令和７年度

高知県メジカ漁場予測システム構築委託業務

要件定義及び基本設計

目次

7. 入出力設計
    1. 7.1. データ管理
    2. 7.2. モデル学習管理
    3. 7.3. 予測結果管理
    4. 7.4. 画面管理

8. 予測モデル設計
    1. 8.1. 予測モデルのアーキテクチャ
    2. 8.2. 評価方法

1. はじめに

1.1. 高知県の漁業の現状について

本県においては、伝統の遠洋・近海かつお・まぐろ漁業をはじめ、釣り・延縄、定置網、中型まき網、魚類養殖などの海面漁業や、うなぎ養殖やあゆ漁などの内水面漁業が営まれている。 このように本県の沿岸・沖合域では多種多様な漁業が営まれており、国際的な操業規制が強化されるなかで漁業生産の場としての重要性が高まっているが、魚価安や燃 油、資材の価格変動などにより漁業経営は総じて不安定であり、高齢化や水産資源の減少と相まって漁業就業者数が減少する厳しい状況が続いている。

1.2. メジカ漁場予測システムについて

高知県では、漁業のデジタル化を推進し、操業の効率化や経営の安定化を図るため、令和元年度に有識者等で構成する高知マリンイノベーション運営協議会を立ち上げ、テーマの 1 つに AI 等を活用した漁船漁業のスマート化を挙げ、メジカ漁業の操業の効率化に向けた漁場予測システムの開発に令和 2 年度から取り組んでいる。 令和 4 年度には、海洋研究開発機構(以下、JAMSTEC)の海洋循環モデル JCOPE-T から得られた水温、塩分、潮流(水深 0～50m)等のデータから漁場を予測するためのアルゴリズムの検証を行い、令和 5 年 1 月からメジカ漁場予測システムの試験運用を開始した。また、令和 4 年度末から、種子島周辺漁業対策事業で開発したメジカ漁獲尾数計測システムで得られた漁獲情報も活用して予測精度向上に取り組み、令和 5 年度秋には漁業現場での実用化に一定の目処が立った。 漁業現場の実用化に向けては、これまでの研究段階のメジカ漁場予測システムとは異なり、プログラム更新機能等を備えた継続的な予測精度の向上が図れるシステムを新たに構築する必要がある。このため、令和 6 年度はシステムの設計を行い、令和 7 年度には、令和 6 年度に行った設計をもとにメジカ漁場予測システムを構築し、令和 8 年度に運用を開始する予定である。 システム完成後は、漁場予測情報を漁業者に配信することで魚群の探索に要する時間を短縮し、操業を効率化することで経営の改善を図る。

1.3. メジカ漁場予測システムの基本設計概要

前述の課題を受け、本書は令和 6 年度に実施されたメジカ漁場システムの基本設計を定義するドキュメントである。 令和 7 年度に予定しているメジカ漁場予測システムの構築では

大きく以下の 2 つの作業を実施する。

1. 外部データソースを用いてメジカ漁場予測モデルを作成し評価・チューニングを行う
2. 高知マリンイノベーション情報発信システム「NABRAS」を通じて、漁業者が効率的にメジカを漁獲できるよう支援するシステムを構築する

2. 業務内容及び要件

本業務は、メジカ漁場予測システムに関する取組みのうち、令和 7 年度に予定している同システムの構築である。

2.1. 業務一覧

令和 7 年度におけるシステム構築時には以下の業務内容をすべて満たすこと。

2.1.1. 構築作業

メジカ漁場予測システムの構築に必要な一連の作業を行うこと。各種データソースからのデータ取得の仕組みの構築や NABRAS での予測結果の表示については各運用保守事業者と連携しながら構築作業を進めること。

1. 開発

   本書をもとにシステム構築に必要な一連の開発作業を行うこと。なお、メジカ漁場予測システムは高知県庁で管理しているクラウドではなく、令和 8 年度以降に運用保守を担当する外部ベンダーが管理する Amazon Web Service (AWS)上に構築することを前提とする。

2. 統合テスト

   外部データリソース及び NABRAS での予測結果表示に関する結合テストを行うこと。

3. リリース方法策定および手順書作成

   メジカ漁場予測システムのリリース方法を策定し手順書を作成すること。なお、CI/CD 環境を構築することを前提とする。

4. 運用手順の作成

   メジカ漁場予測システムを運用していく際に実施する作業を定義し、その手順を策定すること。

2.1.2. 予測モデルの開発およびチューニング作業

メジカ漁場予測システムで用いる漁場予測モデルの作成、および継続的な学習機能を備えるシステム開発に必要な一連の作業を行うこと。

1. 予測モデルの開発

   本書をもとにメジカの漁場を予測する機械学習モデルを実装すること。

2. 予測モデルのデプロイ

   チューニングを行った予測モデルをシステムに組み込むこと。

3. 継続的な学習機能の実装

   初期モデルを使い続けるのではなく、継続的に予測モデルの再学習・評価・再デプロイを行うことができる機能を備えること。

3. 機能要件

本システムは、海況データ、漁船航行データ、漁獲量データを統合的に分析し、高精度なメジカ漁場海域の予測を行い、漁業従事者にメジカ漁場予測分布を表示する HTML ページを RESTful API にて提供し、効率的かつ持続可能な漁業活動を支援することを目的とする。

本章では、本システムに実装する必要がある機能を定義する。

3.1. データ収集機能

メジカの漁場予測に使用する各種データを収集する機能。本システムの外部で管理されているデータソースからデータを定期的に収集し、本システム内で管理するデータベースに保存する。

3.1.1. データの収集

以下 4 つのデータを収集対象とする。ただし、予測精度を向上させていくために将来的に取得するデータ種別が増える可能性があるため、追加で収集するデータを増やす場合は適宜改修を行うこと。 また、データサイズによって AWS lambda のメモリをスケールできるように設計すること。

1. 海況データ
2. 漁船航行データ
3. 漁獲量データ
4. 気象データ

海況データについて

- JCOPE-T のデータは再解析値と予測値の二つがあり、過去と未来のデータは性質が異なる可能性がある
- そのため、本システムにおいては「予測値」をデータベースに格納していくこととし、JAMSTEC が提供するサイトで毎日アップロードされるデータのうち過去分については使用しないこととする
- なお、将来的に再解析値を用いた研究を行う際には、JAMSTEC から必要なデータを

取得できるように高知県から依頼することとする

気象データについて

- 高知県と早稲田大学による過去の研究にて気象データ（土佐清水市）を使用して予測モデルを作成していたが、気象データの有無は予測結果に大きくは影響しないことが確認されている
- また、土佐清水市の気象データを取得可能な API の使用には月額費用がかかるため、運用コストを考慮して本システムでは気象データを使用しないこととする
- ただし、将来的に精度向上の研究を行う際に気象データが必要な可能性があるため、以下の対応を行う方針とし、構築フェーズにて必要なツールの開発は行なっておくこととする
    - 令和 8 年度以降の運用フェーズにおいて、気象庁のサイトからスクレイピング等をすることでデータを蓄積しておくこと
    - 気象データの蓄積場所は、本システムのデータベースではなくクラウドストレージ等にアーカイブしておくこと

注記

※1: 定期実行については「システム設計/バッチ処理」を参照
※2: セキュリティについては「非機能要件/要件一覧」を参照

3.1.2. データの保存

定期的に収集するデータをローデータデータベースに格納し、予測モデルの学習や漁場の予測に使用できるようにする。学習や予測に使用する際にデータの前処理を実施するため、ローデータデータベースにはデータ集計や欠損値の補間等の処理は適用せずに元のデータに極力近い状態で保存しておくこととする。

