H2020 - INDUSTRIAL LEADERSHIP - Leadership in enabling and industrial technologies - Information and Communication Technologies (ICT)

ICT-14-2016: Big Data PPP: cross-sectorial and cross-lingual data integration and experimentation

BigDataOcean

“Exploiting Oceans of Data for Maritime Applications”

D3.2 BigDataOcean Harmonisation, Knowledge Extraction, Business Intelligence and Usage Analytics Services

Workpackage:	WP3 – Cross-Sector Semantics, Analytics and Business Intelligence Algorithms
Authors:	Konstantinos Perakis, Dimitris Miltiadou (UBITECH), Spiros Mouzakitis, Dimitris Papaspyros, Giannis Tsapelas, Panagiotis Kokkinakos (NTUA), Ioanna Lytra, Fabrizio Orlandi (UBONN), Konstantinos Chatzikokolakis (EXMILE), Nuno Amaro (NESTER), Evi Bourma, Maria Sotiropoulou, Antonis Chalkiopoulos (HCMR), Evangelos Trikas (FOINIKAS)
Status:	Final
Date:	31/08/2017
Version:	1.00
Classification:	Public

Disclaimer:

The BigDataOcean project is co-funded by the Horizon 2020 Programme of the European Union. This document reflects only authors’ views. The EC is not liable for any use that may be done of the information contained therein

BigDataOcean Project Profile

Grant Agreement No.: 732310

Acronym:	BigDataOcean
Title:	Exploiting Oceans of Data for Maritime Applications
URL:	http://www.bigdataocean.eu/site/
Start Date:	01/01/2017
Duration:	30 months

Partners

	National Technical University of Athens (NTUA), Decision Support Systems Laboratory, DSSLab Co-ordinator	Greece
	Exmile Solutions Limited (EXMILE)	United Kingdom
	Rheinische Friedrich-Wilhelms-Universitt Bonn (UBONN)	Germany
	Centro de Investigacao em Energia REN – State Grid, S.A. – R&D Nester (NESTER)	Portugal
	Hellenic Centre for Marine Research (HCMR)	Greece
	Ubitech Limited (UBITECH)	Cyprus
	Foinikas Shipping Company (FOINIKAS)	Greece
	Istituto Superiore Mario Boella (ISMB)	Italy
	Instituto de Desenvolvimento de Novas Tecnologias (UNINOVA)	Portugal
	Anonymi Naftiliaki Etaireia Kritis (ANEK)	Greece

Document History

Version	Date	Author (Partner)	Remarks
0.10	08/05/2017	Dimitrios Miltiadou (UBITECH)	ToC
0.20	23/06/2017	Konstantinos Perakis, Dimitris Miltiadou (UBITECH)	Knowledge Extraction & Business Intelligence Algorithms – 1st Draft
0.30	14/07/2017	Konstantinos Chatzikokolakis (EXMILE)	Processing patterns for Maritime Security & Anomaly Detection – 1st Draft
0.30_HCMR	24/07/2017	Evi Bourma, Maria Sotiropoulou, Antonis Chalkiopoulos (HCMR)	Processing patterns for Mare Protection
0.30_NESTER	24/07/2017	Nuno Amaro (NESTER)	Processing patterns for Wave Power Exploitation
0.40	26/07/2017	Konstantinos Perakis (UBITECH)	Version Consolidation plus minor additions
0.50	03/08/2017	Konstantinos Perakis (UBITECH)	Finalisation of Chapter 1, Additions to Chapter 4.1
0.60	10/08/2017	Ioanna Lytra, Fabrizio Orlandi (UBONN)	Delivery of Chapter 2
0.70	10/08/2017	Spiros Mouzakitis, Dimitris Papaspyros, Giannis Tsapelas (NTUA)	Delivery of Chapter 4.2
0.80	11/08/2017	Konstantinos Perakis (UBITECH)	Consolidation of contributions, preparation of review ready version & request for minor additions
0.90	25/08/2017	Dimitrios Miltiadou (UBITECH)	Review ready version
0.90_NESTER	28/08/2017	Nuno Amaro (NESTER)	Official review by NESTER
0.90_NTUA	29/08/17	Spiros Mouzakitis, Dimitris Papaspyros, Giannis Tsapelas, Panagiotis Kokkinakos (NTUA)	Official review by NTUA
1.0	31/08/2017	Dimitrios Miltiadou (UBITECH)	Final version

Executive Summary

The scope of D3.2 is to document the preliminary efforts undertaken within the context of Tasks T3.2 - Multi-Source Big Data Harmonisation and Processing Patterns for Maritime Applications, and T3.3 - Knowledge Extraction, Business Intelligence and Usage Analytics Algorithms.

This report defines the services which will provide the means for collecting and harmonising multi source big maritime data upon receiving as input the maritime information sources identified within the context of Task 2.2 and documented in deliverable D2.1, along with the big data semantic vocabularies analysed and documented within the context of deliverable D3.1.

Furthermore, the maritime data stakeholders needs which were identified within the context of Task 2.2 and documented in deliverable D2.1, were taken into consideration on the definition of the services which will provide the means for processing multi-source big maritime data, as per the processing patterns defined by the project pilot partners.

Finally, this report defines the algorithms that will facilitate the proper execution flow of the four project pilots at first, and of future services as the project evolves upon carefully analysing the maritime data stakeholders needs identified within the context of Task 2.2 and documented in deliverable D2.1, the user stories collected within the context of T4.2 and documented in deliverable D4.1, as well as the initial architectural decisions taken within the context of Task 4.4 and documented in deliverable D4.2.

In particular, the scope of the current report is the preliminary outcome of the aforementioned tasks and the definition of the BigDataOcean platform services related to harmonisation, knowledge extraction, business intelligence and usage analytics taking into consideration that the delivery of these services is a living process that will last until M19, when the second and final version of the current deliverable, namely D3.3 will be delivered, with forthcoming updates on the services based on further identified functional requirements, mainly originating from the pilots of the project, being merged into the final version of this deliverable.

Table of Contents

1 Introduction 10

1.1 Objective of the deliverable 10

1.2 Structure of the deliverable 10

1.3 Positioning within the project 11

2 Harmonisation 12

2.1 Goal 12

2.2 Seamless collection of Big Data Ocean datasets 12

2.2.1 GeoDCAT Application Profile 12

2.2.2 Additional GeoDCAT properties in Big Data Ocean 16

2.2.3 Usage Example 16

2.3 Seamless Harmonisation and Processing of Big Data Ocean Datasets 19

2.4 Examples from BDO Pilots 20

2.4.1 Wave Power 20

2.4.1.1 Datasets 20

2.4.1.2 Query 22

2.4.1.3 Harmonisation 23

2.4.2 Maritime Security 23

2.4.2.1 Datasets 23

2.4.2.2 Query 25

2.4.2.3 Harmonisation 25

2.4.3 Mare Protection 25

2.4.3.1 Datasets 25

2.4.3.2 Query 28

2.4.3.3 Harmonisation 28

2.4.4 Maintenance Recommendations 28

2.4.4.1 Datasets 28

2.4.4.2 Query 30

2.4.4.3 Harmonisation 30

3 Processing Patterns 31

3.1 Processing patterns for Maritime Security and Anomaly Detection 31

3.1.1 Introduction 31

3.1.2 Definition of Anomaly 31

3.1.3 BigDataOcean Approach 34

3.1.3.1 Data sources and data acquisition process 35

3.1.3.2 Raw Data pre-processing / cleaning 35

3.1.3.3 Data enrichment 37
3.1.3.4 BigDataOcean Anomaly Detection approach 38
3.2 Processing patterns for Mare Protection 41
3.2.1 The POSEIDON OSM processing chain 41
3.2.1.1 Input and output files' information 42
3.2.1.2 Description of the oil spill model algorithms 42
3.2.2 Big Data Ocean Approach 44
3.3 Processing patterns for Wave Power exploitation 46
3.3.1 Pilot data services 46
3.3.2 Pilot Methodology 49
3.3.2.1 Data sources and data acquisition process 49
3.3.2.2 Raw Data pre-processing / cleaning 50
3.3.2.3 Handling inconsistent entries and data cleaning 51
3.3.2.4 Data enrichment 52
3.3.2.5 Processing / Knowledge extraction 52
3.3.2.6 Feature Selection / Clustering 52
3.4 Processing patterns for Vessel Fault Prediction 53
3.4.1 Introduction 53
3.4.2 ANEK LINES scope, databases and raw input data 53
3.4.3 FOINIKAS Shipping scope, databases and raw input data 55
3.4.4 Big Data Ocean approach 57
3.5 Other processing patterns for Maritime Applications 60
4 Algorithms 61
4.1 Knowledge extraction and Business Intelligence 61
4.1.1 Basic Statistics 61
4.1.1.1 Summary Statistics 61
4.1.1.2 Correlations 64
4.1.1.3 Sampling 66
4.1.1.4 Hypothesis Testing 67
4.1.1.5 Random Data Generation 69
4.1.2 Machine Learning 69
4.1.2.1 Classification & Regression 70
4.1.2.2 Collaborative Filtering (Recommendation) 78
4.1.2.3 Clustering 80
4.1.3 Featurisation 84
4.1.3.1 Dimensionality Reduction 84

4.1.3.2 Feature Extraction 86
4.1.3.3 Feature Selection 87
4.1.4 List of algorithms 88
4.2 Usage Analytics 96
4.2.1 Platform Usage Analytics 96
4.2.2 Data Usage Analytics 98
4.2.3 Service Usage Analytics 99
5 Conclusions 101
References 102
Annex I: Mare Protection Details 105
Environmental Input files 105
HYDRODYNAMIC DATA 105
METEOROLOGICAL DATA: 106
WAVES DATA: 107
Oil spill Input and Output files: 108
INPUT FILE STRUCTURE: 108
OUTPUT FILE STRUCTURE: 110

List of Figures

Figure 3-1: Maritime Security Anomaly Detection cases 33
Figure 3-2: The POSEIDON Oil Spill Model processing chain 41
Figure 3-3: Mare Protection Pilot through the BDO platform 44
Figure 3-4: Wave power as the next clean energy source: data processing methodology........ 46
Figure 3-5: Wave power as the next clean energy source: functional methodology 46
Figure 3-6: Preliminary resource assessment methodology 47
Figure 3-7: Complete resource assessment methodology........ 48
Figure 3-8: Preliminary environment impact assessment methodology 49
Figure 3-9: Vessel Fault Prediction 58

List of Tables

Table 2-1: GeoDCAT-AP vocabularies 13
Table 2-2: GeoDCAT-AP metadata elements 15
Table 2-3: GeoDCAT-AP additional properties 16
Table 2-4: Usage examples metadata........ 19
Table 2-5: Query example results 20
Table 2-6: Wave power example dataset........ 22
Table 2-7: Wave power query results example........ 23
Table 2-8: Maritime security example dataset 25
Table 2-9: Mare protection example dataset 28
Table 2-10: Mare protection query results example 28
Table 2-11: Maintenance recommendations example dataset........ 30
Table 2-12: Maintenance recommendations example query results 30
Table 3-1: Fuzzy logic rules 39
Table 4-1: List of classification and regression algorithms........ 77
Table 4-2: List of available algorithms 91
Table 4-3: Mapping pilot needs to algorithms available........ 92
Table 4-4: Mapping algorithms available to pilot needs........ 95

Abbreviations

Abbreviation	Description
AIS	Automatic Identification System
ALS	Alternating Least Squares
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BDO	BigDataOcean
CART	Classification And Regression Tree
D	Deliverable
DCAT	Data Catalog Vocabulary
DBSCAN-SD	Density-Based Spatial Clustering of Applications with Noise considering Speed and Direction
FIS	Fuzzy Inference System
GMM	Gaussian Mixture Model
GUI	Graphical User Interface
ICT	Information and Communications Technology
IMO	International Maritime Organisation
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
M	Month
MDA	Maritime Domain Awareness
ML	Machine Learning
MLib	SPARK's Machine Learning Library
MMSI	Maritime Mobile Service Identity
NETCDF	Network Common Data Form
OSM	Oil spill model
PCA	Principal component analysis
RDD	Resilient Distributed Datasets
SRS	Simple Random Sample
SVM	Support vector machines
WP	Workpackage
W3C	World Wide Web Consortium

1 Introduction

1.1 Objective of the deliverable

The scope of D3.2 is to document the preliminary efforts undertaken within the context of Tasks T3.2 – Multi-Source Big Data Harmonisation and Processing Patterns for Maritime Applications, and T3.3 – Knowledge Extraction, Business Intelligence and Usage Analytics Algorithms. Towards this end, the scope of the current deliverable is threefold.

