in the existing publications on little penguins. Finally, this forecast also provides advance information about the date of high activity, which is potentially useful for the operational management of tourist events related to penguin viewing.

The data used for this forecast are the time series reconstructed using the first 5 SSA components. This ensures consistency between the process carried out in Module 1 for peak extraction and the forecasting task. The process for forecasting the peak consists of 2 stages: time series forecasting and identification of the peak DOY. First, we predict the nightly penguin count for the following 365 days (1 year), and then we identify the DOY (day of the year) of the next peak in the time series by detecting the local maximum.

For time series forecasting, we implemented Gaussian processes (GP). This model places a probability distribution over a set of functions, which encodes assumptions about the smoothness and temporal correlation of the forecasted time series (Rasmussen & Nickisch, 2010). All these assumptions are contained in the model kernels, which are its main parameter. When properly defined, this model is able to assume that the observations are noisy and, consequently, it does not have to fit the time series perfectly. Moreover, the noise around each predicted value is assumed to be normally distributed.

GP was chosen because it is a non-parametric model, so it does not assume a predefined functional form for forecasting (Zamanzadeh Talkhouncheh et al., 2024). This allows the complexity of the function to be determined by the data to which it was fitted. Our model was fitted using 3-year windows of time series data to forecast the subsequent year. This model extrapolates the behavior observed in the training data to time windows beyond the training period.

In this section, the necessary knowledge for model implementation and interpretation is described following Rasmussen and Nickisch (2010) where more detailed explanation can also be found. This model searches for the distribution over basis functions (kernels) that best represents a target function, which, in our case, is the time series. The forecasted values of this model are obtained as the mean of a distribution of functions (eq. 3). The nature of these functions is defined through the kernel (eq. 4).

$$m(x) = E[f(x)] \quad \textbf{eq. 3}$$

$$\kappa(x, x') = E[(f(x) - m(x))(f(x') - m(x'))^T]. \quad \textbf{eq. 4}$$

Here $f(x)$is the latent function, $m(x)$ and $\kappa(x, x')$ are its mean and covariance functions, respectively, and $\mathbb{E}[\cdot]$ denotes expectation.

Defining the kernel is a crucial step when building the model, since it defines the shape of the functions. The kernel encodes the similarity between two points, establishing whether the function varies quickly or slowly. Kernels in a Gaussian Process can be broadly categorized into two groups (**Table 4**).

Table 4. Types of kernels in a Gaussian process

Kernel Type	**Characteristics**	**Examples**
Stationary (Shift-Invariant)	The kernel depends only on the separation between inputs, independent of their absolute position.	Squared Exponential (RBF), ARD, Matérn, Periodic, Rational Quadratic, Spectral Mixture
Nonstationary	Similarity measure depends on the location of the inputs, not just their difference	Polynomial kernel, Gibbs kernel (input-dependent length scale/variance), NN-GP kernel derived from infinitely wide neural networks

The hyperparameters of these kernels control the behavior of the model or the noise expected in the data. If the time series is assumed to be noise-free, then the Gaussian process would interpolate the time series by passing through the exact data points of the training set. This makes the model fit the training data points perfectly, which could lead to overfitting in some scenarios. In contrast, if the observed values are considered to be normally distributed around a true underlying value, the model adds an error term that is also assumed to be normal. This gives the model the ability to estimate a function that passes near the observed values, but not necessarily through them.

There are multiple kernels that can be used to construct the model, and the representation of complex functions sometimes requires the combination of multiple kernels. Here, we describe

those relevant to our work. For our time series, we represent the smooth trend through the squared exponential (SE) kernel, a.k.a. RBF:

$$\kappa_{SE}(\boldsymbol{x}, \boldsymbol{x}') = exp\left(-\frac{||\boldsymbol{x}-\boldsymbol{x}'||^2}{2\sigma^2}\right), \quad \textbf{eq. 5}$$

where $x, x' \in \mathbb{R}^d$ are input vectors (e.g., time points in our time series), $\|\cdot\|$ denotes the Euclidean norm, and σ^2 is the length-scale parameter that controls the smoothness of the function. Small σ values lead to highly variable functions, while large values lead to smoother, slowly varying functions.

Another kernel used in the model is the periodic kernel, often known as the Exp-Sine-Squared kernel (**eq. 5**). The product of this kernel with the SE kernel allows the modeling of locally periodic functions. This is important because, in real-world scenarios, periodic patterns are generally not perfectly periodic.

$$\kappa_{Per}(x, x') = exp\left(-\frac{2sin^2(\pi|x-x'|/p)}{\ell^2}\right), \quad \textbf{eq. 6}$$

where p is the period of the function, and ℓ is the length-scale parameter controlling how quickly the correlation decays away from exact periodicity. Variations in amplitude, timing, or width can be represented by the product of the two functions $\kappa_{SE} * \kappa_{Per}$.

When building a GP model, it is important to note that the kernel parameters (σ, ℓ, p) are optimized by the model and are not fixed. Nevertheless, the researcher is free to establish the starting values for such parameters in the optimization process.

For the prediction of the peaks that happen during the breeding season, the model was trained with the previous three years of the nightly count series until day 182 of the year, which corresponds to July 1st or June 30th in a normal and a leap year, respectively. After that, the model forecasts the next 365 days of the nightly count. This is used to identify the timing of the next peak based on the forecasted time series. For the peaks that happen outside the breeding season, the data were observed until December 31st of the previous year. The parameters used for the model were obtained using grid search, and the ones used for the forecast are shown in Table 5.

After peak detection, the predictive performance was primarily assessed using the mean absolute error (MAE), expressed in days, as it provides a direct and interpretable measure of temporal

deviation. Additionally, the root mean squared error (RMSE) was used to penalize larger prediction errors, and the median absolute error (MedAE) was included to assess robustness to outliers. For completeness, additional metrics, including the R^2 score, mean absolute percentage error (MAPE), mean squared error (MSE), and mean squared log error (MSLE), were also computed. The source code can be found in Appendix 2.

Table 5. GP parameters for peak prediction

Parameter	Value
Epochs	100
Likelihood	Gaussian Likelihood
Variance of the Gaussian distribution	0.2
Learning rate	0.1
Optimizer	Adam
κ_{SE} (Trend)	$\sigma = 0.09$
κ_{SE} (Seasonality)	$\sigma = 3$
κ_{Per} (Seasonality)	$p = 0.6$

4.5.3 Nightly penguin count series forecasting

This section aims to evaluate the predictive performance of different model families across multiple temporal granularities and representations of the time series. In particular, it seeks to analyze how data preprocessing and model complexity influence forecasting ability.

The models were trained using four representations of the penguin count series: the original series and its reconstructions using the first 5, 10, 20, and 30 SSA components. Predictions obtained from the SSA-reconstructed series were compared with the corresponding reconstructed test series, avoiding inconsistencies between representation spaces.

All models were trained and evaluated using the same data splits and forecasting horizons. Although the input structures differed according to the requirements of each model, consistent data representations and evaluation protocols were maintained to ensure a fair comparison across model families.

Additionally, three temporal granularities were used for the forecasting evaluation: daily, weekly, and monthly. This was done to assess performance in the short, medium, and long term. For each temporal granularities, the multi-horizon window differed: nightly penguin count used a 14-day (2-week) horizon, the weekly average used a 12-week (~4-month) horizon, and the monthly average used a 6-month horizon.

Prediction error was evaluated independently for each horizon using the RMSE and R^2 metrics. RMSE quantifies the magnitude of the error, while R^2 provides a relative measure of the variance explained by the model. Model performance was analyzed using quantitative metrics and visual inspection of the prediction trajectories. Comparisons with baseline models are presented in later sections.

4.5.4 Backtesting

To ensure a realistic setting for forecast evaluation, a walk-forward backtesting strategy was implemented (Kojevnikov, 2021). **Fig. 18** descriptively shows the backtesting strategy.

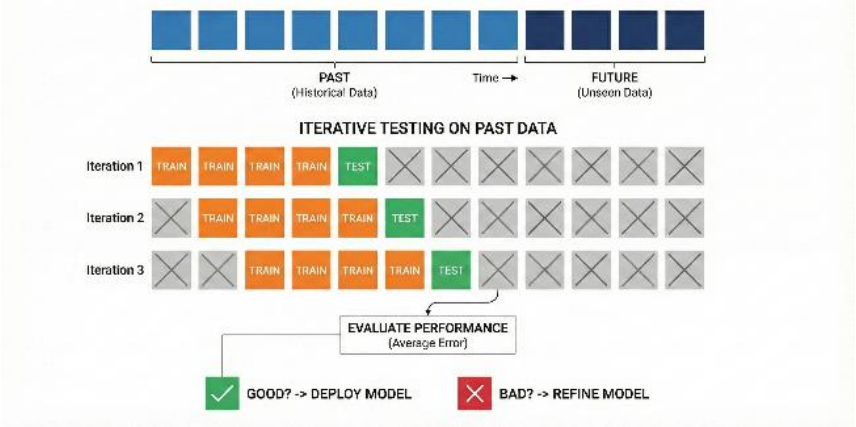

Fig. 18. Backtesting with fixed training window.

This strategy allows models to be evaluated in an environment that simulates real operating conditions, where data become available as they are recorded over time (Wang et al., 2024). Iteratively, the models are trained using a window of historical data and then used to forecast over a defined future horizon.

A direct multi-step forecasting strategy was used, in which an independent model is trained for each step within the prediction horizon. This strategy allows horizon-specific dynamics to be

captured, although it implies a higher computational cost compared with the recursive approach (Taieb et al., 2012).

After each iteration, the training window moves forward in time, incorporating new data. In each iteration, the model is trained only with past information, while the prediction horizon corresponds to unobserved future periods, thus avoiding the use of future information (data leakage).

Although the time window can be expanding, meaning that it retains all historical data while incorporating new data, we used a fixed window. As an advantage, the fixed window avoids the accumulation of historical data that slows model training, while also learning changes in the patterns of the time series (Zhang et al., 2024).

To evaluate model performance, the predictions from all iterations were aggregated and evaluated using RMSE and R2 to estimate the average performance of each model over time.

4.5.5 Forecasting models

For forecasting the nightly penguin count, models from different model families were used to compare statistical, ML, and DL approaches. Specifically, we implemented ARIMA (Hyndman & Athanasopoulos, 2018) and structural time series (STS; Zhang et al., 2024) as statistical models, Ridge (Jisha, 2024) and LightGBM (Ke et al., 2017) as ML models, LSTM (Waqas & Humphries, 2024) as a DL model, and GP (Rasmussen & Nickisch, 2010) as a probabilistic model.

Hyperparameters were tuned using the first 9 years as training data and the following year as validation. After tuning, a fixed training window of 10 years was used during backtesting. This configuration was used for the daily and weekly resolutions. However, the training window used for the monthly resolution was 15 years due to the sample size constraints. The source code can be found in Appendix 3.

4.5.5.1 ARIMA

The Autoregressive Integrated Moving Average (ARIMA) is a statistical model widely used to model linear dependencies over time, and it is usually considered a standard baseline for other models (Bartholomew, 1971). Furthermore, it is known for its high performance in short-term forecasting, and interpretability of relationships between the forecast and past data, which is an advantage over black-box models (Fatima & Rahimi, 2024).

ARIMA combines three components: autoregressive (AR), which captures the dependence of the target on historical values; integrated (I), which transforms a non-stationary series into a stationary one through differencing; and moving average (MA), which models the target by considering errors from past estimates.

Hyperparameter selection was performed automatically by minimizing the Akaike information criterion (AIC), following a standard model-fitting approach (Hyndman & Khandakar, 2008). The differencing order of the series was determined through stationarity tests, while the autoregressive and moving average orders were selected based on predictive performance.

The SARIMA variant was developed to capture seasonality, however, it becomes computationally inefficient when dealing with long seasonal cycles (Hyndman & Athanasopoulos, 2018). In practice, an alternative to this approach is to integrate Fourier terms (**eq. 7**) as input features (Rausch et al., 2022), allowing periodic structures to be modeled. In this work, we implemented this approach not only for ARIMA, but also for Ridge, LSTM, and LightGBM.

$$\text{Fourier}_k(t) = \{\sin(2\pi kt/m),\ \cos(2\pi kt/m)\},\ k = 1, \dots, K. \qquad \textbf{eq. 7}$$

Here, m represents the seasonal period (365 days) and K the number of included harmonics. This formulation allows seasonality to be captured flexibly and efficiently from a computational perspective.

4.5.5.2 Ridge

Ridge is a regularized linear regression model that incorporates a penalty on the magnitude of the coefficients to reduce model overfitting and improve stability in the presence of correlated variables (Jisha, 2024). This model allows the evaluation of the performance of a linear approach against more complex models.

The model was trained using historical values of the time series as input variables, together with Fourier terms to represent annual seasonality. The L2 regularization used by the model allows control of complexity by penalizing large coefficients, which is especially relevant when working with multiple highly correlated variables.

The optimization problem is defined as:

$$\min_w \ \|y - Xw\|^2 + \lambda\|w\|^2, \qquad \textbf{eq. 8}$$

where y represents the target variable, X the input variables, w the model coefficients, and λ is the regularization parameter, which was tuned during the validation phase.

4.5.5.3 Long short-term memory (LSTM)

LSTM is a neural network architecture designed to model complex long-term dependencies in sequential data. This model belongs to the family of recurrent neural networks (RNN), which are specifically designed for sequential data (Waqas & Humphries, 2024). Unlike linear models, LSTM can capture complex dynamic patterns and nonlinear relationships in the time series.

LSTM is characterized by the system of gates used to regulate the flow of information in each cell (**Fig. 19**). Three gates compose each cell, including the input gate, which controls the incorporation of new information; the forget gate, which regulates the information that should be removed; and the output gate, which defines what information is propagated to the next stage.

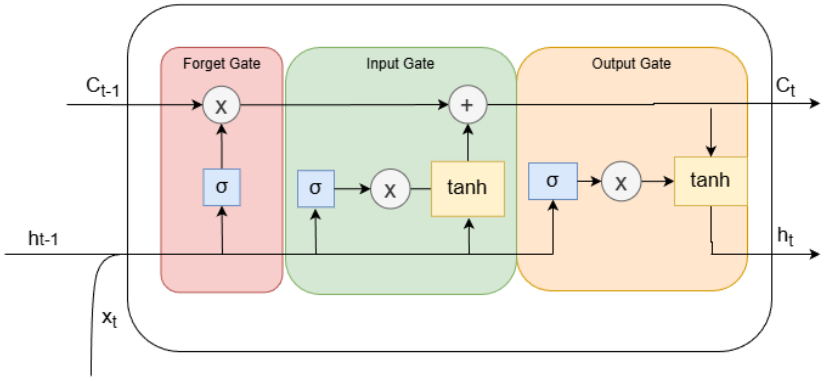

Fig. 19. LSTM architecture.

In this study, LSTM is used to model nonlinear patterns in the nightly penguin count time series that cannot be captured by statistical or linear models. Following the training strategy described previously, this model receives Fourier components and historical data as input.

4.5.5.4 LightGBM

This model builds decision trees sequentially, each with the objective of improving the prediction of the previous tree (Ke et al., 2017).

This model is used due to its ability to represent interactions among the input variables and nonlinear structures in the time series. In general, LightGBM is a tree-based model that implements several optimizations to improve computational cost and model performance (Chen & Guestrin,

2016). These include the use of histograms to discretize continuous variables; leaf-wise growth, which prioritizes the partitioning of nodes that reduce the error the most; and the implementation of Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), which reduce the number of variables and data points processed while minimally affecting model accuracy.

4.5.5.5 Structural time series (STS)

The STS models allow a time series to be represented as a sum of latent components such as trends, seasonality, and short-term variations (Harvey & Harvey, 1990). In this way, the time series is modeled as:

$$f(t) = f_1(t) + f_2(t) + \cdots + f_C(t) + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, \sigma^2)\,, \qquad \textbf{eq. 9}$$

where each $f_{\mathrm{i}}(t)$ corresponds to a structural component of the system.

Based on recommendations by Ravichandran and Prajneshu (2005), this work implemented a model that combines a local level component, an autoregressive component, and a smooth seasonal component, with the aim of representing short-, medium-, and long-term patterns.

The local component models the evolution of the time series as a random walk process:

$$\mu_t = \mu_{t-1} + \epsilon_{\mu,t}, \quad \epsilon_{\mu,t} \sim \mathcal{N}(0, \sigma^2). \qquad \textbf{eq. 10}$$

Meanwhile, the autoregressive component generates a short-term temporal dependence:

$$\mu'_t = \rho \mu'_{t-1} + \epsilon_{\mu',t},\ where\ |\rho| < 1\ and\ \epsilon_{\mu',t} \sim \mathcal{N}(0, \sigma^2). \qquad \textbf{eq. 11}$$

Seasonality is modeled as a smooth component based on trigonometric functions, where the frequencies are determined by the seasonal period *p*:

$$freqs[j] = 2 * \pi * \frac{fm[j]}{p}. \qquad \textbf{eq. 12}$$

The complete model can be represented in state-space form, where the latent states evolve dynamically:

$$\boldsymbol{z}_t = \boldsymbol{F} * \boldsymbol{z}_{t-1} + \boldsymbol{\epsilon}_t. \qquad \textbf{eq. 13}$$

The observations are generated through these states:

$$p(y_t|\boldsymbol{z_t}) = \mathcal{N}(y_t|\boldsymbol{H}_t\boldsymbol{z}_{t-1}, \boldsymbol{R}_t). \quad \textbf{eq. 14}$$

The estimation of the parameters and latent states was carried out through variational inference, which seeks to find an approximate distribution $q_\psi(\boldsymbol{z})$ of the latent states by optimizing the Evidence Lower Bound (ELBO) (Lian et al., 2021):

$$\mathcal{L}(\theta, \psi \mid x) = E_{q_\psi(z)}\left[log p_\theta(x,z) - log q_\psi(z)\right], \quad \textbf{eq. 15}$$

where ψ represents the variational parameters (e.g., means and variances of the latent states).

4.6 Module 3: Climate time series analysis

This module studies the dynamics of the meteorological seasons in the region where the little penguin colony is located. As discussed in Chapter 2, previous studies have shown that key behavioural variables, such as breeding success, MLD, and foraging activity, are associated with climate. Therefore, characterizing the regional climate provides relevant context for interpreting the biological variables analyzed in Module 4.

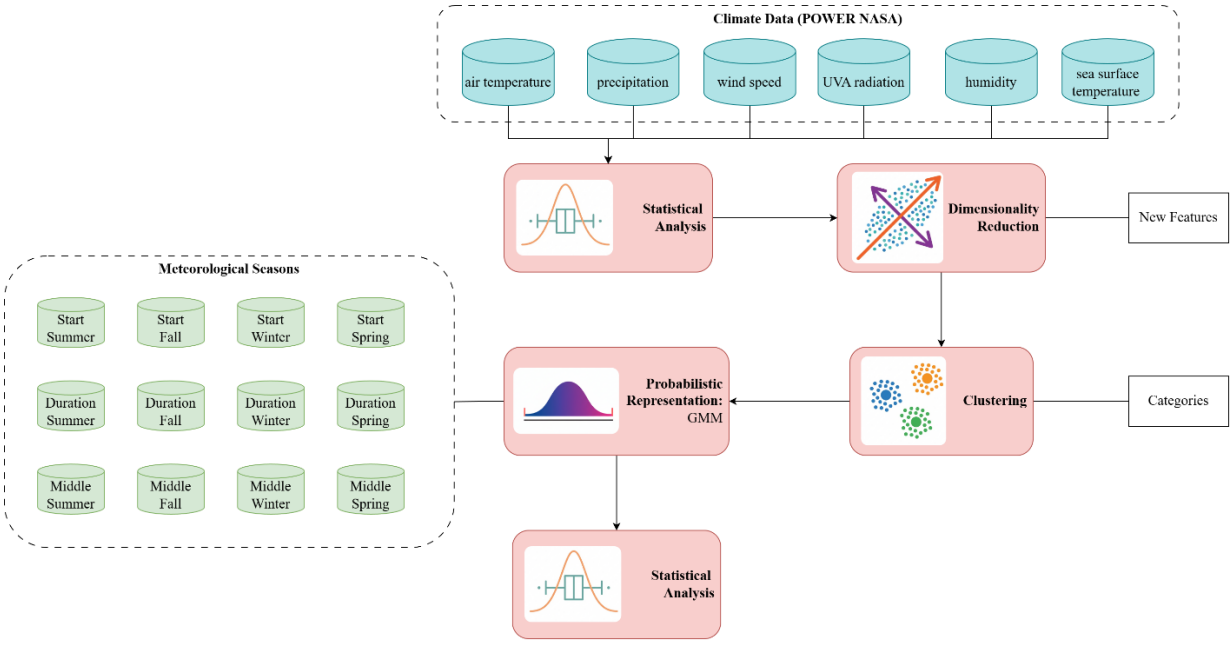

Fig. 20. Module 3 workflow.

This module begins with a statistical description of the climatic variables obtained from NASA POWER, including their mean, range, and variability. These variables are then used to extract meteorological seasons, allowing us to quantify their onset, duration, and midpoint. To the best of

our knowledge, this represents the first numerical portrayal of seasonal dynamics in the studies on little penguins.

Finally, we analyze the relationship between the derived seasonal descriptors and bioclimatic variables, which provide a biologically meaningful representation of environmental conditions. **Fig. 20** shows a summary of the workflow explained in this module.

4.6.1 Characterization of the meteorological variables

The objective of this section is to characterize the statistical and temporal properties of the meteorological variables before their use in later modeling stages. The analyzed variables include air temperature (mean, maximum, and minimum), precipitation, wind speed, relative humidity, and sea surface temperature (SST).

Initially, we analyzed the temporal dynamics of mean, maximum, and minimum air temperature because these are climatic factors related to the penguins' metabolic and thermoregulatory activity (Stahel & Nicol, 1988).

For the analysis, we described the ranges and trends of these daily variables using the Mann-Kendall test, and we complemented the study by describing intra-annual variability using monthly aggregates.

To analyze intra-annual dynamics, the monthly distributions were evaluated using boxplots, which allowed the representation of variability and seasonal patterns throughout the year.

Subsequently, we described the monthly dynamics of precipitation using boxplots, since it is a variable that has been associated with the breeding success of the colony (Chambers et al., 2011). Along with this variable, wind speed, which affects foraging performance and breeding success (Saraux et al., 2016), and humidity were also studied.

Finally, sea surface temperature (SST) was analyzed in terms of its temporal structure and monthly distribution, since it has been associated with the MLD and chick development (Cullen et al., 2009).

4.6.1.1 Bioclimatic variables

To study biologically meaningful climatic characteristics, 19 bioclimatic variables have been used (**Table 6**). These variables are widely used in ecological and species-related studies, as they

provide a more interpretable representation of environmental conditions compared to raw meteorological variables (Amiri et al., 2020). The variables were computed from daily observations of temperature and precipitation, following standard formulas specified in the BIOCLIM framework (https://www.worldclim.org/) and described by boxplots and their mean values in the region from 1981 to 2024.

Table 6. Bioclimatic variables

Bioclimatic variable	Description
BIO1	Annual Mean Temperature
BIO2	Mean Diurnal Range (Mean of monthly (max temp – min temp))
BIO3	Isothermality (BIO2/BIO7) (×100)
BIO4	Temperature Seasonality (standard deviation ×100)
BIO5	Max Temperature of Warmest Month
BIO6	Min Temperature of Coldest Month
BIO7	Temperature Annual Range (BIO5-BIO6)
BIO8	Mean Temperature of Wettest Quarter
BIO9	Mean Temperature of Driest Quarter
BIO10	Mean Temperature of Warmest Quarter
BIO11	Mean Temperature of Coldest Quarter
BIO12	Annual Precipitation
BIO13	Precipitation of Wettest Month
BIO14	Precipitation of Driest Month
BIO15	Precipitation Seasonality (Coefficient of Variation)
BIO16	Precipitation of Wettest Quarter
BIO17	Precipitation of Driest Quarter
BIO18	Precipitation of Warmest Quarter
BIO19	Precipitation of Coldest Quarter

4.6.2 Characterization of the seasons

Characterizing the climatic seasons allows us to move from studying individual meteorological variables to an integrated assessment of climate dynamics. Although these variables provide

relevant information for the species, the interplay between them makes it possible to describe the meteorological seasons to which the colony is exposed.

In the context of this thesis, the temporal characterization of the seasons allows to understand the climatic variability, including changes in the duration, onset, or shift of the seasons on the basis of regionally-collected data.

Specifically, precipitation, temperature (maximum, minimum, and average), and air humidity are used as input variables processed by three unsupervised ML models (namely, principal component analysis, PCA; K-means; and Gaussian Mixture Models, GMMs) to classify each day of the year into a particular "phenological" climatic season (Suleman & Khaiter, 2025). As a result, the onset, duration, and midpoint of each season are determined. The source code can be found in Appendix 4.

The pipeline proposed in the thesis extends a framework by Suleman and Khaiter (2025), and it is more suitable for climatic conditions of Phillip Island. In particular, a feature matrix was constructed having one column for each climatic variable, and each row corresponding to a daily climate record across all years between 1981 and 2024. Each feature was normalized to eliminate the sensitivity of the ML models to different scales. So obtained feature matrix served as input to PCA described in the following section.

4.6.2.1 Dimensionality reduction and climate patterns extraction

To eliminate correlations between climatic variables and reduce dimensionality, PCA was applied, a variance-based model for data transformation (Greenacre et al., 2022). In this model, each calendar day is treated as a datapoint o_i defined by its corresponding climate variables, $o_i = (y_{i1}, \dots, y_{in})$, where y_{in} is the value of variable n on date i. The transformation performed by PCA can be interpreted as a set of new axes in a Euclidean space that capture most of the variance of the data points and change their basis.

Considering matrix $\boldsymbol{C}$ as our input climate dataset, the optimization problem of PCA is to find a vector $\mathbf{e}_j$ that maximizes the variance within the new basis (feature) w_j, as expressed in **eq. 16**, where $\mathbf{S}$ is the covariance matrix of matrix $\boldsymbol{C}$.

$$\boldsymbol{C} = \begin{bmatrix} y_{1,1} & \cdots & y_{1,m} \\ \cdots & \cdots & \cdots \\ y_{n,1} & \cdots & y_{n,m} \end{bmatrix}$$

$$\text{var}(w_j) = \mathbf{e}_j^T \mathbf{S} \mathbf{e}_j, j = 1, \ldots, m \text{ and } m < n. \quad \textbf{eq. 16}$$

These new features correspond to the eigenvectors of the covariance matrix **S**. Each subsequent feature, w_{j+1}, is determined through the same process described in **eq. 17**, after removing the variance contribution of the previous feature w_j from the initial covariance matrix **S** (Hsieh, 2009):

$$\mathbf{S} - \lambda_j \mathbf{e}_j \mathbf{e}_j^T, \quad \textbf{eq. 17}$$

where λ_j are the eigenvalues corresponding to each eigenvector $\mathbf{e}_j$. Similar to the SSA, each λ_j represents the proportion of variance captured by that feature from the original dataset. The first principal components capture the highest proportion of variance in the data and represent the dominant directions of variability in the dataset. In contrast, the last features capture the lowest proportion of variance and are often treated as noise. This proportion of variance is calculated as:

$$\%\text{var} = {\lambda_j}\Big/{\sum_j \lambda_j}, \; j = 1, \ldots, m. \quad \textbf{eq. 18}$$

In practice, the number of retained principal components was selected based on the plot of the cumulative explained variance (Greenacre et al., 2022). With this new representation, datapoints are described as $o_i = (w_{i1}, \ldots, w_{im})$. The new variables were then passed to the K-means clustering algorithm, whose application is explained in the next section.

4.6.2.2 Identification of meteorological seasons

To identify groups of days with similar climatic patterns, the data points transformed with PCA were clustered using a time series adaptation of K-means. K-means is an unsupervised ML algorithm used to partition a dataset into Kclusters. In its standard formulation, the algorithm assigns data points to the centroid (center point of the cluster) by minimizing the distance between each point and its corresponding centroid (eq. 19; Ahmed et al., 2020).

However, as suggested by Suleman and Khaiter (2025), when dealing with temporal patterns, alternative similarity measures can provide a more meaningful comparison between observations. In that study, Dynamic Time Warping (DTW) was used as the distance metric, as it allows the

alignment of temporal sequences that may be shifted in time, making it suitable for capturing similarities in climatic patterns (He et al., 2024).

Under this formulation, the standard Euclidean centroid is replaced by a representative sequence computed through DTW Barycenter Averaging (DBA), which minimizes the average DTW distance to all elements within a cluster. The clustering objective is therefore defined as:

$$J = \sum_{n=1}^{N} \min_{k \in \{1,\dots,K\}} d_{\mathrm{DTW}}(\mathbf{o}_n, \boldsymbol{\mu}_k), \qquad \textbf{eq. 19}$$

where d_{DTW} denotes the Dynamic Time Warping distance, $\mathbf{o}_n$ is the data point n, and $\boldsymbol{\mu}_k$ represents the DTW barycenter of cluster k.

The number of clusters was defined using the elbow criterion (Umargono et al., 2020), based on the reduction of the clustering objective function as K increases. In this study, the resulting clusters are interpreted as meteorological seasons, based on the distribution of the day of year (DOY) associated with each cluster.

4.6.2.3 Separation of similar seasons: GMMs

As mentioned previously, the use of K-means for seasonal clustering based on climatic observations detects three clusters, placing the days belonging to the transitional seasons (fall and spring) in the same cluster. To separate these clusters based on their temporal location, we implemented GMMs.

Additionally, GMMs allow us to characterize each meteorological season by estimating the duration (span) through the standard deviation of the Gaussian models and the mean day of the season using the mean of the models.

Unlike the input of K-means, the input to the GMM is the temporal variable of the observations, that is, the DOY of the datapoints. This model offers a probabilistic perspective of the seasons, not establishing a specific day as the seasonal change, but rather as a probability that transitions from one season to another as time progresses.

This is the last stage of the pipeline, with the estimated season duration and mean DOY as output. Additionally, we calculated the start of the season by subtracting the 2 times the standard deviation of the Gaussian models from the mean DOY, thus capturing most of the distribution.

4.6.3 Relationship between bioclimatic variables and meteorological seasons

To contextualize the environmental conditions associated with the identified meteorological seasons, we analyzed the relationship between the derived seasonal variables and the bioclimatic variables.

The seasonal variables obtained in section 4.6.2 include the onset, midpoint (mean DOY), and duration (span) of each meteorological season. These variables provide a temporal characterization of the climate structure. In contrast, the bioclimatic variables summarize long-term environmental conditions derived from temperature and precipitation data. To assess their relationship, we conducted a correlation analysis between the seasonal descriptors and the bioclimatic variables using the Pearson correlation coefficient. The significance level was set to 0.05.

This analysis allows us to identify associations between shifts in the temporal structure of the seasons (e.g., earlier onset or longer duration) and variations in biologically meaningful climatic conditions. The results of this section provide a descriptive link between climate dynamics and the environmental conditions experienced by the colony, which serves as a basis for the integrated analysis presented in Module 4.

4.7 Module 4: Climate and penguin behaviour.

This module constitutes the integrative stage of the methodological framework, where climatic variables are explicitly linked with the behavioural dynamics of the colony. Unlike previous modules, which characterize biological and climatic variables separately, this stage establishes a direct connection between both domains, with the aim of evaluating the role of climate as a potential driver of animal behaviour.

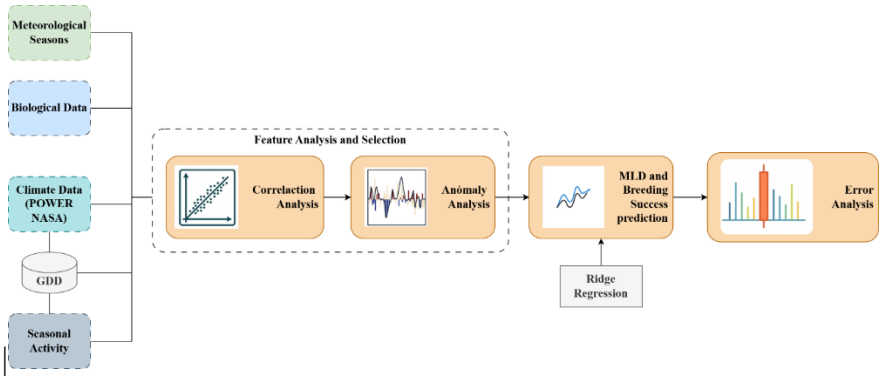

Fig. 21. Module 4 workflow.