また、極端に古いデータについては学習・予測に使用しない方がよい可能性が考えられるが、研究目的として将来的に使用する必要がある可能性があるため、運用開始後に一定期間よりも古いデータについては費用削減のためにアーカイブしておく必要がある。そのため、それが実現できるように構築すること。ただし、アーカイブ方法については自動/手動は問わないため、最低限の機能を備えていればよいこととする。

注記

※3: データ形式については「入出力設計/入出力一覧」を参照

3.2. 予測モデルの学習機能

収集したデータをもとにメジカ漁場予測モデル（以下、モデル）を作成し、評価・デプロイを行う。本システムの運用開始後も、データの蓄積や新規データの追加に応じて学習・評価・デプロイを繰り返すため、一連の処理を行える機能を備えるように構築すること。

なお、学習の実行頻度は漁期に合わせて変更する可能性があり、本システムの運用が始まった後も随時変更される可能性があるため、柔軟に変更できるように構築すること。

3.2.1. 学習データ作成（前処理）

モデルの学習を実行する際に、ローデータデータベースから学習に必要なデータを取得し、学習に使用できる形に前処理を行う。

注記

※1: 定期実行については「システム設計/バッチ処理」を参照

3.2.2. モデルの学習および評価

前処理にて作成したデータをもとに学習を実行し、モデルの評価を行うための評価値を算出する。モデルの評価結果が評価基準を満たした場合はそのモデルを最良のモデルとして次の予測時から使用することとする。評価基準に満たなかった場合はパラメータチューニングを実行し再学習および再評価を複数回繰り返す。それでも評価基準を超えなかった場合は既存のモデルを次回の学習が実行されるまで最良モデルとして使用することとする。

パラメータチューニングを繰り返す回数については、運用保守作業において調整することが考えられるため、容易に変更できるように構築すること。

注記

※2: モデルについては「予測モデル」を参照

3.2.3. モデルの保存およびバージョン管理機能

前述の処理において最良のモデルが確定した場合、ストレージにモデルを保存する。評価基準を満たすモデルが得られなかった場合は、既存のモデルをそのまま最良のモデルとして保存しておく。新しい最良モデルが得られた場合でも、将来的な研究用途のために古いモデルもストレージに残しておくこととする。ただし、本システムの運用時に使用することはないため、運用保守の作業において必要に応じてより保管コストがかからないストレージに移管する可能性があるため、容易にその作業を行える構成とすること。

また、すべてのモデルはバージョン管理を行い、更新履歴と変更履歴を確認できる状態にしておくこと。

注記

※3: モデルの保存については「入出力設計/入出力一覧」を参照

3.2.4. 学習の手動実行機能

通常はモデルの再学習は既定のタイミングで自動的に実行されるようにしておくこととするが、必要に応じて運用保守担当者が手動で一連の処理（学習データの作成～モデルの保存）を行う場合があるため、その機能を備えること。

3.3. 漁場予測機能

ローデータデータベースから予測に必要なデータを取得し予測用データを作成し、最良の学習済みモデルに入力して予測結果を得る。この一連の予測処理は定期的に実行される。

注記

※1: 定期実行については「システム設計/バッチ処理」を参照

3.3.1. 予測用データの作成（前処理）

学習データ作成と同様に、予測を実行する際にローデータデータベースから予測に必要なデータを取得し、予測に使用できる形に前処理を行う。

注記

※2: 定期実行については「システム設計/バッチ処理」を参照

3.3.2. 予測

前処理にて作成したデータを用いて、漁場の予測結果を出力する。なお、予測に使用するモデルは通常は最良モデルを使用することとするが、予測に使用するモデルのバージョンや予測時に使用するパラメータは手動で変更できるような実装とすること。 また、前処理にて作成したデータが欠如していた場合は予測できなかった結果をデータベースに送信すること。

3.3.3. 予測結果の保存

前述の処理にて出力した予測結果を予測結果データベースに保存する。ユーザが過去の予測結果を参照する可能性もあるため、予測結果については過去の情報もすべてデータベースに格納しておくこととする。

注記

※3: データの形式については「入出力設計/入出力一覧」を参照

3.3.4. キャッシュ機能

情報発信システム（NABRAS）からのリクエストの度に予測結果データベースへのアクセスが発生することを避けるために、画面 UI からのリクエストを受け付ける API サーバのメモリ上に最新の予測結果をキャッシュしておくようにする。キャッシュの有効期限は次回のキャッシュ更新までとする。

3.3.5. 予測の手動実行機能

通常、予測処理は既定のタイミングで自動的に実行されるようにしておくこととするが、必要に応じて運用保守担当者が手動で予測を再実行する場合があるため、手動で予測対象日を指定して予測を実行することができるように構築すること。

3.4. 予測結果可視化機能

画面 UI からのリクエストを受け付ける API サーバ上にキャッシュされた予測結果を取得し、Google Map を用いてメジカ漁場予測結果を可視化したページ（以降、可視化ページ）を作成する。

注記

※1: 画面出力 UI に関しては「システム設計/画面 UI」を参照

3.5. 予測結果データ外部出力機能

可視化ページを情報発信システム（NABRAS）から iframe で表示できるようにする。

注記

※1: 出力データの形式については「入出力設計/入出力一覧」を参照
※2: セキュリティ、パフォーマンスについては「非機能要件/要件一覧」を参照

4. 非機能要件

本章では、メジカ漁場予測システムの品質特性、運用条件、および技術的制約を定義する。

4.1. セキュリティ

4.1.1. 通信規格

本システムにおける通信はすべて TLS1.3 を使用し、暗号化通信を行うこととする。

4.1.2. システムセキュリティ

OS パッチを定期的に適用し、OS を最新の状態を保つことでセキュリティを確保する。

4.1.3. 脆弱性対策

Amazon Inspector で脆弱性スキャンを月に 1 回実施すること。脆弱性が発見された場合は発見次第対応すること。

4.1.4. アクセス制御

メジカ漁場予測システムへのアクセスは、WAF（Web Application Firewall）を用いて制御し、継続的に監視すること。また、予測結果は情報発信システム（NABRAS）からの閲覧のみを許可するため、本システム内で作成される可視化ページは閲覧制限を設けること。制限方法については「入出力設計/入出力一覧」を参照すること。

4.2. パフォーマンス

4.2.1. 応答時間

画面 UI からのリクエストを受け付ける API の応答時間は 500ms 以内とする。

4.2.2. バッチ処理時間

毎日の予測結果の表示に必要な一連の処理（最新のデータの収集～予測結果出力～可視化ページの作成）は 5 時間以内に完了することとする。

4.3. 拡張性

4.3.1. 予測モデル

本システムは、将来的な学習モデルの変更やパラメータの調整に対応できる柔軟な設計とすること。

4.3.2. インフラストラクチャー

システムの負荷に応じて、計算リソースを柔軟に拡張できる構成とすること。例えば、Amazon EC2 を使用する場合は、インスタンスタイプを負荷に応じて変更できるような構成にしておくこと。

4.3.3. セキュリティ

新たなセキュリティ要件や脅威に対して、セキュリティ対策を容易に更新できる構成とすること。

4.3.4. 外部サービスとの連携

将来的な外部サービスとの連携追加に対応できる疎結合な設計とすること。

注記

※1: データの追加があった場合は下記の改修が必要となるため、その改修が容易に行える構成とすること。

- データ管理
    - 外部データソースの取得処理およびローデータデータベースへの格納処理
    - ローデータベースのテーブルの修正や追加
- モデル学習管理・予測結果管理
    - ローデータデータベースからのデータ取得処理
    - 学習データ・予測用データを作成する前処理
    - 予測モデルのアルゴリズムの変更および精度検証