Firstly, D3.2 aims at building upon the maritime information sources identified within the context of Task 2.2 and documented in deliverable D2.1, as well as upon the big data semantic vocabularies analysed and documented within the context of deliverable D3.1, allowing stakeholders to reference and use metadata shared by multiple sources and data providers, in order to define the services which will provide the means for collecting and harmonising multi source big maritime data.

Secondly, D3.2 aims at building upon the maritime data stakeholders needs identified within the context of Task 2.2 and documented in deliverable D2.1, in order to define the services which will provide the means for processing multi-source big maritime data, as per the processing patterns defined the project pilot partners.

Thirdly, D3.2 aims at building upon the user stories collected within the context of T4.2 and documented in deliverable D4.1, as well as upon the initial architectural decisions taken within the context of Task 4.4 and documented in deliverable D4.2, in order to define the algorithms that will facilitate the proper execution flow of the four project pilots at first, and of future services as the project evolves.

It should be noted that the delivery of the BigDataOcean platform services related to harmonisation, knowledge extraction, business intelligence and usage analytics is a living process that will last until M19, when the second and final version of the current deliverable, namely D3.3 will be delivered, with updates based upon further identified functional requirements translated into the corresponding services, stemming mainly from feedback received by the project's pilots as well as probably from external stakeholders, being merged into future the final version of this deliverable, updating and customising or even modifying the approaches taken accordingly.

1.2 Structure of the deliverable

Deliverable 3.2 is organised in five main sections as indicated in the table of contents.

1. The first section introduces the deliverable. It documents the scope of the deliverable and briefly describes how the document is structured. It also documents the positioning of the deliverable in the project, namely the relation of the current deliverable with the other deliverables, and how the knowledge produced in the other deliverables and work-packages served as input to the current deliverable.
2. Following the introductory section, section 2 defines the services which will provide the means for collecting and harmonising multi-source big maritime data. To facilitate the collection of the multi-source big maritime data, the main characteristics of the common vocabulary, DCAT, whose terms will be used to describe all datasets in BigDataOcean and GeoDCAT-AP which will be used on top of DCAT in order to include geospatial data, are documented along with the

necessary extensions needed to cover the dataset requirements on the maritime domain. Building upon these metadata as described, the seamless harmonisation and processing of the datasets of BigDataOcean is presented with a query execution example. Finally, examples with real datasets and real needs collected in collaboration with the BigDataOcean pilot partners are presented.

3. Section 3 builds upon the maritime data stakeholders needs identified and defines the services which will provide the means for processing multi-source big maritime data, as per the processing patterns defined by the project pilot partners. Section 3 documents the processing patterns (including the methodology of the pilots) of the four pilot applications / services, and leaves room for additional applications and services which may be identified during the project lifecycle.
4. Section 4 defines the algorithms that will facilitate the proper execution flow of the four project pilots at first, and of future services as the project evolves, describing their purpose and referencing their software implementations, if available. Section 4 also leaves room for additional mathematical algorithms which may also comprise custom implementations, if such needs are identified during the project lifecycle.
5. Section 5 concludes the deliverable. It outlines the main findings of the deliverable which will guide the future research and technological efforts of the consortium.
6. D3.2 also includes one annex documenting details related to the BigDataOcean pilot for Mare Protection. More specific the structure of the data of the environmental input file is described thoroughly along with the structure of the data of the input and output files for the oil spill scenario.

1.3 Positioning within the project

Deliverable D3.2 is the direct, preliminary outcome of Tasks T3.2 – Multi-Source Big Data Harmonisation and Processing Patterns for Maritime Applications, and T3.3 – Knowledge Extraction, Business Intelligence and Usage Analytics Algorithms. D3.2 receives as input the maritime information sources identified within the context of Task 2.2 and documented in deliverable D2.1, as well as the big data semantic vocabularies analysed and documented within the context of deliverable D3.1, so as to define the services which will provide the means for collecting and harmonising multi source big maritime data. D3.2 receives as input also the maritime data stakeholders needs, identified within the context of Task 2.2 and documented in deliverable D2.1, so as to define the services which will provide the means for processing multi source big maritime data, as per the processing patterns defined the project pilot partners. Last but not least D3.2 takes into consideration and integrates the documented outcomes of D4.2, comprising the direct outcome of T4.4, and more specifically capitalises upon the initial architectural decisions so as to define the algorithms that will facilitate the proper execution flow of the four project pilots at first, and of future services as the project evolves.

2 Harmonisation

2.1 Goal

The goal of the harmonisation task in Big Data Ocean is to define methods and tools that will be used for collecting, harmonising, and processing multi-source big maritime data. The objective of this task is to allow seamless collection, harmonisation, and processing of cross-sectorial heterogeneous data, in order to be easily consumable by various types of stakeholders and Big Data Ocean services. Collection refers to gathering metadata in order to describe the various data sources. Harmonisation means the ability to combine heterogeneous data sources.

2.2 Seamless collection of Big Data Ocean datasets

To enable the harmonisation of Big Data Ocean data sources, all datasets will be described in terms of a common vocabulary. DCAT[1] is one of the most popular vocabularies to facilitate interoperability between data catalogs published on the Web. On top of it, GeoDCAT-AP[2] extends its properties, in order to include geospatial data. Many of the Big Data Ocean datasets include measurements over space and time, for instance, vessel routes and positions, Poseidon in-situ datasets, weather data, buoys measurements, and so on[3]. Therefore, GeoDCAT-AP will be used to describe Big Data Ocean datasets; to address the requirements of maritime data, additional properties are suggested.

In the following subsections, the main characteristics of the GeoDCAT-AP vocabulary and the necessary extensions in order to cover the dataset requirements in the maritime domain are presented. In addition, a usage example for describing two datasets from the Copernicus In Situ marine dataset repository[4] is included.

2.2.1 GeoDCAT Application Profile

The objectives of the GeoDCAT-AP are to 1) provide a representation of geospatial metadata conforming to DCAT-AP and 2) provide an RDF-based representation of geospatial metadata, based on widely used vocabularies. GeoDCAT-AP is hence an extension of the DCAT application profile, based on the W3C's Data Catalogue vocabulary (DCAT), used to describe public sector datasets in Europe[5]. It follows the INSPIRE Metadata[6] technical guidelines based on ISO 19115[7] and ISO 19119.

The INSPIRE guidelines and directive include rules and specifications for the description of resources - datasets, series and services and aim at establishing a EU-wide cross-border spatial data infrastructure

[1] https://www.w3.org/TR/vocab-dcat/

[2] https://joinup.ec.europa.eu/node/139283

[3] A complete list is provided in Deliverable D3.1 (M6).

[4] http://marine.copernicus.eu/

[5] W3C, "Data Catalog Vocabulary (DCAT)," 2014. [Online]. Available: http://www.w3.org/TR/vocab-dcat/

[6] European Commission, "INSPIRE Metadata Implementing Rules: Technical Guidelines based on EN ISO 19115 and EN ISO 19119," 2013. [Online]. Available: http://inspire.ec.europa.eu/index.cfm/pageid/101

[7] ISO 19115, a standard of the International Organization for Standardization (ISO), defines how to describe geographical information.

to support EU environmental policies, as well as other policies or activities having an impact on the environment.

The basic use case that GeoDCAT-AP intends to enable is a cross-domain data portal search for datasets and making it easier to share descriptions of spatial datasets between spatial data portals and general data portals. In the context of Big Data Ocean, the use of GeoDCAT-AP will enable the development of services that will integrate and process seamlessly the underlying datasets.

GeoDCAT-AP makes use of various vocabularies in order to describe the dataset metadata, which are summarised in the following table.

adms - http://www.w3.org/ns/adms#	Asset Description Metadata Schema
cnt - http://www.w3.org/2011/content#	Representing Content in RDF 1.0
dc - http://purl.org/dc/elements/1.1/	Dublin Core Metadata Element Set, Version 1.1
dcat - http://www.w3.org/ns/dcat#	Data Catalog Vocabulary
dct - http://purl.org/dc/terms/	DCMI Metadata Terms
dctype - http://purl.org/dc/dcmitype/	DCMI Type Vocabulary
foaf - http://xmlns.com/foaf/0.1/	FOAF Vocabulary
gsp - http://www.opengis.net/ont/geosparql#	OGC GeoSPARQL
locn - http://www.w3.org/ns/locn#	ISA Programme Core Location Vocabulary
owl - http://www.w3.org/2002/07/owl#	OWL Web Ontology Language
prov - http://www.w3.org/ns/prov#	The PROV Ontology
rdf - http://www.w3.org/1999/02/22-rdf-syntax-ns#	Resource Description Framework (RDF): Concepts and Abstract Syntax
rdfs - http://www.w3.org/2000/01/rdf-schema#	RDF Vocabulary Description Language 1.0: RDF Schema
schema - http://schema.org/	schema.org
skos - http://www.w3.org/2004/02/skos/core#	SKOS Simple Knowledge Organisation System - Reference
vcard http://www.w3.org/2006/vcard/ns#	vCard Ontology
xsd - http://www.w3.org/2001/XMLSchema#	XML Schema Part 2: Datatypes Second Edition

Table 2-1: GeoDCAT-AP vocabularies

The metadata elements of the GeoDCAT-AP (Core and Extended) vocabulary that will be used to describe the metadata of the datasets used in the Big Data Ocean platform are listed in the following table, along with a short description. For each element, it is indicated whether the element is mandatory (M), optional (O), conditional (C), or recommended (R).

GeoDCAT-AP Property	Description
dct:title (M)	Dataset title
dct:description (M)	Abstract describing the dataset
dct:type (M)	Data resource type
foaf:homepage (O)	Online resource
dct:identifier (O)	Unique data source identifier
dct:language (O)	Data source language
dct:subject (M)	Topics related to the dataset
dcat:theme (R)	Keywords related to the dataset
dct:spatial (O)	Geographic location covered by the dataset (by four coordinates or by geographic identifier)
dct:temporal (O)	Temporal coverage of the dataset
dct:issued (R)	Date of publication of the dataset
dct:modified (R)	Date of last revision of the dataset
dct:provenance (O)	Provenance of the dataset
dct:conformsTo (O)	Coordinate reference system, Temporal reference system
dct:license (O)	Conditions for access and use
dct:accessRights (O)	Limitations on public access
dct:publisher (R)	Responsible party
dct:format (R)	Encoding
cnt:characterEncoding (R)	Character encoding used in the dataset
dct:accrualPeriodicity (O)	Denotes how often the dataset gets updated

rdfs:comment (C)	Spatial resolution of the dataset
adms:representationTechnique (M)	Spatial representation type

Table 2-2: GeoDCAT-AP metadata elements

For a concrete dataset, some of the aforementioned properties can get values from controlled vocabularies. For instance, according to the GeoDCAT-AP guidelines, the following register has to be used to describe topic categories http://inspire.ec.europa.eu/codelist/TopicCategory, while the register of coordinate reference systems included in the European Petroleum Survey Group (EPSG) Geodetic Parameter Dataset (http://www.opengis.net/def/crs/EPSG/, http://www.epsg-registry.org/) will provide values for coordinate reference systems. In addition, several other vocabularies - described in detail in Deliverable D3.1 -, such as the SWEET Ontology[8] can be used to describe some of the metadata. The following list provides some exemplary external vocabularies (not provided by INSPIRE) that are recommended by GeoDCAT-AP:

- Geographic identifiers
  - For marine regions:
    - Marine Regions http://www.marineregions.org/
    - SeaVoX salt and fresh water body gazetteer - https://www.bodc.ac.uk/data/codes_and_formats/seavox/
  - General:
    - DBpedia for Geographic Placenames - http://dbpedia.org/about
    - National gazetteer vocabularies where feasible
    - SeaVoX salt and fresh water body gazetteer for 'marine geonames' - https://www.bodc.ac.uk/data/codes_and_formats/seavox/
- Keywords (with controlled vocabularies):
  - For discipline: suggested vocabularies are
    - GEneral Multilingual Environmental Thesaurus (GEMET) - https://www.eionet.europa.eu/gemet/
  - General:
    - GEOSS Societal Benefit Areas - https://en.wikipedia.org/wiki/Societal_Benefit_Areas
    - GEneral Multilingual Environmental Thesaurus (GEMET) - https://www.eionet.europa.eu/gemet/
    - British Oceanographic Data centre - http://www.bodc.ac.uk/

Some of these vocabularies are already part of the Big Data Ocean Metadata Repository (see Deliverable D3.1).