In particular, the analysis progresses from an exploratory approach based on correlations toward an evaluation of the predictive power of climatic variables on key breeding season indicators, such as the MLD and breeding success. In this way, this module not only integrates previous results within a unified analytical framework, but also enables a transition from the identification of associations to the development of models capable of capturing the relationship between environmental variability and the biological dynamics of the colony.

4.7.1 Correlation between climate and penguin related variables

Given that in previous modules correlations were only conducted between variables within the same domain, namely, biological variables (Module 2) or climatic variables (Module 3), this section aims to analyze the relationship between climatic variables and variables associated with little penguin behavior. This analysis allows for the exploration of potential dependencies between environmental conditions and the biological dynamics of the colony.

In particular, correlations are evaluated between the previously studied behavioral variables, MLD, breeding success, activity peaks, and the span of the peaks, and a broad set of climatic variables. These include meteorological variables (precipitation, temperature, wind speed, humidity, and SST), bioclimatic variables (BIO1–BIO19), and derived descriptors about the meteorological seasons obtained in Module 3 (onset, mean day, and duration of the seasons).

The analysis is conducted at an annual temporal scale, where each observation corresponds to one year. However, to capture the intra-annual structure of the climate, climatic variables are represented using monthly average values within each year. In this way, twelve monthly features (January to December) are constructed from each climatic variable and are individually correlated with each behavioral variable.

Additionally, for variables related to activity peaks near the moulting season, a shifted temporal window is defined, spanning from July of the previous year to June of the year under study. This decision is based on the temporal location of these biological events, which occur around May, and allows for alignment of the relevant climatic conditions with the observed phenomenon. For the remaining variables, mainly associated with the breeding season, the standard calendar-year structure is maintained.

Correlations are evaluated using the Pearson coefficient, considering as significant those with a p-value less than or equal to 0.05. Results are presented using heatmaps, where significant correlations between variables are visualized, and bar plots that allow comparison of the temporal evolution of the variables over the years.

This type of analysis has been widely used in the little penguin literature to study relationships between climatic and biological variables, such as breeding success–wind (Johnson & Colombelli-Négrel, 2021; Saraux et al., 2016), breeding success–air temperature (Johnson & Colombelli-Négrel, 2021), and MLD–SST (Cullen et al., 2009; Mickelson et al., 1991). In the thesis, this approach is extended through a systematic analysis that considers multiple climatic variables and their intra-annual variation.

Additionally, anomalies (deltas) relative to the mean were calculated for each variable and compared using bar charts, allowing a direct visual comparison of variability patterns between climatic and biological variables.

To further understand the relationship between the climate datasets, the MLD, and breeding success, the Autoregressive Distributed Lag (ARDL) approach was used. ARDL is a time series model that assesses the dependencies between the objective variable and its past values or the present and past valuues of external variables, as shown in **eq. 20** (Kripfganz & Schneider, 2023).

$$Y_t = \sum_{i=1}^{p} \alpha_i Y_{t-i} + \sum_{j=0}^{q} \beta_j X_{t-j} + \epsilon_t, \qquad \textbf{eq. 20}$$

where Y_t denotes the dependent variable at time t, X_t denotes the explanatory variable, α_i and β_j are the short-run dynamic coefficients, p and q represent the lag orders of the dependent and explanatory variables, respectively, and ε_t is the white-noise error term.

Finally, to complement the correlation analysis, temporal trends of variables that show significant relationships are evaluated using the Mann–Kendall test, with a significance level of 0.05. This analysis allows identification of potential systematic changes in the variables over time.

This section is exploratory in nature and focuses on identifying and characterizing potential relationships between climatic and biological variables. The findings obtained here serve as input for variable selection in the following subsection, where predictive models are developed considering a limited number of annual observations.

4.7.2 Breeding variables forecast

In this section, the objective is to evaluate the predictive capacity of climatic variables on biological variables of the breeding season, specifically the MLD and breeding success. This analysis aims to extend the results obtained in the previous section by moving from a correlational relationship to an evaluation of the predictive power of these variables on the reproductive dynamics of the colony.

As input variables, the climatic variables previously analyzed in section 4.6.1 were used. These include meteorological variables represented as monthly averages within each year, where each month constitutes an independent feature at the annual scale, as well as bioclimatic variables (BIO1–BIO19) and descriptors derived from meteorological seasons (onset, duration, and mean day), which are defined at an annual scale. In this way, the set of input variables integrates both the intra-annual structure of climate and aggregated descriptors of environmental conditions.

Variable selection was conducted in two stages. First, only variables that showed a significant correlation with the target variable in the previous section were considered. Subsequently, an additional filter was applied to reduce collinearity among the selected variables. Specifically, variables with a correlation greater than 0.4 with each other were removed, prioritizing in each case those with higher correlation with the target variable.

The final dataset consists of 43 annual observations covering the period from 1982 to 2024. Given the limited number of observations, the number of input variables was restricted to avoid model overfitting. The first 34 years of the dataset were used as the training set, while the last 9 years were used as the testing set.

For modeling, we compared ElasticNet, Ridge, Random Forest, and Gradient Boosting ML models to forecast the MLD. The performances of the models were measured using MAE, RMSE, MAPE, and explained variance. The source code can be found in Appendix 5.

Chapter 5: Results

This chapter presents the experimental outcomes of the modules 1-4 described in Chapter 4. Then, the CEFABC2 management system, integrating every module into a single tool, is introduced.

5.1 Module 1: Time-series data preprocessing

In this section, we present the results obtained from decomposition and reconstruction of the nightly penguin count time series using Singular Spectrum Analysis (SSA). First, the decomposition process is applied to examine the internal structures of the time series. Then, the reconstructed series is used to extract the day of year (DOY) of the time series peaks and to estimate the duration of these high-activity periods.

5.1.1 Decomposition and reconstruction of the time series

In the decomposition process, we performed the spectral analysis of the eigenvalues (a.k.a. singular values), noticing the proportion of variance captured by the SSA components individually (section 4.4.1). The eigenvalues of each component are shown in **Fig. 22**.

Initially, we found that the first component has the largest eigenvalue, which is twice as grater than that of the next component. As mentioned in section 4.4.1, this component is usually associated with the trend in the time series (Wu et al., 2023). This first component is followed by the largest drop in variance in the graph, which leads to the next four components. These four components have very similar values, forming a plateau before the next drop of the graph.

Plateaus in a spectral analysis, as also mentioned in section 4.4.1, are usually associated with the harmonic characteristics of the time series. In the nightly penguin count, harmonic characteristics suggest repetitive seasonal behaviour of the colony. This plateau is followed by the next and last observed drop in variance.

The eigenvalues following this drop decay slowly and smoothly, without abrupt drops in variance. As also mentioned in section 4.4.1, this type of pattern is usually associated with components dominated by high-frequency unstable structures or noise.

The spectral analysis shows that the dominant patterns of the nightly penguin count are captured in the first five components, which is useful information for the reconstruction of the time series. This analysis provides an idea of the possible internal structure of the time series; however, it does

not give details about the harmonic components, such as the period or magnitude. It also does not provide information about the trend of the time series. For that, the eigenvectors are analyzed afterward.

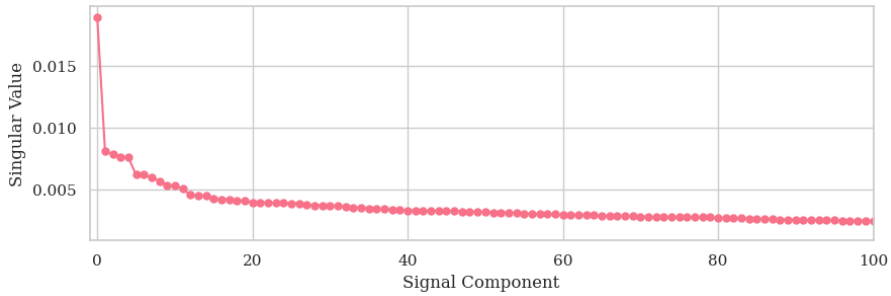

Fig. 22. Singular values of the top 100 components in SSA of penguin nightly count time series.

The visualization of the eigenvector values shows the structures they represent in the time series. **Fig. 23** presents the values of the first seven eigenvectors, which are interpreted as the basis of the temporal patterns dominating the time series structure.

The first eigenvector, which is the one with the largest variance (**Fig. 23**), exhibits an increasing trend and slight oscillatory patterns. The distance between the peaks of these oscillations is approximately 180 steps, suggesting a low-frequency quasi-periodic component superimposed on the long-term trend. The increasing trend of this component indicates an increase in the penguin count.

Additionally, eigenvectors 1 and 2 have oscillatory features with similar frequency but mismatched in phase. According to Hassani (2007), SSA generates pairs of components to represent harmonic structures in the time series. These are observed as complementary functions, such as sine and cosine. The oscillatory components exhibit a period of approximately 180 days (~6 months), indicating a semi-annual cycle in the time series. This periodicity is consistent with the presence of two recurrent biological phases in the colony, likely associated with the breeding and moulting seasons.

Similarly, eigenvectors 3 and 4 form a typical oscillatory pair in SSA that jointly represents the same periodic component of the series. Therefore, they should be grouped to reconstruct a single

underlying oscillation. The following eigenvectors exhibit higher frequencies, representing more detailed high-frequency features of the time series.

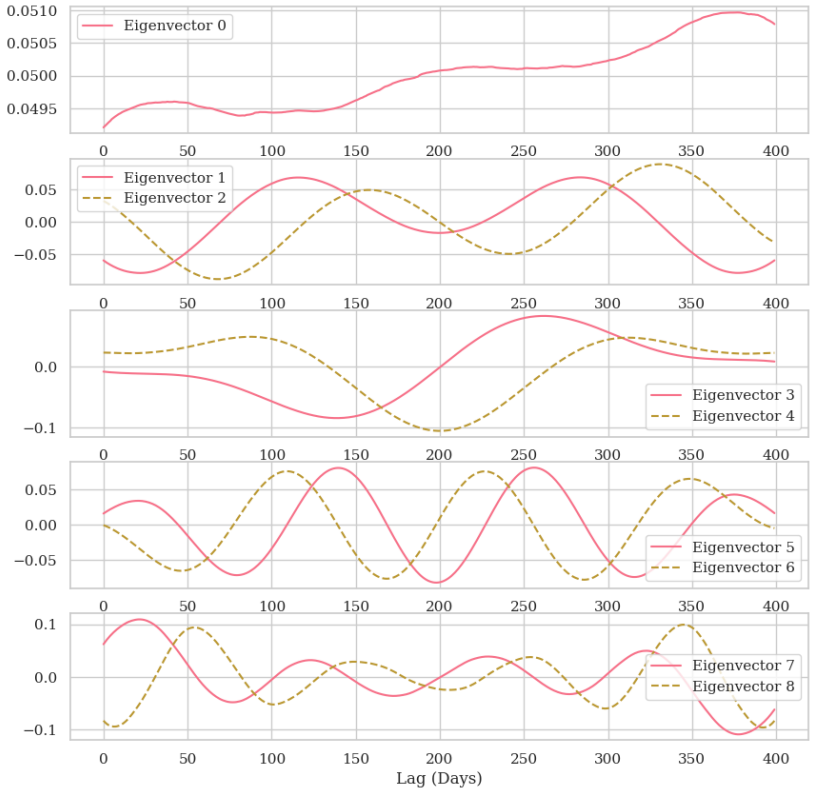

Fig. 23. Eigenvectors of SSA decomposition of penguin nightly count time series.

To analyze the evolution of the time series according to each component, we used the factor vectors, adding each component to the previous ones. **Fig. 24** shows the evolution of the time series as each factor vector is added.

The reconstruction performed using only the first component shows a smooth series with mostly low-frequency variations, which is associated with the trend of the time series. An increasing trend is observed, which agrees with what we spotted in the shape of the eigenvector 0. This indicates a general increase in penguin counts over the years.

The reconstructions performed with the next two components add some oscillating patterns to the series. At this point, adjacent peaks do not vary significantly in magnitude. This shows recurrent count peaks; that is, periods in which the count is higher.

When the next two components are incorporated into the reconstruction, more features of the oscillations are added. Now, adjacent oscillations vary in height and width. Usually, one large oscillation is followed by a smaller one, showing that there are periods with higher counts than others.

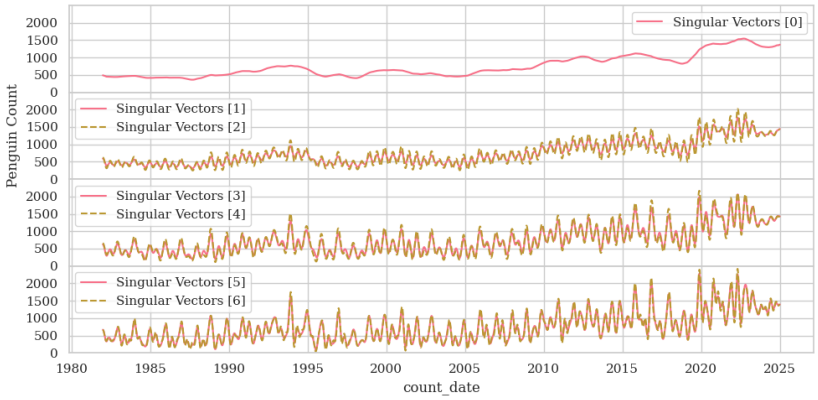

Fig. 24. Progressive reconstruction of the penguin count time series using singular vectors (SSA components).

Fig. 25 shows the reconstruction performed with the aggregated components 6-400. Less apparent structure and finer details of the time series can be seen. This result complements the variance of the components 0-100 in Fig. 22, as it slowly decays as more components are observed.

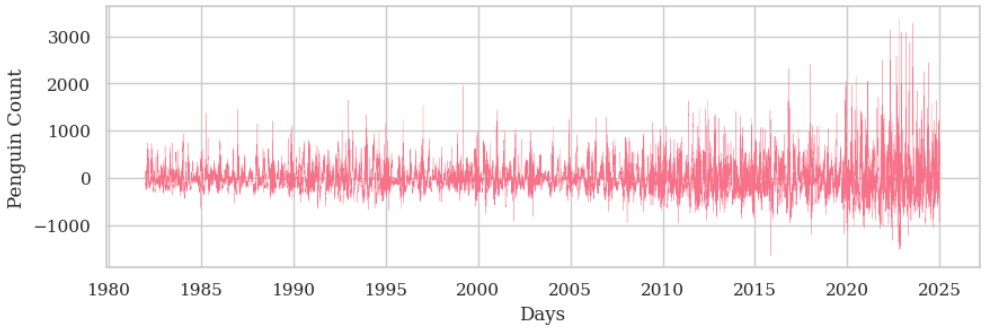

Fig. 25. Reconstruction of the penguin count time series using singular vectors (SSA aggregated components 6-400).

Considering the patterns in the variance and the eigenvectors, we reconstructed the time series of the nightly penguin count time series using only top five components of the SSA. **Fig. 26** shows the reconstructed time series compared to the observed series.

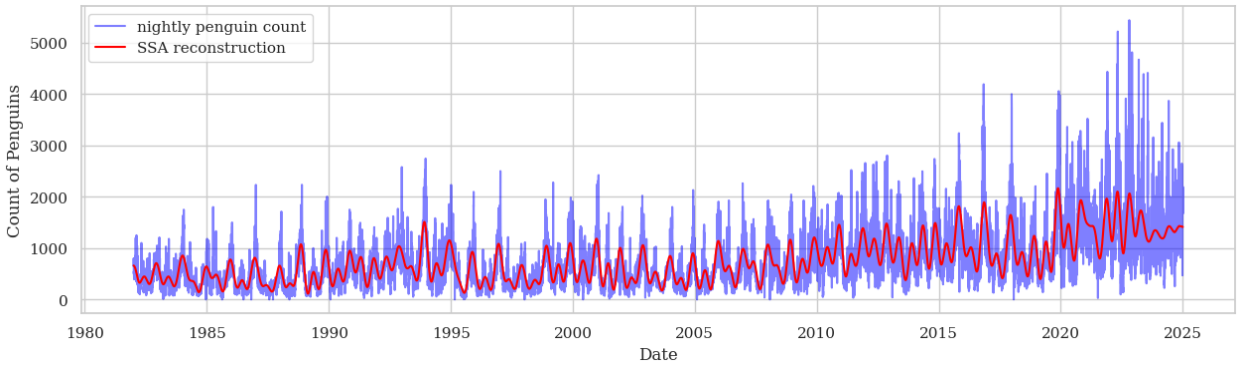

Fig. 26. Observed and reconstructed by the SSA nightly penguin count time series.

5.1.2 Portrayal of the seasonal patterns

The reconstruction performed through the SSA showed that the nightly count time series has two recurrent periodic phenomena representing a structured intra-annual pattern in the colony activity. One peak occurs around the end of the calendar year (November-January) during the breeding season (Dupuis et al., 2023), while the second peak occurs around the moulting period (April-June). **Fig. 27** displays two different years, starting in July, just before the breeding season of the corresponding year, and ending in June of the next year.

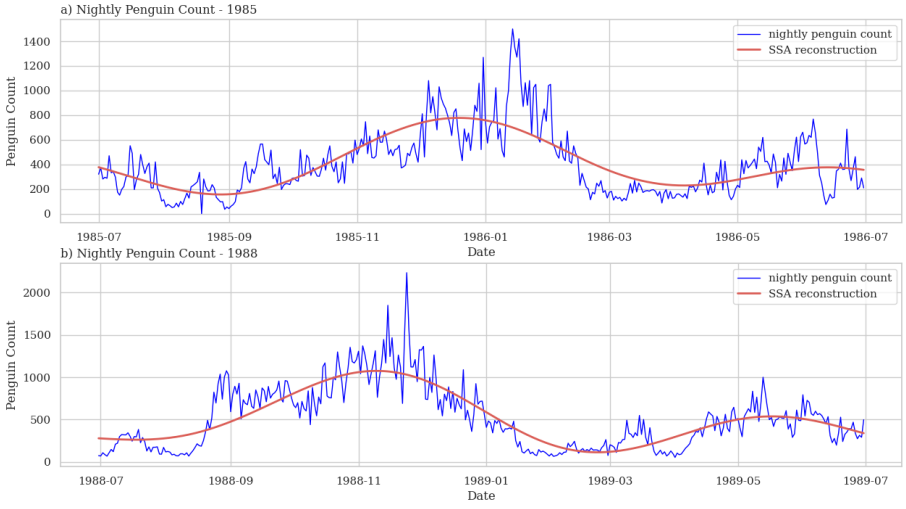

Fig. 27. SSA reconstruction of nightly penguin count time series: a) 1985, b) 1988.

The consistently higher magnitude of the breeding-season peak suggests that this life cycle phase concentrates the highest levels of the colony activity, whereas the subsequent smaller peak likely reflects post-breeding processes. However, inter-annual differences in the timing of the main peak

indicate that, although the seasonal structure remains stable, its temporal occurrence varies, reflecting shifts in the biological activity.

5.1.2.1 Peak extraction

The two peaks in the colony activity are identified from the SSA-reconstructed time series, which preserves the dominant temporal structure of the data while reducing high-frequency fluctuations. As a result, the detected peaks correspond to the main seasonal activity patterns of the colony rather than short-term variability in the observations.

By relying on the reconstructed series, peak detection focuses on the underlying biological processes, providing a more stable and consistent representation that captures the timing of high-activity periods across the year. This approach enables the portrayal of recurrent seasonal behaviour in the colony through filtering out high-frequency fluctuations from the original time series (**Fig. 28)**.

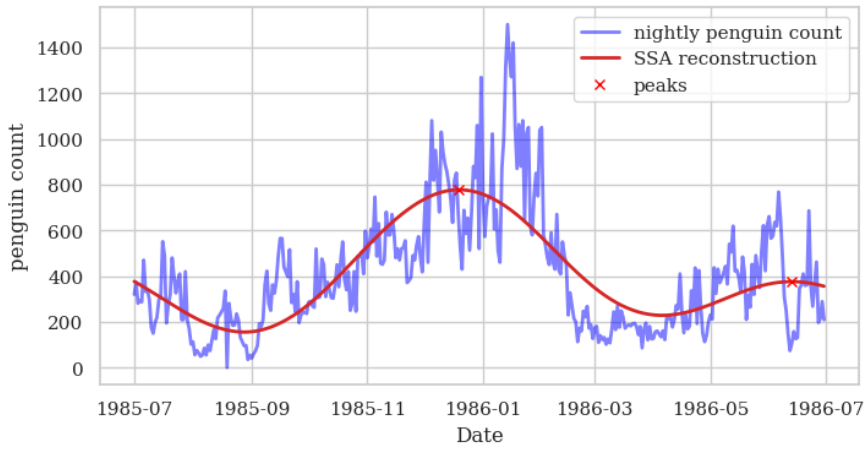

Fig. 28. Peaks (red crosses) identified in the SSA-reconstructed time series.

5.1.2.2 Span of the season: Gaussian mixture models (GMM)

The GMM applied to the SSA-reconstructed nightly count time series successfully identified two components per year, aligning with the two recurrent peaks that were observed in the SSA-reconstructed time series in section 5.1.2.1. Each peak is approximated by a Gaussian component in the GMM, and the standard deviation of such Gaussian components provides a measure of the peak's temporal dispersion, interpreted here as its *span* (Fig. 29). As introduced in the section 4.4.2.2, such span of activity was examined in relation to the MLD and breeding success in Module 2.

Given that one of the peaks occurs during the breeding season, it is referred to throughout this thesis as the *breeding peak*, while the second peak, located near the moulting period, is referred to as the *moulting peak*. It is important to clarify that these spans do not represent the full duration of the breeding or moulting seasons themselves, but rather the temporal extent of increased activity observed in the nightly penguin counts around these biological periods.

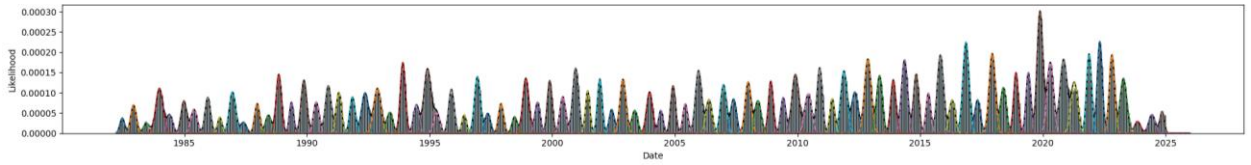

Fig. 29. GMM fitted to the detrended SSA-reconstructed time series of the nightly penguin count.

5.1.3 Module 1 summary

Using SSA to analyze and reconstruct the nightly penguin count time series, we identified the main structural components of the time series, including its trend, seasonal patterns, and higher-frequency variations. The reconstruction using the top 5 SSA components allowed a smoother representation of the time series, enabling the extraction of the peaks and the estimation of their temporal span. To the best of our knowledge, this time series decomposition and feature extraction has not been reported in the literature on little penguins. In this context, the variables derived in this study (i.e., peaks and span) provide a novel and complementary portrayal of the colony biological activity.

5.2 Module 2: Behavioural pattern analysis

In this module, we analyzed the trends, distributions, and correlations among MLD, breeding success, and the nightly counts' peaks related to the Phillip Island colony. Additionally, we evaluated the predictive capacity of statistical, ML, and DL models on the nightly penguin count series. To assess the models, multiple temporal resolutions and number of SSA components were used to reconstruct the time series.

We started by conducting statistical analysis to assess the relationship between the extracted peaks and spans, on the one hand and MLD and breeding success, on the other hand. Then, predictive models are evaluated to analyze how forecasting performance is affected by the number of SSA components used to reconstruct the time series and the temporal resolution of the series.

5.2.1 Statistical analysis of the behavioural variables

5.2.1.1 Peak and span of the biological seasons

The trends of the peak timing and span extracted in section 5.1.2 are shown in Fig. 30. The observed breeding peak records range from 1982 to 2023. On average, they occur on DOY = 334 with the standard deviation of 17 days. The timing of the breeding peaks shows a statistically significant decreasing trend (−0.53 days per year, *p*-value = 0.041), which corresponds to the accumulated total of 22.26 days over the 42-year period. This decreasing trend indicates a progressive shift toward earlier calendar dates. In contrast, the span of the breeding peak shows no significant trend (–0.033 days per year), with the mean value of 42 days and the standard deviation of 5 days.

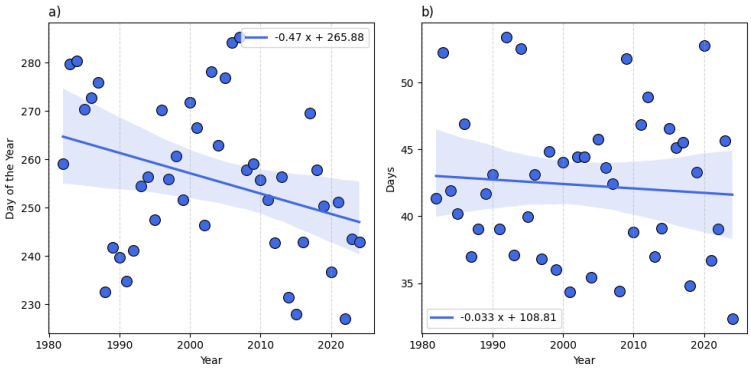

Fig. 30. Temporal evolution of the breeding peaks: a) peak timing (DOY), b) span (days) (standard deviations of the fitted GMM components).

A similar pattern is detected in the moulting peaks, whose records range from 1982 to 2024 (Fig. 31). The peak timing also shows a statistically significant decreasing trend (−0.43 days per year, *p*-value = 0.024), indicating an earlier occurrence of this activity peak totalling to 18.49 days over the 43-year period. However, no significant trend is found in the span of the moulting peaks (–0.07 days per year).

Overall, these results indicate that the main temporal change in the nocturnal activity of the colony occurs in the timing of the peaks, which shifts progressively toward earlier dates, while the duration of the high-activity periods remains relatively stable over time.

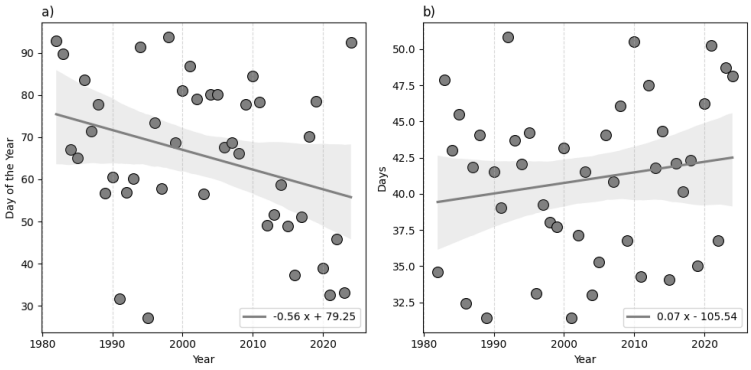

Fig. 31. Temporal evolution of the moulting season: a) peak timing (DOY), b) span (days) (standard deviations of the fitted GMM components).

5.2.1.2 Mean egg-laying date (MLD)

The mean egg-laying date (MLD), analyzed from 1982 to 2023, shows a distribution ranging from mid-August to the end of December, with the mean value of DOY=273 and the standard deviation of 22 days. Consequently, the egg-laying timing occurs primarily during the austral spring (September–November), also showing interannual variability with some years exhibiting earlier dates in late winter and others extending into early summer (Fig. 32). The magnitude of the standard deviation reflects a moderate dispersion in the timing of reproductive onset across years.

After performing the Mann−Kendall test over the MLD records, a statistically significant decreasing trend was detected (−0.9 days per year, *p*-value = 0.0013), indicating a progressive shift toward earlier egg-laying dates, with accumulated total of 37.8 days over the 42-year period. This result suggests a systematic shift in the onset of the breeding season.

Additionally, a strong positive correlation is found between the breeding peak and the MLD (r = 0.79, *p*-value = 2.4e−10), indicating that both variables exhibit a similar temporal behaviour and are closely aligned within the reproductive cycle (**Fig. 33a)**.

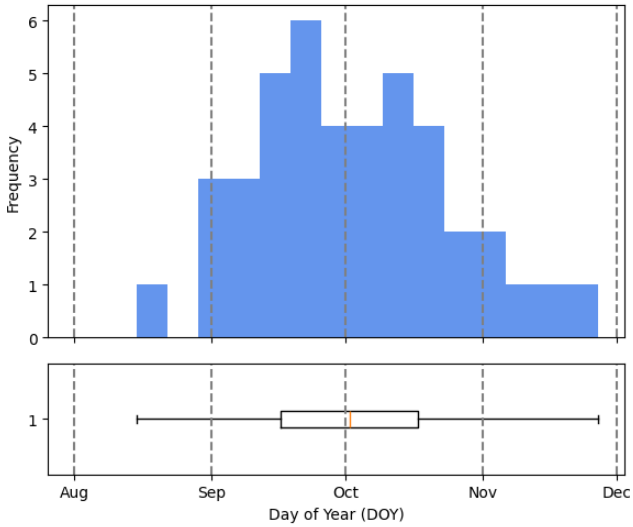

Fig. 32. MLD distribution across the year.

At the same time, the distribution of the mismatch variable, defined as the difference between the breeding peak timing and the MLD, shows a mean of 61 days and a standard deviation of 13.4 days (**Fig. 33b**). Hence, the breeding peak typically occurs two months later than the MLD, with a variation between one and three months after the MLD. The mismatch distribution suggests that the MLD does not occur in synchrony with the breeding peak, but rather precedes it within the breeding cycle.

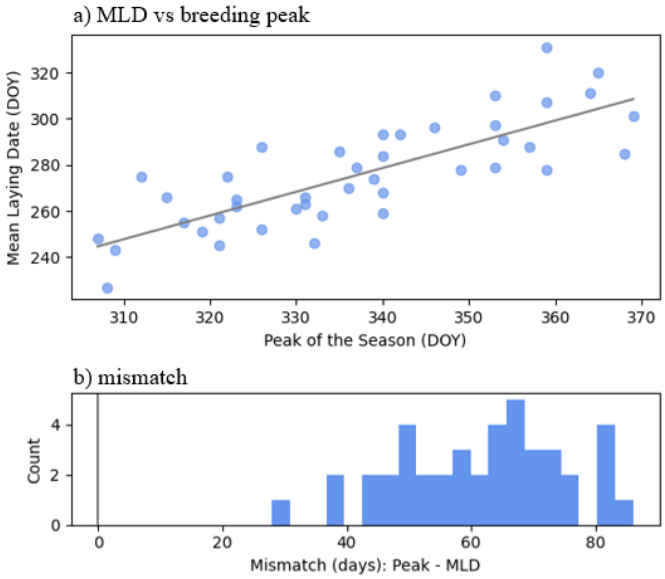

Fig. 33. MLD vs. breeding peak: a) correlation; b) mismatch.

No significant correlation is found between the MLD and the span of the breeding peak (p-value = 0.24). However, a statistically significant correlation is observed between the mismatch and the span of the breeding peak (p-value = 0.031), indicating that when the temporal distance between the MLD and the breeding peak increases, the period of high activity becomes more temporally dispersed (Fig. 34). The breeding peak span increased by 0.115 days for each additional day of mismatch. Hence, years with large temporal distances between the MLD and the breeding peak are associated with the longer span of these biological activities.

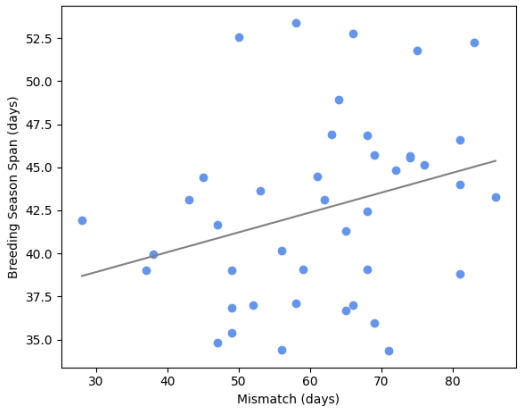

Fig. 34. Span of the breeding peak and mismatch.

Finally, no significant relationships are observed between the MLD and the variables associated with the moulting peak, suggesting that these processes occur independently within the annual cycle.

Overall, statistical analysis involving MLD, breeding peak, mismatch, and span of the breeding peak supports an idea that the variables derived from the SSA-based reconstruction of the nightly penguin count capture relevant aspects of the reproductive dynamics of the colony, providing insights complementary to traditionally reported biological indicators.