4.4. システム環境

本システムはパブリッククラウドである Amazon Web Service（AWS）を使用して構築することを前提とする。

4.4.1. 使用する主要サービス

サービス名	主な用途
Amazon EC2	学習処理
Amazon RDS	ローデータ・予測結果の保存
Amazon S3	学習済みモデルの保存
AWS Lambda	定期実行処理
AWS WAF	API アクセス制御
Amazon CloudWatch	定期実行処理、システムの監視

4.5. 運用保守

4.5.1. システム監視

Amazon CloudWatch を使用し、CPU 使用率、メモリ使用率、ログの監視を行い障害発生時には運用者に通知する。

4.5.2. システム運用

本システムに不具合が発生した場合は、1営業日以内の復旧を目標とする。

管理ツール

ツール	用途	選定理由
GitHub	ソースコード管理	バージョン管理ができ、ロールアップも容易なためGitHbを用いてソースコードの管理を行うこと。
GitHub Actions	CI/CD	ソースコードのデプロイ前にテストできるようにすることでシステムの安定稼働につながり品質を担保できるため、GitHub Actionsを使用してCI/CD環境を構築することとする。
Terraform	インフラ管理	インフラ環境のバージョン管理ができ、柔軟な対応が容易になるためTerraformを用いてインフラ環境を構築することとする。

データ取得管理

外部要因/内部要因の如何によらず、不具合により外部のデータソースからデータが取得できなかった場合、運用者が手動で不足しているデータを取得しローデータベースを最新の状態に保つこと。この作業は次の予測処理がバッチで実行されるまでに行うこと。

学習モデル管理

過去の学習済みモデルを研究目的で将来的に使用する可能性があるため、高知県の管理者から指示があるまでは過去の学習済みモデルはストレージに保持すること。また、古いモデルは別途用意されたストレージに移行させてアーカイブする可能性があるため、過去モデルは手動にて任意のタイミングで取得可能であること。

4.5.3. バックアップ

データのバックアップはAWS Backupを使用し、一元管理する。

1. Amazon RDS

- 方式: 自動スナップショット
- 頻度: 日次
- 保存期間: 12 ヶ月

2. Amazon S3

- 方式: 定期的バックアップ
- 頻度: 月次
- 保存期間: 12 ヶ月

4.6. 互換性

4.6.1. 外部システム連携

外部システムとの連携は RESTful API を使用し、JSON 形式でデータの送受信を行うこととする。

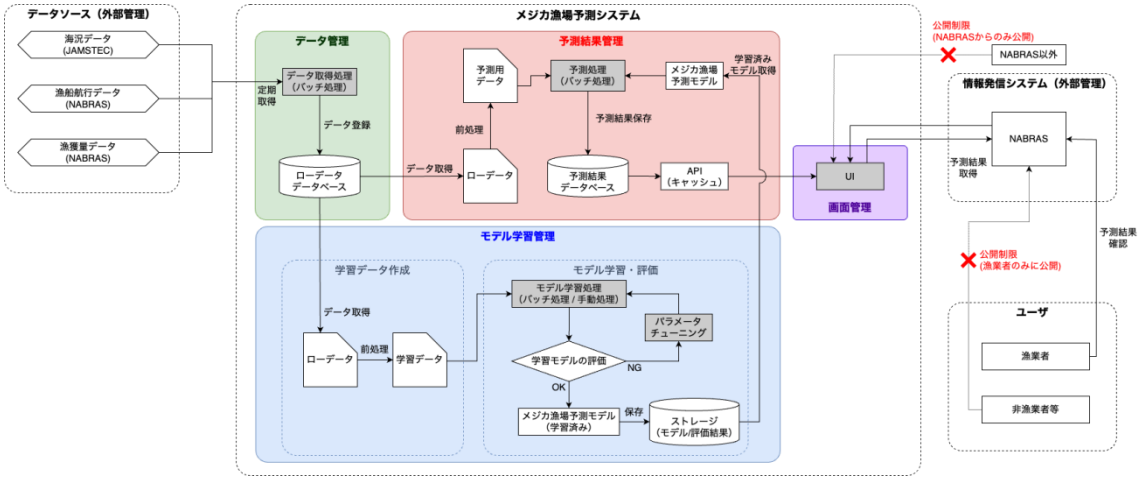
5. システム設計

本章では、メジカ漁場予測システムのシステム構成について記載する。

5.1. システム構成

5.1.1. システム構成図

メジカ漁場予測システムの構成は下図の通り。



5.1.2. 各モジュールの役割

メジカ漁場予測システムを構成するモジュールとその役割は以下の通り。なお、メジカ漁場予測システムは県庁クラウドの外部のクラウド（運用保守を担当するベンダーが管理するクラウド）に構築する。

メジカ漁場予測システム

名称	概要
データ管理	データソースからデータを取得し保存する
モデル学習管理	データ管理にて保存したデータを用いてメジカ漁場予測モデルの学習・評価を行い保存する
予測結果管理	漁場の予測を行い、予測結果を保存する
画面管理	予測結果を元に、漁場の予測分布を表示する HTML ページを作成する

5.1.3. 外部モジュール

以下のモジュールはメジカ漁場予測システムには含まれないが、メジカ漁場予測システムと連携する外部のリソースやシステムである。

データソース（外部管理）

名称	概要
海況データ (JAMSTEC)	JCOPE-T による海況の予測データ。 最新データは公開ページからダウンロード可能。
漁船航行データ (NABRAS)	NABRAS にて用意される API を通して取得可能な各漁船の位置や船首方向等の航行データ。
漁獲量データ (NABRAS)	NABRAS にて用意される API を通して取得可能な各漁船の漁獲量を表すデータ。

情報発信システム（外部管理）

名称	概要
NABRAS	「高知マリンイノベーション」の取組みの一環として構築した情報発信システムであり、以下の情報を一元的に発信する。 ・黒潮牧場ブイや人工衛星の解析データなどの操業の効率化に資するデータ ・赤潮、急潮の予測など、漁業経営のリスク軽減に資する情報 ・県水産振興部が蓄積してきた水産に関する様々な研究・統計データ なお､メジカ漁場システムの予測結果も NABRAS を通して閲覧できるようにする予定である。