[8] https://sweet.jpl.nasa.gov/sweet2.3

2.2.2 Additional GeoDCAT properties in Big Data Ocean

For the needs of Big Data Ocean, GeoDCAT-AP will be extended with additional properties, in order to capture characteristics of marine specific datasets. The following table includes a list of these additional properties. These additional properties were extracted from interviewing the pilot partners. This list is expected to be extended if additional properties are identified that cannot be expressed using GeoDCAT-AP.

Property	Description
dbo:verticalCoverage	Water depth covered by the dataset (for measurements in the sea)
dbo:temporalResolution	Granularity of measurements
dbo:gridResolution	Horizontal/spatial grid resolution

Table 2-3: GeoDCAT-AP additional properties

2.2.3 Usage Example

For providing a usage example of GeoDCAT to describe metadata in the context of Big Data Ocean, two datasets from Copernicus marine data repository were selected: (1) **MEDSEA_ANALYSIS_FORECAST_WAVES_006_011** composed by hourly wave parameters at 1/24º horizontal resolution covering the Mediterranean Sea and extending up to -18.125W into the Atlantic Ocean and (2) **MEDSEA_ANALYSIS_FORECAST_PHYS_006_001**, a coupled hydrodynamic-wave model implemented over the whole Mediterranean Basin. The following tables include metadata for the aforementioned datasets.

Property	Dataset 1	Dataset 2
dct:title	Mediterranean Sea Physics Analysis and Forecast	Mediterranean Sea Waves Hindcast and Forecast

dct:description	The physical component of the Mediterranean Forecasting System (Med-currents) is a coupled hydrodynamic-wave model implemented over the whole Mediterranean Basin. The model horizontal grid resolution is 1/16˚ (ca. 6-7 km) and has 72 unevenly spaced vertical levels.The hydrodynamics are supplied by the Nucleous for European Modelling of the Ocean (NEMO) while the wave component is provided by WaveWatch-III. The model solutions are corrected by the variational assimilation (based on a 3DVAR scheme) of temperature and salinity vertical profiles and along track satellite Sea Level Anomaly observations.	MEDSEA_ANALYSIS_FORECAST_WAVES_006_011 is the nominal product of the Mediterranean Sea Waves Forecasting system, composed by hourly wave parameters at 1/24º horizontal resolution covering the Mediterranean Sea and extending up to -18.125W into the Atlantic Ocean. The Mediterranean Forecasting System, waves forecast component, is a wave model based on WAM Cycle 4.5.4, a state of the art third generation wave model successfully used for the last 20 years for wave hindcasting and forecasting. In the wave model the continuous wave spectrum is approximated by means of step functions which are constant in a frequency-direction bin. The Med-waves modelling system resolves the prognostic part of the wave spectrum with 24 directional and 32 logarithmically distributed frequency bins. The Med-waves set up includes a coarse grid domain with a resolution of 1/6° covering the North Atlantic Ocean from 75° W to 10° E and from 70° N to 10° S and a nested fine grid domain with a resolution of 1/24° covering the Mediterranean Sea from 18.125° W to 36.2917° E and from 30.1875° N to 45.9792° S.
dct:type	http://inspire.ec.europa.eu/metadata-codelist/ResourceType/dataset	http://inspire.ec.europa.eu/metadata-codelist/ResourceType/dataset
foaf:homepage	http://marine.copernicus.eu/services-portfolio/access-to-products/?option=com_csw&view=details&product_id=MEDSEA_ANALYSIS_FORECAST_PHYS_006_001	http://marine.copernicus.eu/services-portfolio/access-to-products/?option=com_csw&view=details&product_id=MEDSEA_ANALYSIS_FORECAST_WAV_006_011

dct:identifier	MEDSEA_ANALYSIS_FORECAST_PHYS_006_001	MEDSEA_ANALYSIS_FORECAST_WAV_006_011
dct:language	en	en
dct:subject	https://inspire.ec.europa.eu/metadata-codelist/TopicCategory/oceans	https://inspire.ec.europa.eu/metadata-codelist/TopicCategory/oceans
dcat:theme	https://www.eionet.europa.eu/gemet/en/concept/14844	https://www.eionet.europa.eu/gemet/en/concept/9262 https://www.eionet.europa.eu/gemet/en/concept/14844
dct:spatial	15°W -> 36.25°E ; 30.1875°S -> 45.9375°N	18.12°W -> 36.30°E ; 30.17°S -> 45.98°N
dct:temporal	from 2013-01-01T00:00:00Z to Present	from 2016-08-01T00:00:00Z to Present
dct:issued	-	-
dct:modified	-	-
dct:provenance	-	-
dct:conformsTo	Spherical 3D CS (EPSG 6404)	WGS 84 (EPSG 4326)
dct:license	-	-
dct:accessRights	-	-
dct:publisher	CMEMS	CMEMS
dct:format	NetCDF https://www.unidata.ucar.edu/software/netcdf/	NetCDF https://www.unidata.ucar.edu/software/netcdf/
cnt:characterEncoding	-	-
dct:accrualPeriodicity	daily	daily
rdfs:comment	Good quality dataset	Good quality dataset

adms:representationTechnique	-	-
bdo:verticalCoverage	from -5500.0 to 0.0 (72 levels)	Surface
bdo:temporalResolution	daily-mean hourly-mean	hourly-instantaneous
bdo:gridResolution	0.063degree x 0.063degree	0.042degree x 0.042degree

Table 2-4: Usage examples metadata

2.3 Seamless Harmonisation and Processing of Big Data Ocean Datasets

In order to demonstrate the seamless harmonisation and processing of datasets in Big Data Ocean given the metadata descriptions described in the previous sections, the following query is used as an example:

Query the **sea water temperature**, **sea water salinity** on the sea surface and the **sea surface significant height** in the **Mediterranean Sea** since **2017-01**.

In order to evaluate this query, first of all, the right datasets need to be selected in terms of:

- *Variables measured*
- *Spatial coverage*
- *Temporal coverage*
- *Vertical coverage*

In this example, the two datasets described in the previous subsection should be selected since they meet the query restrictions, as illustrated in the following table.

	Mediterranean Sea Physics Analysis and Forecast	**Mediterranean Sea Waves Hindcast and Forecast**
Variables measured	ocean_mixed_layer_thickness () **sea_water_salinity (S)**	**sea_surface_wave_significant_height (SWH)** sea_surface_wave_period_at_variance_spectral_density_maximum ()

	sea_surface_height_above_sea_level (SSH) sea_water_potential_temperature (T) **sea_water_temperature (T)** ...	sea_surface_wave_mean_period_from_variance_spectral_density_inverse_frequency_moment (MWT) ...
Spatial coverage	**Mediterranean Sea**	**Mediterranean Sea**
Temporal coverage	2013-01 until today (**2017-01 - today included**)	2016-08 until today (**2017-01 - today included**)
Vertical coverage	from -5500.0 to **0.0** (72 levels)	**Surface**

Table 2-5: Query example results

As a second step, the corresponding queries will run against the two datasets and the partial results will be merged. Other use cases of harmonisation of Big Data Ocean datasets include:

1. Search of datasets according to specific criteria (datasets in a specific sea area, datasets including measurements in a specific depth, etc.)
2. Run analytics and visualisations over time and space (retrieve measurements for different time spans and in different geographical areas).

2.4 Examples from BDO Pilots

The following examples have been created with real datasets and real needs collected together with the BigDataOcean pilot partners.

2.4.1 Wave Power

2.4.1.1 Datasets

Property	Dataset 1	Dataset 2
dct:title	ATLANTIC-IBERIAN BISCAY IRISH- OCEAN PHYSICS ANALYSIS AND FORECAST	ATLANTIC-IBERIAN BISCAY IRISH- OCEAN WAVE ANALYSIS AND FORECAST
dct:description	The operational IBI (Iberian Biscay Irish) Ocean Analysis and Forecasting system, daily run by Puertos del Estado provides a 5-day hydrodynamic forecast including high frequency processes of paramount importance to characterise regional scale marine processes (i.e. tidal forcing, surges and high frequency atmospheric forcing, fresh water river discharge, etc). A weekly update of IBI downscaled analysis is also delivered as historic IBI best estimates. The system is based on a (eddy-resolving) NEMO model application run at 1/36° horizontal resolution.	The IBI MFC provides a short term (5-days) high-resolution wave forecast product for the IBI (Iberian Bis-cay Irish) area. The IBI MFC wave model system is daily run by Puertos del Estado and it is based on the MFWAM model, run on a grid of 10 km of horizontal resolution and forced with the ECMWMF wind data. Apart of generating the 5-day forecast product delivered to CMEMS users, the IBI MFC wave system is setting up to provide internally some coupling parameters adequate to be used as forcing in a coupled IBI NEMO ocean model forecast run.
dct:type	http://inspire.ec.europa.eu/metadata-codelist/ResourceType/dataset	http://inspire.ec.europa.eu/metadata-codelist/ResourceType/dataset
foaf:homepage	http://marine.copernicus.eu/services-portfolio/access-to-products/?option=com_csw&view=details&product_id=IBI_ANALYSIS_FORECAST_PHYS_005_001	http://marine.copernicus.eu/services-portfolio/access-to-products/?option=com_csw&view=details&product_id=IBI_ANALYSIS_FORECAST_WAV_005_005
dct:identifier	IBI_ANALYSIS_FORECAST_PHYS_005_001	IBI_ANALYSIS_FORECAST_WAV_005_005
dct:language	en	en
dct:subject	https://inspire.ec.europa.eu/metadata-codelist/TopicCategory/oceans	https://inspire.ec.europa.eu/metadata-codelist/TopicCategory/oceans
dcat:theme		https://www.eionet.europa.eu/gemet/en/concept/9262
dct:spatial	(19°W : 5°E) ; (26°N : 56°N)	(19°W : 5°E) ; (26°N : 56°N)
dct:temporal	from 2013-01-01T00:00:00Z to Present	from 2015-05-01T00:00:00Z to Present
dct:issued	-	-
dct:modified	-	-

dct:provenance	-	-
dct:conformsTo	WGS 84 (EPSG 32662)	WGS 84 (EPSG 4326)
dct:license	-	-
dct:accessRights	-	-
dct:publisher	CMEMS	CMEMS
dct:format	NetCDF https://www.unidata.ucar.edu/software/netcdf/	NetCDF https://www.unidata.ucar.edu/software/netcdf/
cnt:characterEncoding	-	-
dct:accrualPeriodicity	daily	daily
rdfs:comment	-	-
adms:representationTechnique	-	-
bdo:verticalCoverage	from -5500.0 to 0.0 (50 levels)	Surface
bdo:temporalResolution	daily-mean hourly-mean	hourly-mean
bdo:gridResolution	0.028degree x 0.028degree	0.1degree x 0.1degree

Table 2-6: Wave power example dataset

2.4.1.2 Query

What are the **eastward and northward sea water velocities** at sea surface level, the **sea surface height**, the **sea mean wave period** and **sea wave significant height** at the location (9.21°W: 39.56°N) since **2015-01**?