5.2.1.3 Breeding success

The records of breeding success, defined as the average number of chicks per pair, encompass the period from 1982 to 2023. The breeding success values range from 0.07 to 2.25, with a mean of 1.04 and a standard deviation of 0.43 (**Fig. 35**). Therefore, each pair successfully raises one chick per breeding season on average. However, the observed range reveals a substantial interannual

variability, with some years exhibiting success rates lower than half of the mean value, while other years show values that double the mean.

Mann−Kendall test indicates that breeding success shows a statistically significant increasing trend of 0.01 chicks per pair per year (p-value = 0.044), suggesting a gradual improvement in the reproductive performance of the colony over time.

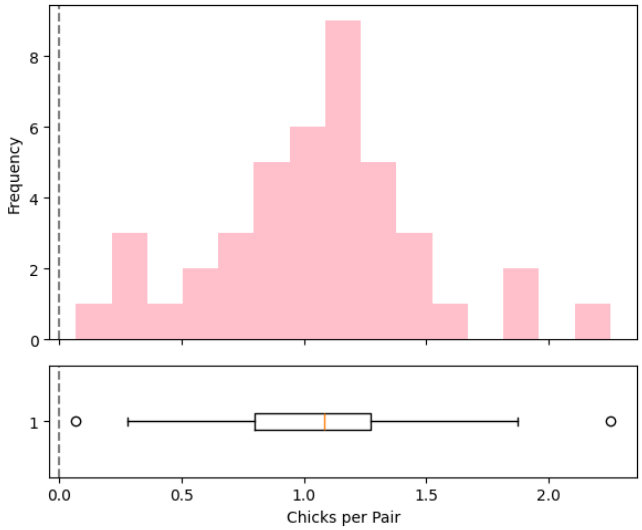

Fig. 35. Breeding success distribution.

Significant correlations are detected between breeding success and the other variables associated with the breeding season (**Fig. 36**). A strong negative correlation is found with MLD (r = −0.655, p-value = 2.51e−06), indicating that an earlier MLD is associated with higher reproductive success. Additionally, breeding success is positively correlated with the span of the breeding season (r = 0.51, p-value = 0.00055), suggesting that years with longer peaks tend to correspond to higher reproductive output. A negative correlation is also observed between the timing of the breeding peak and the breeding success (r = −0.42, p-value = 0.005), indicating that earlier breeding peaks are associated with improved breeding success. In sum, breeding success decreased by 0.012 chicks per pair for each additional DOY of MLD, increased by 0.041 chicks per pair for each additional day of breeding peak span, and decreased by 0.010 chicks per pair for each additional DOY of the breeding peak.

No significant relationships were observed between breeding success and variables associated with the moulting peak, suggesting that reproductive performance is primarily linked to the temporal dynamics of the breeding period.

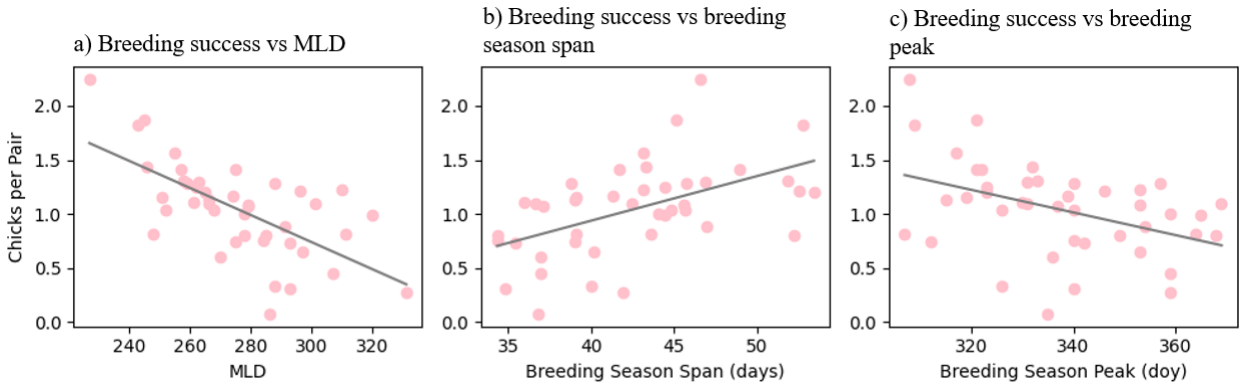

Fig. 36. Relationships between breeding success and a) mean egg-laying date (MLD), b) breeding season span, and c) breeding season peak timing.

Overall, correlations between the breeding peak and breeding success indicate a strong relationship. Consequently, breeding peaks extracted from the SSA-reconstructed time series serve as relevant descriptors of the colony's reproductive dynamics.

5.2.2 Prediction of the peaks

To predict the moulting and breeding peaks, we implemented Gaussian processes (GP) described in section 4.4.2.1. The model initially forecasted the SSA-reconstructed time series and then we extracted the peaks from that forecast.

The forecast of the moulting peak in 2008 is shown in Fig. 37. The orange curve represents the mean function of the GP model, i.e., the forecast of the nightly penguin count time series, while the shaded area corresponds to the associated variance (uncertainty). The red crosses on the reconstructed nightly penguin count forecast curve indicate the detected peaks, and the blue curve represents the SSA-reconstructed time series (5 components).

During the first 7 months of the 2008 forecast, the model captures the observed pattern, enclosing the SSA-reconstructed time series within the uncertainty area. Beyond this period, the model begins to deviate from the true series, and the forecast uncertainty no longer encompasses the true values. This shows that the model is useful for extracting the peaks that occur within a few months

from the last observed data point, but its performance decreases for peaks located farther ahead in time.

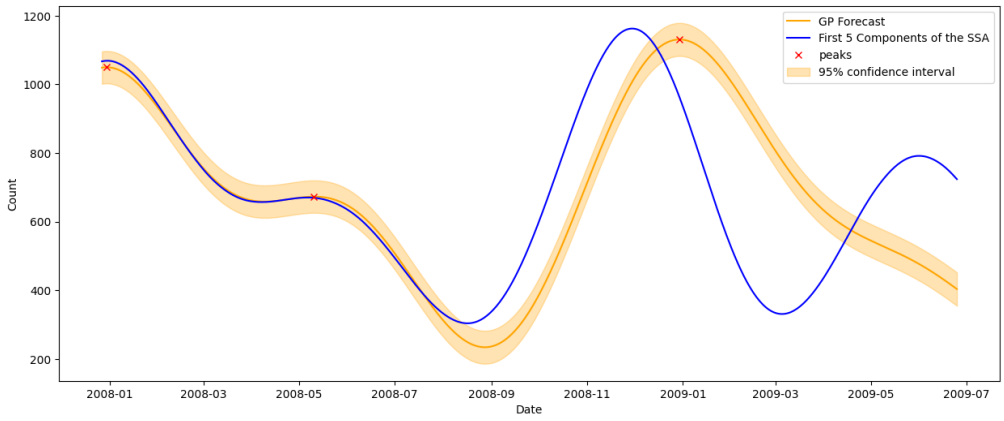

Fig. 37. Forecast of the moulting peak for 2008 with GP model.

Fig. 38 shows the forecast of the breeding peak in 2008. Similar to the moulting peak, the model captures the seasonal dynamics of the SSA-reconstructed time series during the initial months but progressively deviates from the observed data points as the forecast horizon extends. Hence, the breeding peak can also be extracted from the GP forecast.

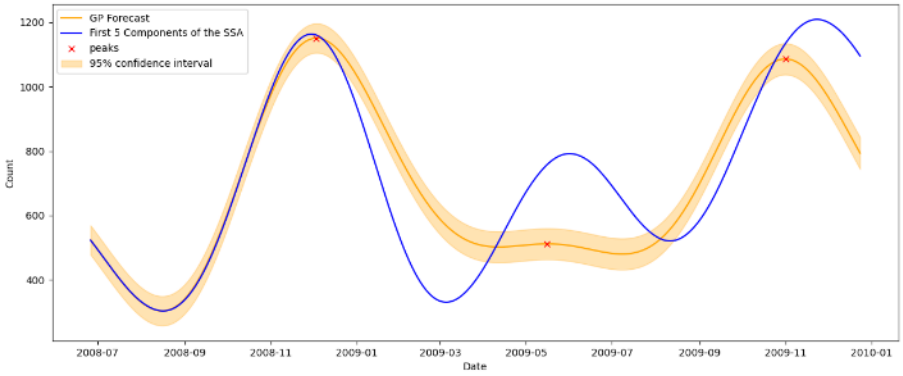

Fig. 38. Forecast of the breeding peak for 2008 with GP model.

Fig. 39 depicts the relationships between predicted and observed peaks. For both the moulting and breeding peaks, the dashed line represents perfect agreement between predictions and observations. The points are clustered around the diagonal, indicating a high level of agreement between predicted and observed peak timings.

Peak prediction errors are summarized in **Table 7**. On average, observed breeding peaks occur on DOY=334 with the standard deviation of 17 days. Hence, the model has an RMSE lower than the spread of the data and forecasts the occurrence of the breeding peak approximately 152 days (~5 months) in advance. Similarly, the observed moulting peaks occur on average on DOY=145 with the standard deviation of 15.9 days. Consequently, the model has an RMSE lower than the spread of the moulting peaks and forecasts their occurrence approximately 110 days (~5 months) in advance. In general, the chosen training window, described in section 4.4.2.1, allows a gap of several months between the last observed value and the predicted peaks.

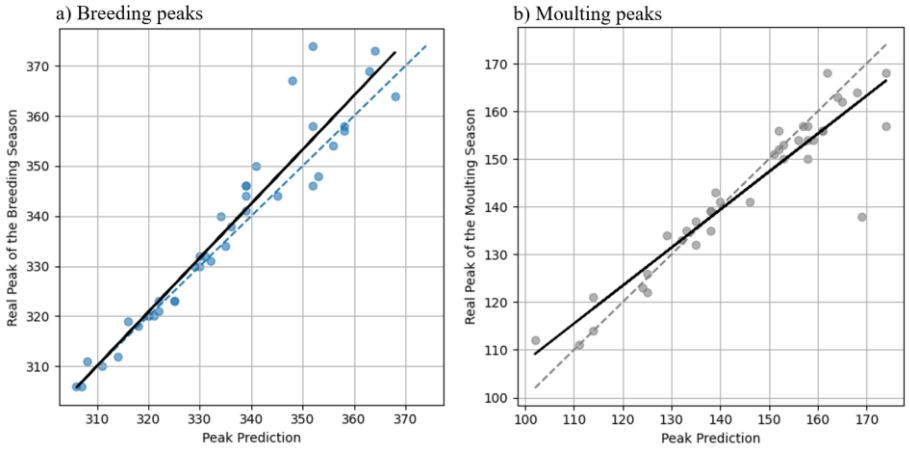

Fig. 39. Relationships between observed and predicted values of a) breeding peaks, b) moulting peaks.

Table 7. Peak prediction error

Error	Moulting Peak	Breeding Peak
Mean absolute error	4.103	3.718
Root mean squared error	6.645	5.664
R^2 score	0.826	0.887
Median absolute error	2.000	2.000
Mean absolute percentage error	0.029	0.011
Mean squared error	44.154	32.077
Mean squared log error	0.002	0.000

5.2.3 Count forecast

5.2.3.1 Granularities of the series

Fig. 40 shows different temporal granularities and SSA-based reconstructions used to evaluate forecasting performance. As described in section 4.4.1, SSA allows the decomposition of the time series into components associated with dominant patterns and higher-frequency variability.

As the number of SSA components increases, the reconstructed series incorporates more details and becomes less smooth. The reconstruction with 5 SSA components captures the main seasonal patterns associated with the annual phenology (section 5.1.1). However, when additional components are included (10, 20, and 30), more oscillatory features are captured. As well, the difference between the reconstructed time series and the raw data is getting smaller at the coarser temporal granularity (i.e., monthly vs. weekly vs. daily).

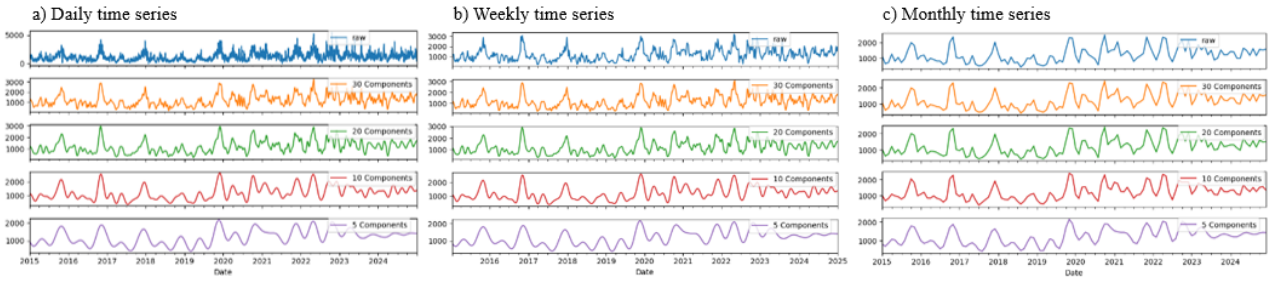

Fig. 40. Nightly penguin count: raw time series and reconstruction with 5, 10, 20, and 30 SSA components for a) daily, b) weekly, and c) monthly granularity.

Overall, SSA reconstruction and temporal aggregation provide different levels of detail in the representations of the time series, which later in the study are used to evaluate forecasting performance.

5.2.3.2 Behavioural time series forecasting

5.2.3.2.1 Nightly granularity

The performance of the models at nightly granularity (**Fig. 41**) decreases as the forecast horizon extends. Across all SSA reconstructions, ARIMA and Ridge show the best performance, while GP and STS consistently underperform. In particular, STS exhibits unstable behaviour, with errors substantially increasing for prolonged horizons and more SSA components.

LightGBM outperforms GP and STS in all SSA-reconstructed series and shows greater robustness to noise compared to LSTM. LSTM performs better than LightGBM, STS, and GP for SSA-reconstructed series, but underperforms them when using the raw data, suggesting that it learns irrelevant patterns in the presence of noise. In contrast, ARIMA and Ridge remain robust across all reconstructions, outperforming the rest of the models.

As described in section 4.4.1, SSA reconstruction isolates the dominant patterns of the observed time series while reducing high-frequency variability when a reduced number of components is used. In this context, increasing the number of components introduces additional details to the reconstructed time series, which affects model performance. While ARIMA and Ridge remain stable, more complex models, such as LSTM, are more sensitive to this additional variability.

When using the raw time series, the differences between models become less noticeable, and similar performances are observed across horizons (except for STS due to its high error rate). This result supports an idea that the noise present in the original series limits the ability of the models to capture the underlying temporal structure.

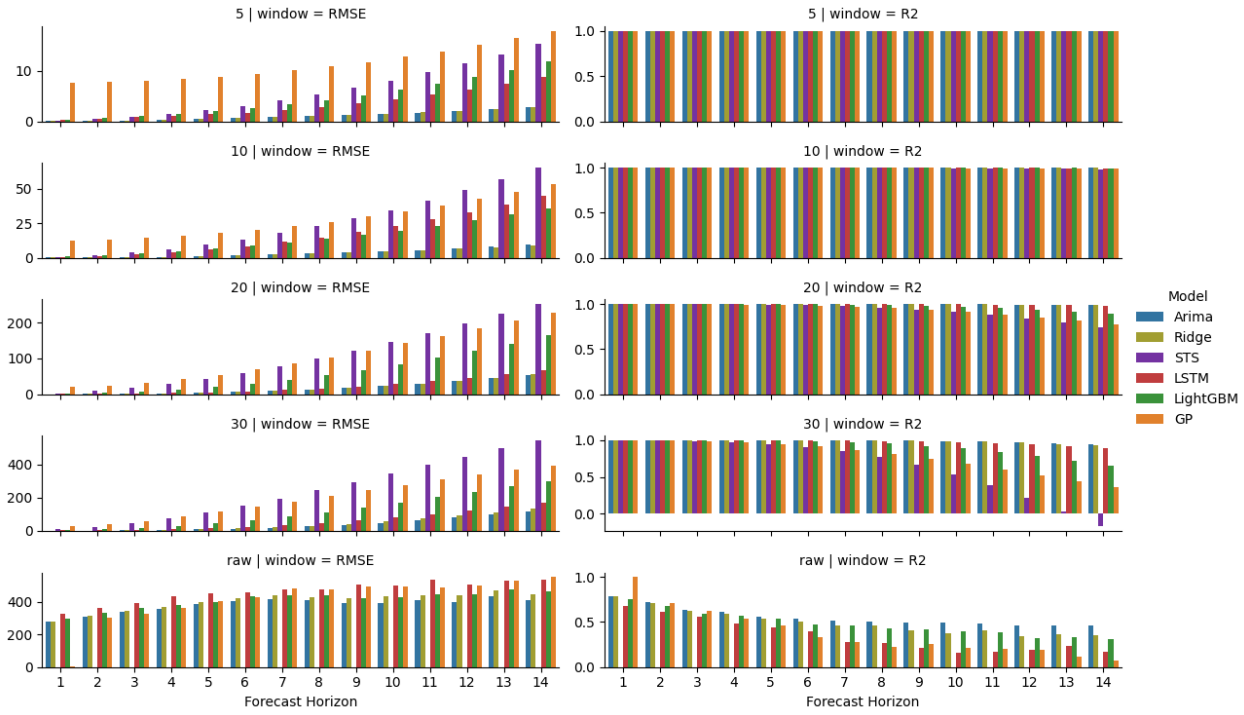

Fig. 41. Performance error at nightly granularity using 5, 10, 20, and 30 SSA components for the time series reconstruction.

In general, the forecasting performance is strongly affected by the high-frequency variability of the raw nightly penguin count, and that preprocessing through SSA is key to improving model performance.

5.2.3.2.2 Weekly granularity

The performance of the models at weekly granularity shows a different behaviour compared to the nightly resolution (**Fig. 42**). In this setting, LightGBM consistently outperforms all other models across all SSA-based reconstructions. Similar to the nightly case, GP exhibit the lowest performance when using 5 SSA components. However, as more SSA components are used for the reconstruction (10, 20, 30), the performance of GP becomes comparable to that of LSTM, ARIMA, and Ridge at longer forecasting horizons, with this convergence occurring earlier when more components are included in the reconstruction.

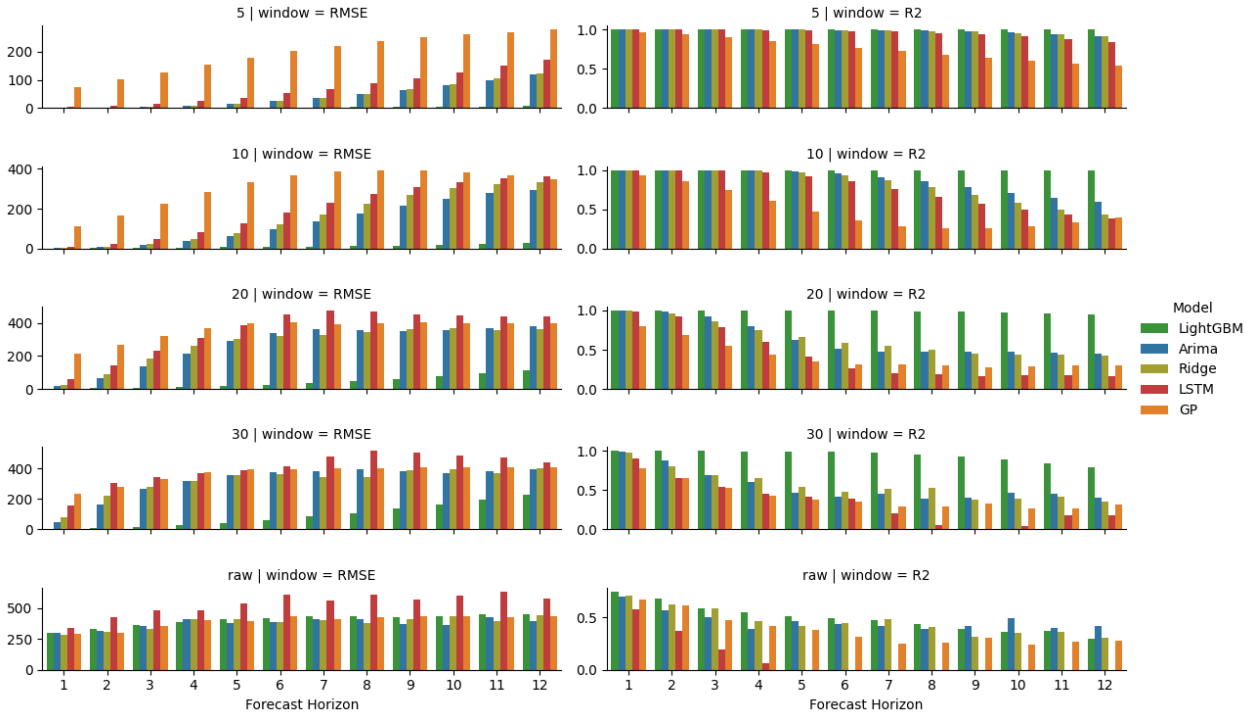

Fig. 42. Performance error at weekly granularity using 5, 10, 20, and 30 SSA components for the time series reconstruction.

LSTM continues to underperform relative to the other models as more SSA components are used to reconstruct the series, indicating sensitivity to additional oscillatory behaviours introduced by

the new components. When using the raw weekly time series, the differences between models decrease, and all models, except LSTM, exhibit similar performance.

5.2.3.2.3 Monthly granularity

The performance of the models at monthly granularity shows a similar behaviour across all model families (**Fig. 43**). In contrast to the nightly and weekly granularities, the differences between the raw and the SSA-reconstructed time series are less noticeable.

As a result, the performance differences between models become less prominent, suggesting that most models capture similar underlying patterns at this temporal scale. Additionally, the performance of LightGBM decreases faster than that of the other models, indicating a reduced advantage of more complex models when the series is aggregated at a monthly level.

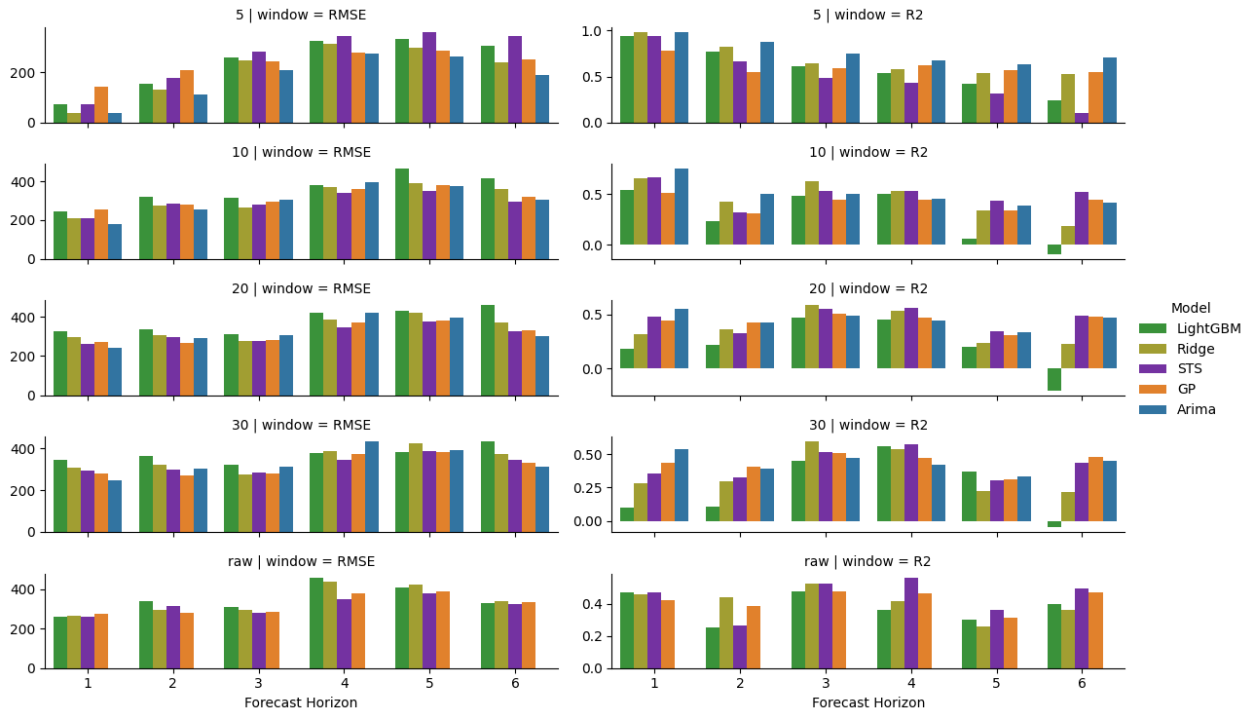

Fig. 43. Performance error at monthly granularity using 5, 10, 20, and 30 SSA components for the time series reconstruction.

5.2.3 Module 2 summary

In this section, we analyzed the statistical properties of the biological indicators: moulting and breeding peaks. Their relationship with MLD and breeding success, two variables that have been widely studied in the literature (Cannell et al., 2012; Saraux & Chiaradia, 2022), was also analyzed.

Subsequently, we predicted the nightly penguin count time series with the aim of extracting the peak and evaluating predictive performance from months before the peak. Finally, we evaluated the predictions of model families used in time series forecasting using a multi-horizon strategy.

5.3 Module 3: Climate time series analysis

This section presents the trends and statistical characteristics of the climatic variables on Phillip Island. As discussed in the background, several climatic variables, such as SST, wind speed, humidity, precipitation, and air temperature, have been associated with breeding success and MLD (Dann & Chambers, 2013). Therefore, understanding their variability provides the necessary context for interpreting the relationships explored in Module 4.

Additionally, this module presents the variations in the duration, mean day, and starting day of the meteorological seasons in the region. These variables were derived using the unsupervised ML models presented in section 4.4.2.

5.3.1 Characterization of the meteorological variables

The region exhibits an oceanic climate, with mild winters and cool summers. Air temperature variables (i.e., T2M, T2M_MAX, T2M_MIN) range between 5.2°C and 31.0°C, showing a clear seasonal pattern with heat peaks in summer (January–February) and minima in winter (July) (**Fig. 44c–e**).

Throughout the year, this region experiences monthly accumulated precipitation (PRECTOTCORR) ranging from 4.9 to 192.8 mm (**Fig. 44a**), relative humidity (RH2M) from 48.7 to 95.3% (**Fig. 44b**), and wind speed (WS2M) from 1.1 to 11.3 m/s (**Fig. 44f**).

After performing Mann-Kendall test on the monthly average values (accumulated for precipitation) of each climate variable and testing each month independently, we found the following annual trends (**Fig. 45**):

- precipitation (PRECTOTCORR) has a decreasing trend in June and July.
- relative humidity (RH2M) has an increasing trend in January and February, but a decreasing trend from May to September.
- daily average temperature (T2M) and maximum daily temperature (T2M_MAX) have an increasing trend from June to January (excluding November).

- minimum daily temperature (T2M_MIN) has an increasing trend from June to January and in March.
- wind speed (WS2M) has a decreasing trend in January.

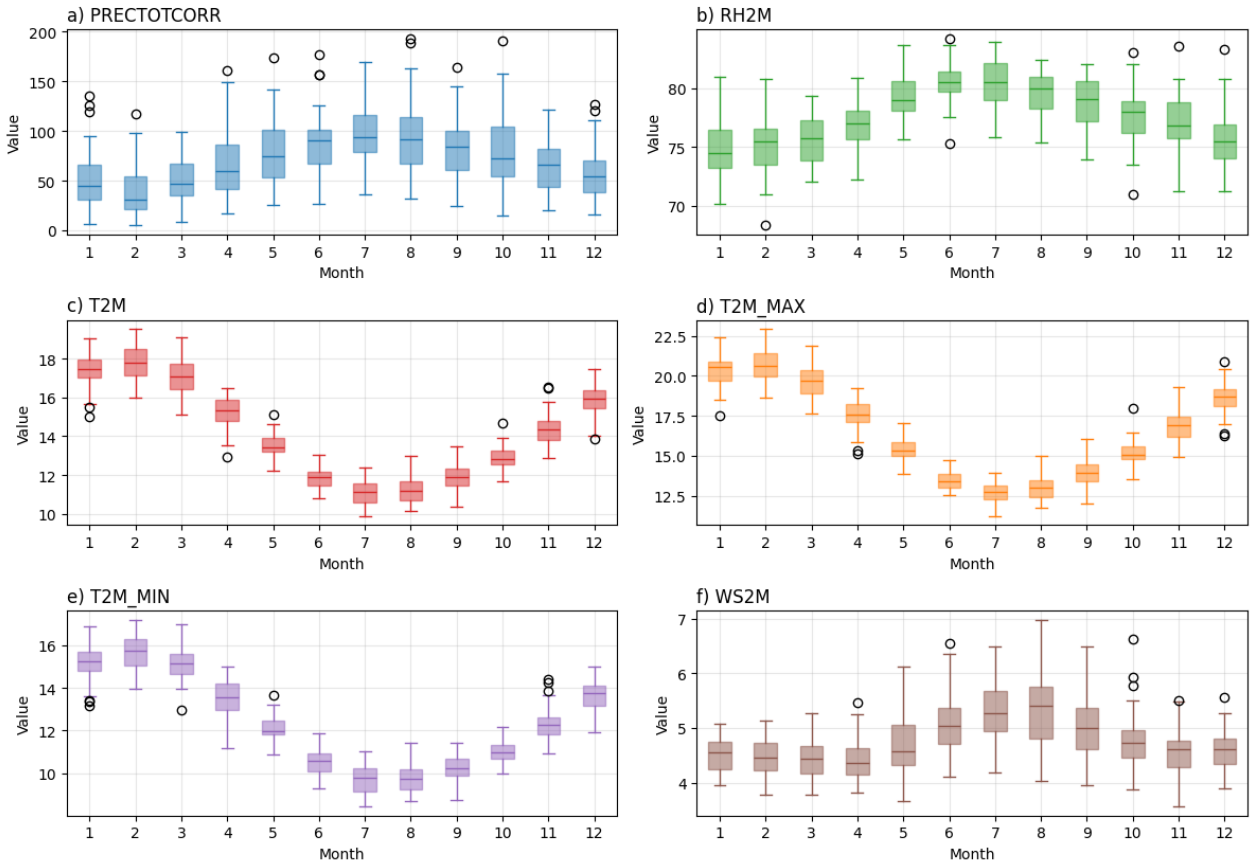

Fig. 44. Monthly variability of climate variables: a) precipitation (PRECTOTCORR), b) relative humidity (RH2M), c) air temperature (T2M), d) maximum temperature (T2M_MAX), e) minimum temperature (T2M_MIN), and f) wind speed (WS2M).

Fig. 45 depicts the months with the highest trend for each climate variable.

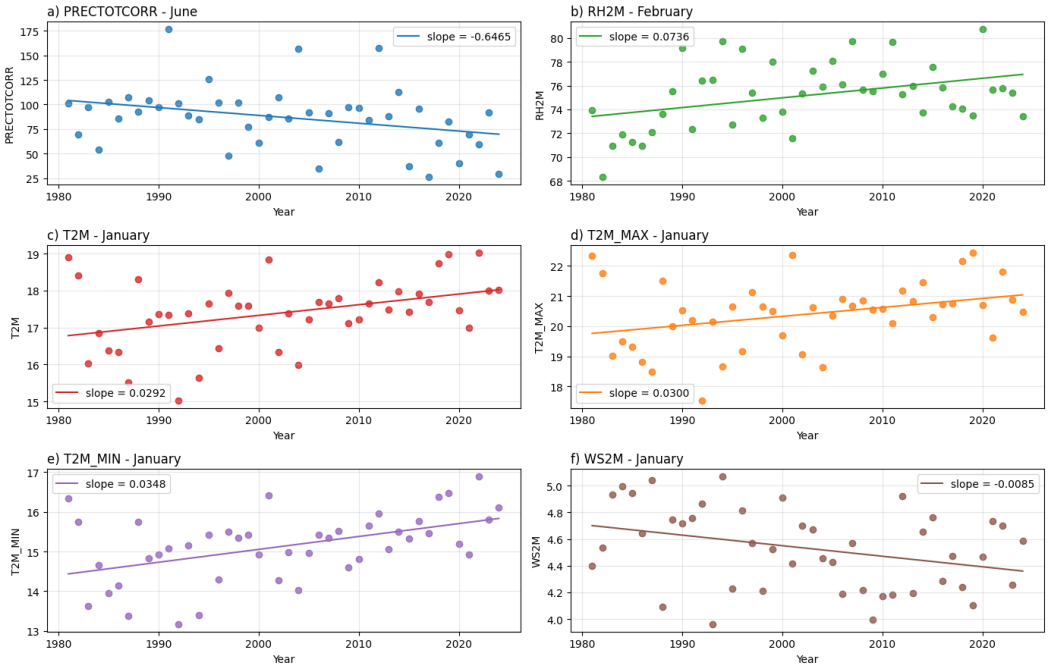

Fig. 45. Monthly trends of climate variables: a) precipitation (June), b) relative humidity (February), c) air temperature (January), d) maximum temperature (January), e) minimum temperature (January), and f) wind speed (January)

Sea surface temperature (SST) ranges from 12.2 to 19.5 °C (**Fig. 46b**), having an increasing trend in every month, except for May, which has no trend.