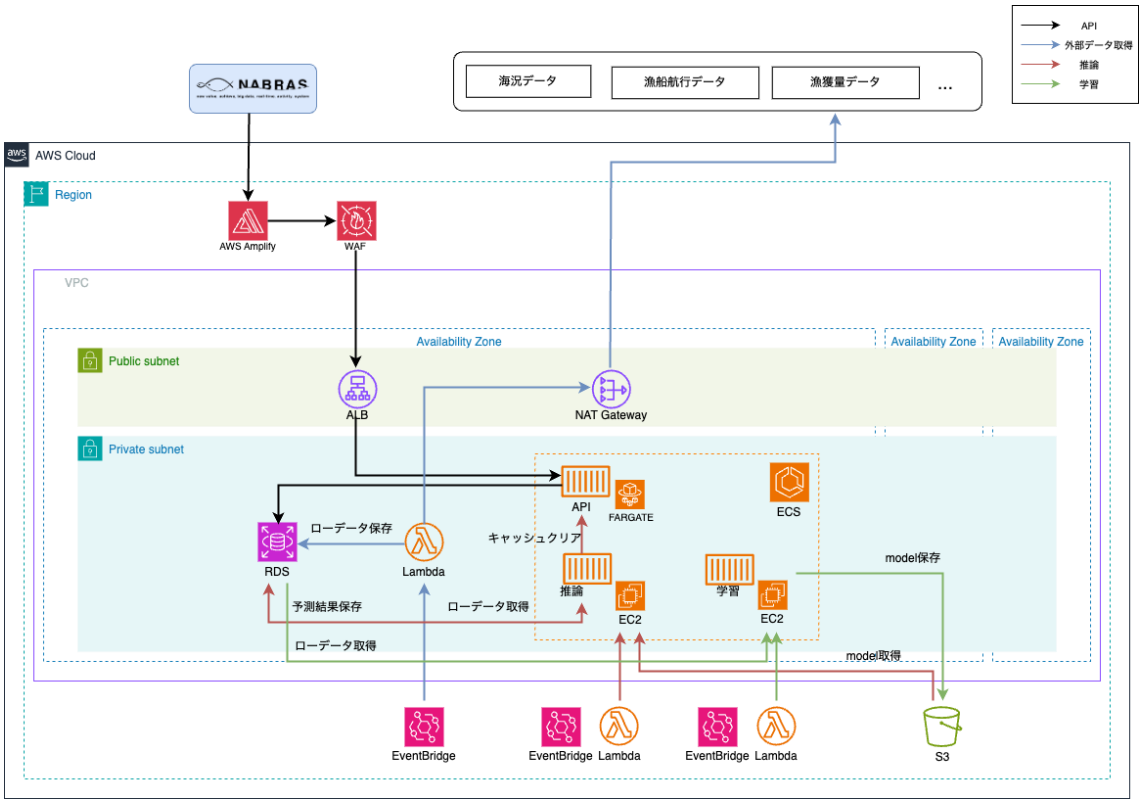
ユーザ

名称	概要
漁業者	NABRAS およびメジカ漁場予測システムの予測結果を使用して漁業を行う、予測結果の直接的な利用者。
非漁業者等	上記以外の人々。予測結果は漁業者のみに閲覧を限定するため公開制限を行う