2.4.1.3 Harmonisation

	Dataset 1	Dataset 2
Variables measured	sea_water_salinity (S) **eastward_sea_water_velocity (3DUV)** **northward_sea_water_velocity (3DUV)** sea_water_potential_temperature_at_sea_floor (bottomT) **sea_surface_height_above_geoid (SSH)** (…)	sea_surface_primary_swell_wave_significant_height (SW1) **sea_surface_wave_mean_period_from_variance_spectral_density_second_frequency_moment (MWT)** sea_surface_wave_mean_period_from_variance_spectral_density_inverse_frequency_moment (MWT) sea_surface_wave_stokes_drift_y_velocity (VSDXY) sea_surface_wave_stokes_drift_x_velocity (VSDXY) **sea_surface_wave_significant_height (SWH)** (…)
Spatial coverage	**Atlantic - Iberian Biscay Irish - Ocean**	**Atlantic - Iberian Biscay Irish - Ocean**
Temporal coverage	from 2013-01-01T00:00:00Z to Present	from 2015-05-01T00:00:00Z to Present
Vertical coverage	from -5500.0 to 0.0 (50 levels)	Surface

Table 2-7: Wave power query results example

2.4.2 Maritime Security

2.4.2.1 Datasets

Property	Dataset 1	Dataset 2
dct:title	Vessel Positions	World Port Index
dct:description	Vessel position collected through the AIS by MarineTraffic.com	Location and physical characteristics of, and the facilities and services offered by major ports and terminals world-wide

dct:type	https://inspire.ec.europa.eu/metadata-codelist/ResourceType/dataset	https://inspire.ec.europa.eu/metadata-codelist/ResourceType/dataset
foaf:homepage	www.marinetraffic.com	https://msi.nga.mil/NGAPortal/MSI.portal?_nfpb=true&_pageLabel=msi_portal_page_62&pubCode=0015
dct:identifier	-	World Port Index (Pub 150)
dct:language	en	en
dct:subject	https://inspire.ec.europa.eu/metadata-codelist/TopicCategory/transportation	https://inspire.ec.europa.eu/metadata-codelist/TopicCategory/location
dcat:theme	https://www.eionet.europa.eu/gemet/en/theme/37	-
dct:spatial	Global coverage	Global coverage
dct:temporal	2011	2017
dct:issued	-	-
dct:modified	-	-
dct:provenance	-	-
dct:conformsTo	-	
dct:license	Proprietary	https://www.nga.mil/About/Pages/FOIA.aspx
dct:accessRights	-	-
dct:publisher	MarineTraffic	National Geospatial Intelligence Agency (USA)
dct:format	csv	Shapefile, pdf or MS Access DB
cnt:characterEncoding	UTF-8	-
dct:accrualPeriodicity	Per minute	-
rdfs:comment	-	-
adms:representationTechnique	-	-

bdo:verticalCoverage	Surface	Surface
bdo:temporalResolution	-	-
bdo:gridResolution	-	-

Table 2-8: Maritime security example dataset

2.4.2.2 Query

What are **port latitude**, **port longitude** and **vessel's position** of all vessels and ports from **01-01-2011** to **31-12-2011?**

2.4.2.3 Harmonisation

	Dataset 1	Dataset 2
Variables measured	vessel_id, vessel_status, vessel_speed, vessel_name, **vessel_latitude**, **vessel_longitude**, vessel_course, vessel_heading, **vessel_timestamp**	port_id, port_name, port_country, port_latitude, port_longitude, **port_geometry**
Spatial coverage	Global coverage	Global coverage
Temporal coverage	2011	2017
Vertical coverage	Surface	Surface

2.4.3 Mare Protection

2.4.3.1 Datasets

Property	Dataset 1	Dataset 2
dct:title	POSEIDON High resolution Aegean Model – Hindcast and Forecast datasets	POSEIDON WAM Cycle 4 for the Aegean – Hindcast and Forecast datasets
dct:description	The model domain covers the geographical area 19.5◦E – 30◦E and 30.4◦N – 41◦N) with a horizontal resolution of 1/30◦ and 24 sigma layers	The wave forecasting system was set-up as a nested configuration with a coarse grid covering the entire Mediterranean Sea at a spatial resolution of 0.1◦×0.1◦

	along the vertical with a logarithmic distribution near the surface and the bottom. The model includes parameterisation of the main Greek rivers (Axios, Aliakmonas, Nestos, Evros) while the inflow/outflow at the Dardanelles is treated with open boundary techniques. The Aegean Sea model is forced with hourly surface fluxes of momentum, heat and water provided by the Poseidon - ETA high resolution (1/20°) regional atmospheric model issuing forecasts for 5 days ahead. Boundary conditions at the western and eastern open boundaries of the Aegean Sea hydrodynamic model are provided on a daily basis (daily averaged fields) by the OGCM OPA model covering the whole Mediterranean Sea with a resolution of 1/16° and 72 levels in the vertical. The Aegean Sea model is re-initialised from the OGCM results once every week. The assimilation scheme is based on the Singular Evolutive Extended Kalman (SEEK) filter which is an error subspace extended Kalman filter that operates with low-rank error covariance matrices as a way to reduce the computational burden.	and a fine grid nested within the coarse grid. The domain of the fine grid covers the Aegean Sea between 30.4°N and 41°N, and between 19.5°E and 30°E at a spatial resolution of 1/30° ×1/30° resolving the wave spectrum at each grid point in 24 directional and 30 frequency (0.05Hz->0.79316Hz) bins. The wave models are based on the WAM Cycle-4 code. It is a third generation wave model, which computes spectra of random short-crested wind-generated waves. The WAM code can be used for shallow and deep-water calculations and can account for depth and current refraction. The following basic wave physics are accounted for in the WAM code: • Wave propagation in time and space • Wave generation by the wind • Shoaling and refraction due to depth • Shoaling and refraction due to current • White-capping and bottom friction • Quadruplet wave-wave interactions The wave forecasting system issues wave forecasts for the next 5 days forced with hourly analysis and forecast winds produced by the POSEIDON weather prediction system.
dct:type	http://inspire.ec.europa.eu/metadata-codelist/ResourceType/dataset	http://inspire.ec.europa.eu/metadata-codelist/ResourceType/dataset
foaf:homepage	http://tethys.hcmr.gr/opendap/hyrax/medess4ms/OCEAN/contents.html	http://tethys.hcmr.gr/opendap/hyrax/medess4ms/WAVES/contents.html

dct:identifier	20170804_hi-HCMR-OCEAN-POSEIDON-AEG-b20170802_FC03-fv01.00	20170806_hi-HCMR-WAVES-POSEIDON-AEG-b20170802_FC05-fv01.00
dct:language	en	en
dct:subject	https://inspire.ec.europa.eu/metadata-codelist/TopicCategory/oceans	https://inspire.ec.europa.eu/metadata-codelist/TopicCategory/oceans
dcat:theme	Aegean Sea	Aegean Sea
dct:spatial	Lon.[19.5 30]°E; Lat.[30.4 41] °N	Lon.[19.5 30]°E; Lat.[30.4 41]°N
dct:temporal	One year historical data and 5 days forecast	One year historical data and 5 days forecast
dct:issued		
dct:modified		
dct:provenance	POSEIDON High resolution Aegean Model	POSEIDON WAM Cycle 4 for the Aegean
dct:conformsTo		
dct:license	copyright	copyright
dct:accessRights		
dct:publisher	HCMR	HCMR
dct:format	NetCDF https://www.unidata.ucar.edu/software/netcdf/	NetCDF https://www.unidata.ucar.edu/software/netcdf/
cnt:characterEncoding		
dct:accrualPeriodicity	daily	daily
rdfs:comment	-	-
adms:representationTechnique	-	-
bdo:verticalCoverage	15 levels from surface to the bottom, keeping a higher vertical resolution close to the surface.	15 levels from surface to the bottom, keeping a higher vertical resolution close to the surface.
bdo:temporalResolution	1-h resolution (forecast) and 6-h resolution (analysis/historical data)	1-h resolution (forecast) and 6-h resolution (analysis/historical data)

bdo:gridResolution	1°/30 × 1°/30 (~3.5km × 3.5 km)	1°/30 × 1°/30 (~3.5km × 3.5 km)

Table 2-9: Mare protection example dataset

2.4.3.2 Query

What are the **Potential Temperature**, **Velocity Zonal Component** on the sea surface and the **Mean Wave Period** in the **Aegean Sea** since **2017-01-01**?

2.4.3.3 Harmonisation

	Dataset 1	Dataset 2
Variables measured	sea_water_salinity (psal) **sea_water_potential_temperature (potemp)** **sea_water_x_velocity (uvel)** sea_water_y_velocity (vvel)	sea_surface_wave_significant_height (whs) **sea_surface_wave_zero_upcrossing_period (wper)** sea_surface_wave_to_direction (wdir)
Spatial coverage	**Aegean Sea**	**Aegean Sea**
Temporal coverage	One year back and 5 days forecast	One year back and 5 days forecast
Vertical coverage	0, 5, 10, 15, 20, 30, 50, 80, 150, 300, 650, 1000, 1500, 2500, 3500	Sea surface

Table 2-10: Mare protection query results example

2.4.4 Maintenance Recommendations

2.4.4.1 Datasets

Property	Dataset 1	Dataset 2
dct:title	Planned maintenance system (PMS)	WEB SIGNALS Database
dct:description	PMS data are used for the technical ship management system for both planned and unplanned maintenance, defect reporting and	Operational data from the WEB SIGNALS Database (engine speed, consumptions, distance traveled, loading performance, discharging

	technical asset and data management. The planned Maintenance System data are used to streamline the planning, documentation and implementation of maintenance work and surveys onboard ship.	performance, weather condition, lubricants etc)
dct:type	http://inspire.ec.europa.eu/metadata-codelist/ResourceType/dataset	http://inspire.ec.europa.eu/metadata-codelist/ResourceType/dataset
foaf:homepage	Not Public	Not Public
dct:identifier	PMS_2012_2016	WEBSIGNALS_2012_2016
dct:language	en	en
dct:subject	https://inspire.ec.europa.eu/metadata-codelist/TopicCategory/transportation	https://inspire.ec.europa.eu/metadata-codelist/TopicCategory/transportation
dcat:theme	https://www.eionet.europa.eu/gemet/en/theme/37	https://www.eionet.europa.eu/gemet/en/theme/37
dct:spatial	Global coverage	Global coverage
dct:temporal	From 01.01.2012 to 31.12.2016	From 01.01.2012 to 31.12.2016
dct:issued	2.8.2017	2.8.2017
dct:modified	2.8.2017	2.8.2017
dct:provenance	-	-
dct:conformsTo	-	-
dct:license	Data are property of the company and may only be used confidentially within the project and the limits described in the Grant Agreement and the Consortium Agreement	Data are property of the company and may only be used confidentially within the project and the limits described in the Grant Agreement and the Consortium Agreement
dct:accessRights	Consortium members only	Consortium members only
dct:publisher	FOINIKAS	FOINIKAS
dct:format	Excel file(output from database query)	Excel file(output from database query)

cnt:characterEncoding	UTF-8	UTF-8
dct:accrualPeriodicity	daily	daily
rdfs:comment	-	-
adms:representationTechnique	-	-
bdo:verticalCoverage	-	-
bdo:temporalResolution	hourly	hourly
bdo:gridResolution	-	-

Table 2-11: Maintenance recommendations example dataset

2.4.4.2 Query

What is the **Sea Condition**, **Wind direction**, **average ship speed** and **fuel consumption** at the Diesel Engine 1 performance checking on the 02/01/2012?