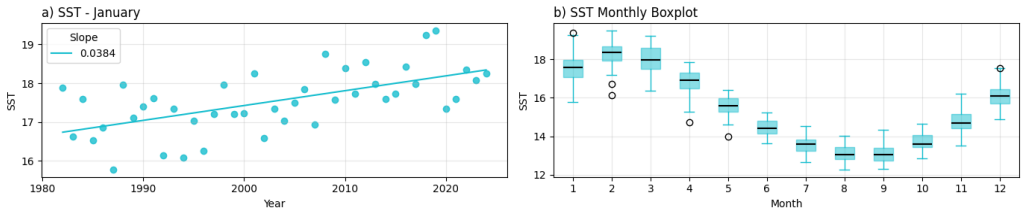

Fig. 46. Sea surface temperature (SST): a) January trend and b) monthly distribution

In general, these results show that the region experiences warming trends for several months with reduced precipitation in the middle of the year.

5.3.2 Bioclimatic variables

Bioclimatic variables are ecologically meaningful descriptors derived from temperature and precipitation series. These variables summarize different aspects of the thermal and precipitation

regimes of the region, providing a more integrated representation of environmental conditions than raw meteorological variables.

To the best of our knowledge, although these variables are biologically relevant, they have not been used to study the little penguins. Their inclusion in this work extends the set of environmental descriptors typically considered in the literature.

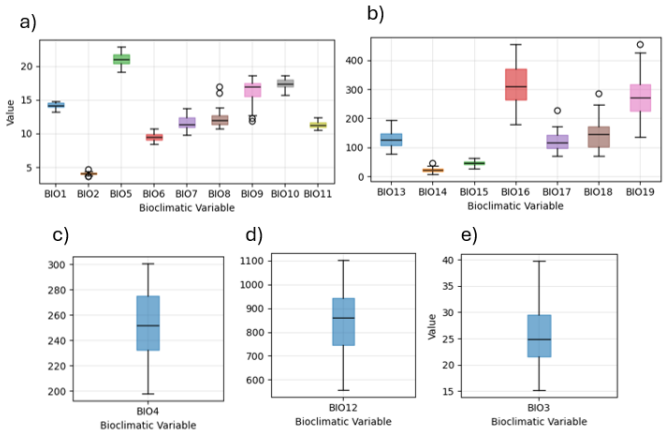

Fig. 47. Bioclimatic variables: a) temperature-related variables, b) precipitation-related variables, c) BIO4, d) BIO12, and e) BIO3.

In this module, they are presented as part of the climatic characterization of the region. Their role becomes relevant in Module 4, where they are integrated with the biological variables of the colony. **Fig. 47** shows their distribution.

5.3.3 Analysis of the seasonal dynamics

As explained in section 4.4.2, the data points of our climate dataset were clustered using K-means. As a result, we obtained three well-defined clusters (**Fig. 48a**) that distribute along different ranges of temperature. **Fig. 49** presents the temporal distribution of these clusters over the period of two years, where red dots are summer-related days, blue dots are winter-related days, and yellow dots are transitional seasons. The fact that spring and fall are grouped within the same, transitional, cluster suggests that, based on their climatic characteristics, both seasons exhibit similar environmental conditions.

Fig. 48b shows the separation of this yellow cluster into two clusters performed using GMM.

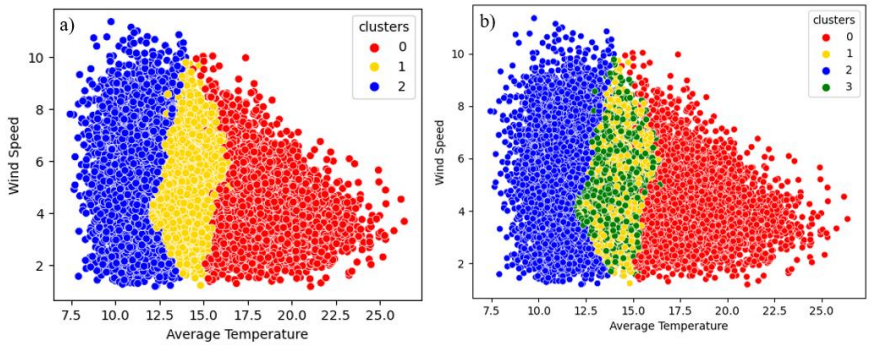

Fig. 48. Distribution of the climate data points a) clustered by K-means, b) separated by GMM.

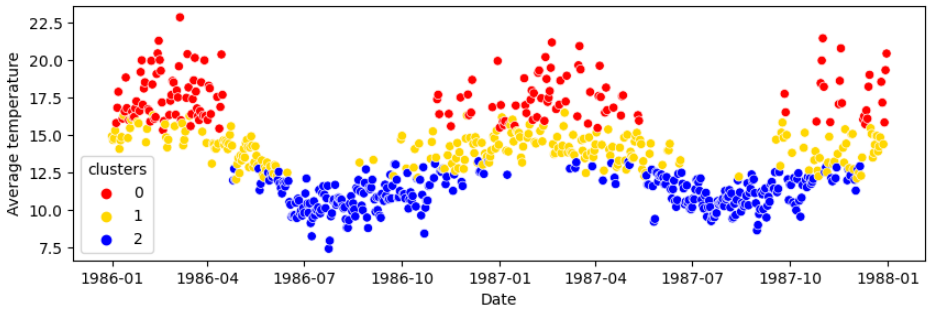

Fig. 49. K-means seasonal clustering of 1986 and 1987 data points (cluster 0 represents Summer, cluster 1 combines Spring and Fall, cluster 2 represents Winter).

Fig. 50 shows the histogram of the points related to the transitional seasons and how the GMM fits it. In the middle of the graph, we observe a bimodal distribution, where the two modes correspond to spring and fall. This indicates that, although both seasons may share similar climatic characteristics, their temporal distributions allow to separate them, as they occur in different periods of the year.

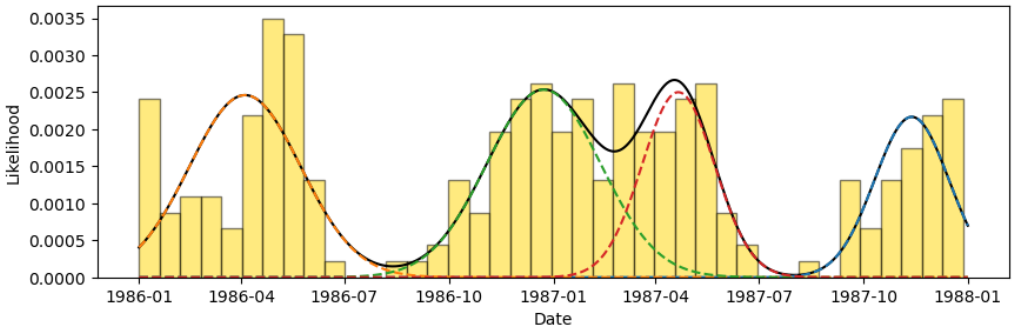

Fig. 50. GMM of the fall–spring mixed cluster distribution of 1986 and 1987 data points.

Fig. 51 depicts the same two years after the split of the transitional seasons was performed. The green points are the days attributed to spring, while the yellow points belong to fall. The dashed lines are the starting dates of the seasons established by the Bureau of Meteorology (BoM) of Australia.

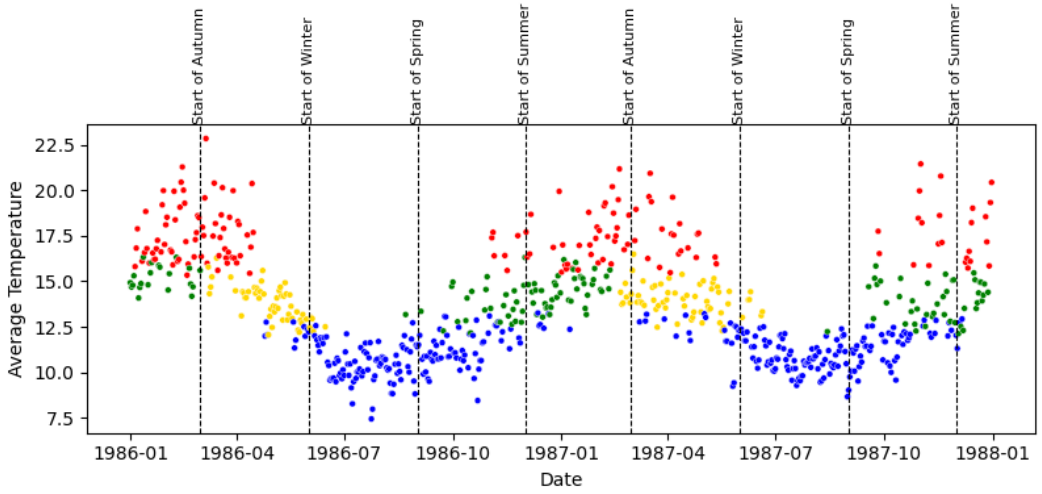

Fig. 51. Seasonal clusters of 1986 and 1987 formed by data points after GMM transformation.

GMM was also used to extract the duration and middle DOY of each season. The mean of each Gaussian distribution provides the seasonal middle DOY, the standard deviation represents the span, and the start of the season was calculated by subtracting the span from the mean DOY.

Fig. 52 shows the trends in fall (yellow), spring (green), winter (blue), and summer (red) seasons. The middle DOY of the fall seasons shows an increasing trend with a slope of 0.28 days/year and the standard deviation remaining relatively stable, with a slope of -0.02 days/year. The fall start days show an increasing trend of 0.31 days/year, meaning a tendency of a later fall start.

In contrast to fall, springs exhibit a contrasting behaviour. The middle DOY has a decreasing trend of -0.79 days/year, while the standard deviation shows a neutral trend of 0.08 days/year. The start day also shows a decreasing trend of -0.95 days/year, suggesting a shift towards earlier spring occurrence.

For winters, the middle DOY displays an insignificant trend with a slope of -0.06 days/year, while the standard deviation decreases with a slope of -0.14 days/year. The start day shows an increasing trend of 0.23 days/year, suggesting that winters are getting shorter over time.

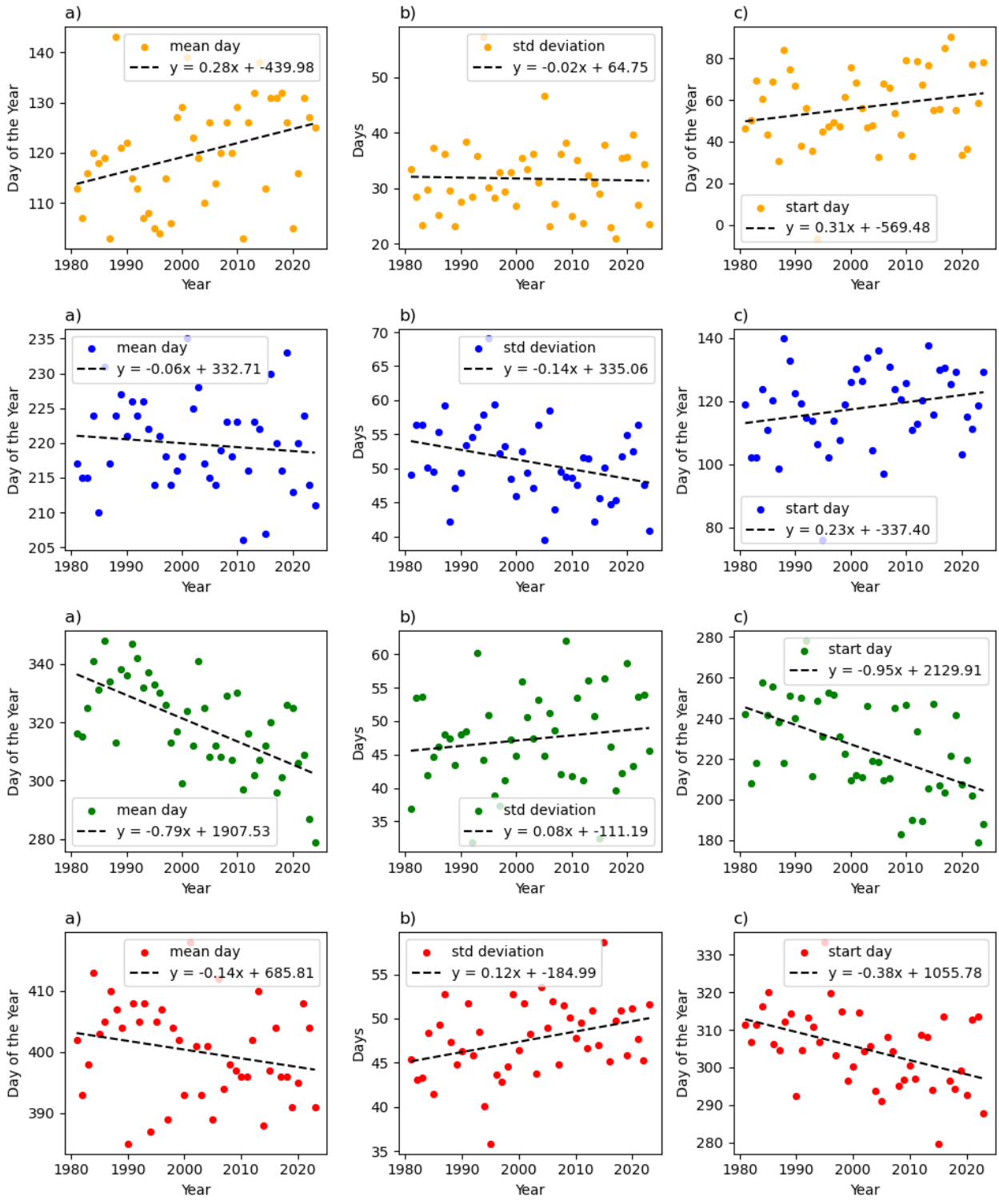

Fig. 52. Trends of seasonal parameters for fall (yellow), spring (green), winter (blue), and summer (red): a) middle DOY, b) standard deviation (span), c) start day.

The summer seasons show a decreasing trend of -0.14 days/year for the middle DOY. The standard deviation increases with a slope of 0.12 days/year, and the start day shows a decreasing trend of -0.38 days/year, implying that summers start earlier and last longer.

Together, these results indicate a consistent pattern in the temporal structure of the seasons: summers expand over time, while winters become shorter. At the same time, springs and falls tend to shift towards earlier and later start DOY, respectively.

5.3.4 Correlation analysis of seasonal dynamics and bioclimatic variables

In this section, we explored how the obtained seasonal characteristics are correlated with the bioclimatic variables. **Table 8** summarizes the significant (p-value > 0.05) correlations, grouping the variables into positive and negative associations for each seasonal parameter.

The fall middle DOY is positively linked to temperature-related variables (BIO1, BIO5, BIO6, BIO10, BIO11), and negatively associated with precipitation during the warmest quarter (BIO18). The fall span is negatively correlated with temperature variability (BIO4, BIO7) and warm-period temperatures (BIO5, BIO10).

The spring middle DOY has the opposite behaviour compared with fall being negatively related to temperature variables (BIO1, BIO6, BIO10, BIO11), but positively correlated with precipitation variables (BIO12, BIO16, BIO19). The spring span is positively correlated with a subset of temperature variables (BIO6, BIO11) and wet-period temperature (BIO8).

The winter middle DOY is positively correlated with BIO9 and BIO19. The winter span is mostly negatively associated with temperature variables (BIO1, BIO6, BIO10, BIO11), while being positively related to precipitation of the coldest month (BIO13). In contrast, the summer middle DOY has no significant association with any of the BIO variables, yet the summer span is positively correlated with BIO1, BIO6, and BIO11.

The start of the seasons follows similar patterns. Fall and winter starts are mainly linked to temperature-related variables. Spring start shows strong negative correlation with cold-related temperature variables (BIO6, BIO11), while summer start presents weaker and mixed relationships, involving both temperature and precipitation BIO variables.

Table 8. Summary of correlations between seasonal parameters and bioclimatic variables

Seasonal Parameter	Bioclimatic variables	
	Positive Correlation	*Negative Correlation*
Fall Mean DOY	BIO1, BIO5, BIO6, BIO9, BIO10, BIO11	BIO18
Fall Span (Variance)		BIO4, BIO5, BIO7, BIO10
Spring Mean DOY	BIO12, BIO16, BIO19	BIO1, BIO6, BIO8, BIO10, BIO11
Spring Span (Variance)	BIO6, BIO8, BIO11	
Winter Mean DOY	BIO9, BIO19	
Winter Span (Variance)	BIO13	BIO1, BIO6, BIO10, BIO11
Summer Span (Variance)	BIO1, BIO6, BIO11	
Fall Start	BIO1, BIO4, BIO5, BIO7, BIO10	BIO18
Spring Start	BIO19	BIO1, BIO6, BIO8, BIO10, BIO11
Winter Start	BIO1, BIO6, BIO10, BIO11	
Summer Start	BIO9, BIO12, BIO16, BIO19	BIO1, BIO8, BIO11

In general, BIO1 appears across all seasons, being linked to the timing of fall and spring, and to the span of winter and summer. A similar pattern is observed for BIO6 and BIO11, which are also related to the spring span.

5.3.5 Module 3 summary

This section describes the climatic conditions and seasonal dynamics on Phillip Island. Bioclimatic variables are introduced as integrated descriptors of environmental conditions, expanding beyond

those traditionally used in little penguin studies. Seasonal patterns were identified revealing shifts in seasonal timing, including earlier springs, later falls, expanding summers, and shortening winters. Finally, correlations between seasonal parameters and bioclimatic variables indicate that temperature-related variables play a dominant role in defining seasonal timing and duration, with precipitation acting as a complementary factor.

5.4 Module 4: Climate and penguin behaviour.

In this section, we studied the relationship between the biological variables and the climatic variables analyzed in Modules 1, 2, and 3. Considering the biological variables extracted from Module 1 (breeding and moulting peaks and span), the relationships analyzed in Module 2, and the seasonal and climatic variables from Module 3, this module targets to examine the interactions between the variables and develop predictive models based on the correlations found. We start with correlation analysis to identify the climatic variables relevant to the biological variables. Subsequently, we construct and evaluate the predictive models. In this section, all computations were performed using anomalies of the climate variables, i.e., deviations from their mean values.

5.4.1 Correlation between MLD and climate

5.4.1.1 Correlation between MLD and climate variables

This section begins by evaluating the associations between MLD and climate variables. **Fig. 53** shows statistically significant correlation coefficients (p-value < 0.05) between the MLD records and monthly climate variables over the 33-year period from 1982 to 2014 These data points are considered as the training set for the predictive model built in sections 5.4.1.5 and 5.4.2.6, avoiding data leakage from the subsequent period from 2015 to 2023 considered as the testing set.

MLD is primarily associated with temperature-related variables, which were the only variables in the experiment showing statistically significant Pearson correlation coefficients – all of which are negative. Consequently, earlier MLDs are statistically related to the years with higher monthly temperatures.

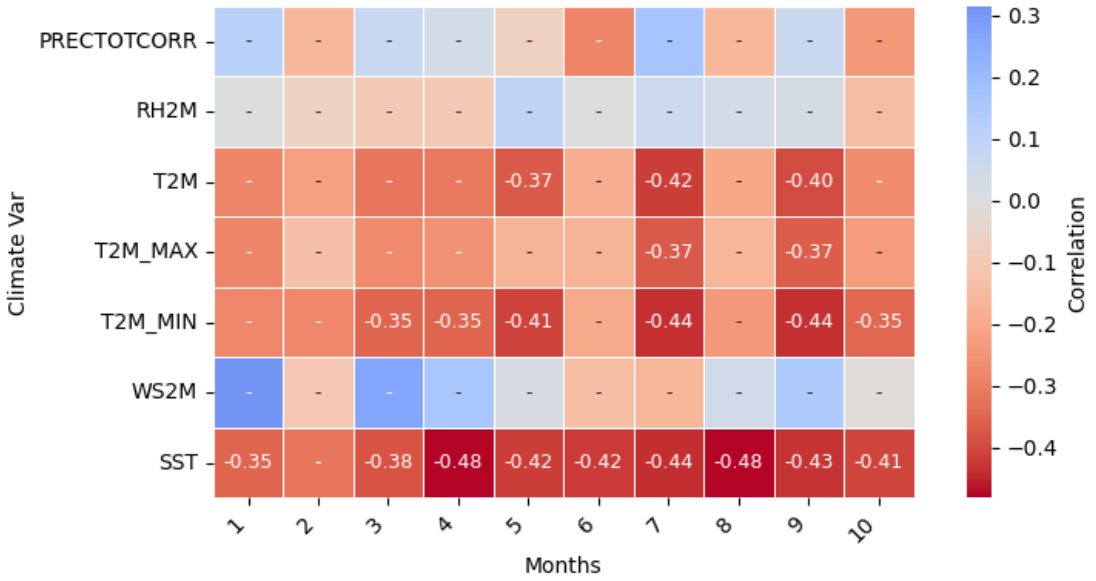

Fig. 53. Heat map of significant correlation coefficients between MLD and climate variables.

A special case is the SST, given that MLD demonstrated statistically significant correlation with SST during the largest number of months (9 out of 10 evaluated months of a year). Moreover, the strongest correlation coefficient in the analysis (r = -0.48) was detected in April and August. Similar observations were reported by Cullen et al. (2009), who developed a predictive model based only on SST.

The next most influential variable is the mean minimum temperature, with which MLD has significant correlations in 6 out of 10 evaluated months, reaching the maximum value of r = -0.44 in July and September.

Months closer to the MLD (section 5.2.1.2) show stronger correlations; therefore, the thermal conditions prior to the breeding season are indicators of MLD status. The corresponding scatter plots of the correlations are shown in Fig 54. These scatterplots use the anomalies of the climate variables.

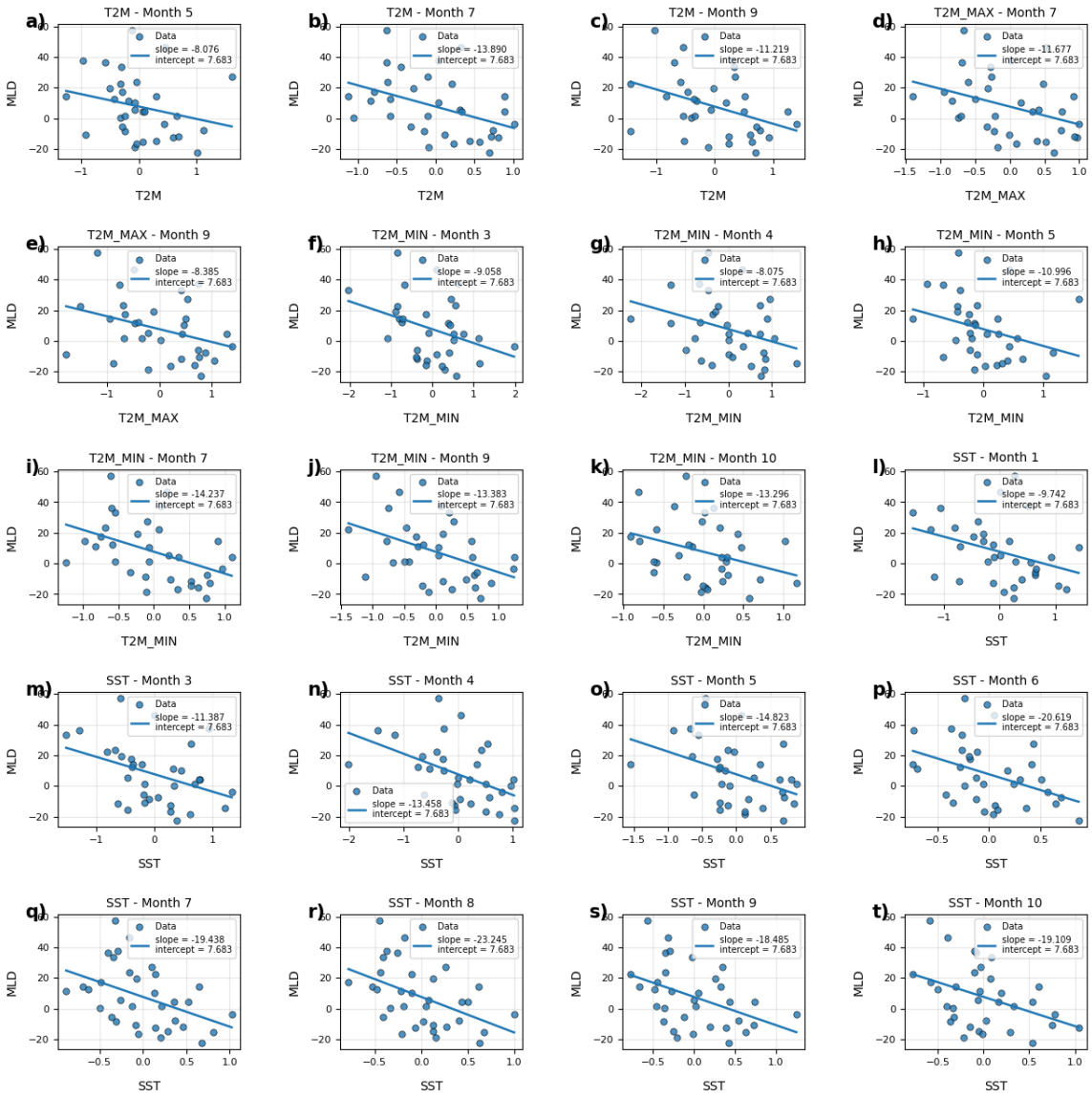

Fig. 54. Scatterplots of correlations between MLD and climate variable anomalies.

5.4.1.2 Correlation between MLD and bioclimatic variables

The correlations between MLD and bioclimatic variables show a pattern similar to that of climatic variables in the previous section (**Fig. 55**). MLD is negatively correlated with temperature-related bioclimatic variables, including annual mean temperature (BIO1), mean temperature of the wettest quarter (BIO8), and mean temperature of the warmest (BIO10) and coldest (BIO11) quarters. Overall, this outcome corresponds with the results found in section 5.4.1.1, where earlier MLD is associated with warmer years. The corresponding scatter plots of the correlations are shown in **Fig 56.** These scatterplots use the anomalies of the bioclimatic variables.

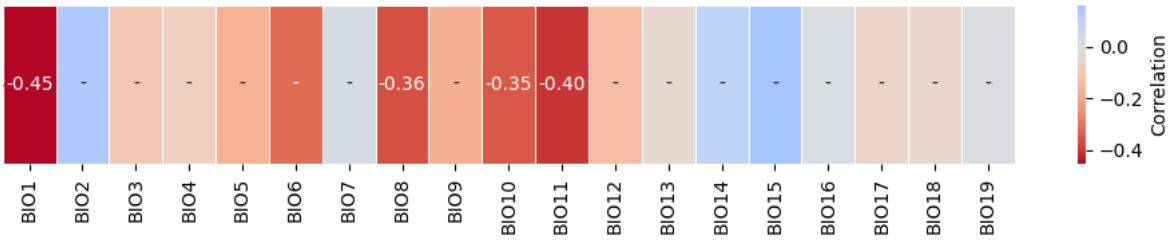

Fig. 55. Heat map of correlation coefficient between MLD and bioclimatic variables.

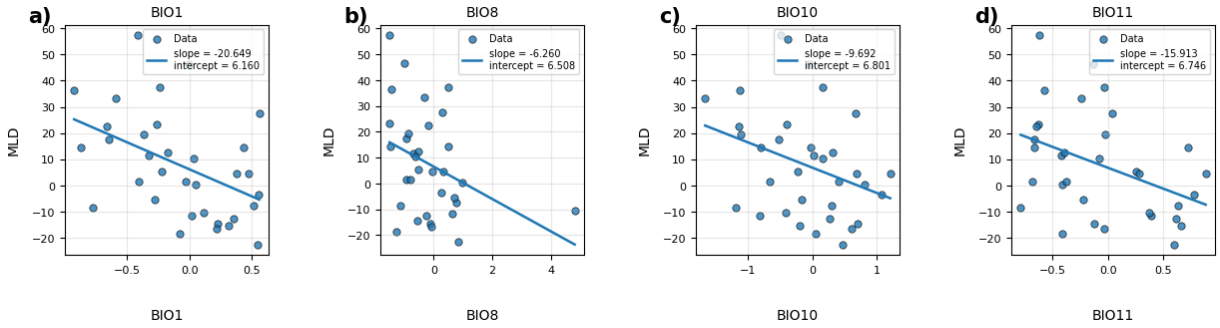

Fig. 56. Scatterplots of correlations between MLD and bioclimatic variable anomalies.

5.4.1.3 Correlation between seasonality characteristics and MLD

Correlation between MLD and seasonality characteristics is particularly strong during fall, winter, and spring (**Fig. 57**). For example, MLD is positively correlated with winter duration, meaning that years with shorter winters display earlier MLDs.

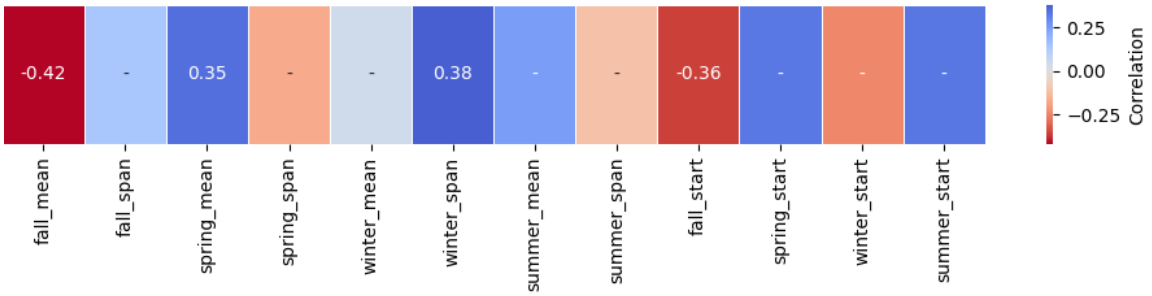

Fig. 57. Heat map of correlation coefficients between MLD and seasonal characteristics.

MLD is also correlated with the shifts in the transitional seasons, particularly, negatively with the fall mean DOY and positively – with the spring mean DOY. That is, earlier MLDs are related to years with shorter winters, later falls, and earlier springs. The corresponding scatter plots of the

correlations are shown in **Fig. 58**. These scatterplots use the anomalies of the seasonal characteristics.

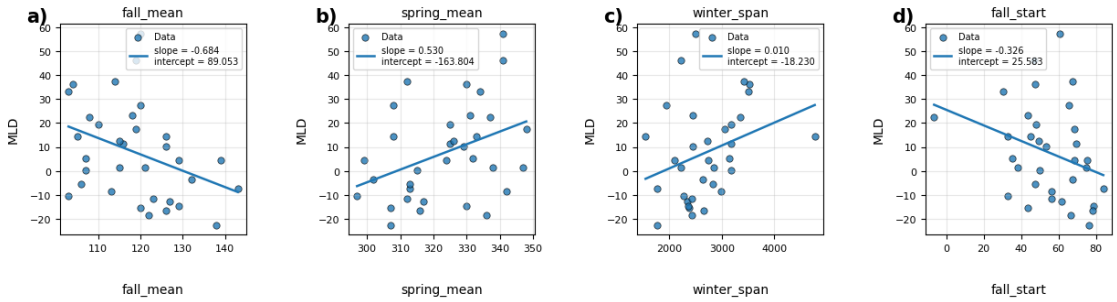

Fig. 58. Scatterplots of correlations between MLD and seasonal characteristic anomalies

5.4.1.4 Correlation between MLD and the composite of significant variables

Considering all pairwise products of bioclimatic, climate, and seasonal variables, we examined their correlations with MLD. In general, the June precipitation appeared in many such products, and was the most frequent variable among the calculated correlations. Although MLD has not been explicitly reported in the literature on little penguins as positively correlated with precipitation, similar relationships have been observed in other bird species (Haller et al., 2025). The remaining variables were temperature-related once from different months.