5.2. サーバ構成

5.2.1. サーバ構成図

メジカ漁場予測システムのサーバ構成は下図の通り。



5.2.2. 使用するクラウドについて

パブリッククラウドである Amazon Web Services（AWS）上に構築すること。

5.3. 画面 UI

5.3.1. 前提

ここでは、NABRAS で表示する予測結果の表示部の画面を定義する。なお、具体的な文言や表示・デザインの調整については詳細設計フェーズにて対応することとする。

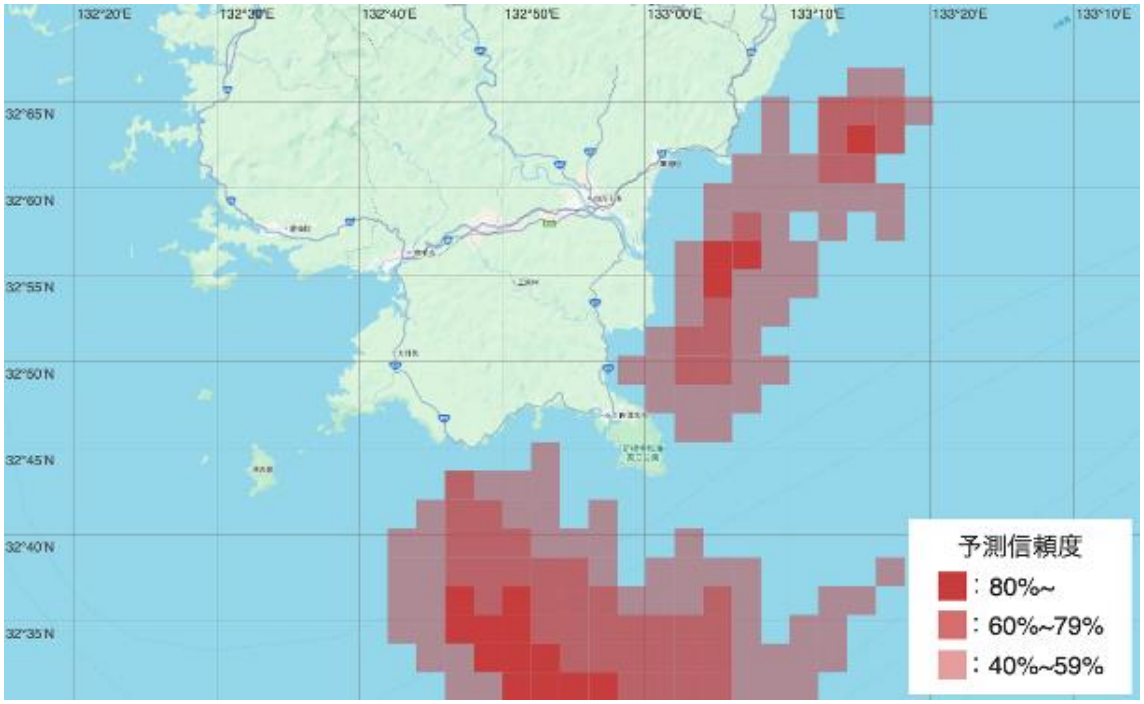
5.3.2. 予測結果表示 UI

正常時の UI

Google map 上に以下の情報を表示する。

- 予測の信頼度別に配色された予測結果のマップ
- 配色ごとの信頼度の凡例
- 緯度・経度のグリッド線およびその値

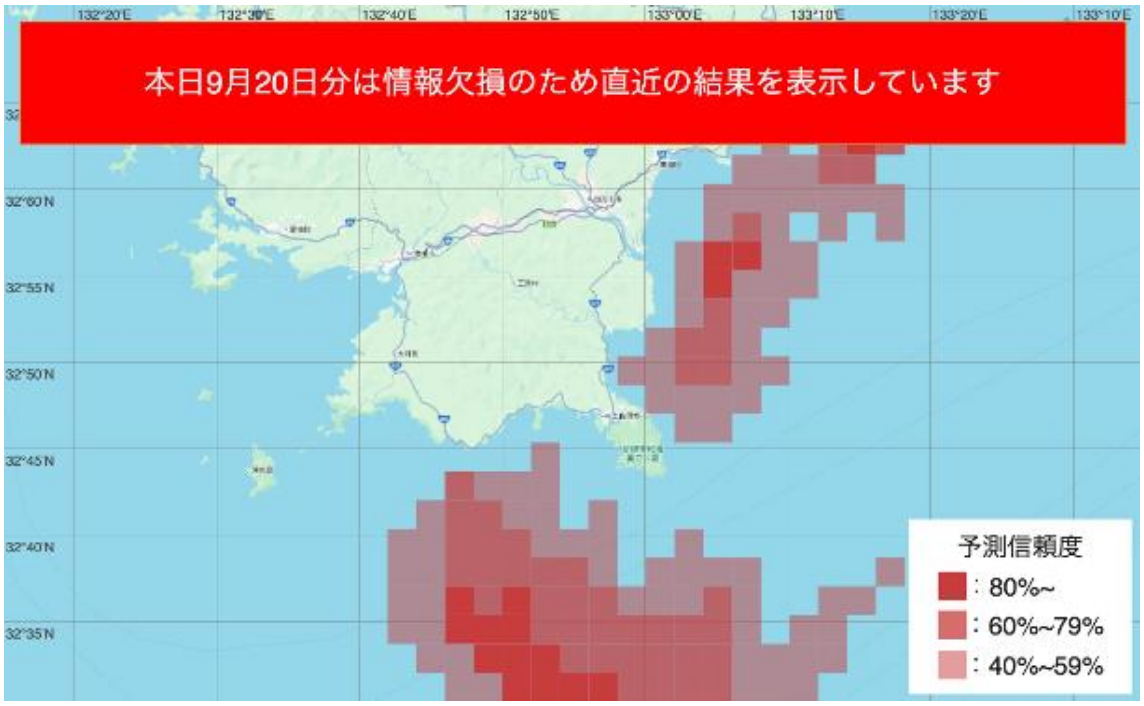
注記：タイムゾーンを出力する必要がある場合は JST にすること



不具合発生時の UI

予測処理が正常に完了出来なかった場合など、メジカ漁場予測システムに問題が発生しユーザに正しい情報を提供出来ない場合は以下の対応を行う。

- 正しく処理が完了した直近の予測結果を表示する
- マップの上部にエラーメッセージを目立つように表示する



5.4. システムフロー

5.4.1. 概要

メジカ漁場予測システムは以下に記述する 4 つのモジュールから大きくは構成される。各々のモジュールは原則的に独立して動作するため、モジュールごとに処理の流れの概要を以下に記載する。

5.4.2. データ管理

1. 定期実行でデータソース（外部管理）からデータを取得する
    - 取得するデータについては機能要件/要件一覧/データーの収集を参照
2. ローデータベースに格納できる形式に変換する
3. ローデータデータベースに保存する
4. 予測結果管理の処理をトリガーする

注記

※1: 定期実行については「システム詳細/バッチ」処理を参照
※2: ローデータデータベースに保存するデータ形式については「入出力設計/入出力一覧」を参照

5.4.3. モデル学習管理

1. 学習を実行するEC2インスタンスを起動する
    - 起動方法1：EventBridgeで定期自動起動
    - 起動方法2：Lambdaから起動(curl等の起動リクエストコマンドを受け付けるAPIサーバをLambda上に実装する)
    - 起動方法3：AWSコンソールから運用担当者が手動起動
2. データ管理のローデータデータベースからデータを取得する
3. 取得したデータを学習に使用できるように前処理を行う
4. 学習データを使用して予測モデルの学習を行う
5. 学習した予測モデルを既定の評価指標で評価する
6. 既存の最良モデルよりも良い評価結果が得られた場合は、後続の処理4を行う
    - 既存の最良モデルよりも低い評価結果となった場合は、パラメータの変更を行い処理4～6を再度実施する
    - 既定の回数だけ繰り返しても良い評価結果が得られない場合は中止し、既存の最良モデルを継続使用する
7. 学習した予測モデルを最新の最良モデルとしてストレージに保存し、評価結果、学習ログも合わせて保存する

注記

※3: ローデータデータベースから取得するデータ形式に関しては「入出力設計/入出力一覧」を参照

※4: 機械学習については「機械学習」を参照
※5: ストレージに保存するデータに関しては「入出力設計/入出力一覧」を参照

5.4.4. 予測結果管理

5.4.4.1. 予測実行

1. データ管理のローデータデータベースからデータを取得する
2. 前処理を行い、予測用データを作成する
3. モデル学習管理のストレージから予測モデルを取得する
4. 予測用データと予測モデルを用いて予測結果を出力する
    - 予測用データが欠如していた場合、予測せず実行日分の結果は欠如として予測結果データベースに保存する
5. 予測結果を予測結果データベースに保存する

5.4.4.2. API

最新(Request した日)の予測結果を取得する場合

1. UI からの Request を受信する
2. 対象日の予測結果データを取得する
    - 対象日の予測結果データがキャッシュされている場合はキャッシュ上のデータを取得する
    - 対象日の予測結果データがキャッシュされていない場合
        - 既存のキャッシュを破棄
        - 対象日のデータを予測結果データベースから取得しキャッシュ後、そのデータを取得する
    - 対象日の予測結果データが欠如していた場合、その旨を UI への Response に記載する
3. UI への Response に対象日の予測結果データを記載し提供する

過去の予測結果を取得する場合

1. UI からの Request を受信する
2. 対象日の予測結果データを予測結果データベースから取得する
3. UI への Response に対象日の予測結果データを記載し提供する

注記

※6: ローデータデータベースのデータ形式については「入出力設計/入出力一覧」を参照

5.4.5. 画面管理

1. 情報発信システム（NABRAS）から Request を受信する
2. 予測結果管理の API へ Request を送信し、予測対象日の予測結果を受信する
3. 予測結果を用いて、緯度・経度ごとに確信度に応じて色分けされた漁場の予測分布を表示する HTML ページを作成する
    - 予測結果が欠如していた場合、エラーメッセージとキャッシュデータを使用した画面を作成し、情報発信システム（NABRAS）への Response として HTML ページ URI を返す
4. 情報発信システム（NABRAS）への Response として HTML ページ URI を返す

5.5 バッチ処理

メジカ漁場予測システムではデータの取得、予測モデルの学習、予測結果の出力は原則的にそれぞれ独立したバッチ処理として実行される。ここでは各々のモジュールにて行う処理について記述する。

なお、時刻については特に言及がない限りは日本標準時の時刻とする。

5.5.1. モジュール一覧

モジュール	実施処理	実行頻度
データ管理	データ取得処理	日次
モデル学習管理	モデル学習処理	漁期によって設定を変更
予測結果管理	予測処理	日次

5.5.2. データ取得処理

実行時刻

毎日午前 3 時に実行する。

エラーハンドリング

リトライエラー

- 外部データリソースからのデータ取得においてエラーが発生した場合は再試行を最大で 3 回実施する。それでもエラーが発生した場合はログに記録し、管理者にアラートを送信する。管理者は不足分を補うため手動で次の予測実行時間になるまでにデータ取得すること。

データ保存エラー

- データの加工およびローデータベースへの保存時にエラーが発生した場合は、ログに記録し管理者にアラートを送信する。

5.5.3. モデル学習処理

実行時刻

実行時刻および実行頻度は、メジカ漁の漁期に合わせて適切な値を設定することとする。なお、実行時刻と実行頻度は本システムの運用を開始した後も随時変更される可能性があるため、本番リリース後に高知県と協議しながら適切な設定値を決めていくこと。

エラーハンドリング

学習時エラー

- ログに記録し、管理者にアラートを送信する

5.5.4. 予測処理

実行時刻

毎日午前 4 時頃に実行する。ただし、予測処理はデータ取得処理が完了した後に実行する必要があるため、データ取得処理に要する時間を考慮して実行時刻を確定することとする。