2.4.4.3 Harmonisation

	Dataset 1	**Dataset 2**
Variables measured	Diesel generator engine no 1 Status	Sea Condition, Wind direction, average ship speed and fuel consumption
Spatial coverage	FROM MALTA TO KALILIMENE	FROM MALTA TO KALILIMENE
Temporal coverage	02/01/2012	02/01/2012
Vertical coverage	-	-

Table 2-12: Maintenance recommendations example query results

3 Processing Patterns

The current section provides information about the processing patterns of each of the four pilots of the project. More specifically, the current section shortly introduces each of the pilots in terms of their scope and analyses the required inputs and desired outputs, describing the methodology (including the processing of the initial raw data) followed in order to reach the aspired results.

3.1 Processing patterns for Maritime Security and Anomaly Detection

3.1.1 Introduction

This section provides a detailed analysis of the processing patterns to be developed for the maritime security and anomaly detection pilot. Specifically, the methodology and algorithmic approach to be followed for the realisation of this pilot will be described in the following subsections. The various types of anomalies in maritime shipping are firstly analysed and classified to either static or dynamic depending on the vessels' and voyages' characteristics. Following this, a brief state of the art analysis of anomaly detection techniques is presented before the algorithmic analysis and the tools used in BigDataOcean are described.

3.1.2 Definition of Anomaly

Maritime Domain Awareness (MDA) is the effective understanding of activities, events and threats in the maritime environment that could impact the global safety, security, economic activity or the environment [1]. Recent advancements in Information and Communications Technologies (ICT) have created opportunities for increasing MDA, through better monitoring and understanding of vessel movements. The International Maritime Organisation (IMO) identified this issue as affecting the safety and efficiency of navigation and initiated a work program named e-Navigation to reduce the "confusion of profusion". The IMO defines e-Navigation as: "the harmonised collection, integration, exchange, presentation and analysis of maritime information onboard and ashore by electronic means to enhance berth to berth navigation and related services, for safety and security at sea and protection of the marine environment" (RD1). e-Navigation is expected to contribute to safer waterways, reducing accidents and environmental incidents through improved situational and traffic awareness both afloat and ashore [2].

While in the past though sea transport surveillance had suffered from a lack of data, current tracking technology (i.e., Automatic Identification System, AIS) has transformed the problem into one of data overload [3]. For the last decade AIS has been inseparable part of the modern maritime industry. The original purpose of the system was to provide vessels' crews the necessary information in order to avoid collisions in open seas and was designed to operate in range of a few kilometres (i.e., less than 50km). Even though the AIS system was not designed to be monitored in a centralised method, the maritime industry has been extremely interested in such systems (e.g., MarineTraffic, etc.).

Positional data together with the departure and destination ports transmitted in AIS messages can be used for route prediction and in conjunction with vessel's speed, time of arrival prediction is possible. Performing complex operations over such large datasets, though can give extra insights besides route prediction. For instance, combining route forecasts for multiple vessels can provide early warnings of possible collisions (by determining whether vessels' routes will meet in space and time) and actual route data can be used to perform various kinds of complex analytics (e.g. root-cause analysis in case of forensic investigation). In addition, improving the route analysis process can offer to various maritime stakeholders (e.g., shipping companies, charterers, insurance companies and port authorities) the opportunity to perform risk analysis and understand better any possible threats of vessels' manoeuvres, or even perform environment impact assessment, providing CO_2 emissions and fuel consumption predictions. Ultimately, AIS historical data can be used to determine actual sea lanes, their capacity and port connections, produce realistic vessel operational profiles that determine the normal behaviour of specific vessel types, detect any anomalies (i.e. irregular behaviour) and much more.

Anomaly is a "strange" deviation from a vessels' normal behaviour, meaning that it is inconsistent with, or straying from what is usual, normal or expected, or because it is not conforming to rules, laws or regulations [4] . Detecting an anomaly can be defined as a method that supports situation assessment by indicating objects and situations that deviate from the expected behaviour and thus may be of interest for further investigation. The understanding of the complex maritime environment and a vessel normal behaviour though, can never be limited to simply adding up and connecting various vessel positions as they travel across the seas. A combination of static information such as reporting information, vessel's flag (i.e., country), ship's owner, vessel's name, IMO and MMSI and destination port with dynamic information such as speed/course changes, proximity with other vessels or structures, etc. is needed to classify possible abnormal ship's behaviour. An anomaly can be classified as either static or dynamic depending on the vessel's characteristics that distinguish the behaviour as anomaly. Static anomalies are related to vessel's identification information mismatches or irregular changes. This information includes vessel's flag, IMO, MMSI, vessel's name and owner company. In addition, irregular changes in destination reported from AIS messages (particularly when the vessel is under-way) is a potential indicator of risk. Combining such information with port inspections or incident reports that prove vessels' are not conforming to regulations can also assist in static anomaly detection; thus classifying a vessel as potentially higher risk than others. Dynamic anomalies are mostly related to vessels' voyages and deviations from these. Speed or course changes, proximity with other vessels, and mismatches between the ship type and the sea lane (or zone) travelling are aspects that could constitute a dynamic anomaly. Figure 3-1 below captures all the dynamic and static data needed for performing Anomaly Detection through AIS data.

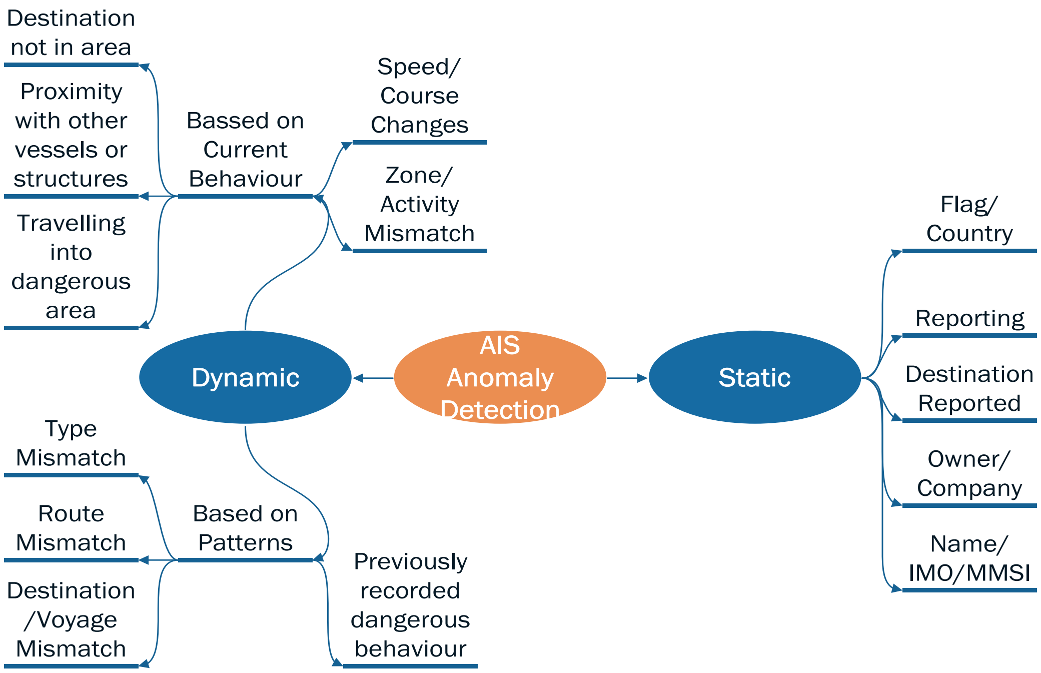


Figure 3-1: Maritime Security Anomaly Detection cases

Static anomaly detection is mostly treated as a decision-making process driven by risk identification/assessment in the related literature. Two classes of solutions are dominant in this perspective; the ones relying on probabilistic risk assessment and the ones using fuzzy logic as a relaxation approach to the definite boundaries of probabilistic approaches. Probabilistic risk assessment has been introduced as a solution for the assessment of risk in the maritime domain in [5]. Merrick and Dorp [6] applied a Bayesian simulation for the occurrence of situations with accident potential and a Bayesian multivariate regression analysis of the relationship between factors describing these situations and expert judgments of accident risk, to perform a full-scale assessment of risk and uncertainty. In their work [7], Balmat et al. propose a fuzzy approach in order to evaluate the maritime risk assessment applied to safety at sea and more particularly, the pollution prevention on the open sea. The proposed decision-making system exploits a set of open datasets (i.e., Lloyd's Register, IMO, EQUASIS, Paris MOU) combined with human expert experience to perform information analysis and define the risk factor. Besides Balmat's solution, other approaches [8,9] also rely on Fuzzy-Bayesian networks to model maritime security risks.

Dynamic anomaly detection is highly related to efficiently handling vast amount of (mostly positional) data. Previous works have been focused on extracting knowledge regarding motion patterns from AIS data in support of MDA including numerous methods of supervised and unsupervised clustering data mining techniques. In their work [10], Pallotta et al. propose the TREAD methodology as a method of automatically learning a statistical model for maritime traffic from AIS data in an unsupervised way as a framework for anomaly detection and route prediction. In [11], Ristic et al. use AIS data to extract motion patterns which are then used to construct the corresponding motion anomaly detectors. In relation to sea ports research and AIS, Ricci et al. made use of AIS to model maritime terminals operations, specifically focusing on The Port of Messina [12]. In their work, Wang et al. attempt to tackle the big data issue caused by the AIS data for anomaly detection purposes [13]. They implement a two-step process, where they firstly use an unsupervised technique, based upon the Density-Based Spatial Clustering of Applications with Noise considering Speed and Direction (DBSCAN-SD) incorporating non-spatial attributes, such as speed and direction, to label normal and abnormal position points of vessels based on the raw AIS data. Secondly, they train a supervised learning algorithm designed with the MapReduce paradigm running on Hadoop using the labelled data generated in from the first step. Spatial join queries, which combine trajectory datasets and a spatial objects dataset based on spatio-temporal predicates, have high computational requirements, which often lead to long query latencies. In [14], Ray et al. propose a parallel in-memory trajectory-based Spatiotemporal Topological join (PISTON), a parallel main memory query execution infrastructure designed specifically to address the difficulties of spatio-temporal joins. Generally, the methods which are used in the context of anomaly detection are based on statistical/probabilistic models [15–18], such as the Gaussian Mixture Model (GMM) and the adaptive Kernel Density Estimator (KDE) [11,19], Bayesian networks [20–23], but also neural networks [24–26] and hybrid approaches [27].

A number of prototype systems have been developed for experimental and operational reasons. For example, SeeCoast [28] is installed at Kount Harbor Operations Center in Portsmouth, Virginia. The system uses the Hawkeye system to fuse video data with radar signals and AIS messages to produce fused vessel tracks in or close by the port and reliably detect anomalies on such tracks. SCANMARIS [29] is a feedback based system tested at "Centre Régional Opérationnel de Surveillance et Sauvetage Corsen" on Ouessant traffic management. It uses a rule engine and a learning engine to process data fused from maritime traffic pictures, alert operators based on the rules defining anomalies and adapt its operation through the operators' feedback. LEPER [30], which was tested successfully at the Joint Interagency Task Force South (JIATF South), is a system that performs primitive geohashing using a military grid reference system upon which it decomposes ship's trajectories into sequences of discrete squares and uses Hidden Markov Model to calculate transition probabilities between grid locations. The predicted location is compared with the vessel's position (determined by the speed and heading of the vessel) and if the distance between these two positions is above a predefined threshold, an anomaly is raised. Other notable prototypes that currently exist are SECMAR [31], FastC^{2}AP [32] and MALEF [33].