The combination of *PRECTOTCORR_6 * SST_4* produced one of the strongest correlation coefficients of -0.57 which is higher than those of the individual contributing indicators (not significant, -0.48), suggesting that phenological patterns of MLD in little penguins are strongly associated with the compounded influence of both climatic variables. In general, this type of relationship is typical of different bird species, where temperature and precipitation act as important climate drivers, with complex and non-linear effects on bird phenology, depending on the ecosystem and biotic interactions (Koshelev et al., 2021).

5.4.1.5 Forecasting of the MLD

The variables selected for the predictive model were *PRECTOTCORR_6 * SST_4, breeding peak, SST_1*, *BIO8,* and *SST_5*, as they showed low collinearity and were related to MLD. After performing hyperparameter tuning using grid search, we compared ElasticNet, Ridge, Random Forest, and Gradient Boosting ML models to forecast the MLD (**Fig. 59**). The performance of the tree-based models (i.e., Random Forest, and Gradient Boosting) on the testing dataset was the

lowest in terms of MAE and RMSE. On the contrary, ElasticNet and Ridge demonstrated the highest performance, yielding identical error-metric values.

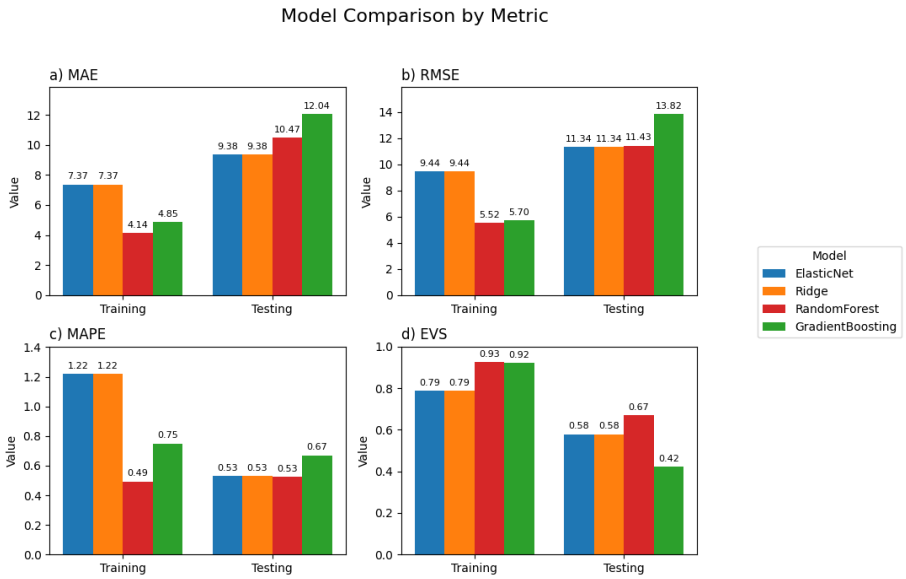

Fig. 59. Training and testing performance of ML models for MLD prediction evaluated using a) MAE, b) RMSE, c) MAPE, and d) explained variance.

In general, all four models showed a good predictive performance compared to the linear regression method reported in the literature (Cullen et al., 2009). **Fig. 60** shows the performance of the Ridge model. The model captures the general timing of MLD, having the largest prediction errors in the years with atypical conditions (i.e., unusually early or late MLD).

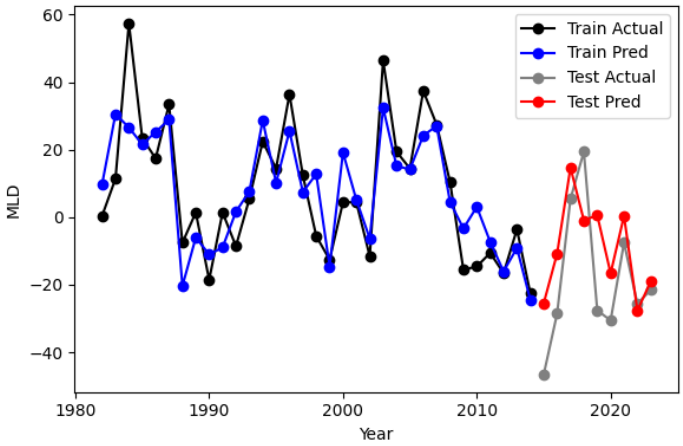

Fig. 60. Ridge model prediction of MLD. The MLD is centered on its mean.

5.4.1.6 Trend analysis

To perform trend analysis, we examined variables with which MLD was correlated. We subtracted the mean from each value, resulting in anomalies of each variable.

SST in April shows an increasing trend in anomalies (p-value = 0.026, slope = 0.019). **Fig. 61** illustrates this behaviour: orange bars represent MLD anomalies, while blue and red bars correspond to negative and positive SST anomalies, respectively, with more negative values in earlier years. The correlation between April SST and MLD is negative, which is reflected in the opposite directions of the anomalies.

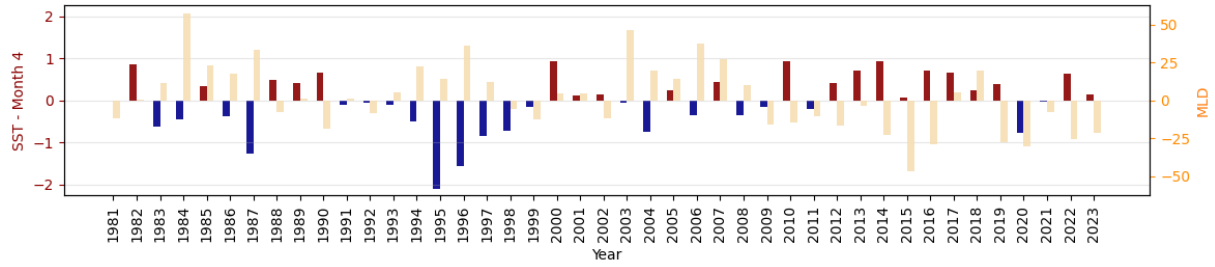

Fig. 61. Annual comparison of SST anomalies in April and MLD anomaly.

Finally, minimum temperature in September shows an increasing trend (p-value = 0.005, slope = 0.020), with more blue bars in earlier years (**Fig. 62**). MLD is negatively correlated with this variable, meaning that their anomalies are generally located in opposite directions.

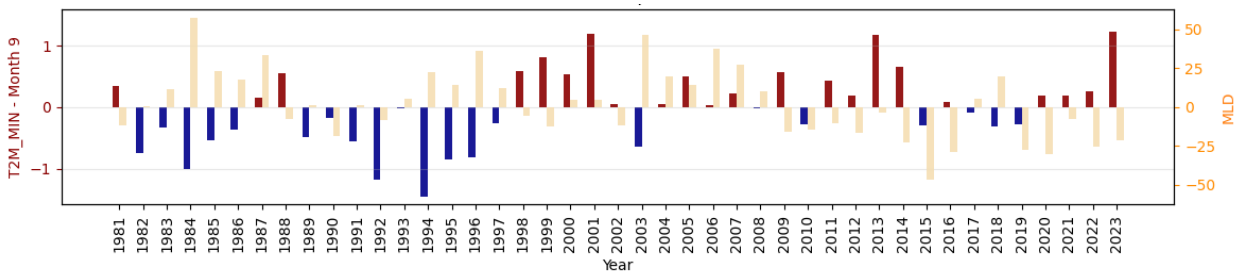

Fig. 62. Annual comparison of minimum temperature anomalies in September and MLD anomaly.

5.4.1.7 ARDL analysis for MLD

As mentioned in section 4.7.1, an ARDL analysis was performed on 42 annual records of MLD over the period from 1982 to 2023. Initially, each climate variable was evaluated as an MLD

predictor. The performance of the model was measured using the Akaike Information Criterion (AIC), where variables producing lower AIC values are better at describing MLD.

Fig. 63a presents top 20 out of 115 studied variables with the lowest AIC. Most of the climate variables at the top of the list are related to temperature during the second semester of the year (July-December). Ten of the thirteen temperature-related variables ranked in the top 20 correspond to months between June and October, preceding the MLD period or overlapping with its first half (please see section 5.2.1.2 for MLD distribution). The rest of the variables is related to precipitation, wind speed, relative humidity, and those characterizing meteorological seasons.

On the next step, a second climate variable was added iteratively to the ARDL model built on the top three variables identified initially. Most of the second variables are non-temperature-related, such as precipitation, wind speed, relative humidity, and characteristics of meteorological seasons. **Fig. 63b-d** shows top 20 variables with the lowest AIC when using T2M_MIN_9, SST_8, and T2M_MIN_7 as the first variable, respectively. PRECTOTCORR_7 was the top variable in all three settings with two variables.

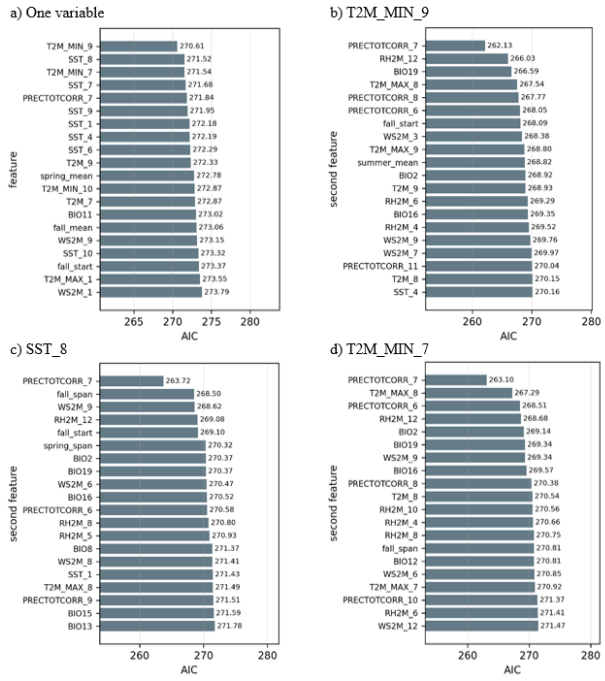

Fig. 63. Top 20 ARDL models with MLD as the target variable: a) models with one predictor; and two-predictor models where the first predictor is b) T2M_MIN_9, c) SST_8, and d) T2M_MIN_7.

These results indicate that the MLD timing is consistently determined by temperature-related variables, better explaining its dynamics compared to other types of variables, with precipitation contributing supplementary information. Particularly, among all the tested models, the combination T2M_MIN_9 + PRECTOTCORR_7 was the one with the lowest AIC.

5.4.2 Correlation between breeding success and climate characteristics

5.4.2.1 Correlation between breeding success and climate variables

In contrast to MLD, breeding success does not show a dominant correlation with the temperature-related variables, having statistically significant correlation coefficients only in 4 out of 12 months (**Fig. 64**). Conversely, variables insignificant for MLD are significant for breeding success, particularly wind speed and relative humidity.

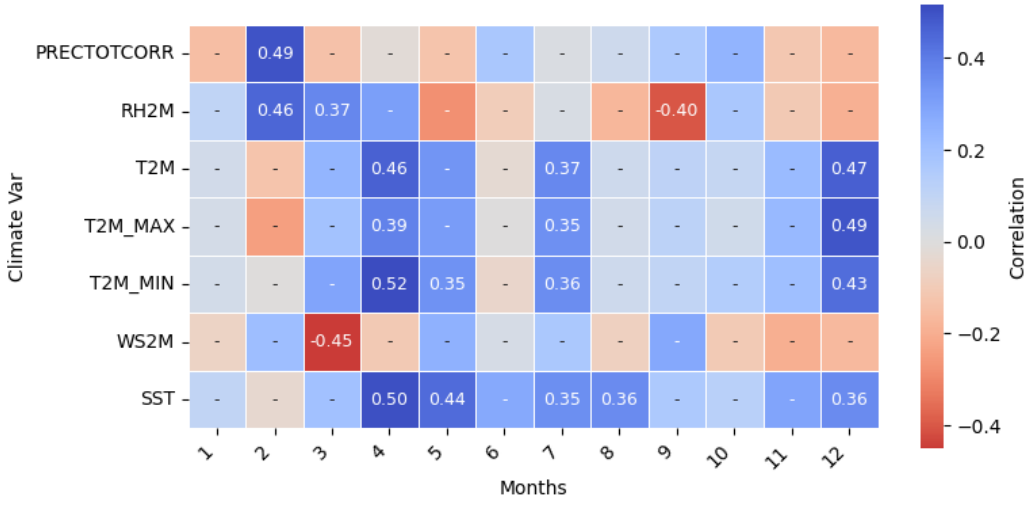

Fig. 64. Heat map of significant correlation coefficients between breeding success and climate variables.

During the breeding period, particularly in summer, precipitation and humidity become relevant. Breeding success is negatively correlated with both variables, meaning that lower breeding success occurs in years with higher accumulated precipitation.

Breeding success is positively correlated with mean and maximum temperatures during the breeding season. Thus, warmer conditions during chick rearing are associated with higher breeding success.

Breeding success also shows positive correlation with climate variables during the months outside the breeding season, including precipitation and humidity early in the year, as well as minimum temperature and SST in April. This result suggests that better breeding performance is linked to higher fall temperatures. The corresponding scatter plots of the correlations are shown in **Fig. 65**. These scatterplots use the anomalies of the climate variables.

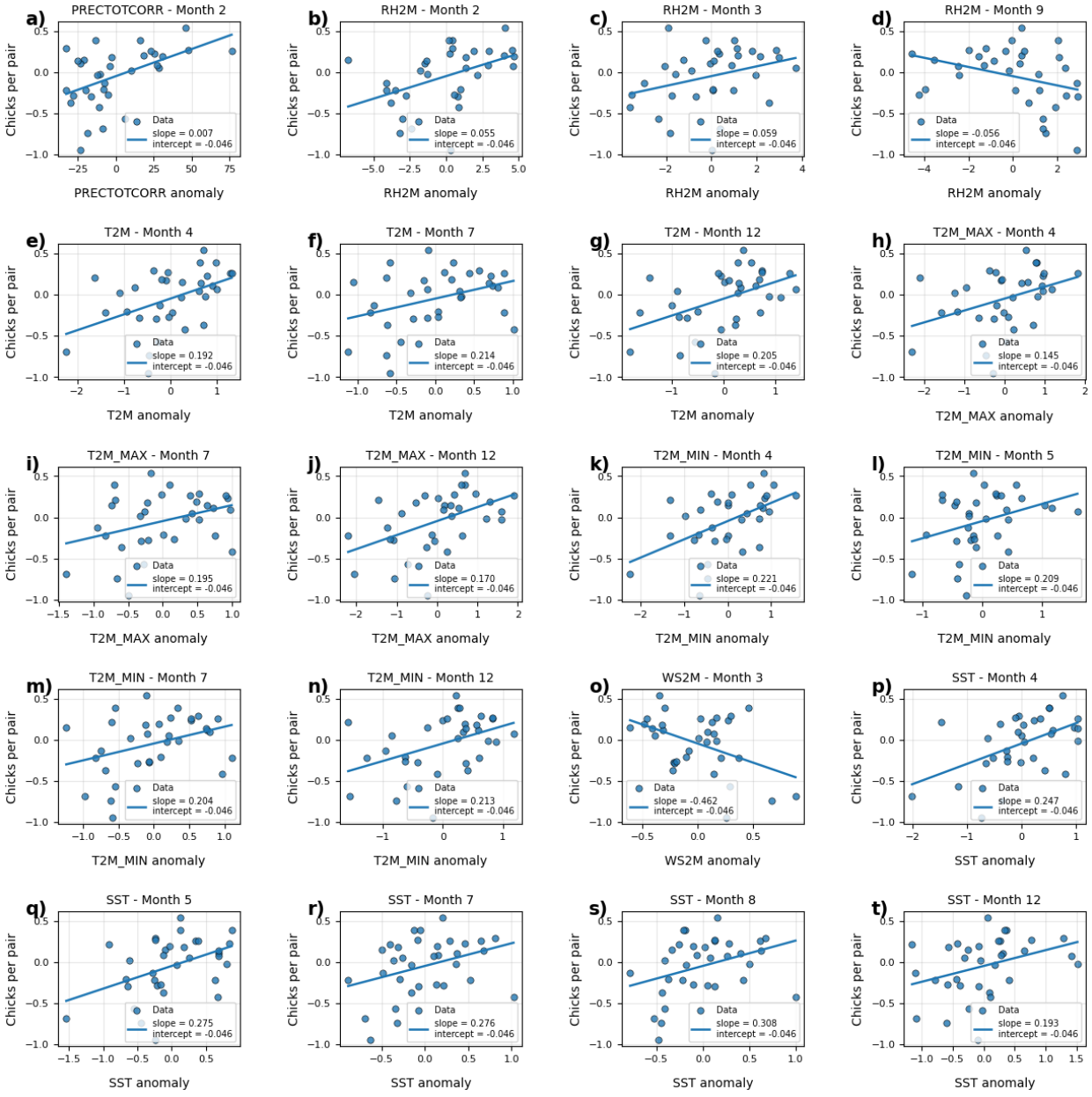

Fig. 65. Scatterplots of correlations between breeding success and climate variable anomalies.

5.4.2.2 Correlation between breeding success and seasonal parameters

The experiment with seasonal parameters showed negative correlations between breeding success and mean summer DOY and winter duration (**Fig. 66**). This result indicates that, historically, better breeding success is associated with shorter winters and earlier summers. The corresponding scatter plots of the correlations are shown in **Fig. 67**. These scatterplots use the anomalies of the seasonal parameters.

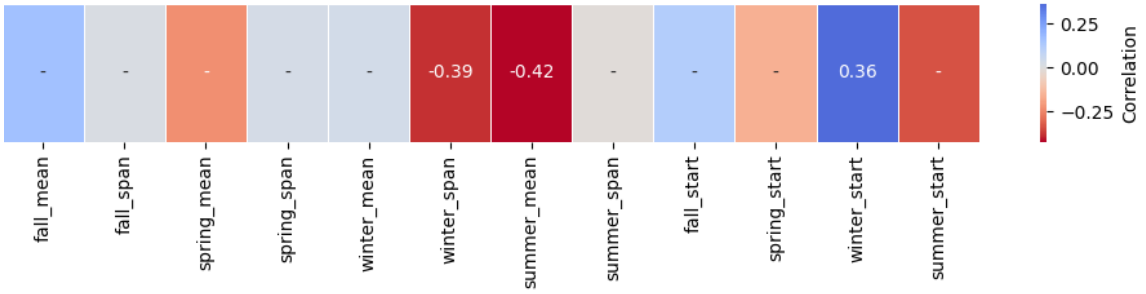

Fig. 66. Heat map of significant correlation coefficients between breeding success and seasonal parameters.

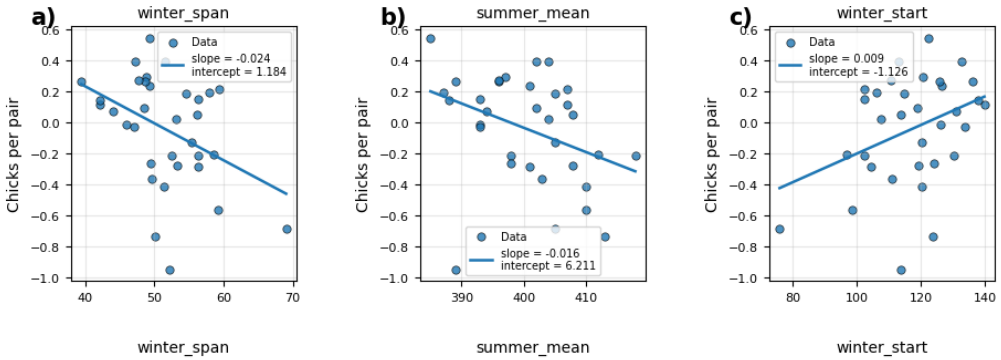

Fig. 67. Scatterplots of correlations between breeding success and seasonal parameter anomalies.

5.4.2.3 Correlation between breeding success and bioclimate variables

No significant correlations were found between breeding success and bioclimate variables.

5.4.2.4 Correlation between breeding success and accumulated variables

Breeding success is significantly correlated (r = 0.47, p-value: 0.005) with the accumulated mean daily temperature calculated over the temporal span of the breeding season peak identified in Module 1. This variable is known as GDD (growing degree days) in the literature and has been shown to be related to the reproduction of other bird species (Boom et al., 2023). **Fig. 68** shows

the annual comparison between these variables, where blue and red bars represent negative and positive GDD anomalies, respectively. The orange bars represent breeding success anomalies.

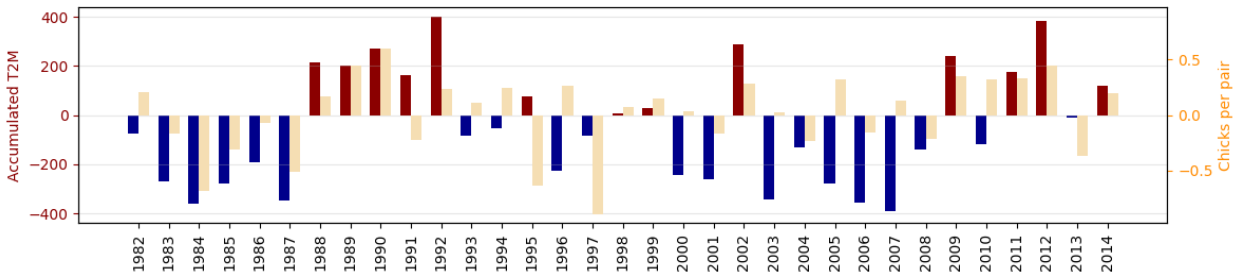

Fig. 68. Comparison of annual anomalies of breeding success vs. accumulated mean daily temperature (GDD).

5.4.2.5 Correlation between breeding success and composite of significant variables

Considering all pairwise products of bioclimatic, climate, and seasonal variables, we examined their correlations with breeding success. The results showed that breeding success was most strongly correlated with the product T2M_MAX_12 * accumulated_T2M ($r = 0.66$). Additionally, these two variables are non-collinear, which avoids redundancies between the predictors.

5.4.2.6 Forecasting of the breeding success

The variables selected for the predictive models (section 4.7.2) were T2M_MAX_12 * accumulated_T2M, RH2M_3, and WS2M_3, as they showed low collinearity and were correlated with breeding success. **Fig. 69** shows the comparison between ElasticNet, Ridge, Random Forest, and Gradient Boosting ML models when forecasting breeding success. The tree-based models (Random Forest, and Gradient Boosting) exhibited the lowest performance on the testing dataset. On the contrary, Ridge had the best performance on all four metrics.

Ridge and ElasticNet models showed good predictive performance compared to the linear regression method reported the literature (Cullen et al., 2009). **Fig. 70** shows the performance of the Ridge model. These results indicate that the model captures the general pattern of breeding success, however producing the largest prediction errors in the years with atypical conditions (e.g., unusually low or high chick survival rate).

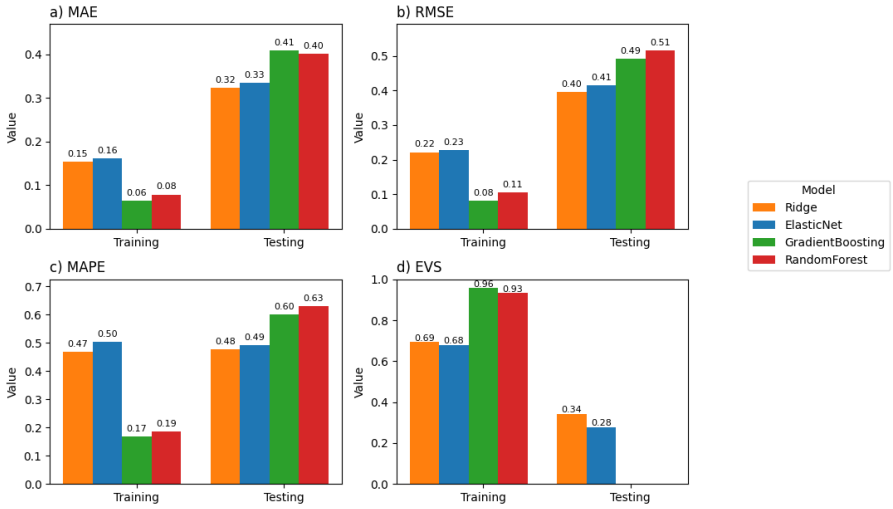

Fig. 69. Training and testing performance of ML models for breeding success prediction evaluated using a) MAE, b) RMSE, c) MAPE, and d) explained variance.

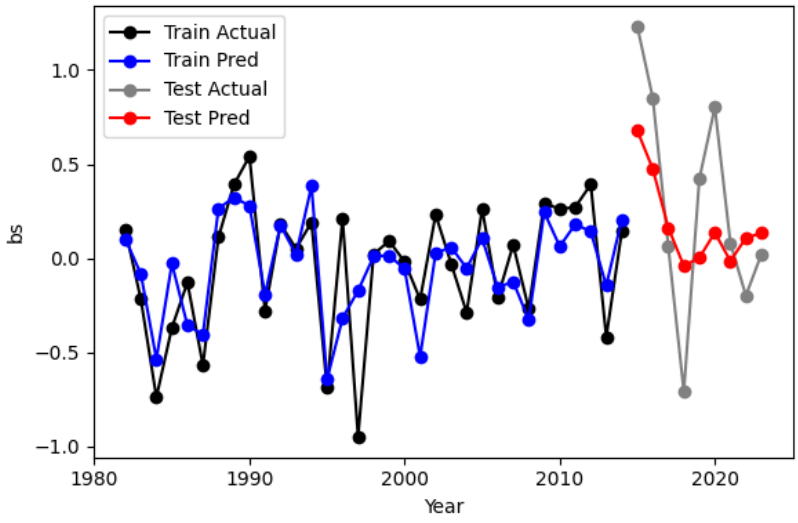

Fig. 70. Ridge model prediction of breeding success (centered on its mean).

5.4.2.7 Trend analysis

As seen in **Fig. 71**, the anomaly values of T2M_MAX exhibit an increasing trend in December (*p*-value = 0.046, slope = 0.0205). Here, orange bars represent breeding success anomalies, while blue and red bars correspond to T2M_MAX anomalies (negative and positive, respectively), with a stronger presence of negative values in earlier years (1982 - 2002).

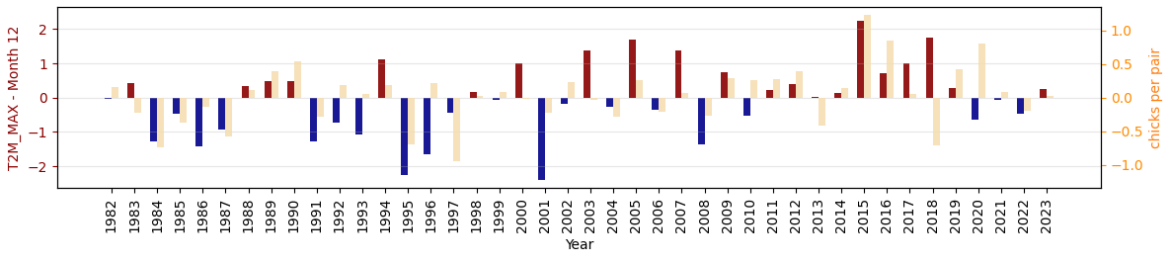

Fig. 71. Comparison of annual anomalies of breeding success vs. December maximum temperature.

5.4.2.8 ARDL analysis for breeding success

ARDL analysis was performed on 42 annual records of breeding success over the period from 1982 to 2023. **Fig. 72a** shows the top 20 climate variables with the lowest AIC when studied individually. Ten out of the twenty variables are temperature-related, yet in contrast to the MLD case, the majority of them is below the top 10 list. Four of the top 20 variables are associated with the meteorological seasons, while those related to precipitation, relative humidity, and wind speed occur twice each.

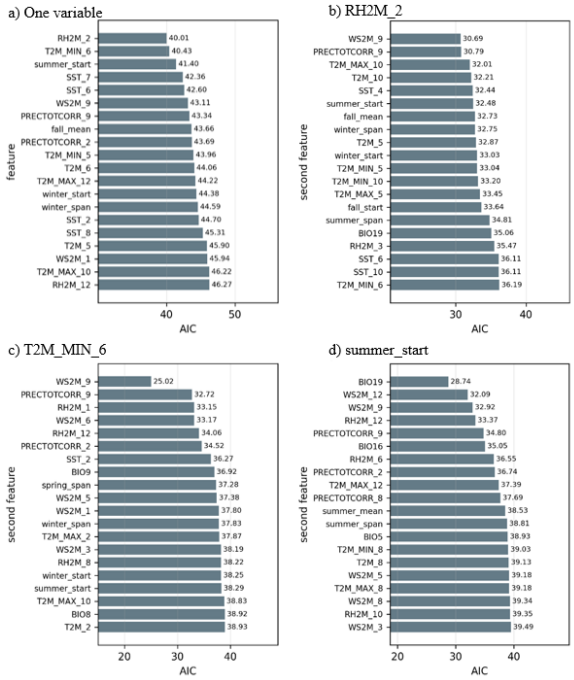

Fig. 72. Top 20 ARDL models with breeding success as the target variable: a) models with one predictor; and two-predictor models where the first predictor is b) T2M_MIN_9, c) SST_8, and d) T2M_MIN_7.

Fig. 72b-d shows the ARDL performance on two variables having RH2M, T2M_MIN_6, and summer_start as the first variable, respectively. T2M_MIN_6 and summer_start show the pairs with the lowest AIC values. All three model settings contain diverse and evenly distributed types of variables, i.e., no dominant type of variable is detected. The two-variable model with T2M_MIN_6 and WS2M_9 had the lowest AIC, suggesting that the combination of temperature and wind speed better explains the variation in breeding success.

5.4.3 Correlation between breeding and moulting characteristics and climate variables

No significant correlation was detected between the breeding and moulting peaks identified in Module 1 and the climate variables. However, significant correlation was found between the span of breeding peaks and the climate variables.

5.4.3.1 Correlation between breeding peak span and climate variables

The breeding peak span shows significant correlations with temperature, precipitation, relative humidity, and wind speed (**Fig. 73**). No correlation was found with SST, indicating a dominance of terrestrial climate variables. In general, the span depends on temperature variables at the beginning, middle, and end of the year, as well as during the MLD month.

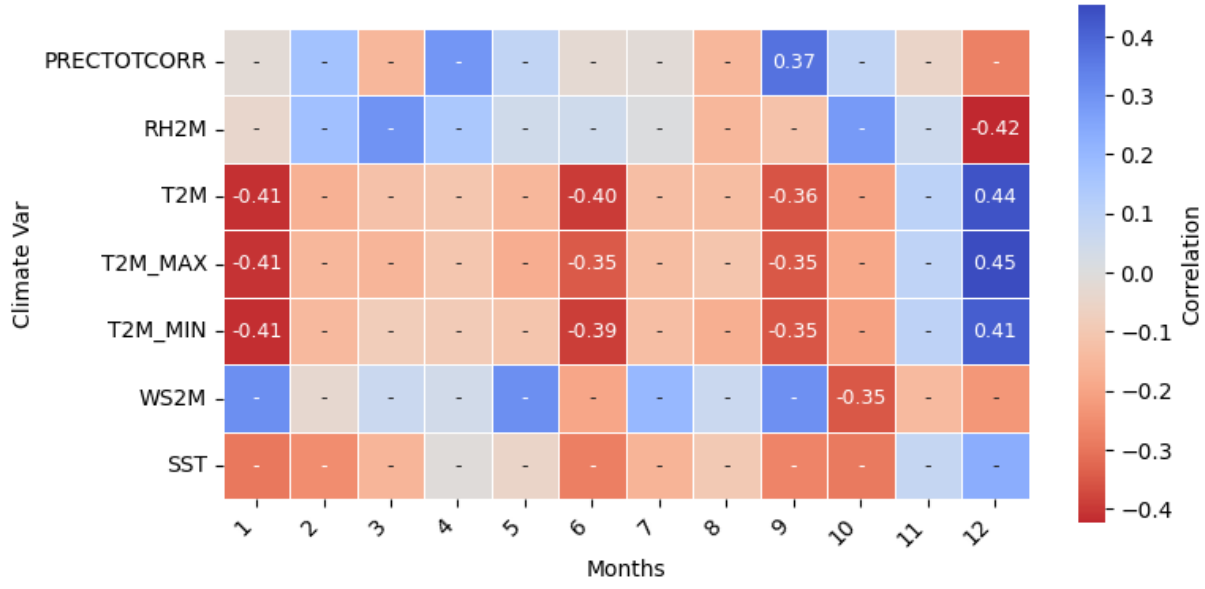

Fig. 73. Heat map of significant correlation coefficients between breeding peak span and climatic variables.