エラーハンドリング

データ保存エラー

- ログに記録し、管理者にアラートを送信する

予測時エラー

- ログに記録し、管理者にアラートを送信する

6. データベース設計

本章では、データベースのテーブル設計やデータ形式を定義する。

データベースおよび各種ファイルの格納には Amazon Web Service の以下を使用する。

- Amazon RDS
- Amazon S3

6.1. データ管理

6.1.1. ローデータデータベース

外部データソースから取得したデータを保持する。

- Service: Amazon RDS
- DB name: raw_data
- Tables:
    - sea_condition
    - fishing_boat_navigation
    - haul_medica

sea_condition テーブル

海況データを保持するテーブルは以下のように定義する。

フィールド名	データ型	制約	備考
id	BIGINT	PRIMARY KEY, AUTO_INCREMENT, NOT NULL	識別子
timestamp	TIMESTAMP	NOT NULL	データ取得日時

latitude	DOUBLE	NOT NULL	緯度
longitude	DOUBLE	NOT NULL	経度
depth	FLOAT	NOT NULL	深度 (m)
egt	FLOAT	NOT NULL	海面高度 (m)
st	FLOAT	NOT NULL	塩分 (PSU)
tt	FLOAT	NOT NULL	水温 (度)
ui	FLOAT	NOT NULL	東西流速 (m/s)
vi	FLOAT	NOT NULL	南北流速 (m/s)
created_at	TIMESTAMP	NOT NULL	データが作成された日時

fishing_boat_navigation テーブル

漁船航行データを保持するテーブルは以下のように定義する。

フィールド名	データ型	制約	備考
id	BIGINT	PRIMARY KEY,AUTO_INCREMENT,NOT NULL	識別子
timestamp	TIMESTAMP	NOT NULL	データ取得日時
vessel	VARCHAR(25)	NOT NULL	船名
vessel_speed	FLOAT	NOT NULL	船速 (m/s)
vessel_head	FLOAT	NOT NULL	船首 (度)
latitude	DOUBLE	NOT NULL	緯度
longitude	DOUBLE	NOT NULL	経度
speed	SMALLINT	NOT NULL	速度 (m/s)
line_cord	TINYINT	NOT NULL	ラインコード
temperature	DOUBLE	NOT NULL	水温 (度)
depth	DOUBLE	NOT NULL	水深 (m)
depth_1	DOUBLE	NOT NULL	第一層水深 (m)
flow_direction_1	DOUBLE	NOT NULL	第一層流向 (度)
flow_rate_1	DOUBLE	NOT NULL	第一層流速 (m/s)
depth_2	DOUBLE	NOT NULL	第二層水深 (m)
flow_direction_2	DOUBLE	NOT NULL	第二層流向 (度)
flow_rate_2	DOUBLE	NOT NULL	第二層流速 (m/s)
depth_3	DOUBLE	NOT NULL	第三層水深 (m)

flow_direction_3	DOUBLE	NOT NULL	第三層流向（度）
flow_rate_3	DOUBLE	NOT NULL	第三層流速（m/s）
created_at	TIMESTAMP	NOT NULL	データが作成された日時

haul_medica テーブル

漁獲量データを保持するテーブルは以下のように定義する。

フィールド名	データ型	制約	備考
id	BIGINT	PRIMARY KEY, AUTO_INCREMENT, NOT NULL	識別子
date	DATE	NOT NULL	処理日
vessel	VARCHAR(25)	NOT NULL	漁船名称
market	VARCHAR(25)	-	市場名称
species	VARCHAR(25)	-	魚の種類
fish_name	VARCHAR(25)	-	魚の名称
amount	SMALLINT	-	漁獲数量（重量）
created_at	TIMESTAMP	NOT NULL	データが作成された日時

6.2. モデル学習管理

6.2.1. ストレージ

年月日ごとにフォルダを作成し、学習済みの予測モデルおよび評価結果、学習時のログを保持する。

- Service: Amazon S3
- Bucket Name: medica-model
- Type: csv

保持対象ファイル一覧

file 名	用途
weight.pth	学習済みモデル
metrics.csv	評価結果
train.log	学習時のログ

S3 バケットおよび各種ファイルの構成

```
medica-model/
├─── 2026-04-01/
|    ├─── weight.pth
|    ├─── metrics.csv
|    └─── train.log
├─── 2026-04-02/
|    ├─── weight.pth
|    ├─── metrics.csv
|    └─── train.log
├─── ...
└─── YYYY-MM-DD/
     ├─── weight.pth
     ├─── metrics.csv
     └─── train.log
```

6.3. 予測結果管理

6.3.1. 予測結果データベース

メジカ漁場予測結果を保持する。

- Service: Amazon RDS
- DB Name: prediction
- Table: medica_result

medica_result テーブル

メジカ漁場の予測結果を保持するテーブルは以下のように定義する。

フィールド名	データ型	制約	備考
id	BIGINT	PRIMARY KEY, AUTO_INCREMENT, NOT NULL	識別子
latitude	DOUBLE	NOT NULL	緯度
longitude	DOUBLE	NOT NULL	経度
confidence	FROAT	NOT NULL	確信度
model_version	VARCHAR(8)	NOT NULL	v1, v2 ...
created_at	TIMESTAMP	NOT NULL	データが作成された日時
forecast_at	TIMESTAMP	NOT NULL	予測対象日の日時

7. 入出力設計

本章では下記項目の入出力のデータ形式を定義する。

- データソースとデータ管理におけるインターフェイス
- 予測結果管理と情報発信システム(NABRAS)におけるインターフェイス
- データ管理、モデル学習管理、予測結果管理の各種サービスにおけるインターフェイス

なお、本章で定めるエラーハンドリングは各々の個別処理におけるエラーハンドリングであり、バッチ処理としてのエラーハンドリングについては別途システム設計/バッチ処理にて定めることとする。

7.1. データ管理

7.1.1. データ収集

7.1.1.1. 海況データ

JAMSTEC が提供するサイトから海況データ(NetCDF)を取得する。ここでは、当該サイトからダウンロードしたデータの形式について記述する。

No.	略称	説明
1	EGT	海面高度
2	ST	塩分
3	TT	水温
4	UI	東西流速
5	VI	南北流速

取得方法

以下のコマンドを使用して、最新のデータをダウンロードする。

wget https://www.jamstec.go.jp/jcope/data/jcopet.eas.var/{yyyymmdd}/{FILE_NAME}.nc

注記

- 毎日データがアップロードされるため、{yyyymmdd}配下のファイルを用いてローデータデータベースを毎日更新する
- データを最新状態に保つため、毎日アップロードされる±8 日分のデータのうち{yyyymmdd}の日付以降のデータすべてを用いてデータベースの更新を行う(使用するデータについては機能要件/データ収集も参照すること)
    - 例：20241126 にアップロードされたデータの場合は、*_20241126.nc~*_20241204.nc の日付のデータのみを用いる
- 全データを一度に Lambda を用いて処理しようとするとタイムアウトが発生する可能性が高くストレージリソースも十分に必要になるため、一度の Lambda の実行で 1 ファイルのみダウンロード・処理・データ保存を行うように構築する

- タイムゾーンは UTC である

取得データ形式

NetCDF はデータ自身に以下の情報を保持しているため、実データを直接確認することでも以下の情報の確認は可能である。

1. EGT (sea surface height: 海面高度)

フィールド名	データ型	説明
lon	float32	経度
lat	float32	緯度
depth	float32	深度
time	datetime64(ns)	時刻
sea surface height	float32	海面高度 (m)

2. ST (salinity: 塩分)

フィールド名	データ型	説明
lon	float32	経度
lat	float32	緯度
depth	float32	深度
time	datetime64(ns)	時刻
salinity	float32	塩分 (PSU)

3. TT (temperature: 水温)

フィールド名	データ型	説明
lon	float32	経度
lat	float32	緯度
depth	float32	深度
time	datetime64(ns)	時刻
temperature	float32	水温 (° C)

4. UI (zonal current: 東西流速)

フィールド名	データ型	説明
lon	float32	経度
lat	float32	緯度
depth	float32	深度
time	datetime64(ns)	時刻
zonal current	float32	東西流速 (m/s)

5. VI (meridional current: 南北流速)

フィールド名	データ型	説明
lon	float32	経度
lat	float32	緯度
depth	float32	深度
time	datetime64(ns)	時刻
meridional current	float32	南北流速 (m/s)