3.1.3 BigDataOcean Approach

As the anomaly detection depends on the vessels' characteristics monitored, in BigDataOcean we follow a similar methodology having both static and dynamic anomaly detection capabilities. For static anomaly detection, we examine the aforementioned static characteristics of a vessel. These are considered static in the sense of slowly (or even never) changing and we use an inference engine that determines possible anomalies. This engine relies on the concept of "rules", similar to the ones used in the SeeCoast system and the SCANMARIS project described afore, but as the boundaries of the inputs are vague, using Boolean logic in the ruleset has been avoided. Fuzzy logic and linguistic variables have been used instead, to define the rules and express the notion of "degree of membership". Dynamic anomalies are linked to the vessels' voyages. In order to identify such anomalies, we use the AIS location messages to semantically enrich simple location signals and transform them into port-to-port trajectories. Then these are used to extract knowledge using clustering algorithms that will identify common routes that vessels follow when traveling on the same itinerary. Figure 2 below illustrates this methodology both for static and dynamic anomaly detection, including all the steps necessary to transform raw data received from various data sources into actionable intelligence through a series of algorithms and tools used for the considered pilot. In the following subsections, all these steps illustrated in Figure 3-2 are further analysed.

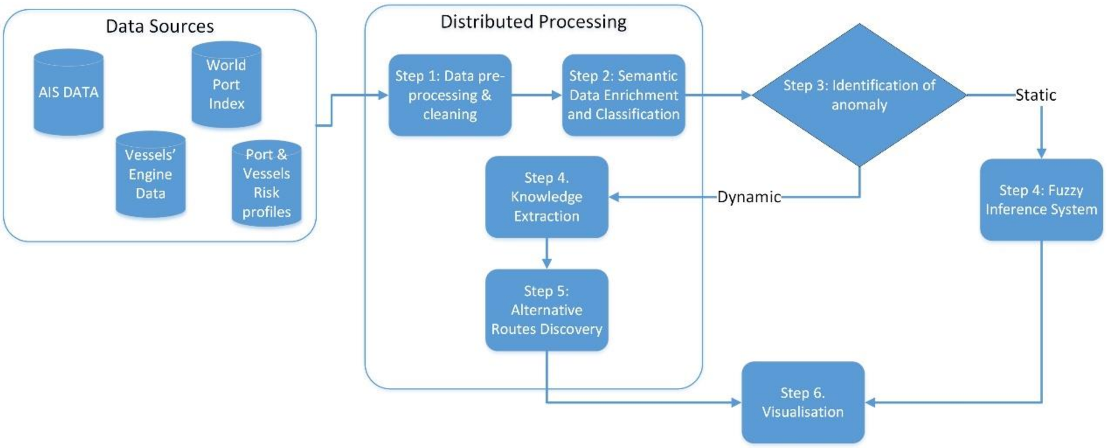

Data Sources
AIS DATA
World Port Index
Vessels' Engine Data
Port & Vessels Risk profiles
Distributed Processing
Step 1: Data pre-processing & cleaning
Step 2: Semantic Data Enrichment and Classification
Step 3: Identification of anomaly
Static
Dynamic
Step 4. Knowledge Extraction
Step 4: Fuzzy Inference System
Step 5: Alternative Routes Discovery
Step 6. Visualisation

Figure 3-2: Maritime Security and Anomaly Detection data processing methodology

3.1.3.1 Data sources and data acquisition process

The key data that will be used for the maritime security and anomaly detection pilot of BigDataOcean are data transceived (i.e., transmitted and received) through the AIS. The existence of centralised AIS monitoring systems has disruptively changed the way the maritime industry operates. However, these systems also exposed the limitation of the original AIS design. Limited range of AIS transceivers and the transmission frequency has made it difficult to have uniform coverage of the globe. Even after the adoption of satellite-AIS, there are still large areas of the sea that have limited, or none, coverage. Moreover, the collection of a global dataset is challenging itself due to several reasons. Network delays, different time-zones and the volume of data that must be synchronised, stored and processed in near real time are only some of them. In addition to technical limitations, human factor has significantly impacted data consistency, due to lack of a global regulation that covers maritime industry in total (i.e. not common policy about MMSI changes, IMO changes), due to illegal actions or due to human errors. All these factors are taken into account during the AIS data analysis as they may affect its outcome.

To limit (and if possible completely avoid) the drawbacks of the AIS system, data available from other data sources are needed, either to complement or to validate the accuracy of the AIS data. A global set of ports that includes a reference location is required for further processing in conjunction with AIS positional data. In addition, vessels operational data, weather data, oceans currents data, if available, can be helpful for the maritime security and anomaly detection pilot. AIS dataset will be collected from EXMILE through its platform (i.e., MarineTraffic). For the rest of the data sources, the original datasets will be collected and stored in the BigDataOcean platform infrastructure in the platform's default text file format (e.g., txt or csv). However, data transceived through the AIS, as well as data acquired from other data sources, include various information that are not relevant to the maritime security and anomaly detection application scenario. Thus, besides data cleaning which is a step that will remove invalid/inaccurate entries from the dataset, a pre-processing step is also needed during which the attributes needed for the considered application are filtered from the original dataset. In the following subsection, the schema definition of the raw data and the future selection are provided.

3.1.3.2 Raw Data pre-processing / cleaning

Schema definition of raw data

Detecting anomalies in vessels' operations requires accurate modelling of the "normal" behaviour of the vessels. To perform such modelling, features available from the data sources that are essential for further processing need to be identified. From the AIS system all the data that are relevant to vessel's position are of interest for this scenario. These include the following features:

- *ship_id*: long (nullable = false): Vessel's unique identifier;
- *status:* integer (nullable = true): Identifier of the vessel's status. Its value is mapped to a status message that includes a generic description of the vessel's status (e.g., "Underway using Engine", etc.);
- *speed:* integer (nullable = true): Vessel's speed over ground measured in knots (i.e. nautical miles/hour);
- longitude: double (nullable = false)
- latitude: double (nullable = false)
- course: integer (nullable = true): Vessel's course over ground measured in degrees (accepted values 0-360)
- heading: integer (nullable = true): Vessel's true heading measured in degrees (accepted values 0-360)
- time_stamp: timestamp (nullable = false)
- *flag_code*: integer (nullable = true): Vessel's flag code
- *ship_type*: integer (nullable = true): type of vessel

Besides vessels' AIS data collected, port data are also critical for the maritime security and anomaly detection scenario. The following port data are needed for this scenario:

- *port_id*: long (nullable = false)
- *port_name*: string (nullable = true)
- *port_country*: string (nullable = true)
- *longitude*: double (nullable = false)
- *latitude*: double (nullable = false)
- *Geometry:* (nullable = false): bounding box based on the latitude/longitude values, or radius defining a circle with center the latitude/longitude coordinates.

Similarly, weather and sea data could be useful for analysing shipping patterns. Such data may include the following attributes:

- *longitude*: double (nullable = true)
- *latitude*: double (nullable = true)
- *time_stamp*: timestamp (nullable = false)
- *weather/sea attributes*
    - *air temperature*
    - *visibility*
    - *wind direction*
    - *wind power*
    - *wave height*
    - *wave direction*
    - *water temperature*
    - *atmospheric pressure*

Vessels' operational data could be used to validate the accuracy of the information extracted from the aforementioned data sets. Such data should include the following:

- *ship_id: long (nullable = false)*

- *status: integer (nullable = true)*
- *time_stamp: timestamp (nullable = false)*
- *Datasets from vessel fault prediction and recommendations (see 3.4)*

Port inspection reports is another source of data that can be used to identify high risk vessels and in conjunction with other static information detect effectively static anomalies. Such data in accordance with directives on port state control [34] that design vessels' risk profiles, include the following:

- *ship_type*: string (nullable=true): Ship type identification (i.e., passenger ship, oil tanker ship, bulk carrier, gas carrier, etc.
- *flag_country*: string (nullable=false): Vessel's flag
- *IMO*: integer (nullable = false): ship's identification number
- *Deficiency reasons*: string (nullable = false): list of vessel deficiencies that led to detention
- *Past_detentions*: string (nullable: false): Number of detentions over the past 36 months
- *Port of detention*: string (nullable: false)
- *Duration of detention*: integer (nullable:): Days of detention
- *Date of release*: Date (nullable: false)
- *Gross tonnage*: integer (nullable: false)
- *Keeldate*: integer (nullable: false)

Finally, it should be noted that other data may also be identified during the project's lifetime and used for the Maritime Security and Anomaly Detection pilot.

Handling inconsistent entries

Data collected from these data sources may include empty or incomplete entries. Thus, it is mandatory to define a strategy for handling inconsistent records. Initially, all rows that are empty or do not match the schema defined above are excluded from further processing. Then, attribute values are nullified in cases of entries with missing values or values out of accepted range. Then, for the remainder of the dataset a set of reasoning processes is applied to mark and exclude spoofing events. Spoofing events are entries that correspond to impossible vessels' movements. For instance,

- simultaneous appearance of the vessel in more than one locations,
- acceleration that goes beyond the capabilities of the vessel, or
- speed reported that does not match the distance travelled between the timestamps of two consecutive messages.

Finally, the dataset is sorted based on timestamp and entries that break the monotonality of the dataset are excluded. At this point the data have been collected and cleaned and are ready for further processing to transform them to meaningful information. In the next subsection, we describe the process of enriching the value of data through spatio-temporal correlation.

3.1.3.3 Data enrichment

AIS messages do not provide trustworthy voyage information with respect to departure and destination ports. In fact, the only relevant data collected from AIS are type 5 messages which include the destination port information. However, this is manually entered by the ship's crew and prone to errors, inconsistencies, etc. Thus, it is fundamental to discover such knowledge (i.e., departure and destination ports) from the AIS positional data to perform accurate route analytics. Port destination and departure for each AIS message is calculated based on the distance between vessel positions and port's reference coordinates as used from the World Port Index data source. Due to the volume of the AIS data this assignment has to be performed in a distributed processing system (i.e. Apache SPARK). The purpose of such correlation is to assign accurately the departure port, the destination port based on the timestamp of each AIS message and the departure time and arrival time respectively.

In addition to route assignment, for each message the time elapsed (measured in minutes) since the vessel's departure based on the reported timestamps (i.e. from AIS messages) is computed. The time elapsed field adds the time dimension in the analysis of AIS messages and groups them into time-aligned routes with the same <departure, destination> port and vessel type.

3.1.3.4 BigDataOcean Anomaly Detection approach

Fuzzy Logic reasoning for Static Anomaly Detection

Fuzzy logic relies on the theory of fuzzy sets, first introduced by Lotfi Zadeh in [35], which define sets that contain element with degree of membership. This approach exploits the notion of degree in the verification of a condition, enabling conditions to be in intermediate states between the states of conventional evaluations, thus allowing variables to be "partially" true, or "not definitely yes" etc. Such notions can be formulated mathematically and processed by machines, giving thus a more human-like interaction between the programmer and the computers [36]. In the context of BigDataOcean fuzzy logic has been selected for static anomaly detection as it is considered to be an ideal tool when dealing with imprecise or contradictive data, which can be modelled adequately with fuzzy sets, and combined with human logic [37].

Fuzzy Inference System (FIS) is the fundamental implementation of fuzzy logic schemes comprising three key elements, namely the fuzzifier, the inference engine and the defuzzifier. The first element is responsible to transform crisp values (e.g. real, integer, natural number, etc.) to fuzzy degrees of membership to states (i.e., values between the [0,1] interval). Then, the inference engine collects all the outputs of every rule into one fuzzy set. Several aggregation schemes have been proposed and applied, namely the maximum, the probabilistic or, the sum, etc. with the latter being the most common one. Finally, the defuzzifier undertakes the defuzzification process, which is the aggregation of the outcomes of all the rules and the production of a single number; the single number is a crisp value.