Since breeding peaks occur at the end of the year (section 5.2.1.1), their span is associated with temperature and humidity during this period. In particular, correlation with temperature is positive, meaning that broader activity peaks are associated with warmer Decembers.

The breeding peak span is also positively correlated with precipitation in September, suggesting that stronger rainfall during the MLD period may extend the breeding peak span. In contrast, temperatures in the months preceding the breeding season have a negative correlation, so wider peaks occur in the years when these months are colder. The corresponding scatter plots of the correlations are shown in **Fig. 74**. These scatterplots use the anomalies of the climate variables.

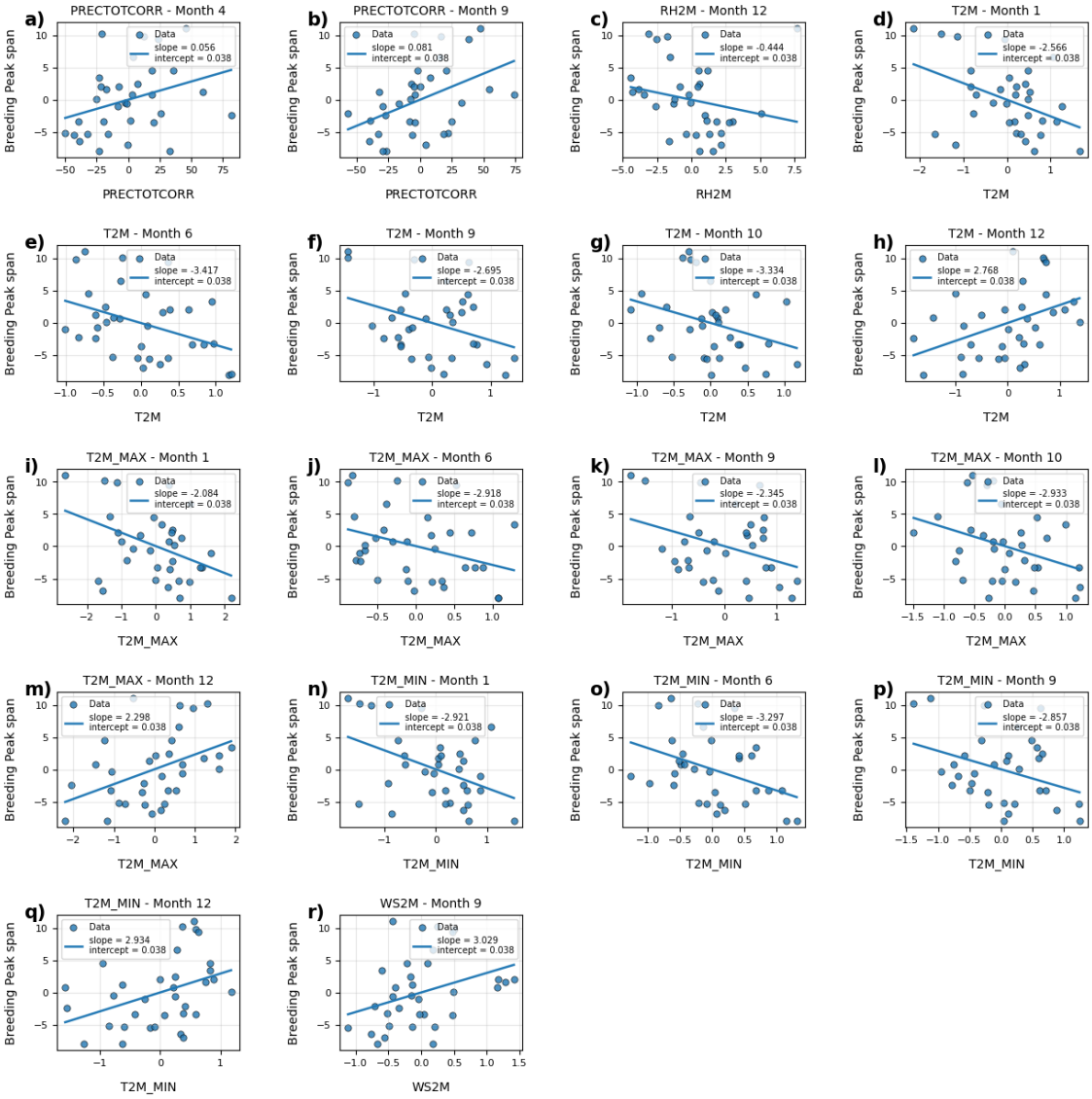

Fig. 74. Scatterplots of correlations between breeding peak span and climate variable anomalies.

5.4.3.2 Correlation between breeding peak span and bioclimatic variables

Among the bioclimatic variables, breeding peak span is only correlated with BIO10 (Mean Temperature of Warmest Quarter) (**Fig. 75**). Its negative correlation suggests that shorter breeding peak span is associated with higher temperatures. The corresponding scatter plot of the correlation is shown in **Fig. 76**. These scatterplots use the anomalies of the BIO10.

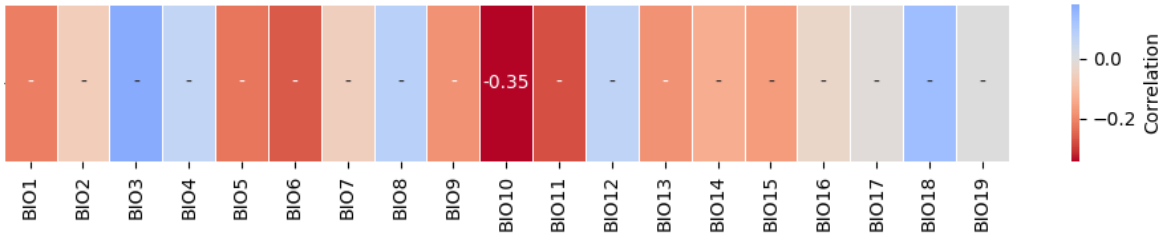

Fig. 75. Correlation between breeding peak span and bioclimatic variables.

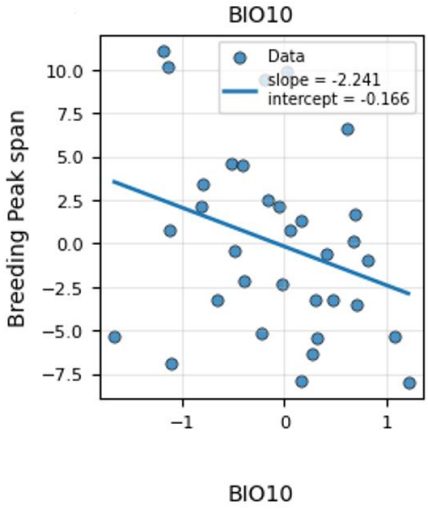

Fig. 76. Scatterplot of correlation between breeding peak span and BIO10 anomalies**.**

5.4.3.3 Correlation between breeding peak span and seasonal parameters

No significant correlation was found between the breeding peak span and seasonal parameters.

5.4.3.4 Correlation between breeding peak span and the composite of significant variables

Considering all pairwise products of bioclimatic, climate, and seasonal variables, we examined their correlations with breeding peak span. Within the set of pairwise products, breeding success exhibited the highest correlation value (r = 0.61) with PRECTOTCORR_9 * T2M_MAX_12. Among the top 20 most correlated composites, precipitation and humidity appear repeatedly,

usually accompanied by a temperature variable at the beginning (January) or end of the year (December), indicating a multivariable pattern in which the relevant climatic signal arises from the combination of factors rather than from individual contributors.

5.4.3.5 Forecasting of the breeding peak span

The variables selected for the predictive models (section 4.7.2) were T2M_MAX_12, T2M_MIN_6, PRECTOTCORR_9 * T2M_MAX_12, WS2M_10, and BIO10, as they showed low collinearity and were correlated with the breeding peak span. **Fig. 77** shows the comparison between ElasticNet, Ridge, Random Forest, and Gradient Boosting ML models in forecasting breeding success. Gradient Boosting demonstrated the best performance among the models by having the lowest errors and the highest explained variance. **Fig. 78** shows the forecast by the Gradient Boosting model.

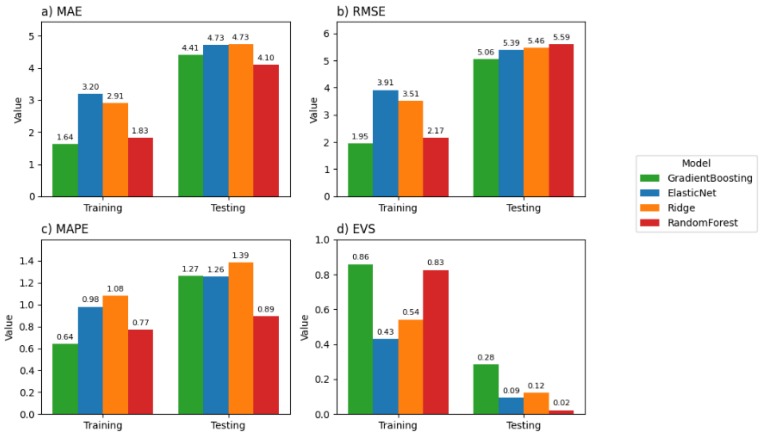

Fig. 77. Training and testing performance of ML models in breeding peak span prediction evaluated using a) MAE, b) RMSE, c) MAPE, and d) explained variance.

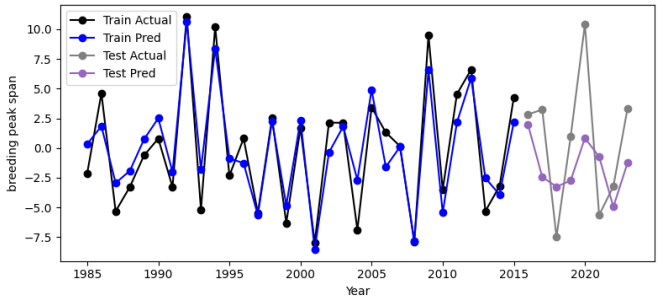

Fig. 78. Gradient boosting model prediction of breeding peak span (centered on its mean).

5.4.3.6 ARDL analysis for breeding peak span

ARDL analysis was performed on 42 annual records of breeding success over the period from 1982 to 2023. **Fig. 79a** shows the top 20 climate variables with the lowest AIC when evaluated as a single-variable model. Most of the variables are temperature-related, although precipitation, wind speed, relative humidity, and meteorological season variables are also observed in the list. In the top three, we have T2M_MIN_6, PRECTOTCORR_4, and BIO14, which are temperature-related and two precipitation-related variables.

Fig. 79b-d shows the performance of the ARDL models with two variables. Wind speed, relative humidity, and temperature are among the most frequent variables. The model using PRECTOTCORR_4 and T2M_MAX_12 had the lowest AIC. This shows that the breeding peak span is mainly associated with precipitation and temperature.

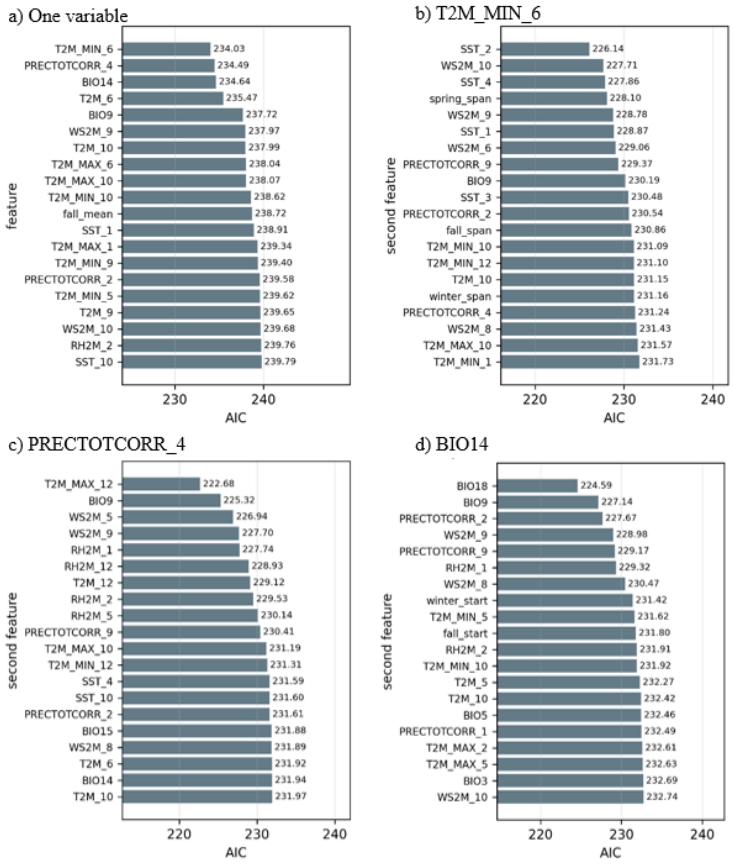

Fig. 79. Top 20 ARDL models with breeding peak span as the target variable: a) models with one predictor; and two-predictor models where the first predictor is b) T2M_MIN_6, c) PRECTOTCORR_4, and d) BIO1

5.4.4 Module 4 summary

In this module, we observed that temperature is the main driver of MLD, with warmer conditions leading to earlier laying dates. In contrast, breeding success depends on a broader set of factors, with temperature and accumulated heat showing positive correlations. In prediction exercise, the MLD model explained a substantial proportion of the variance, while the breeding success model had a moderate performance when compared with the MLD model.

5.5 CEFABC2 management system

The results obtained in the Modules 1-4 provide a useful background for building a tool, Computational Ethology Framework for Animal Behaviour study under Climate Change (CEFABC2), that informs the parks management on the state of the little penguin colony on Phillip Island. CEFABC2 inputs the biological and climatic variables presented in the Modules, generating the forecasts of MLD, breeding success, or penguin count values.

Fig. 80 shows an example of the tool application to predict the nightly penguin count, MLD, breeding success and breeding peak. The user chooses the temporal granularity for the penguin count forecast (all temporal granularities – daily, weekly, monthly – are included in this case). The number of months prior to the forecast start can also be selected (1 month in this case). The number of SSA components for the time series forecasting is specified in the "# SSA components" panel (5 components in this example).

Two models can be selected for the breeding success forecast: the model developed in Section 5.4.2.6 (Model 1) and an adapted version of this model (Model 2) that uses variable values over the period preceding the breeding season, avoiding variables, such as T2M_MAX_12, that require a complete year of data collection before the models can be run. Model 2 uses the variables *PRECTOTCORR_2 * SST_4*, *predicted breeding peak*, and *RH2M_3* (**Fig. 81**).

Additionally, the MLD is predicted using the model described in section 5.4.1.5 and is shown as a vertical dashed red line in the dashboard (**Fig. 80**). Finally, the breeding peak is predicted using the GP model described in section 5.2.2 and is displayed as a vertical dashed purple line in the dashboard.

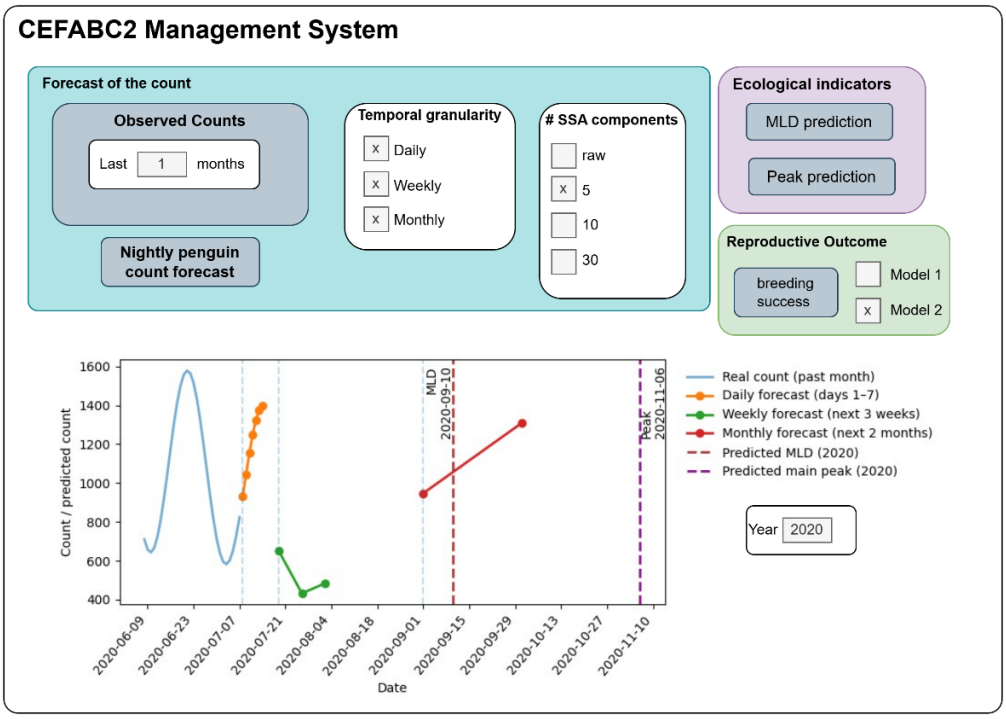

Fig. 80. Dashboard of CEFABC2 management system.

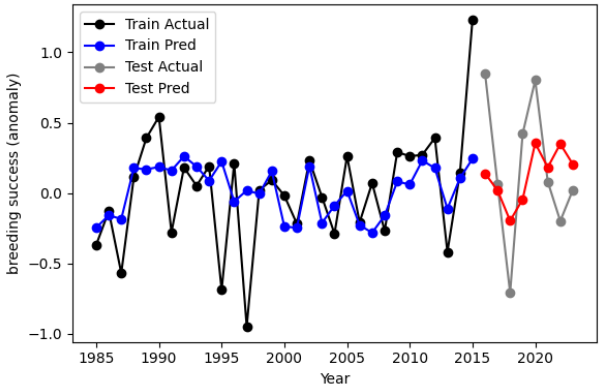

Fig. 81. Adapted Ridge model prediction of breeding success (centered on its mean).

Chapter 6: Conclusions

This thesis proposed a computational ethology framework on animal behaviour study under climate change (CEFABC2), developed to study and monitor the dynamics of animal behaviour in a colony. This thesis was focused on the little penguin colony located on Phillip Island in Australia, forecasting and studying the biological variables: nightly penguin count, mean egg-laying date (MLD), and breeding success. We leveraged climate variables, bioclimate, and meteorological season variables to add explanation to the biological variables. The contributions made by the presented work are enumerated below:

- **New biological behaviour variables proposition**. This thesis proposed the breeding and moulting peak variables, which included the timing and span of these phenomena. These biological variables are indicators of the calendar periods when the nightly penguin count is high. Additionally, it was shown that the breeding peak is a biological variable significantly associated with the mean egg-laying date (MLD) and breeding success, two representative indicators used in literature to study the species (Saraux & Chiaradia, 2022).
- **Climate assessment**. We implemented unsupervised ML models, such as K-means, Principal Component Analysis (PCA), and Gaussian Mixture Model (GMM), to extract the duration, start and middle date of the meteorological seasons in Phillip Island. Using Sen's slope estimator and Mann-Kendall tests, the trends of the meteorological seasons, bioclimatic, and climatic variables were evaluated.
- **Forecast of the biological variables**. ML, DL, and statistical models were used to forecast the biological variables. Gaussian processes (GP) were used to forecast the moulting and breeding peak. Ridge regression was used to forecast the MLD and breeding success. Structural time series (STS), ARIMA, LSTM, GP, LightGBM, and Ridge models were used to forecast the nightly penguin count under multiple temporal granularities. Additionally, Singular Spectrum Analysis (SSA) components were used for the time series reconstruction.
- **Novel framework that integrates biological and climate-related variables - CEFABC2**. This thesis provides four modules to analyze biological variables of the little penguin colony and climate variables of the region. The first two modules study the

biological variables, the third module analyzes the regional climate, and the fourth module integrates both variables.

- **CEFABC2 management system**. As a result of the analyses performed in the modules, this thesis proposes a system that provides information about the state and forecast of each biological variable.

Future studies could focus on the expansion of the framework to other little penguin colonies, subject to the data availability. Multiple colonies can be found in Australia and New Zealand, such as Rabbit Island, Kanowna Island, and Motunau Island (Taylor, 2000; Hoskins et al., 2008). Accordingly, CEFABC2 may be expanded to other colonies that register all these biological variables. This would allow the comparison between the colonies and would provide a broader perspective about little penguins. Additionally, this expansion may also be performed to other penguin species, such as Adélie, Chinstrap, and Gentoo (Bost & Jouventin, 1991; Phillips, 2021; Watanabe et al., 2020).

Future studies would also explore the set of hybrid and ensemble ML models. Combining two or more models with the goal of creating a more robust model is a common practice used to leverage the strengths of each model (Buyo et al., 2025). Combining the models or adding different models, such as Temporal Convolutional Networks (TCN), may improve the nightly penguin count forecast.

A limitation of this thesis is the use of other biological variables. In literature, we found variables such as adult body mass (Robinson et al., 2005), chick weight (Chiaradia & Nisbet, 2006), and second clutches (Rowe et al., 2020). Adding these variables would extend the applicability of the CEFABC2 management system.

In general, CEFABC2 permits the study of behavioural dynamics in a colony. This framework shows the species adaptations to environmental changes in the region they inhabit. The obtained insights from the proposed framework may inform decisions made by animal protection entities in charge of the colony or the organization of eco-touristic activities.

References

1. Adam, T., Griffiths, C. A., Leos-Barajas, V., Meese, E. N., Lowe, C. G., Blackwell, P. G., … & Langrock, R. (2019). Joint modelling of multi-scale animal movement data using hierarchical hidden Markov models. Methods in Ecology and Evolution, 10(9), 1536–1550.
2. Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. Electronics, 9(8), 1295.
3. Alcock, J. (2009). Animal behavior: An evolutionary approach. Sinauer associates.
4. Amiri, M., Tarkesh, M., Jafari, R., & Jetschke, G. (2020). Bioclimatic variables from precipitation and temperature records vs. remote sensing-based bioclimatic variables: Which side can perform better in species distribution modeling? Ecological Informatics, 57, 101060.
5. Anderson, D. J., & Perona, P. (2014). Toward a science of computational ethology. *Neuron*, *84*(1), 18-31.
6. Arablouei, R., Currie, L., Kusy, B., Ingham, A., Greenwood, P. L., & Bishop-Hurley, G. (2021). In-situ classification of cattle behavior using accelerometry data. Computers and Electronics in Agriculture, 183, 106045.
7. Arndt, R. J., O'Keefe, J. M., Mitchell, W. A., Holmes, J. B., & Lima, S. L. (2018). Do predators influence the behaviour of temperate-zone bats? An analysis of competing models of roost emergence times. Animal Behaviour, 145, 161–170.
8. Aulsebrook, A. E., Jacques-Hamilton, R., & Kempenaers, B. (2024). Quantifying mating behaviour using accelerometry and machine learning: challenges and opportunities. Animal Behaviour, 207, 55–76.
9. Bartholomew, D. J. (1971). Time Series Analysis Forecasting and Control.
10. Berger-Tal, O., & Saltz, D. (2016). Indirect behavioral indicators and their uses in conservation and management. In O. Berger-Tal & D. Saltz (Eds.), Conservation behavior: Applying behavioral ecology to wildlife conservation and management (pp. 352–375). Cambridge University Press.
11. Berger-Tal, O., Polak, T., Oron, A., Lubin, Y., Kotler, B. P., & Saltz, D. (2011). Integrating animal behavior and conservation biology: a conceptual framework. Behavioral Ecology, 22(2), 236-239.
12. Berman, G. J. (2018). Measuring behavior across scales. BMC biology, 16(1), 23.
13. Bilodeau, S. M., Schwartz, A. W., Xu, B., Paúl Pauca, V., & Silman, M. R. (2022). A low-cost, long-term underwater camera trap network coupled with deep residual learning image analysis. PLOS ONE, 17(2), e0263377.
14. Bist, R. B., Yang, X., Subedi, S., & Chai, L. (2023). Mislaying behavior detection in cage-free hens with deep learning technologies. Poultry Science, 102(7), 102729.
15. Boom, M. P., Schreven, K. H., Buitendijk, N. H., Moonen, S., Nolet, B. A., Eichhorn, G., … & Lameris, T. K. (2023). Earlier springs increase goose breeding propensity and nesting

success at Arctic but not at temperate latitudes. Journal of Animal Ecology, 92(12), 2399–2411.
16. Bost, C. A., & Jouventin, P. (1991). The breeding performance of the Gentoo Penguin Pygoscelis papua at the northern edge of its range. Ibis, 133(1), 14–25.
17. Buyo, N., Sheikh-Akbari, A., & Saleem, F. (2025). An ensemble approach to predict a sustainable energy plan for London households. Sustainability, 17(2), 500.
18. Cannell, B. L., Chambers, L. E., Wooller, R. D., & Bradley, J. S. (2012). Poorer breeding by little penguins near Perth, Western Australia is correlated with above average sea surface temperatures and a stronger Leeuwin Current. Marine and Freshwater Research, 63(10), 914–925.
19. Carroll, G., Harcourt, R., Pitcher, B. J., Slip, D., & Jonsen, I. (2018). Recent prey capture experience and dynamic habitat quality mediate short-term foraging site fidelity in a seabird. Proceedings of the Royal Society B: Biological Sciences, 285(1883).
20. Carroll, G., Slip, D., Jonsen, I., & Harcourt, R. (2014). Supervised accelerometry analysis can identify prey capture by penguins at sea. Journal of Experimental Biology, 217(24), 4295–4302.
21. Cavallo, C., Chiaradia, A., Deagle, B. E., Hays, G. C., Jarman, S., McInnes, J. C., … & Reina, R. D. (2020). Quantifying prey availability using the foraging plasticity of a marine predator, the little penguin. Functional Ecology, 34(8), 1626–1639.
22. Chambers, L. E., Dann, P., Cannell, B., & Woehler, E. J. (2014). Climate as a driver of phenological change in southern seabirds. International Journal of Biometeorology, 58(4), 603–612.
23. Chambers, L. E., Devney, C. A., Congdon, B. C., Dunlop, N., Woehler, E. J., & Dann, P. (2011). Observed and predicted effects of climate on Australian seabirds. Emu, 111(3), 235–251.
24. Chen, C. (2014). The citespace manual. College of Computing and Informatics, 1(1), 1–84.
25. Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 785–794).
26. Chiaradia, A. F., & Kerry, K. R. (1999). Daily nest attendance and breeding performance in the little penguin Eudyptula minor at Phillip Island, Australia. Marine Ornithology, 27, 13–20.
27. Chiaradia, A., & Nisbet, I. C. (2006). Plasticity in parental provisioning and chick growth in little penguins Eudyptula minor in years of high and low breeding success. ARDEA-WAGENINGEN-, 94(2), 257.
28. Chiaradia, A., McBride, J., Murray, T., & Dann, P. (2007). Effect of fog on the arrival time of little penguins Eudyptula minor: a clue for visual orientation? Journal of Ornithology, 148(2), 229–233.

29. Chilvers, B. L., Morgan, K. M., Finlayson, G., & Sievwright, K. A. (2015). Diving behaviour of wildlife impacted by an oil spill: A clean-up and rehabilitation success? Marine Pollution Bulletin, 100(1), 128–133.
30. Christin, S., Hervet, É., & Lecomte, N. (2019). Applications for deep learning in ecology. Methods in Ecology and Evolution, 10(10), 1632–1644.
31. Cobo Martín, M. J. (2012). SciMAT: herramienta software para el análisis de la evolución del conocimiento científico. Propuesta de una metodología de evaluación. Granada: Universidad de Granada.
32. Cucari, N., Tutore, I., Montera, R., & Profita, S. (2023). A bibliometric performance analysis of publication productivity in the corporate social responsibility field: Outcomes of SciVal analytics. Corporate Social Responsibility and Environmental Management, 30(1), 1–16.
33. Couzin, I. D., & Heins, C. (2023). Emerging technologies for behavioral research in changing environments. Trends in Ecology & Evolution, 38(4), 346-354.
34. Cui, Z., Qing, X., Chai, H., Yang, S., Zhu, Y., & Wang, F. (2021). Real-time rainfall-runoff prediction using light gradient boosting machine coupled with singular spectrum analysis. Journal of Hydrology, 603, 127124.
35. Cullen, J. M., Chambers, L. E., Coutin, P. C., & Dann, P. (2009). Predicting onset and success of breeding in little penguins Eudyptula minor from ocean temperatures. Marine Ecology Progress Series, 378, 269–278.
36. Dann, P., & Chambers, L. (2013). Ecological effects of climate change on little penguins Eudyptula minor and the potential economic impact on tourism. Climate Research, 58, 67–79.
37. Dann, P., Norman, F. I., Cullen, J. M., Neira, F. J., & Chiaradia, A. (2000). Mortality and breeding failure of little penguins, Eudyptula minor, in Victoria, 1995–96, following a widespread mortality of pilchard, Sardinops sagax. Marine and Freshwater Research, 51(4), 355–362.
38. Davies, N. B., Krebs, J. R., & West, S. A. (2012). An introduction to behavioural ecology. John Wiley & Sons.
39. Derviş, H. (2019). Bibliometric analysis using bibliometrix an R package. Journal of Scientometric Research, 8(3), 156–160.
40. Donthu, N., Kumar, S., Mukherjee, D., Pandey, N., & Lim, W. M. (2021). How to conduct a bibliometric analysis: An overview and guidelines. Journal of business research, 133, 285–296.
41. Du, L., Shi, Y., Chen, Y., Li, F., Zhou, P., Duan, L., & Qian, Y. (2025). A dual mixture-of-experts framework for multi-view K-means clustering with view balance and regional sparsity. Information Fusion, 103449.
42. Dupont, F., Marcoux, M., Hussey, N., & Auger-Méthé, M. (2025). Improved order selection method for hidden Markov models: a case study with movement data. Methods in Ecology and Evolution, 16(6), 1215–1227.

43. Dupuis, B., Kato, A., Joly, N., Saraux, C., Ropert-Coudert, Y., Chiaradia, A., & Chimienti, M. (2023). COVID-related anthropause highlights the impact of marine traffic but not of tourism on breeding little penguins. Biological Conservation, 287, 110323.
44. Fatima, S. S. W., & Rahimi, A. (2024). A review of time-series forecasting algorithms for industrial manufacturing systems. Machines, 12(6), 380.
45. Fazzari, E., Romano, D., Falchi, F., & Stefanini, C. (2025). Animal behavior analysis methods using deep learning: A survey. Expert Systems With Applications, 289, 128330.
46. Forester, J. D., Im, H. K., & Rathouz, P. J. (2009). Accounting for animal movement in estimation of resource selection functions: sampling and data analysis. Ecology, 90(12), 3554–3565.
47. Fukasawa, K., & Higashide, D. (2025). Mechanistic home range capture–recapture models for the estimation of population density and landscape connectivity. Ecology, 106(2), e70046.
48. Ganendran, L. B., Sidhu, L. A., Catchpole, E. A., Chambers, L. E., & Dann, P. (2016). Effects of ambient air temperature, humidity and rainfall on annual survival of adult little penguins Eudyptula minor in southeastern Australia. International Journal of Biometeorology, 60(8), 1237–1245.
49. Gillis, W. F., & Datta, S. R. (2017). Knowing where the nose is. BMC Biology, 15(1), 42.
50. Golyandina, N., & Korobeynikov, A. (2014). Basic singular spectrum analysis and forecasting with R. Computational Statistics & Data Analysis, 71, 934–954.
51. Goodwin, N. L., Nilsson, S. R., & Golden, S. A. (2020). Rage Against the Machine: Advancing the study of aggression ethology via machine learning. Psychopharmacology, 237(9), 2569–2588.
52. Greenacre, M., Groenen, P. J., Hastie, T., d'Enza, A. I., Markos, A., & Tuzhilina, E. (2022). Principal component analysis. Nature Reviews Methods Primers, 2(1), 100.
53. Grove, M., Borg, J. M., & Polack, F. (2020, July). Coloured noise time series as appropriate models for environmental variation in artificial evolutionary systems. In Artificial Life Conference Proceedings 32 (pp. 292–299). One Rogers Street, Cambridge, MA 02142–1209, USA journals-info@ mit. edu: MIT Press.
54. Haller, A., Ciolete, T., Lopes, L. E., & Cunha, F. (2025). Weather conditions influence breeding initiation of an intra-tropical migrant bird, Sporophila lineola (Lined Seedeater). Ornithology, 142(3), ukaf017.
55. Harvey, A. C., & Harvey, A. (1990). Forecasting, structural time series models and the Kalman filter.
56. Hashimoto, Y., Zin, T. T., Tin, P., Kobayashi, I., & Hama, H. (2025). Generating Accurate Activity Patterns for Cattle Farm Management Using MCMC Simulation of Multiple-Sensor Data System. Sensors, 25(21), 6781.
57. Hassani, H. (2007). Singular spectrum analysis: methodology and comparison.