エラーハンドリング

ダウンロードエラー：データの取得処理中に発生するエラー

- ログに記録する

リトライエラー：ダウンロードエラーが発生した場合に、再試行を最大 3 回繰り返す処理に関するエラー

- ダウンロードエラーが発生した場合に再試行を 3 回行う、それでもデータが取得で

きなかった場合はログに記録し、管理者にアラートを送信する。

- Request した日付分はデータ欠如とする。

7.1.1.2. 漁船航行データ (GPS)

情報発信システム(NABRAS)が管理するデータベース上に保存されている船舶の GPS データから位置、針路、速力、航行、水深状態情報を取得する API。なお、情報発信システム(NABRAS)からのデータ取得については認証を設定することとしているため、認証情報については高知県経由で担当ベンダーに問い合わせて確認すること。

Request

GET https://{NABRAS_DOMAIN}?date=yyyy-mm-dd

- date: データを取得したい対象日

Response

Header

ヘッダー名	値
Content-Type	application/json
Content-Encoding	gzip
Access-Control-Allow-Origin	https://{MEJICA_SYSTEM_DOMAIN} ※{MEJICA_SYSTEM_DOMAIN}: 本システムのドメイン

Body

Top Level フィールド:

フィールド名	データ型	説明
timestamp	DateTime (ISO 8601)	レスポンスの取得日時
data	List	GPS データのリスト(詳細は下表を参照)

data 配列の各要素:

フィールド名	データ型	説明
timestamp	DateTime (ISO 8601)	漁船データ取得日時
vessel	String	船名
vessel_speed	Float	船速 (m/s)
vessel_heading_angle	Integer	船首 (度)
latitude	Double	緯度
longitude	Double	経度
speed	Double	速度 (m/s)
line_cord	Integer	ラインコード

JSON Schema

API レスポンスのスキーマ例は以下の通り。


```
{
  "timestamp":"2024-01-03T15:45:00Z",
  "data":[
    {
      "timestamp":"2024-01-03T15:45:00Z",
      "vessel": "漁船 A",
      "vessel_speed": 0.3,
      "vessel_head": 4,
      "latitude": 38.9231,
      "longitude": 141.6742,
      "speed": 7.5,
      "line_cord": 1
    }
  ]
}
```


エラーハンドリング

リトライエラー

- 再試行を 3 回行う、それでもデータが取得できなかった場合はログに記録し、管理

者にアラートを送信する。

- Request した日付分はデータ欠如とする
- 管理者は不足分を補うため手動で次の予測実行時間になるまでにデータ取得すること。

7.1.1.3. 漁獲量データ

漁で得られた魚種別の漁獲尾数を取得するための API。なお、情報発信システム(NABRAS)からのデータ取得については認証を設定することとしているため、認証情報については高知県経由で担当ベンダーに問い合わせて確認すること。

Request

GET https://{NABRAS_DOMAIN}/amount?date=yyyy-mm-dd

- date: データを取得したい対象日

Response

Header

ヘッダー名	値
Content-Type	application/json
Content-Encoding	gzip
Access-Control-Allow-Origin	https://{MEJICA_SYSTEM_DOMAIN} ※{MEJICA_SYSTEM_DOMAIN}: 本システムのドメイン

Body

Top Level フィールド:

フィールド名	データ型	説明
timestamp	DateTime (ISO 8601)	response が返ってきた時刻
data	object	漁獲量データのリスト(詳細は下表を参照)

data の各要素:

フィールド名	データ型	説明
date	DateTime (ISO 8601)	データを取得した時刻
vessel	String	漁船名称
market	String	市場名称
fishery_type	String	漁業種名称
fish_name	String	魚種名称
amount	Integer	漁獲量(kg)

JSON Schema

API レスポンスのスキーマ例は以下の通り。

```
{
  "timestamp": "2024-01-03T18:30:00Z",
  "data": [
    {
      "date": "2024/01/03",
      "vessel": "漁船 A",
      "market": "下ノ加江",
      "speaies": "目近漁",
      "fish_name": "マルソウダガツオ",
      "amount": 126
    }, ...
  ]
}
```

エラーハンドリング

リトライエラー

- 再試行を 3 回行う、それでもデータが取得できなかった場合はログに記録し、管理者にアラートを送信する。
- Request した日付分はデータ欠如とする
- 管理者は、不足分を補うため手動で次の予測実行時間になるまでにデータ取得すること。

7.2. モデル学習管理

7.2.1. データ取得

学習データを作成するため、ローデータデータベースの各テーブルからデータを既定の期間分取得する。使用する期間は予測モデルの構造に依存するため、構築フェーズにおいて適切な値を検証して設定することとする。

データ取得対象となるテーブル

データを取得する候補となるテーブルは以下の通り。ただし、実際に使用するデータ(対象テーブル)は予測モデルに依存するため、使用するテーブルは構築フェーズにおいて決めることとする。

- *sea_condition*
- *fishing_boat_navigation*
- *haul_medica*

注記

※1: 各テーブルの定義は「データベース設計/データ管理」を参照。

7.2.2. ストレージ

モデル学習処理が完了した際に出力される以下のファイルをストレージに保存する。

1. メジカ漁場予測モデル (weight.pth)
2. 評価結果 (metrics.csv)
3. 学習ログ (train.log)

注記

※2: フォルダ構成については「データベース設計/モデル学習管理」を参照。

7.3. 予測結果管理

7.3.1. データ取得

予測用データを作成するため、ローデータデータベースの各テーブルからデータを既定の期間分取得する。使用する期間は予測モデルの構造に依存するため、構築フェーズにおいて適切な値を検証して設定することとする。

データ取得対象となるテーブル

データを取得する候補となるテーブルは以下の通り。ただし、実際に使用するデータ(対象テーブル)は予測モデルに依存するため、使用するテーブルは構築フェーズにおいて決めることとする。

- *sea_condition*
- *fishing_boat_navigation*
- *haul_medica*

7.3.2. 予測処理

予測データ

7.3.1. データ取得で取得したデータをもとに予測データを作成し入力データとする。

学習済み予測モデル

ストレージから最新の学習済み予測モデル（weight.pth）を取得しロードする。

注記

※1: ストレージの詳細はデータベース設計/モデル学習管理を参照。

予測結果

各緯度・経度ごとの confidence （確信度）の値を出力し、予測結果データベースに格納する。予測結果データベースのテーブル詳細はデータベース設計/予測結果管理を参照。

7.3.3. API

予測対象日の予測結果データを予測結果データベースから取得する。

Header

ヘッダー名	値
Content-Type	application/json
Content-Encoding	gzip

Request

GET api/v1/visualization?target-date=yyyymmdd

- target-date: 取得したい予測対象日(yyyymmdd の形式で指定する)

注記

Response

Top level フィールド:

フィールド名	データ型	説明
timestamp	Timestamp (ISO 8601)	取得日時
data	List	詳細データのリスト

data 配列の各要素:

フィールド名	データ型	説明
id	Integer	識別子
latitude	Double	緯度
longitude	Double	経度

confidence	Float	予測の確信度
color	String	予測結果を描画する際に使用する色

JSON Schema

API レスポンスのスキーマ例は以下の通り。

```
{
  "timestamp": "2024-01-03T10:30:00Z",
  "data": [
    {
      "id": 1,
      "latitude": 35.123456,
      "longitide": 35.234567,
      "confidence": 0.3,
      "color": "#FF0000"
    },
    {
      "id": 2,
      "latitude": 34.987654,
      "longitude": 35.098765,
      "confidence": 0.9,
      "color": "#00FF00"
    }
  ]
}
```