Thus, in order to define the FIS the set of inputs and the output of the rule set should be defined. In the context of static anomaly detection, the inputs are the vessels' static characteristics and the output is the fuzzy anomaly detection indicator. Table 3-1 below sums up the rule set that drives the Fuzzy Inference System. Each rule is a union of conditions that when met the corresponding output is triggered (based also on the fuzzy degree). Thus, each set of input values may match to multiple rules with a certain degree. The defuzzifier will take this fact into account when transforming the fuzzy values into a crisp output.

The output of the Fuzzy Inference engine is an indicator for anomaly. In order to empower the validity of the result, vessels with high anomaly detection indicator values are further investigated by querying the "high risk profile vessels and port" dataset to check whether these ships meet international safety, security and environmental standards and that crew members have adequate living and working conditions.

Inputs				Output
Flag change frequency	Destination change frequency[9]	MMSI/IMO inconsistencies	Destination port/ship type inconsistencies	Anomaly Detection indicator
Low	Low	Low	Low	Low
Low	Low	Low	High	Medium
Low	Low	High	Low	Low
Low	Low	High	High	Medium
Low	High	Low	Low	Low
Low	High	Low	High	Medium
Low	High	High	Low	Low
Low	High	High	High	Medium
High	Low	Low	Low	Medium
High	Low	Low	High	Medium
High	Low	High	Low	Medium
High	Low	High	High	High
High	High	Low	Low	Medium
High	High	Low	High	High
High	High	High	Low	High
High	High	High	High	High

Table 3-1: Fuzzy logic rules

BigDataOcean approach for Dynamic Anomaly Detection

As previously described, dynamic anomaly can be considered as a "strange" deviation in what could be considered as a vessels' normal behaviour, meaning that it is inconsistent with, or straying from what is usual, normal or expected for the specific vessel or vessels of its type. The challenge in this approach is defining the baseline behaviour accurately so as to detect potential outliers from this. In the following sections, we describe the steps we initially take in the BDO approach to define the "normal" behaviour.

[9] The boundaries of this input will also depend on the ship type and the route/geographic region that the vessel is travelling. For instance, a passenger ship travelling in Aegean is expected to have frequent destination change (i.e. once every few hours, or once per day), but this is normal, as the distance travelled between the vessel's itineraries is short. On the other hand, a tanker vessel that travels in the Mediterranean sea is not expected to change its destination every few hours.

This can be understood as identifying the common routes vessels between ports of specific categories/types follow under most conditions.

Processing / Knowledge extraction

In order to take advantage of the parallelisation ability of distributed processing (i.e., SPARK) each enriched message is mapped to a key-value pair. The key uniquely identifies the route per vessel type and it is generated as the concatenated unique identifier of departure – destination pair and the vessel type, while the value is the enriched message itself.

Up to this point each record of the dataset has been enriched with additional voyage related information. To prepare data for further processing, all records are organised on lists based on the key defined in the map phase. This is performed by a reduce-by-key procedure and produces a set of key-valued pairs organised in a set of rows equal to the distinct number of keys (i.e. Unique routes per ship type).

The result of this process is stored and is accessible for additional processing or enrichment with extra features when needed.

Simplify/Interpolate dataset

Depending on the clustering algorithm selected in the next step, a second pre-processing step may be needed. Some algorithms require uniform distribution of data where some interpolation algorithm should be applied before. Others are computationally demanding and under certain circumstances might be useful to subsample or simplify the original dataset.

Feature Selection / Clustering

In order to capture a unique route for all the points included in each set we apply clustering techniques (i.e. k-means, DBSCAN, EM etc.). The required features selected for the clustering are the following:

- latitude,
- longitude,
- relative timestamp.

This enables clusters of vessels' positions to be detected based on location and time with respect to the departure timestamp, enhancing route perception with average elapsed time (or within a time range) from departure for vessels of the same type with the same route that sail in close by trajectories.

It is assumed that same vessel types sailing on the same route will follow similar trajectories. However, this is not always accurate, as various factors such as weather conditions, draught, etc. may vastly differentiate the vessel's course. In most cases the actual data indicated the existence of (at least) two different courses for the same route. The clustering result may require additional processing to deliver maritime meaningful trajectories.

3.2 Processing patterns for Mare Protection

The current section provides in detail an analysis of the methodology that is being followed in order to perform oil spill dispersion simulations for marine protection, contingency planning and prevention. The following schema is representing the basic steps for oil spill dispersion simulations.

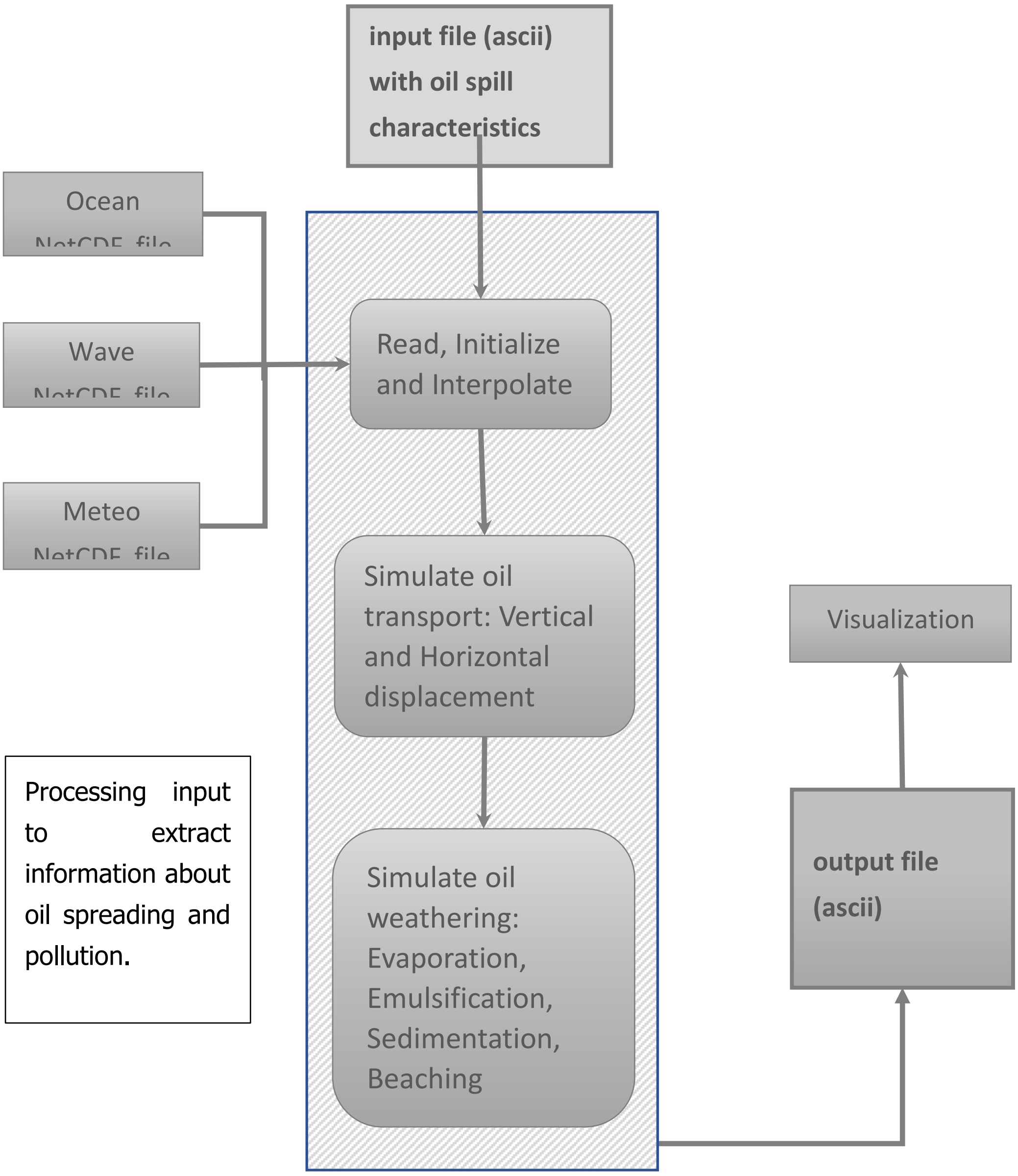

input file (ascii) with oil spill characteristics
Ocean NetCDF file
Wave NetCDF file
Meteo NetCDF file
Read, Initialize and Interpolate
Simulate oil transport: Vertical and Horizontal displacement
Simulate oil weathering: Evaporation, Emulsification, Sedimentation, Beaching
Processing input to extract information about oil spreading and pollution.
Visualization
output file (ascii)

Figure 3-2: The POSEIDON Oil Spill Model processing chain

3.2.1 The POSEIDON OSM processing chain

Oil spill forecasting models constitute a fundamental means in marine safety. This valuable tool has a key role in contingency planning and response strategies in case of marine pollution alerts due to oil spill accidents. Meteorological and oceanographic forecasts as well as the modelling of oil weathering processes are used in order to predict the fate and track the spill in a crucial period of time. Dedicated numerical models are employed to predict the direction and route of the spill, which resources are threatened, and the expected state of the oil after the first critical hours.

POSEIDON OSM is the oil spill model operationally used by the Hellenic Centre for Marine Research (HCMR) in the Aegean and Ionian Seas to provide oil spill dispersion and fate simulations using the atmospheric, oceanographic and sea state forecasting results that are produced during the daily operation of POSEIDON System. It is a 3-D Langragian, numerical model that simulates the pollutant transport (physical movement of the oil in the marine environment) and weathering (transformation of the oil due to interaction with the sea and atmosphere: evaporation, emulsification, sedimentation, beaching), while the oil slick is represented as "parcels" with time dependent chemical and physical characteristics.

3.2.1.1 Input and output files' information

The required input information consists of data that specify the event and the oil spill:

- location of the event (Lat/Lon),
- date and time of the event,
- total volume of the oil released into the sea,
- number of particles describing the volume,
- critical density for evaporation and emulsification,
- evacuation time (instant disposal in the sea or not),
- total time of model integration.

Bathymetric data are also introduced into the model, while the required meteorological and oceanographic data include:

- wind speed/direction and air temperature from the atmospheric forcing,
- significant wave height/ direction and wave period from the wave data,
- temperature, salinity and flow field for the whole water column together with the advection terms and the vertical mixing coefficients from the hydrodynamic data.

The output file describes the status of the oil quantity spilled at sea, in every time step through the simulation period requested by the user. Each time step describes:

- position of the spill
- evaporated oil percentage
- percentage of oil at surface
- emulsified oil percentage
- percentage of oil on coast
- percentage of oil at the bottom of the sea
- updated density of oil

The input files' information that are imported to the POSEIDON OSM and simulated in order to produce the forecast, along with the model output, are fully described in the appendix at the end of this paragraph.

3.2.1.2 Description of the oil spill model algorithms

The OSM model is developed in Fortran programming language and is able to handle NetCDF environmental input files that follow certain specifications (described in detail in the appendix), along with an ASCII file with the necessary oil spill information.

The main program of the POSEIDON oil spill model reads the required information in order the model's main routine to allocate the arrays needed to facilitate the specific simulation scenario and handle the input data. It handles the routines that read information for the model's initialisation (event's date/time, number of spills, total number of particles), the dimension indices of the 3D circulation data (im, jm, km), the 2D wave (imw, jmw) and meteorological data (imm, jmm). Reading the indices, the number of spills and the total number of particles for each spill, array memory allocation is feasible. The program is called and the above information is passed parametrically.

The program also calls a main routine, which performs all main calculations and handles the subroutines operations. Several subroutines are called and executed (31 in total) in order to perform several actions and simulate all the physical and chemical processes that will represent the dispersion and fate of the oil spill in the sea.