58. He, M., Jiang, S., Ren, L., Cui, H., Qin, T., Du, S., … & Xu, C. Y. (2024). Streamflow prediction in ungauged catchments through use of catchment classification and deep learning. Journal of Hydrology, 639, 131638.
59. Hayden, B. Y., Park, H. S., & Zimmermann, J. (2022). Automated pose estimation in primates. American journal of primatology, 84(10), e23348.
60. Hoskins, A. J., Dann, P., Ropert-Coudert, Y., Kato, A., Chiaradia, A., Costa, D. P., & Arnould, J. P. (2008). Foraging behaviour and habitat selection of the little penguin Eudyptula minor during early chick rearing in Bass Strait, Australia. Marine Ecology Progress Series, 366, 293–303.
61. Hounslow, J. L., Fossette, S., Chong, W., Bali, R., Tucker, A. D., Whiting, S. D., & Gleiss, A. C. (2023). Behaviour-specific spatiotemporal patterns of habitat use by sea turtles revealed using biologging and supervised machine learning. Journal of applied ecology, 60(9), 1828-1840.
62. Hsieh, W. W. (2009). Machine learning methods in the environmental sciences: Neural networks and kernels. Cambridge University Press.
63. Hu, C., Elbroch, M., Meyer, T., Pozdnyakov, V., & Yan, J. (2021). Moving-resting process with measurement error in animal movement modeling. Methods in Ecology and Evolution, 12(11), 2221–2233.
64. Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice. OTexts.
65. Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. Journal of statistical software, 27, 1–22.
66. Immelmann, K. (2012). Introduction to ethology. Springer Science & Business Media.
67. Iqbal, U., Riaz, M. Z. B., Zhao, J., Barthelemy, J., & Perez, P. (2023). Drones for flood monitoring, mapping and detection: A bibliometric review. Drones, 7(1), 32.
68. Ives, A. R., Dennis, B., Cottingham, K. L., & Carpenter, S. R. (2003). Estimating community stability and ecological interactions from time-series data. Ecological Monographs, 73(2), 301–330.
69. Jacobson, O. T., Barrett, B. J., Perry, S. E., Finerty, G. E., Tiedeman, K. M., & Crofoot, M. C. (2024). A new approach to geostatistical synthesis of historical records reveals capuchin spatial responses to climate and demographic change. Ecology Letters, 27(5), e14443.
70. Jisha, G. (2024). Enhanced weather prediction with feature engineered, time series cross validated ridge regression model. In 2024 Control Instrumentation System Conference (CISCON) (pp. 1–6). IEEE.
71. Johnson, B., & Colombelli-Négrel, D. (2021). Breeding success in Southern Australian little penguins is negatively correlated with high wind speeds and sea surface temperatures. The Condor, 123(1), duaa062.
72. Kabra, M., Robie, A. A., Rivera-Alba, M., Branson, S., & Branson, K. (2013). JAABA: Interactive machine learning for automatic annotation of animal behavior. Nature Methods, 10(1), 64–67.

73. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., … & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30.
74. Kleanthous, N., Hussain, A., Mason, A., Sneddon, J., Shaw, A., Fergus, P., … & Al-Jumeily, D. (2018, November). Machine learning techniques for classification of livestock behavior. In International Conference on Neural Information Processing (pp. 304–315). Cham: Springer International Publishing.
75. Klomp, N. I., Meathrel, C. E., Wienecke, B. C., & Wooller, R. D. (1991). Surface nesting by little penguins on Penguin Island, Western Australia. Emu-Austral Ornithology, 91(3), 190–193.
76. Kojevnikov, D. (2021). The validity of backtesting for evaluation of autoregressive time series predictions (Doctoral dissertation, Ph. D. thesis, Tilburg University).
77. Koshelev, O. I., Koshelev, V. O., Fedushko, M. P., & Zhukov, O. V. (2021). Annual course of temperature and precipitation as proximal predictors of birds' responses to climatic changes on the species and community level. Folia Oecologica, 48(2), 118–135.
78. Kripfganz, S., & Schneider, D. C. (2023). ardl: Estimating autoregressive distributed lag and equilibrium correction models. The Stata Journal, 23(4), 983–1019.
79. Leos-Barajas, V., Gangloff, E. J., Adam, T., Langrock, R., Van Beest, F. M., Nabe-Nielsen, J., & Morales, J. M. (2017). Multi-scale modeling of animal movement and general behavior data using hidden Markov models with hierarchical structures. Journal of Agricultural, Biological and Environmental Statistics, 22(3), 232–248.
80. Levitis, D. A., Lidicker Jr, W. Z., & Freund, G. (2009). Behavioural biologists do not agree on what constitutes behaviour. Animal behaviour, 78(1), 103-110.
81. Li, M., & Bolker, B. M. (2017). Incorporating periodic variability in hidden Markov models for animal movement. Movement Ecology, 5(1), 1.
82. Lian, H., Liu, B., & Li, P. (2021). A fuel sales forecast method based on variational Bayesian structural time series. Journal of High Speed Networks, 27(1), 45–66.
83. Martin, J. R. (2004). A portrait of locomotor behaviour in Drosophila determined by a video-tracking paradigm. Behavioural processes, 67(2), 207–219.
84. Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning. Nature Neuroscience, 21(9), 1281–1289.
85. McClintock, B. T., Langrock, R., Gimenez, O., Cam, E., Borchers, D. L., Glennie, R., & Patterson, T. A. (2020). Uncovering ecological state dynamics with hidden Markov models. Ecology letters, 23(12), 1878–1903.
86. McKenzie-Smith, G. C., Wolf, S. W., Ayroles, J. F., & Shaevitz, J. W. (2025). Capturing continuous, long timescale behavioral changes in Drosophila melanogaster postural data. PLoS computational biology, 21(2), e1012753.
87. McLachlan, G. J., Lee, S. X., & Rathnayake, S. I. (2019). Finite mixture models. Annual Review of Statistics and Its Application, 6(1), 355–378.

88. Meyer, X., MacIntosh, A. J., Chiaradia, A., Kato, A., Ramírez, F., Sueur, C., & Ropert-Coudert, Y. (2020). Oceanic thermal structure mediates dive sequences in a foraging seabird. Ecology and Evolution, 10(13), 6610–6622.
89. Mickelson, M. J., Dann, P., & Cullen, J. M. (1991). Sea Temperature in Bass Strait and Breeding Success of the Little Penguin Eudyptula minorai Phillip Island, South-eastern Australia. Emu-Austral Ornithology, 91(5), 355–368.
90. Nicholls, D. B. (1918). An introduction to the study of penguins on the Nobbies, Phillip Island, Western Port, Victoria. Emu-Austral Ornithology, 17(3), 118–144.
91. Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., & Clune, J. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proceedings of the National Academy of Sciences, 115(25), E5716–E5725.
92. Northrup, J. M., Avrin, A., Anderson Jr, C. R., Brown, E., & Wittemyer, G. (2019). On-animal acoustic monitoring provides insight to ungulate foraging behavior. Journal of Mammalogy, 100(5), 1479–1489.
93. Odo, A., Muns, R., Boyle, L., & Kyriazakis, I. (2023). Video analysis using deep learning for automated quantification of ear biting in pigs. IEEE Access, 11, 59744–59757.
94. Otsuka, R., Yoshimura, N., Tanigaki, K., Koyama, S., Mizutani, Y., Yoda, K., & Maekawa, T. (2024). Exploring deep learning techniques for wild animal behaviour classification using animal-borne accelerometers. Methods in Ecology and Evolution, 15(4), 716–731.
95. Pereira, T. D., Aldarondo, D. E., Willmore, L., Kislin, M., Wang, S. S. H., Murthy, M., & Shaevitz, J. W. (2019). Fast animal pose estimation using deep neural networks. Nature methods, 16(1), 117-125.
96. Phillips, J. A., Fayet, A. L., Guilford, T., Manco, F., Warwick-Evans, V., & Trathan, P. (2021). Foraging conditions for breeding penguins improve with distance from colony and progression of the breeding season at the South Orkney Islands. Movement Ecology, 9(1), 22.
97. Piper, W. H., Dahl, K., Kebeh, M., & Piper, A. J. (2023). Site familiarity makes an enduring impact on breeding success. Animal Behaviour, 205, 79–87.
98. Rasmussen, C. E., & Nickisch, H. (2010). Gaussian processes for machine learning (GPML) toolbox. The Journal of Machine Learning Research, 11, 3011–3015.
99. Rausch, T. M., Albrecht, T., & Baier, D. (2022). Beyond the beaten paths of forecasting call center arrivals: on the use of dynamic harmonic regression with predictor variables. Journal of Business Economics, 92(4), 675–706.
100. Ravichandran, S., & Prajneshu, P. (2005). On the modelling of seasonal fluctuations in fish landings using structural time-series approach.
101. Reilly, P. N., & Cullen, J. M. (1981). The little penguin Eudyptula minor in Victoria, II: Breeding. Emu-Austral Ornithology, 81(1), 1–19.
102. Reynolds, R. W., Banzon, V. F., & NOAA CDR Program. (2008). NOAA Optimum Interpolation 1/4 Degree Daily Sea Surface Temperature (OISST) Analysis, Version 2 [sea

surface temperature (SST), 1981–2025]. NOAA National Centers for Environmental Information.
103. Riaz, M. U., O'Grady, L., McAloon, C. G., Logan, F., & Gormley, I. C. (2025). Comparison of machine learning and validation methods for high-dimensional accelerometer data to detect foot lesions in dairy cattle. PLOS ONE, 20(6), e0325927.
104. Robinson, S., Chiaradia, A., & Hindell, M. A. (2005). The effect of body condition on the timing and success of breeding in Little Penguins Eudyptula minor. Ibis, 147(3), 483–489.
105. Rodríguez, A., Chiaradia, A., Wasiak, P., Renwick, L., & Dann, P. (2016). Waddling on the dark side: Ambient light affects attendance behavior of little penguins. Journal of Biological Rhythms, 31(2), 194–204.
106. Rodríguez, A., Holmberg, R., Dann, P., & Chiaradia, A. (2018). Penguin colony attendance under artificial lights for ecotourism. Journal of Experimental Zoology Part A: Ecological and Integrative Physiology, 329(8–9), 457–464.
107. Rollinson, C. R., Finley, A. O., Alexander, M. R., Banerjee, S., Dixon Hamil, K. A., Koenig, L. E., … & Zipkin, E. F. (2021). Working across space and time: nonstationarity in ecological research and application. Frontiers in Ecology and the Environment, 19(1), 66–72.
108. Romero-Ferrero, F., Bergomi, M. G., Hinz, R. C., Heras, F. J., & De Polavieja, G. G. (2019). Idtracker. ai: tracking all individuals in small or large collectives of unmarked animals. Nature methods, 16(2), 179-182.
109. Rowe, L. K., Weir, J. S., & Judkins, A. G. (2020). Breeding of little penguins (Eudyptula minor), including multiple brooding, at South Bay, Kaikōura, New Zealand, 2006–2017. Notornis, 67(2), 451–458.
110. Salton, M., Saraux, C., Dann, P., & Chiaradia, A. (2015). Carry-over body mass effect from winter to breeding in a resident seabird, the little penguin. Royal Society Open Science, 2(1).
111. Saraux, C., & Chiaradia, A. (2022). Age-related breeding success in little penguins: a result of selection and ontogenetic changes in foraging and phenology. Ecological Monographs, 92(1), e01495.
112. Saraux, C., Chiaradia, A., Salton, M., Dann, P., & Viblanc, V. A. (2016). Negative effects of wind speed on individual foraging performance and breeding success in little penguins. Ecological Monographs, 86(1), 61–77.
113. Saraux, C., Robinson-Laverick, S. M., Le Maho, Y., Ropert-Coudert, Y., & Chiaradia, A. (2011). Plasticity in foraging strategies of inshore birds: how little penguins maintain body reserves while feeding offspring. Ecology, 92(10), 1909–1916.
114. Scharf, H. R., & Buderman, F. E. (2020). Animal movement models for multiple individuals. Wiley Interdisciplinary Reviews: Computational Statistics, 12(6), e1506.
115. Sievwright, K. A., Battley, P. F., McConnell, H. M., Chilvers, B. L., & Morgan, K. J. (2019). Post-release breeding success of oil-rehabilitated and non-rehabilitated little blue

penguins, Eudyptula minor, following the M/V Rena oil spill, New Zealand. Marine Pollution Bulletin, 149, 110553.
116. Sparks, A. H. (2018). nasapower: a NASA POWER global meteorology, surface solar energy and climatology data client for R. Journal of Open Source Software, 3(30), 1035.
117. Stahel, C. D., & Nicol, S. C. (1988). Ventilation and oxygen extraction in the little penguin (Eudyptula minor), at different temperatures in air and water. Respiration Physiology, 71(3), 387–398.
118. Stoye, F. V., Hoyer, A., & Langrock, R. (2025). Autoregressive Hidden Markov Models for High-Resolution Animal Movement Data: FV Stoye et al. Journal of Agricultural, Biological and Environmental Statistics, 1–16.
119. Stratigakos, A., Bachoumis, A., Vita, V., & Zafiropoulos, E. (2021). Short-term net load forecasting with singular spectrum analysis and LSTM neural networks. Energies, 14(14), 4107.
120. Suleman, M., & Khaiter, P. A. (2025). A hybrid framework for regional climate seasonality study and trend analysis. Environmental Modelling & Software, 188, 106429.
121. Sur, M., Hall, J. C., Brandt, J., Astell, M., Poessel, S. A., & Katzner, T. E. (2023). Supervised versus unsupervised approaches to classification of accelerometry data. Ecology and Evolution, 13(5), e10035.
122. Sutherland, D. R., & Dann, P. (2012). Improving the accuracy of population size estimates for burrow-nesting seabirds. Ibis, 154(3), 488–498.
123. Sutton, G. J., Hoskins, A. J., Berlincourt, M., & Arnould, J. P. (2017). Departure time influences foraging associations in little penguins. PLoS One, 12(8), e0182734.
124. Taieb, S. B., Bontempi, G., Atiya, A. F., & Sorjamaa, A. (2012). A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. Expert systems with applications, 39(8), 7067–7083.
125. Takagi, T., Tanaka, Y., Sasano, E., Kanda, K., & Sakai, Y. (2025). Data assimilation reveals behavioral dynamics of sea cucumbers as a model for slow-moving benthic animals. Scientific Reports.
126. Taylor, G. A. (2000). Action plan for seabird conservation in New Zealand. Part A: Threatened seabirds. Department of Conservation.
127. Thorson, J. T. (2022). Development and simulation testing for a new approach to density dependence in species distribution models. ICES Journal of Marine Science, 79(1), 117–128.
128. Trendafilova, I. (2021). Singular spectrum analysis for the investigation of structural vibrations. Engineering Structures, 242, 112531.
129. Umargono, E., Suseno, J. E., & Gunawan, S. V. (2020, October). K-means clustering optimization using the elbow method and early centroid determination based on mean and median formula. In The 2nd International Seminar on Science and Technology (ISSTEC 2019) (pp. 121–129). Atlantis Press.

130. Urbaniak, R., Xie, M., & Mackevicius, E. (2024). Linking cognitive strategy, neural mechanism, and movement statistics in group foraging behaviors. Scientific Reports, 14(1), 21770.
131. Valletta, J. J., Torney, C., Kings, M., Thornton, A., & Madden, J. (2017). Applications of machine learning in animal behaviour studies. Animal Behaviour, 124, 203–220.
132. Van Eck, N. J., & Waltman, L. (2016, September). VOSviewer manual.
133. Wang, F., Shao, W., Yu, H., Kan, G., He, X., Zhang, D., … & Wang, G. (2020). Re-evaluation of the power of the Mann-Kendall test for detecting monotonic trends in hydrometeorological time series. Frontiers in Earth Science, 8, 14.
134. Wang, Q., Zhou, J., & Wang, G. (2024). Adaptive filtering over complex networks in intricate environments. IEEE Transactions on Circuits and Systems I: Regular Papers, 71(12), 6044–6057.
135. Wang, S., Li, K., Liu, Y., Chen, Y., & Tang, X. (2024). VIX constant maturity futures trading strategy: A walk-forward machine learning study. PLOS ONE, 19(4), e0302289.
136. Waqas, M., & Humphries, U. W. (2024). A critical review of RNN and LSTM variants in hydrological time series predictions. MethodsX, 13, 102946.
137. Watanabe, Y. Y., Ito, K., Kokubun, N., & Takahashi, A. (2020). Foraging behavior links sea ice to breeding success in Antarctic penguins. Sci Adv 6: eaba4828.
138. Weavers, B. W. (1991). Seasonal foraging ranges and travels at sea of Little Penguins Eudyptula minor, determined by radiotracking. Emu-Austral Ornithology, 91(5), 302–317.
139. Wong, B. B., & Candolin, U. (2015). Behavioral responses to changing environments. Behavioral Ecology, 26(3), 665-673.
140. Wu, S., Li, Z., Li, H., Bian, S., Ouyang, H., Yao, Y., … & He, Y. (2023). Detection of periodic signals with time-varying coefficients from CMONOC stations in China by singular spectrum analysis. IEEE Transactions on Geoscience and Remote Sensing, 61, 1–12.
141. Yu, H., Deng, J., Nathan, R., Kröschel, M., Pekarsky, S., Li, G., & Klaassen, M. (2021). An evaluation of machine learning classifiers for next-generation, continuous-ethogram smart trackers. Movement Ecology, 9(1), 15.
142. Zamanzadeh Talkhouncheh, M., Davoodi, S., Wood, D. A., Mehrad, M., Rukavishnikov, V. S., & Bakhshi, R. (2024). Robust machine learning predictive models for real-time determination of confined compressive strength of rock using mudlogging data. Rock Mechanics and Rock Engineering, 57(9), 6881–6907.
143. Zhang, C., Zhang, Y., Cucuringu, M., & Qian, Z. (2024). Volatility forecasting with machine learning and intraday commonality. Journal of Financial Econometrics, 22(2), 492–530.
144. Zhang, F., Li, Y., Li, X., Zhang, B., Xue, C., & Wang, Y. (2024). Comparison of ARIMA and Bayesian structural time series models for predicting the trend of syphilis epidemic in Jiangsu Province. Infection and Drug Resistance, 5745–5754.

145. Zhang, M., Han, Y., Wang, C., Yang, P., Wang, C., & Zalhaf, A. S. (2024). Ultra-short-term photovoltaic power prediction based on similar day clustering and temporal convolutional network with bidirectional long short-term memory model: A case study using DKASC data. Applied Energy, 375, 124085.
146. Zimmer, I., Ropert-Coudert, Y., Poulin, N., Kato, A., & Chiaradia, A. (2011). Evaluating the relative importance of intrinsic and extrinsic factors on the foraging activity of top predators: a case study on female little penguins. Marine Biology, 158(4), 715–722.

Appendices

Appendix 1. Python Code for Singular Spectrum Analysis of Nightly Penguin Counts

```
# ================================================================================
# Code for Singular Spectrum Analysis (SSA)
# Nightly penguin count decomposition and reconstruction
# ================================================================================
# ================================================================================
# 1. Import libraries
# ================================================================================
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import seaborn as sns
import pymannkendall as mk
import matplotlib.pyplot as plt
from moabb.pipelines.features import AugmentedDataset
import warnings
from scipy.signal import find_peaks
from scipy.linalg import svd
from statsmodels.tsa.stattools import adfuller, kpss
warnings.filterwarnings("ignore", category=FutureWarning)
from statsmodels.tools.sm_exceptions import InterpolationWarning
warnings.filterwarnings("ignore", category=InterpolationWarning)
from scipy.stats import norm
# ================================================================================
# 2. Load nightly penguin count data
# ================================================================================
csv_nightly_count_path = r'..\data\penguin_count.csv'
df_total_penguin_count = pd.read_csv(csv_nightly_count_path)
df_total_penguin_count.count_date = df_total_penguin_count.count_date.apply(pd.to_datetime)
np_total_penguin_count = df_total_penguin_count.nightly_crossing_number.to_numpy()
# ================================================================================
# 3. Step 1: Construct the trajectory matrix
```

```
# ================================================================================

'''Step 1: Trajectory Matrix '''

augmented_dataset = AugmentedDataset(order=400, lag=1)

np_trajectory_matrix = augmented_dataset.fit_transform(np.array([[np_total_penguin_count]]),
y=np.ones(1))[0].transpose()

# ================================================================================

# 4. Define SSA decomposition and reconstruction functions

# ================================================================================

def decompose_and_grouping_step(X,groups):

    ''' Returns the list of groupings, the eigenvector, and the eigenvalues'''

    list_A = []

    U, s, Vh = np.linalg.svd(X, full_matrices=False)

    current = set([item for sublist in groups for item in sublist])

    rest = [[x for x in list(range(len(s))) if x not in current]]

    all_groups = groups + rest

    for g in all_groups:

        U_r = U[:,g]

        s_r = s[g]

        Vh_r = Vh[g, :]


        A_r = U_r @ np.diag(s_r) @ Vh_r

        list_A.append(A_r)


    variance_pct = np.sqrt(s)/np.sqrt(s).sum()

    return list_A, U, Vh, variance_pct, all_groups

def diagonal_averaging(matrix):

    n_rows, n_cols = matrix.shape

    N = n_rows + n_cols - 1

    result = np.zeros(N)

    counts = np.zeros(N)

    for i in range(n_rows):

        for j in range(n_cols):

            result[i + j] += matrix[i, j]

            counts[i + j] += 1
```

```
    return result / counts
# ============================================================================
# 5. Steps 2 and 3: SSA decomposition and grouping
# ============================================================================
'''Step 2 and 3: Decomposition and grouping'''
groups = [[i] for i in range(0,5)]
reconstructed_matrices, decomposition, eigenvectors, variance_pct, all_groups =
decompose_and_grouping_step(np_trajectory_matrix, groups)
# ============================================================================
# 6. Step 4: Reconstruction using diagonal averaging
# ============================================================================
''' Step 4: Reconstruction'''
reconstructions = []
for i,matrix in enumerate(reconstructed_matrices):
    reconstructions.append(diagonal_averaging(matrix))
    print(f'reconstruction {i}')
reconstructions = np.array(reconstructions)
# ============================================================================
# 7. Plot singular value contribution
# ============================================================================
plt.figure(figsize=(10,3))
plt.plot(variance_pct, 'o-',markersize=5)
plt.xlabel("Signal Component")
plt.ylabel("Singular Value")
plt.xlim(-1,100)
plt.grid(True)
plt.show()
# ============================================================================
# 8. Plot eigenvectors
# ============================================================================
fig, ax = plt.subplots(5,1,figsize=(10,10))
sns.lineplot(-1*eigenvectors[0], ax=ax[0], label = f'Eigenvector {0}')
for i in range(2,9,2):
    sns.lineplot(-1*eigenvectors[i-1], ax=ax[int(i/2)], label = f'Eigenvector {i-1}')
```

```
    sns.lineplot(-1*eigenvectors[i], linestyle='--', ax=ax[int(i/2)], label = f'Eigenvector {i}')
ax[4].set_xlabel('Lag (Days)')
plt.show()
# ================================================================================
# 9. Plot SSA decompositions
# ================================================================================
fig, ax = plt.subplots(5,1,figsize=(10,10))
sns.lineplot(x=df_total_penguin_count.count_date[:-400], y = -1*decomposition[:,0], ax=ax[0], label =
f'Decomposition {0}')
for i in range(2,9,2):
    sns.lineplot(x=df_total_penguin_count.count_date[:-400], y = -1*decomposition[:,i-1], ax=ax[int(i/2)], label =
f'Decomposition {i-1}')
    sns.lineplot(x=df_total_penguin_count.count_date[:-400], y = -1*decomposition[:,i], linestyle='--',
ax=ax[int(i/2)], label = f'Decomposition {i}')
ax[0].set_title(f'First 9 SSA Decompositions')
ax[4].set_xlabel('Lag (Days)')
plt.show()
# ================================================================================
# 10. Evaluate reconstruction fit
# ================================================================================
from sklearn.metrics import r2_score
r2_score(df_total_penguin_count.nightly_crossing_number[:-1],reconstructions[:-1].sum(axis=0))
# ================================================================================
# 11. Plot residual component
# ================================================================================
''' Rest '''
fig, ax = plt.subplots(figsize=(10,3))
sns.lineplot(x=df_total_penguin_count.count_date[:-1], y = reconstructions[len(reconstructions)-1], ax=ax,
linewidth=0.2)
ax.set_ylabel('Penguin Count')
ax.set_xlabel('Days')
plt.show()
# ================================================================================
# 12. Plot original series and SSA reconstruction
# ================================================================================
```

```
fig, ax = plt.subplots(figsize=(15,4))

sns.lineplot(x=df_total_penguin_count.count_date,y=np_total_penguin_count, ax=ax, color='blue', alpha=0.5,label = 'nightly penguin count')

sns.lineplot(x=df_total_penguin_count.count_date[:-1],y=reconstructions[:-1].sum(axis=0), ax=ax,label = 'SSA reconstruction', color = 'red')

ax.set_ylabel('Count of Penguins')

ax.set_xlabel('Date')

plt.show()

# ================================================================================

# 13. Save SSA decomposition output

# ================================================================================

df_nightly_count_decomposition = pd.DataFrame(reconstructions.T, columns=[f'component_{i}' for i in range(5)] + ['noise'])

df_nightly_count_decomposition['count_date'] = df_total_penguin_count.count_date.copy()

df_nightly_count_decomposition = df_nightly_count_decomposition[['count_date']+[f'component_{i}' for i in range(5)] + ['noise']]

df_nightly_count_decomposition.to_csv(r'..\data\penguin_count_decomposition_30.csv', index=False)

# ================================================================================

# 14. Prepare annual reconstructed and original count matrices

# ================================================================================

df_reconstruction = df_total_penguin_count[:-1].copy()

df_reconstruction['reconstructed'] = reconstructions[:-1].sum(axis=0)

df_reconstruction['DOY'] = df_reconstruction.count_date.dt.day_of_year

df_reconstruction['YEAR'] = df_reconstruction.count_date.dt.year

df_reconstructed_pivot = df_reconstruction.pivot_table(index='YEAR', columns='DOY', values='reconstructed')

df_original_pivot = df_reconstruction.pivot_table(index='YEAR', columns='DOY', values='nightly_crossing_number')

df_time_pivot = df_reconstruction.reset_index().pivot_table(index='YEAR', columns='DOY', values='count_date')

X_time = np.concatenate((df_time_pivot.loc[1982:2023,182:365].to_numpy(), df_time_pivot.loc[1983:2024,:181].to_numpy()), axis=1)

X_original = np.concatenate((df_original_pivot.loc[1982:2023,182:365].to_numpy(), df_original_pivot.loc[1983:2024,:181].to_numpy()), axis=1)

X_reconstructed = np.concatenate((df_reconstructed_pivot.loc[1982:2023,182:365].to_numpy(), df_reconstructed_pivot.loc[1983:2024,:181].to_numpy()), axis=1)

# ================================================================================

# 15. Scale reconstructed and original annual matrices
```

```
# ============================================================================

maxmin_scaler = MinMaxScaler()

X_reconstructed_scaled = maxmin_scaler.fit_transform(X_reconstructed.T).T

X_reconstructed_scaled.shape

X_original_scaled = maxmin_scaler.transform(X_original.T).T

# ============================================================================

# 16. Plot selected annual examples

# ============================================================================

fig, ax = plt.subplots(2,figsize=(15,8))

sns.lineplot(x=X_time[1985-1982],y= X_original[1985-1982], color='blue', linewidth=1, ax=ax[0],label = 'nightly penguin count')

sns.lineplot(x=X_time[1985-1982],y= X_reconstructed[1985-1982], color=(0.86, 0.37, 0.34), linewidth=2, ax=ax[0],label = 'SSA reconstruction')

ax[0].set_title('a) Nightly Penguin Count - 1985', loc='left')

ax[0].set_ylabel('Penguin Count')

ax[0].set_xlabel('Date')

sns.lineplot(x=X_time[1988-1982],y= X_original[1988-1982], color='blue', linewidth=1, ax=ax[1],label = 'nightly penguin count')

sns.lineplot(x=X_time[1988-1982],y= X_reconstructed[1988-1982], color=(0.86, 0.37, 0.34), linewidth=2, ax=ax[1],label = 'SSA reconstruction')

ax[1].set_title('b) Nightly Penguin Count - 1988', loc='left')

ax[1].set_ylabel('Penguin Count')

ax[1].set_xlabel('Date')

plt.show()

# ============================================================================

# 17. Detect and plot annual peaks

# ============================================================================

all_peaks = []

for i,y in enumerate(list(range(1982,2024))):

    plt.figure(figsize=(8,4))

    sns.lineplot(x=X_time[i],y=X_original[i], color='blue', linewidth=2, alpha=0.5,label = 'nightly penguin count')

    sns.lineplot(x=X_time[i],y= X_reconstructed[i], color='#d62728', linewidth=2,label = 'SSA reconstruction')


    hight =- X_reconstructed[i].std()/2 + X_reconstructed[0].mean()

    peaks, properties = find_peaks(X_reconstructed[i], height=hight)
```

```
all_peaks.append(peaks[0]+182)


plt.plot(X_time[i,peaks], X_reconstructed[i][peaks], "x", label='peaks', color='red')


print(properties)
plt.ylabel('penguin count')
plt.xlabel('Date')
plt.legend()
plt.show()
```