7.4. 画面管理

情報発信システム(NABRAS)からリクエストを受け付けて、予測結果を表示する URL を返す。

注記

※1: 取得データ不足などで予測対象日の予測結果がない場合は、エラーメッセージを前日の出力 UI とともに表示する。
※2: 出力 UI の詳細は「システム設計/画面 UI」を参照。

UI

公開範囲

- Basic 認証を設定すること。

Header

ヘッダー名	値
Content-Type	application/json

Request

GET https://{MEJICA_SYSTEM_DOMAIN}/ui?date=yyyymmdd

- date: 取得したい予測対象日の日付。yyyymmdd の形式で指定する。

Response

JSON Schema

API レスポンスのスキーマ例は以下の通り。

```
{
  "status": "success",
  "data": "https://{MEJICA_SYSTEM_DOMAIN}/visualization/yyyymmdd"
```

8. 予測モデル設計

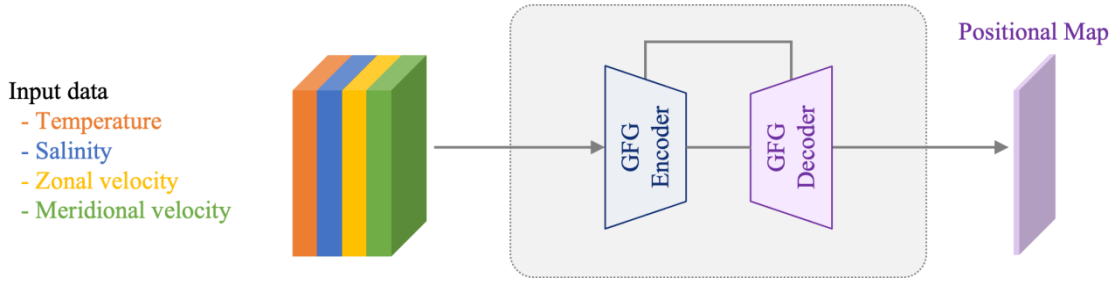
本章では、メジカ漁場予測システムにおける予測モデルについて記載する。

8.1. 予測モデルのアーキテクチャ

8.1.1. 全体構造

令和 7 年度の本システムの構築フェーズにて採用するモデルの構造は以下をベースとする。ただし、以下の点に注意すること。

- 本システムは令和 8 年度以降も継続的にデータ収集・モデルの再学習を行なっていくため、容易に改変可能なモデルとすること
- 構築フェーズで作成するモデルは以下の構造をベースとするが、高精度な結果が得られるモデル構造が得られた場合は以下の構造から改変しても良い。ただし、出力結果として得られるデータはデータベースと整合性が取れるものとすること
    - 予測結果のデータ形式については「データベース設計/予測結果管理」を参照すること



8.1.2. 各種モジュールについて

入力データ

令和 7 年度の構築フェーズで使用する入力データは海況データの以下の 4 種類である。

- 水温（Temperature）

- 塩分（Salinity）
- 東西流速（Zonal velocity）
- 南北流速（Meridional velocity）

なお、他のデータについては、予測モデルの精度がより向上することが確認できた場合は使用しても良いこととする。

Good Fishing Ground (GFG) Encoder

入力データから特徴量を抽出するためのエンコーダーモジュールである。主な機能は以下の通り。

- 空間的特徴の抽出
    - 畳み込み層による局所的な特徴抽出
- 活性化関数
    - 活性化関数による非線形変換
- スキップコネクション
    - Encoder から Decoder への接続

Good Fishing Ground (GFG) Decoder

エンコードされた特徴量から予測結果を生成するデコーダーモジュールである。主な機能は以下の通り。

- アップサンプリングによる特徴マップの拡大
    - 空間解像度の段階的な復元
- 特徴量の統合
    - 空間特徴と Encoder の中間特徴量の統合
- 予測値の生成
    - 確率分布への変換

出力データ

本システムの出力データは以下の形式で出力すること。

- 形式
    - 緯度経度ごとにおける確率値の 2 次元マップ
- 出力値
    - 各点における良い漁場である確率値（0～1）

8.2. 評価方法

8.2.1. 前提条件

予測モデルを学習・評価するため、漁船の GPS 情報から実際に漁をしていた緯度経度を良い漁場である地点として 1、その他の地点を 0 とする 2 値マップを使用する。

2 値マップの値が 1 の地点は、実際に漁をしていることから良い漁場=正例である。一方、その他の地点は良い漁場であるが漁をしていないのか、良い漁場でないから漁をしていないのかが判別できないため、ラベル無しデータとなる。すなわち、漁場ではない=負例の情報がないため、一般的に用いられることが多い評価指標である F 値を求めることができない。

そこで、正例とラベル無しデータから計算可能な F 値に相当する指標[1]を、予測モデルの評価に使用する。

[1]Learning with Positive and Unlabeled Examples Using Weighted Logistic Regression

8.2.2. F 値に相当する指標

F 値は式①で表されるが、precision と recall の両方が大きくならなければ、F 値は大きくならない。

$$F\text{値} = \frac{2 \cdot \mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \quad \cdots ①$$

precision はモデルが正と予測した地点の内の正例の割合であることから、式②で表される。

$$\mathrm{precision} = \Pr[Y = 1 | f(X) = 1] \cdots ②$$

recall は正例である地点の内、モデルが正と予測した地点の割合であることから式③で表される。

$$recall = Pr[f(X) = 1|Y = 1] \cdots ③$$

前提条件で述べた通り、正例の情報はあるが負例の情報がないため、recall は計算できるが precision は求めることができない。

ここで、ベイズの定理より式④が成り立つ。

$$Pr[Y = 1|f(X) = 1]Pr[f(X) = 1] = Pr[f(X) = 1|Y = 1]Pr[Y = 1] \cdots ④$$

式④の両辺に recall を掛け、式②、③を代入し整理すると、式⑤が導かれる。

$$\frac{precision \cdot recall}{Pr[Y = 1]} = \frac{recall^2}{Pr[f(X) = 1]} \cdots ⑤$$

式⑤の左辺の分母はデータの正例の割合であり、評価対象となる緯度・経度の範囲が固定であれば定数であるため、左辺はＦ値のように precision と recall の両方が大きくならなければ値が大きくならない指標となる。すなわち、Ｆ値と同様の特性を持つことがわかる。

よって、式⑤の右辺を計算することでＦ値に相当する指標を計算でき、これをもって同じデータに対するモデルの評価が可能である。

8.2.3. 評価対象の範囲と精度目標

評価対象の範囲は、以下の条件を満たす必要がある。

1. 漁をした地点(=正例)全てを含む

   正例の情報を最大限に活用し、モデルの評価精度を向上させるため。

2. 評価対象範囲を統一する

評価対象となる範囲（緯度・経度）を変更すると、ラベル無しデータの数が増減し式⑤右辺の分母の定数値が変わるため、F 値に相当する指標も変化する。そのため、モデル評価の一貫性を極力保つために、評価対象の範囲（緯度・経度）は原則固定する。評価範囲を変更してモデルの再学習を行う必要がある場合には、定量的な評価指標だけではなく定性的に予測結果が妥当かを評価する。

初期の構築では評価対象の範囲を高知県沿岸の以下の範囲とする。ただし、より適切な範囲が設定できるのであれば、高知県と協議の上変更しても良いこととする。

- 北緯 32.0°～33.6°（58 点）
- 東経 132.0°～134.0°（72 点）

また、モデルが満たすべき精度として、上記の条件において評価用データに対して F 値に相当する指標⑤は 5 以上を目標とすることとする。