Appendix 2. Gaussian Process Forecasting of SSA-Reconstructed Penguin Count Time Series

```
# ================================================================================
# Appendix X. Code for Gaussian Process forecasting
# Forecasting SSA-reconstructed nightly penguin counts
# ================================================================================
# ================================================================================
# 1. Import libraries
# ================================================================================
from sklearn.metrics import (
    mean_absolute_error,
    root_mean_squared_error,
    r2_score,
    median_absolute_error,
    mean_absolute_percentage_error,
    mean_squared_log_error,
)
import matplotlib.pyplot as plt
import random
import numpy as np
import torch
from sklearn.preprocessing import MinMaxScaler
import pandas as pd
from tqdm import tqdm
def set_seed(seed=42):
    random.seed(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    if torch.cuda.is_available():
        torch.cuda.manual_seed(seed)
        torch.cuda.manual_seed_all(seed)
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False
set_seed(42)
```

```
import gpytorch
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
from sklearn.metrics import (
    mean_absolute_error,
    root_mean_squared_error,
    r2_score,
    median_absolute_error,
    mean_absolute_percentage_error,
    mean_squared_log_error,
    mean_squared_error
)
from scipy.signal import find_peaks
import seaborn as sns
# ================================================================================
# 2. Load SSA decomposition data
# ================================================================================
decomposition_path = '/content/penguin_count_decomposition_30.csv'
raw_path = '/content/penguin_count_no_outliers.csv'
penguin_count = '/content/penguin_count.csv'
time_series_ = pd.read_csv(decomposition_path)
time_series_.count_date = time_series_.count_date.apply(pd.to_datetime)
time_series_.set_index('count_date', inplace=True)
time_series_ = time_series_.resample('D').mean()
n_components = 5
time_series_ = time_series_[[f'component_{i}' for i in
range(n_components)]].sum(axis=1).rename('nightly_crossing_number')
time_series_ = pd.DataFrame(time_series_)
# ================================================================================
# 3. Define training and testing periods
# ================================================================================
time_series = time_series_
starting_year = 1982
lower_date = f'{starting_year}-01'
upper_date = f'{starting_year+43}-01'
```

```
delimiter = f'{starting_year}'
delimited_data = time_series[(time_series.index >= lower_date)&(time_series.index < upper_date)].copy()
data_train = delimited_data[delimited_data.index < delimiter].copy()
data_test = delimited_data[delimited_data.index >= delimiter].copy()
target = 'nightly_crossing_number'
# ================================================================================
# 4. Define the exact Gaussian Process model
# ================================================================================
class ExactGPModel(gpytorch.models.ExactGP):
    def __init__(self, train_x, train_y, likelihood, rbf_prior_length1=0.09, rbf_prior_length2=3):
        super(ExactGPModel, self).__init__(train_x, train_y, likelihood)
        self.mean_module = gpytorch.means.ConstantMean()
        P = 0.6
        k_trend = gpytorch.kernels.ScaleKernel(gpytorch.kernels.RBFKernel(lengthscale=rbf_prior_length1))
        k_season = gpytorch.kernels.PeriodicKernel(period_length=P,
period_length_constraint=gpytorch.constraints.Interval(0.5,0.7)) \
                * gpytorch.kernels.RBFKernel(lengthscale=rbf_prior_length2)
        gpytorch_kernel = k_trend + k_season
        self.covar_module = gpytorch.kernels.ScaleKernel(gpytorch_kernel)
    def forward(self, x):
        mean_x = self.mean_module(x)
        covar_x = self.covar_module(x)
        return gpytorch.distributions.MultivariateNormal(mean_x, covar_x)
# ================================================================================
# 5. Recursive yearly refitting and forecasting
# ================================================================================
chunk_size = 365 + 182
window_size = len(X_train)
epochs = 100
noise = 0.2
X_hist = X_train.clone()
y_hist = y_train.clone()
predictions = []
errors = []
```

```
for i in tqdm(range(0, len(X_test), 365)):
    if len(y_hist) > window_size:
        X_hist = X_hist[-window_size:]
        y_hist = y_hist[-window_size:]
    likelihood = gpytorch.likelihoods.GaussianLikelihood()
    likelihood.noise_covar.register_constraint("raw_noise", gpytorch.constraints.GreaterThan(noise))
    model = ExactGPModel(X_hist, y_hist, likelihood).double().to(device)
    model.train()
    likelihood.train()
    optimizer = torch.optim.Adam(model.parameters(), lr=0.1)
    mll = gpytorch.mlls.ExactMarginalLogLikelihood(likelihood, model)
    for _ in range(epochs):
        optimizer.zero_grad()
        output = model(X_hist)
        loss = -mll(output, y_hist)
        loss.backward()
        optimizer.step()
    model.eval()
    likelihood.eval()
    X_future = X_test[i : i + chunk_size]
    with torch.no_grad(), gpytorch.settings.fast_pred_var():
        preds = likelihood(model(X_future))
    y_pred = preds.mean.detach().cpu().numpy()
    y_var = preds.variance.detach().cpu().numpy()
    predictions.append(y_pred)
    errors.append(y_var)
    X_new = X_test[i : i + chunk_size]
    y_new = y_test[i : i + chunk_size]
    X_hist = torch.cat([X_hist, X_new])
    y_hist = torch.cat([y_hist, y_new])
# ============================================================================
# 6. Inverse-transform predictions and uncertainty estimates
# ============================================================================
```

```
predictions_inv = [
    scaler_y.inverse_transform(p.reshape(-1, 1)).flatten()
    for p in predictions
]
errors_inv = [
    scaler_y.inverse_transform(p.reshape(-1, 1)).flatten()
    for p in errors
]
# ================================================================================
# 7. Plot yearly forecasts and detect annual peaks
# ================================================================================
all_peaks = []
for i in range(41):
  date_index = predictions.index
  plt.figure(figsize=(15,6))
  sns.lineplot( x = data_test.index[i*365:i*365+len(predictions[i])], y = predictions_inv[i], color='orange', label='GP Forecast')
  sns.lineplot( x = data_test.index[i*365:i*365+len(predictions[i])], y= data_test['nightly_crossing_number'][i*365:i*365+len(predictions[i])],color='blue',label='First 5 Components of the SSA')
  y_mean = predictions_inv[i].flatten()
  hight =  -0.9
  peaks, properties = find_peaks(predictions[i], height=hight)
  if peaks[0] < 300:
    if peaks[0] < 50:
      all_peaks.append(peaks[1])
    else:
      all_peaks.append(peaks[0])
  else:
    all_peaks.append(None)
  plt.plot(data_test.index[i*365:i*365+len(predictions[i])][peaks], predictions_inv[i][peaks], "x", label='peaks', color='red')
  y_std = np.sqrt(errors_inv[i]).flatten()
  upper = y_mean + 1.96 * y_std
  lower = y_mean - 1.96 * y_std
```

```
plt.fill_between(
    data_test.index[i*365:i*365+len(predictions[i])],
    lower,
    upper,
    alpha=0.3,
    label="95% confidence interval"
    , color='orange'
)
plt.xlabel('Date')
plt.ylabel('Count')
plt.legend()
plt.show()
```

Appendix 3. Python Code for LightGBM Forecasting of SSA-Reconstructed Penguin Counts

```
# ================================================================================
# 1. Import libraries
# ================================================================================
import pandas as pd
import numpy as np
from lightgbm import LGBMRegressor
from skforecast.direct import ForecasterDirect
from skforecast.recursive import ForecasterRecursive
from sklearn.metrics import (
    mean_absolute_error,
    root_mean_squared_error,
    r2_score,
    median_absolute_error,
    mean_absolute_percentage_error,
    mean_squared_log_error,
    mean_squared_error
)
from sklearn.linear_model import Ridge
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
from skforecast.model_selection import grid_search_forecaster, TimeSeriesFold, backtesting_forecaster
from pmdarima.preprocessing import FourierFeaturizer
from skforecast.preprocessing import RollingFeatures
# ================================================================================
# 2. Load SSA decomposition data
# ================================================================================
decomposition_path = '/content/penguin_count_decomposition_30.csv'
raw_path = '/content/penguin_count_no_outliers.csv'
penguin_count = '/content/penguin_count.csv'
time_series_ = pd.read_csv(decomposition_path)
time_series_.count_date = time_series_.count_date.apply(pd.to_datetime)
time_series_.set_index('count_date', inplace=True)
time_series_ = time_series_.resample('D').mean()
n_components = 30
time_series_ = time_series_[[f'component_{i}' for i in range(n_components)]].sum(axis=1).rename('nightly_crossing_number')
time_series_ = pd.DataFrame(time_series_)
```

```
time_series = time_series_
# ==============================================================================
# 3. Define training and testing periods
# ==============================================================================
starting_year = 1982
lower_date = f'{starting_year}-01'
upper_date = f'{starting_year+43}-01'
delimiter = f'{starting_year+10}'
delimited_data = time_series[(time_series.index >= lower_date)&(time_series.index < upper_date)].copy()
data_train = delimited_data[delimited_data.index < delimiter].copy()
data_test = delimited_data[delimited_data.index >= delimiter]
# ==============================================================================
# 4. Generate Fourier exogenous variables
# ==============================================================================
trans = FourierFeaturizer(m=365, k=2)
trans.fit(delimited_data['nightly_crossing_number'].index)
y_prime, exog_ = trans.transform(delimited_data['nightly_crossing_number'].index, return_x=True)
exog_.index  = delimited_data.index
exog = exog_
# ==============================================================================
# 5. Define direct multi-step LightGBM forecaster
# ==============================================================================
steps = 14
lags = 4
forecaster = ForecasterDirect(
                estimator       = LGBMRegressor(random_state=42, verbose=-1, learning_rate= 0.1, n_estimators=200, max_depth=5),
                steps           = steps,
                lags            = lags,
                window_features = [RollingFeatures(stats=['mean'], window_sizes=3)],
                differentiation = 1,
            )
# ==============================================================================
# 6. Backtesting configuration
# ==============================================================================
cv = TimeSeriesFold(
        steps              = steps,
        initial_train_size = len(data_train),
        fold_stride        = steps,
```

```
        fixed_train_size   = True,
        refit              = 26,
        differentiation = 1,
    )
# ==============================================================================
# 7. Run backtesting
# ==============================================================================
metric, predictions = backtesting_forecaster(
                        forecaster    = forecaster,
                        y             = delimited_data['nightly_crossing_number'],
                        cv            = cv,
                        metric        = 'mean_squared_error',
                        n_jobs        = 'auto',
                        verbose       = True,
                        show_progress = True,
                        exog=exog[(delimited_data.index >= lower_date)&(delimited_data.index < upper_date)],
                    )
# ==============================================================================
# 8. Compute evaluation metrics by forecast horizon
# ==============================================================================
metrics = []
for i in range(steps):
    horizon_index = range(i, len(predictions), steps)
    predictions.iloc[horizon_index,2] = i
    y_pred = predictions.pred.iloc[horizon_index]
    y = data_test.nightly_crossing_number.iloc[horizon_index]
    mse = mean_squared_error(y, y_pred)
    rmse = np.sqrt(mse)
    mape = mean_absolute_percentage_error(y, y_pred)
    r2 = r2_score(y, y_pred)
    mae = mean_absolute_error(y, y_pred)
    metrics.append({
        "horizon": i,
        "MSE": mse,
        "RMSE": rmse,
        "MAPE": mape,
        "R2": r2,
        "MAE": mae
```

```
    })
final_metrics_df = pd.DataFrame(metrics)
# ==============================================================================
# 9. Plot LightGBM forecast against observed values
# ==============================================================================
date_index = predictions.index
plt.figure(figsize=(15,6))
plt.plot(predictions.pred, label='Forecast (LightGBM)', alpha=0.7, color='red')
plt.plot(data_test, label='Actual', alpha=0.5)
plt.legend()
plt.title("Forecast with Fourier Terms & Annual Retraining")
plt.show()
# ==============================================================================
# 10. Define recursive LightGBM forecaster for grid search
# ==============================================================================
steps = 26
lags = 10
forecaster = ForecasterRecursive(
                estimator       = LGBMRegressor(random_state=42, verbose=-1, learning_rate= 0.1, n_estimators=100, max_depth=4),
                lags            = 10,
                window_features = RollingFeatures(stats=['std'], window_sizes=10),
                differentiation = 1,
            )
# ==============================================================================
# 11. Define grid search parameters
# ==============================================================================
lags_grid = {
    'lags_1': 9,
    'lags_2': 10,
    'lags_3': 11
}
param_grid = {
    'learning_rate': [0.1,0.01],
    'max_depth':[4,5,6,7],
    'n_estimators': [50,100,200]
}
# ==============================================================================
# 12. Define grid search cross-validation folds
```

```
# ============================================================================
cv = TimeSeriesFold(
        steps              = 14,
        initial_train_size = len(data_train),
        fold_stride        = 14,
        fixed_train_size   = True,
        refit              = True,
       differentiation = 1,
    )
# ============================================================================
# 13. Run grid search
# ============================================================================
results = grid_search_forecaster(
            forecaster    = forecaster,
            y             = delimited_data['nightly_crossing_number'],
            param_grid    = param_grid,
            lags_grid     = lags_grid,
            cv            = cv,
            metric        = 'mean_squared_error',
            return_best   = True,
            n_jobs        = 'auto',
            verbose       = False,
            show_progress = True,
           exog=exog,
         )
```

Appendix 4. Python Code for PCA, Time-Series K-Means, and Gaussian Mixture Modeling for Seasonal Climate Classification

```
# 1. Import libraries
#=========================================================================
===
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from tslearn.clustering import TimeSeriesKMeans
#
=========================================================================
==
# 2. Define plotting function
#
=========================================================================
==
def plot_features(data, column_names, plot_name):
    plt.figure(figsize=(5,3))
    sns.violinplot(data=data, inner="quartile", palette="Set2")
    plt.xticks(ticks=np.arange(len(column_names)), labels=column_names, rotation=30)
    plt.title(plot_name, fontsize=14)
    plt.ylabel("Value", fontsize=12)
    plt.xlabel("Features", fontsize=12)
    plt.tight_layout()
    plt.show()
#
=========================================================================
==
# 3. Load climate variables and create feature matrix
```

```
#
==========================================================================
==
variables = [
'PRECTOTCORR',
 'T2M',
 'T2M_MAX',
 'T2M_MIN',
 'WS2M']
features = []
doys = None
for variable in variables:
    climate = pd.read_csv(fr'..\resources\{variable}\complete_dataset_{variable}.csv')
    new_feature = climate\
            .groupby(['YEAR','DOY'])\
            .mean()[climate.columns[-1]]\
            .to_list()
    if type(doys) == type(None) :
        doys = climate\
            .groupby(['YEAR','DOY'])\
            .mean().reset_index().DOY.to_numpy()
        years = climate\
            .groupby(['YEAR','DOY'])\
            .mean().reset_index().YEAR.to_numpy()
    features.append(new_feature)
    print(new_feature)
features = np.array(features).T
#
==========================================================================
==
# 4. Normalize climate features
#
==========================================================================
==
```

```
normalizer = MinMaxScaler((0,1))
normalized_features = normalizer.fit_transform(features)
#
==========================================================================
==
# 5. Apply PCA dimensionality reduction
#
==========================================================================
==
pca = PCA(n_components=3)
reduced_features = pca.fit_transform(normalized_features)
#
==========================================================================
==
# 6. Plot accumulated explained variance
#
==========================================================================
==
accumulated = np.cumsum(pca.explained_variance_ratio_)
plt.figure(figsize=(8,5))
plt.bar(range(len(accumulated)), accumulated, color="darkred")
for i, val in enumerate(accumulated):
    plt.text(i, val , str(round(val,2)), ha='center', va='bottom', fontsize=10)
plt.title("Variance", fontsize=14)
plt.xlabel("Index", fontsize=12)
plt.ylabel("Captured Variance", fontsize=12)
plt.xticks(range(len(pca.explained_variance_ratio_)))
plt.tight_layout()
plt.show()
#
==========================================================================
==
# 7. Apply time-series K-means clustering with DTW
#
==========================================================================
==
```

```
model = TimeSeriesKMeans(n_clusters=3, metric="dtw", n_jobs = -1)
model.fit(reduced_features)
cltrs = model.predict(reduced_features)
df_data_seasons = pd.DataFrame(features, columns=variables)
df_data_seasons['SEASON'] = cltrs
df_data_seasons['DOY'] = doys
df_data_seasons['YEAR'] = years
df_data_seasons['DATE'] = pd.to_datetime(df_data_seasons["YEAR"].astype(str) +
df_data_seasons["DOY"].astype(str), format="%Y%j")
df_data_seasons["month"] = df_data_seasons['DATE'].dt.month
df_data_seasons['kmeans_dtw'] = cltrs
data = df_data_seasons.copy()
data['seasons_kmeans'] = data.apply(lambda x:  x['kmeans_dtw'], axis=1)
Av_values_seasons = data[['T2M','kmeans_dtw']].groupby('kmeans_dtw').mean().sort_values(by= 'T2M')
palette = {}
for value, color in zip(Av_values_seasons.index, ['blue','green','red']):
    palette[value] = color
sns.scatterplot(data = data, x='T2M', y='WS2M',hue='seasons_kmeans', palette=palette)
plt.title('Seasonal Clustering')
plt.xlabel('Average Temperature')
plt.ylabel('Wind Speed')
plt.legend(title='clusters')
plt.show()
#
==========================================================================
==
# 8. Recreate seasonal clustering dataframe using predicted centers
#
==========================================================================
==
centers = model.predict(reduced_features)
df_data_seasons = pd.DataFrame(features, columns=variables)
df_data_seasons['SEASON'] = centers
```

```
df_data_seasons['DOY'] = doys

df_data_seasons['YEAR'] = years

df_data_seasons['DATE'] = pd.to_datetime(df_data_seasons["YEAR"].astype(str) +
df_data_seasons["DOY"].astype(str), format="%Y%j")

df_data_seasons["month"] = df_data_seasons['DATE'].dt.month

df_data_seasons['kmeans_dtw'] = centers

data = df_data_seasons.copy()

data['seasons_kmeans'] = data.apply(lambda x:  x['kmeans_dtw'], axis=1)

#
==========================================================================
==

# 9. Define seasonal color palette

#
==========================================================================
==

Av_values_seasons = data[['T2M','kmeans_dtw']].groupby('kmeans_dtw').mean().sort_values(by= 'T2M')

palette = {}

for value, color in zip(Av_values_seasons.index, ['blue','green','red']):

    palette[value] = color

#
==========================================================================
==

# 10. Plot climate clusters

#
==========================================================================
==

sns.scatterplot(data = data, x='T2M', y='WS2M',hue='seasons_kmeans', palette=palette)

plt.title('Seasonal Clustering')

plt.xlabel('Average Temperature')

plt.ylabel('Wind Speed')

plt.legend(title='clusters')

plt.show()

#
==========================================================================
==

# 11. Extract transition-season cluster
```

```
#
==========================================================================
==
fall_spring_cluster = data[data.kmeans_dtw == 1].copy()
dates_fall_spring = fall_spring_cluster.DATE.values
plt.figure(figsize=(10,3))
plt.hist(dates_fall_spring, bins=300, density=True, alpha=0.5, color="darkgreen", edgecolor="black")
plt.show()
#
==========================================================================
==
# 12. Fit Gaussian Mixture Model to transition-season dates
#
==========================================================================
==
x_train = fall_spring_cluster.index.values.reshape(-1, 1)
gmm = GaussianMixture(n_components=88, init_params='kmeans', n_init=15, max_iter=700,
random_state=47)
gmm.means_ = df_gaussians['mean']
gmm.covariances_ = df_gaussians['covariance']
gmm.weights_= df_gaussians['weight']
gmm.fit(x_train)
#
==========================================================================
==
# 13. Plot Gaussian mixture and component distributions
#
==========================================================================
==
plt.figure(figsize=(30,3))
plt.hist(dates_fall_spring, bins=600, density=True, alpha=0.5, color="darkgreen", edgecolor="black")
date_range = pd.date_range(pd.to_datetime('1981')+pd.Timedelta(days = x_train.min()),f'2025')
x_range = np.arange(x_train.min() , x_train.min()+len(date_range)).reshape(-1, 1)
logprob = gmm.score_samples(x_range)
pdf = np.exp(logprob)
plt.plot(date_range, pdf, "-k", label="Mixture")
```

```
for mean, cov, weight in zip(gmm.means_, gmm.covariances_, gmm.weights_):
    std = np.sqrt(cov[0][0])
    component_pdf = weight * norm.pdf(x_range, mean[0], std)
    plt.plot(date_range, component_pdf, "--", label=f"Component μ={mean[0]:.2f}")
plt.ylabel('Likelihood')
plt.xlabel('Date')
plt.show()
#
==========================================================================
==
# 14. Extract ordered Gaussian parameters and model criteria
#
==========================================================================
==
indexes_of_interest = np.argsort(gmm.means_.flatten())
fall_spring_means = np.int32(gmm.means_[indexes_of_interest].flatten())
fall_spring_covariances = np.int32(gmm.covariances_[indexes_of_interest].flatten())
print('aic: ',gmm.aic(x_train), ', bic: ', gmm.bic(x_train))
#
==========================================================================
==
# 15. Predict Gaussian classes for transition-season observations
#
==========================================================================
==
spring_fall_predictions = gmm.predict(x_train)
df_predictions = pd.DataFrame({'predictions':spring_fall_predictions})
#
==========================================================================
==
# 16. Order Gaussian components and assign transition seasons
#
==========================================================================
==
ordered_means = np.argsort(gmm.means_.flatten())
means = gmm.means_[ordered_means].flatten()
```

```
covariances = gmm.covariances_[ordered_means].flatten()
weights = gmm.weights_[ordered_means].flatten()
fall_means = ordered_means[list(range(0,len(ordered_means),2))]
spring_means = ordered_means[list(range(1,len(ordered_means),2))]
for i, cluster in enumerate(ordered_means):
    if i%2 == 0:
        df_predictions.replace(cluster,100,inplace=True)
    else:
        df_predictions.replace(cluster,300, inplace=True)
df_predictions.replace({300:3, 100:1}, inplace=True)
data.loc[data.seasons_kmeans == 1,'seasons_kmeans'] = df_predictions.values
#
==========================================================================
==
# 17. Plot refined seasonal clusters
#
==========================================================================
==
sns.scatterplot(data = data, x='T2M', y='WS2M',hue='seasons_kmeans', palette=palette)
plt.title('Seasonal Clustering')
plt.xlabel('Average Temperature')
plt.ylabel('Wind Speed')
plt.legend(title='clusters')
plt.show()
#
==========================================================================
==
# 18. Prepare seasonal color mapping for time plots
#
==========================================================================
==
import pandas as pd
import matplotlib.pyplot as plt
palette = [ "red", "orange","blue","green" ]
```

```
data = data.copy()
data["DATE"] = pd.to_datetime(data["DATE"])
data = data.sort_values("DATE")
data["seasons_kmeans"] = pd.Categorical(
    data["seasons_kmeans"],
    categories=sorted(data["seasons_kmeans"].unique()),
    ordered=True
)
categories = data["seasons_kmeans"].cat.categories
color_map = dict(zip(categories, palette))
data["color"] = data["seasons_kmeans"].map(color_map)
data_i = data.set_index("DATE").sort_index()
#
==========================================================================
==
# 19. Define seasonal boundary reference dates
#
==========================================================================
==
season_boundaries = [
    ("12-01", "Start of Summer"),
    ("03-01", "Start of Autumn"),
    ("06-01", "Start of Winter"),
    ("09-01", "Start of Spring"),
]
boundary_dates = {
    y: [(pd.Timestamp(f"{y}-{md}"), label) for md, label in season_boundaries]
    for y in range(1981, 2026)
}
#
==========================================================================
==
# 20. Plot seasonal assignments over two-year windows
```

```
#
==========================================================================
==
for start_year in range(1981, 2025):
    start = pd.Timestamp(start_year, 1, 1)
    end   = pd.Timestamp(start_year + 2, 1, 1)
    filtered = data_i.loc[start:end - pd.Timedelta(days=1)].reset_index()
    fig, ax = plt.subplots(figsize=(8, 4))
    ax.scatter(
        filtered["DATE"],
        filtered["T2M"],
        c=filtered["color"],
        s=12
    )
    for vdate, label in boundary_dates[start_year] + boundary_dates[start_year + 1]:
        if start <= vdate < end:
            ax.axvline(
                vdate,
                color="black",
                linestyle="--",
                linewidth=0.8
            )
            ax.text(
                vdate,
                1.01,
                label,
                rotation=90,
                ha="center",
                va="bottom",
                fontsize=8,
                transform=ax.get_xaxis_transform()
            )
```

```
    ax.set_ylabel("Average Temperature")
    ax.set_xlabel("Date")
    plt.show()
```

Appendix 5. Python Code for Machine Learning Model Comparison for MLD Prediction

```
# 1. Import libraries
# ============================================================================
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, GridSearchCV, TimeSeriesSplit
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import Ridge, ElasticNet
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.metrics import (
    mean_absolute_error,
    mean_squared_error,
    r2_score,
    mean_absolute_percentage_error,
    explained_variance_score
)
# ============================================================================
# 2. Prepare input data
# ============================================================================
X = selected_df.copy()
y = all_mld.loc[1985:, :].copy()
if isinstance(y, pd.DataFrame) and y.shape[1] == 1:
    y = y.iloc[:, 0]
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=42,
```

```
    shuffle=False
)
# ============================================================================
# 3. Define evaluation metrics function
# ============================================================================
def get_metrics(y_true, y_pred):
    mae = mean_absolute_error(y_true, y_pred)
    mse = mean_squared_error(y_true, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_true, y_pred)
    mape = mean_absolute_percentage_error(y_true, y_pred)
    evs = explained_variance_score(y_true, y_pred)
    return {
        "MAE": mae,
        "MSE": mse,
        "RMSE": rmse,
        "R2": r2,
        "MAPE": mape,
        "EVS": evs
    }
# ============================================================================
# 4. Define temporal cross-validation
# ============================================================================
tscv = TimeSeriesSplit(n_splits=3)
# ============================================================================
# 5. Define models and hyperparameter grids
# ============================================================================
model_configs = {
    "Ridge": {
        "estimator": Pipeline([
            ("scaler", StandardScaler()),
            ("model", Ridge())
        ]),
```

```
        "param_grid": {
            "model__alpha": [0.01, 0.1, 1.0, 10.0, 100.0]
        }
    },
    "ElasticNet": {
        "estimator": Pipeline([
            ("scaler", StandardScaler()),
            ("model", ElasticNet(max_iter=20000, random_state=42))
        ]),
        "param_grid": {
            "model__alpha": [0.001, 0.01, 0.1, 1.0],
            "model__l1_ratio": [0.2, 0.5, 0.8]
        }
    },
    "RandomForest": {
        "estimator": RandomForestRegressor(random_state=42),
        "param_grid": {
            "n_estimators": [10, 20, 40, 80],
            "max_depth": [2, 3, None],
            "max_features": ["sqrt", None]
        }
    },
    "GradientBoosting": {
        "estimator": GradientBoostingRegressor(random_state=42),
        "param_grid": {
            "n_estimators": [10, 20, 40, 80],
            "learning_rate": [0.03, 0.05, 0.1],
            "max_depth": [1, 2, 3],
            "subsample": [0.8, 1.0]
        }
    }
}
# ============================================================================
```

```
# 6. Run grid search for each model
# ================================================================================
best_models = {}
predictions = {}
results = []
cv_results_dict = {}
for name, cfg in model_configs.items():
    print(f"\nSearching hyperparameters for: {name}")
    grid = GridSearchCV(
        estimator=cfg["estimator"],
        param_grid=cfg["param_grid"],
        scoring="neg_root_mean_squared_error",
        cv=tscv,
        n_jobs=-1,
        refit=True,
        return_train_score=True
    )
    grid.fit(X_train, y_train)
    best_model = grid.best_estimator_
    best_models[name] = best_model
    cv_results = pd.DataFrame(grid.cv_results_).sort_values("rank_test_score")
    cv_results_dict[name] = cv_results
    y_pred_train = best_model.predict(X_train)
    y_pred_test = best_model.predict(X_test)
    predictions[name] = {
        "train": y_pred_train,
        "test": y_pred_test
    }
    train_metrics = get_metrics(y_train, y_pred_train)
    test_metrics = get_metrics(y_test, y_pred_test)
    results.append({
        "Model": name,
        "Best Params": grid.best_params_,
```

```
        "Best CV RMSE": -grid.best_score_,
        "Train MAE": train_metrics["MAE"],
        "Train MSE": train_metrics["MSE"],
        "Train RMSE": train_metrics["RMSE"],
        "Train R2": train_metrics["R2"],
        "Train MAPE": train_metrics["MAPE"],
        "Train EVS": train_metrics["EVS"],
        "Test MAE": test_metrics["MAE"],
        "Test MSE": test_metrics["MSE"],
        "Test RMSE": test_metrics["RMSE"],
        "Test R2": test_metrics["R2"],
        "Test MAPE": test_metrics["MAPE"],
        "Test EVS": test_metrics["EVS"]
    })
# ================================================================================
# 7. Create final results summary
# ================================================================================
results_df = pd.DataFrame(results).sort_values("Test RMSE").reset_index(drop=True)
print("\n" + "="*80)
print("FINAL SUMMARY ORDERED BY TEST RMSE")
print("="*80)
print(results_df)
# ================================================================================
# 8. Print detailed model results
# ================================================================================
for _, row in results_df.iterrows():
    print("\n" + "="*80)
    print(f"Model: {row['Model']}")
    print("="*80)
    print(f"Best Params: {row['Best Params']}")
    print(f"Best CV RMSE: {row['Best CV RMSE']:.6f}")
    print("\n--- TRAIN ---")
    print(f"MAE : {row['Train MAE']:.6f}")
```

```
    print(f"MSE : {row['Train MSE']:.6f}")
    print(f"RMSE: {row['Train RMSE']:.6f}")
    print(f"R2  : {row['Train R2']:.6f}")
    print(f"MAPE: {row['Train MAPE']:.6f}")
    print(f"EVS : {row['Train EVS']:.6f}")
    print("\n--- TEST ---")
    print(f"MAE : {row['Test MAE']:.6f}")
    print(f"MSE : {row['Test MSE']:.6f}")
    print(f"RMSE: {row['Test RMSE']:.6f}")
    print(f"R2  : {row['Test R2']:.6f}")
    print(f"MAPE: {row['Test MAPE']:.6f}")
    print(f"EVS : {row['Test EVS']:.6f}")
# ============================================================================
# 9. Print top configurations for each model
# ============================================================================
for name, df_cv in cv_results_dict.items():
    print("\n" + "="*80)
    print(f"TOP 5 CONFIGURATIONS FOR {name}")
    print("="*80)
    cols_to_show = [
        "params",
        "mean_test_score",
        "std_test_score",
        "mean_train_score",
        "std_train_score",
        "rank_test_score"
    ]
    print(df_cv[cols_to_show].head(5))
# ============================================================================
# 10. Print compact final table
# ============================================================================
summary_cols = [
    "Model",
```

```
    "Best Params",
    "Best CV RMSE",
    "Test MAE",
    "Test RMSE",
    "Test R2",
    "Test MAPE"
]
print("\n" + "="*80)
print("COMPACT TABLE")
print("="*80)
print(results_df[summary_cols])
# ============================================================================
# 11. Plot test comparison across models
# ============================================================================
plt.figure(figsize=(10, 5))
plt.plot(y_train.index, y_train, label="Train Actual", marker="o", color="black")
plt.plot(y_test.index, y_test, label="Test Actual", marker="o", color="gray", linewidth=2)
for name, preds in predictions.items():
    plt.plot(y_test.index, preds["test"], marker="o", label=f"{name} Test Pred")
plt.xlabel("Year")
plt.ylabel("MLD")
plt.title("Model comparison on test data using best hyperparameters")
plt.legend()
plt.tight_layout()
plt.show()
# ============================================================================
# 12. Plot individual model predictions
# ============================================================================
for name, preds in predictions.items():
    plt.figure(figsize=(8, 4))
    plt.plot(y_train.index, y_train, label="Train Actual", marker="o", color="black")
    plt.plot(y_train.index, preds["train"], label="Train Pred", marker="o")
    plt.plot(y_test.index, y_test, label="Test Actual", marker="o", color="gray")
```

```
    plt.plot(y_test.index, preds["test"], label="Test Pred", marker="o")
    plt.xlabel("Year")
    plt.ylabel("MLD")
    plt.title(f"{name} - best hyperparameters")
    plt.legend()
    plt.tight_layout()
    plt.show()
# ================================================================================
# 13. Plot metric comparison between training and testing sets
# ================================================================================
import numpy as np
import matplotlib.pyplot as plt
metrics = ["MAE", "RMSE", "MAPE", "EVS"]
order = {"MAE":"a) ","RMSE":"b) ", "MAPE":"c) ", "EVS":"d) "}
models = results_df["Model"].tolist()
model_colors = {
    "ElasticNet": "tab:blue",
    "Ridge": "tab:orange",
    "GradientBoosting": "tab:green",
    "RandomForest": "tab:red"
}
fig, axes = plt.subplots(2, 2, figsize=(10, 7))
axes = axes.flatten()
for i, metric in enumerate(metrics):
    ax = axes[i]
    train_col = f"Train {metric}"
    test_col = f"Test {metric}"
    x = np.array([0, 1])
    width = 0.8 / len(models)
    all_values = []
    for j, model in enumerate(models):
        row = results_df[results_df["Model"] == model].iloc[0]
        values = [row[train_col], row[test_col]]
```

```
        all_values.extend(values)
        positions = x - 0.4 + width / 2 + j * width
        bars = ax.bar(
            positions,
            values,
            width=width,
            label=model,
            color=model_colors.get(model, "gray")
        )
        for bar in bars:
            height = bar.get_height()
            ax.annotate(
                f"{height:.2f}",
                xy=(bar.get_x() + bar.get_width() / 2, height),
                xytext=(0, 3),
                textcoords="offset points",
                ha="center",
                va="bottom",
                fontsize=8
            )
    ax.set_title(order[metric]+ metric, loc='left')
    ax.set_ylabel("Value")
    ax.set_xticks(x)
    ax.set_xticklabels(["Training", "Testing"])
    if metric == "EVS":
        ax.set_ylim(0, 1)
    else:
        max_val = max(all_values)
        ax.set_ylim(0, max_val * 1.15)
handles, labels = axes[0].get_legend_handles_labels()
fig.legend(handles, labels, title="Model", bbox_to_anchor=(0.9, 0.5), loc="center left")
plt.suptitle("Model Comparison by Metric", fontsize=16)
plt.show()
```