Basic assumptions and initial spreading

As described by Johansen (1985) and Elliott (1986) [38-39], the oil might be represented by a large number of material particles or parcels, each of which represents in turn a group of oil droplets of like size and composition. For modelling purposes, the whole mass of oil slick is simulated with `parcels' characterised by evolving physicochemical properties and may represent, in reality, many m^3 of oil, occupying a considerable area on the sea surface. The parameters used to attribute the physicochemical properties for each parcel and to initialise the OSM are the *x, y, z* coordinates, the initial volume, the density, and the droplet diameter.

The process of initial spreading, dominated by the inertial gravity, viscosity and surface tension forces, has been described satisfactory by Fay's law (Rasmussen, 1985) [40].

Other physical processes are described and simulated using several Fortran subroutines:

Oil transport

The hydrodynamic processes are described by two modules, the circulation module and the wind generated waves module. The circulation module used is the Princeton Ocean Model (POM) (Mellor, 1991) [41]. The wave field is used for Stokes drift computation.

Horizontal and vertical diffusion

Two very important processes, resulting from the turbulent nature of the current fields, are the horizontal and vertical diffusion (Mellor and Yamada, 1982) [42].

Evaporation

The evaporation is a process in influencing mainly the lighter fractions of mixture of hydrocarbons and can result in the transfer of 20-40 % of spilled oil from the sea surface to the atmosphere, depending on the type of the oil. The most volatile (low carbon number) hydrocarbons evaporate most rapidly, typically in less than a day and sometimes in under an hour. The method used to characterise evaporation of oil has suggested by Stiver et al. (1989) [43].

Emulsification

Emulsification is a process, during which the water is mixing with the oil up to concentrations of 80% water and 20% oil. Emulsification effects to the mixing of water in the heavier fractions of the hydrocarbons and to the formation of a 'chocolate mousse' with a tendency to remain on the surface as a thick slice. Factors that influence the emulsification process are the wind speed, the wave characteristics, the make-up of the oil, the degree of weathering of the oil, the environmental temperature, the local thickness, and the time (Rasmussen, 1985) [44].

Beaching and sedimentation

According to field observations, the oil has different behaviour when it contacts various forms of coastline. This can be described by a mechanical process, called 'beaching', that affects the spatial evolution of the oil slick in coastal waters by defining the trapping time that the oil quantity remains on a specific type of coastline. The OSM follows the approach of Gundlach (1987).

Other routines are also used in order realise necessary actions like reading input data in NetCDF format, interpolate in time and space, etc.

3.2.2 Big Data Ocean Approach

Through the BDO scenario of the Mare Protection Pilot, the end user will have access to combined information and products available through the platform.

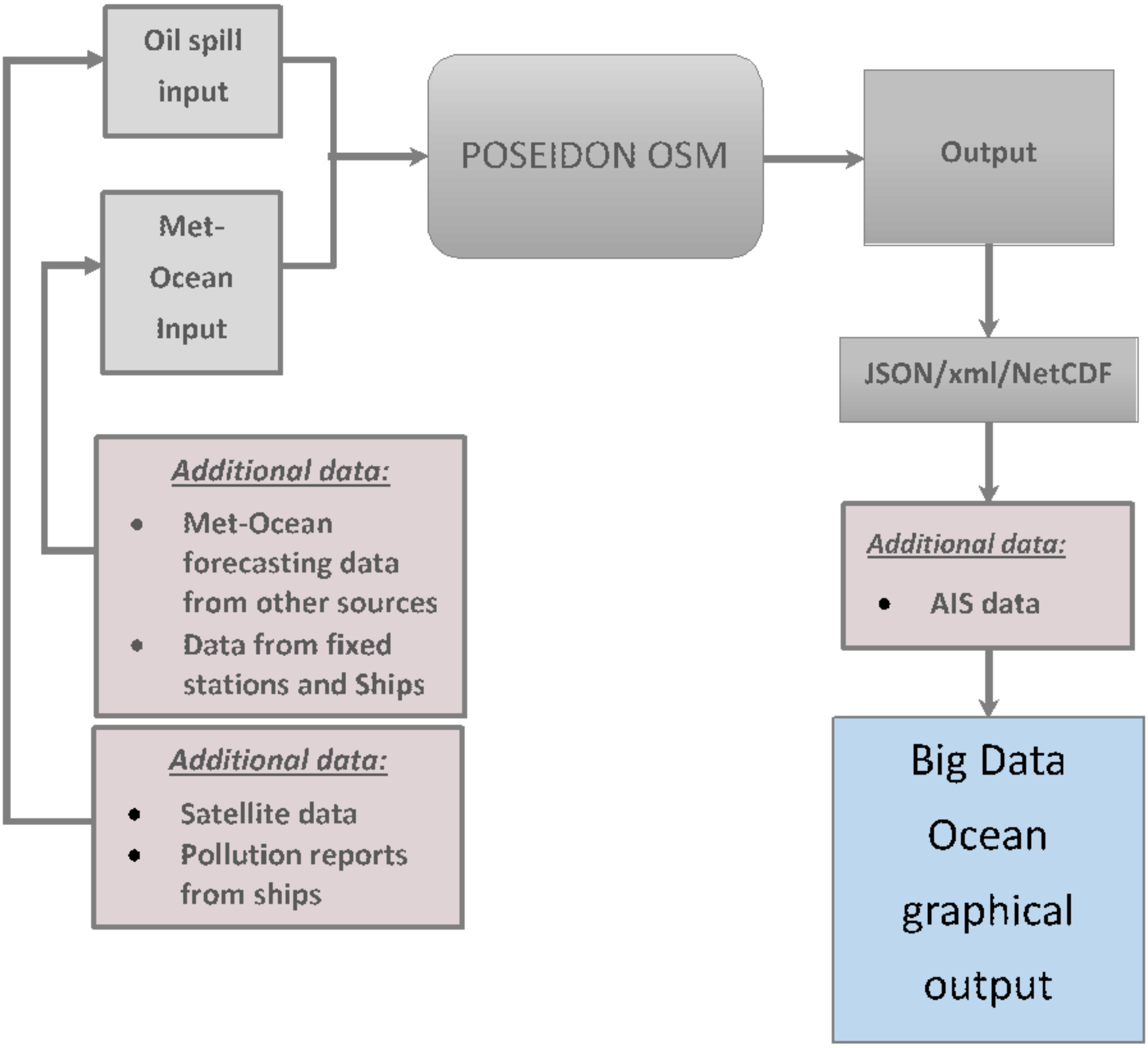

Oil spill input
Met-Ocean Input
POSEIDON OSM
Output
JSON/xml/NetCDF
Additional data:
Met-Ocean forecasting data from other sources
Data from fixed stations and Ships
Additional data:
Satellite data
Pollution reports from ships
Additional data:
AIS data
Big Data Ocean graphical output

Figure 3-3: Mare Protection Pilot through the BDO platform

The basic scenario of the Mare protection pilot will be: oil spill dispersion simulations can be triggered and performed through the BDO platform by POSEIDON OSM. Additional features will be available to the user through the BDO platform, if applicable:

1. Ability to use forecast data from other sources than HCMR products, available in the BDO platform.
2. In situ data from fixed stations and ships. If in-situ data are available to the area of the accident, in order to enrich the input and optimise simulation's precision. In-situ data must be available throughout the simulation period. Therefore, this ability will be offered upon availability and for historical runs only.
3. Satellite images of oil spills in the sea can be used to perform real time simulations that could support decisions and enhance effectiveness in response strategies. Depending on the time of occurrence, forecasting or historical runs could either be performed.
4. Pollution reports from ships can also be used as an input in order to trigger a simulation and predict oil fate and dispersion.
5. AIS data can be presented in parallel along with the OSM output in order to offer valuable indications for possible polluters.

In the BDO context the requests and results will be handled by the BDO platform and the model execution will be handled by HCMR. The user will make a request in the BDO platform providing as many information as possible for the oil spill event. The input data together with the forecasting data and all the other relevant data available in the BDO database will be pushed to POSEIDON OSM through an API call. The results (in JSON or NetCDF format) will be sent back to the BDO platform and will be available to the user in download and view mode.

3.3 Processing patterns for Wave Power exploitation

This section provides a detailed analysis of the methodology followed in the implementation of the wave power assessment pilot. Figure 3-4 depicted below illustrates this methodology, including all the steps necessary to transform raw data collected from multiple data sources into actionable intelligence through a series of algorithms and tools used for the considered scenario. In the following subsections, all steps illustrated in this figure are further described.

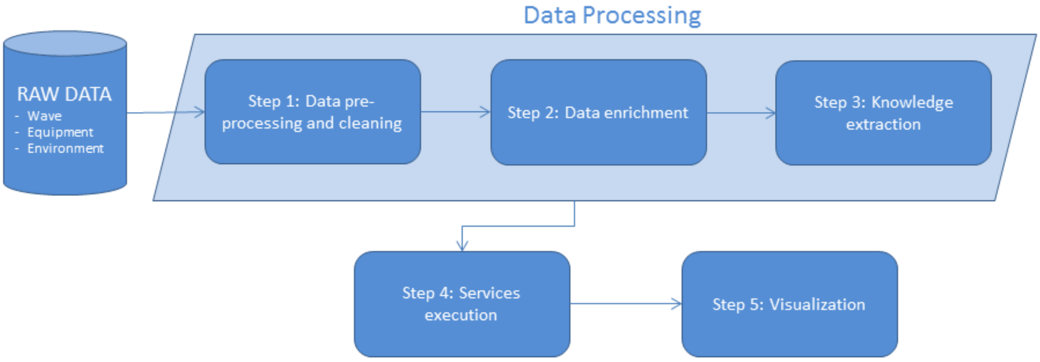


Figure 3-4: Wave power as the next clean energy source: data processing methodology

3.3.1 Pilot data services

Based upon the data collected from multiple sources, this pilot aims to deliver several services related to the assessment of wave energy in specific locations. Figure 3-5 depicts the detailed pilot methodology, including the data sources, data processing algorithms (from the functional point of view) and visualisation of provided services.

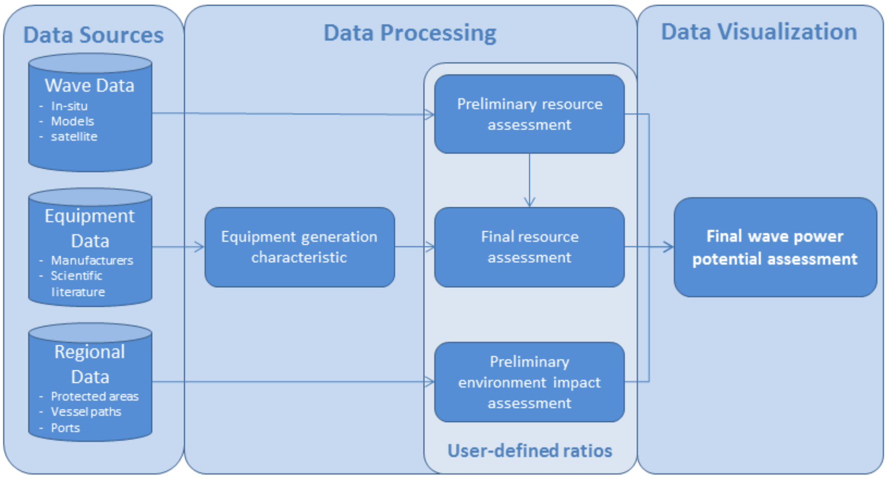


Figure 3-5: Wave power as the next clean energy source: functional methodology

The main goal of the pilot is to allow the user to perform a wave power potential assessment in a specific location. To materialise this, one fully completed example will show the wave potential at an offshore renewables pilot zone in the Atlantic Coast of Portugal considering at least two different technologies (for wave farm converters). Despite this location specific application example, the pilot will achieve higher flexibility by allowing the user to select other areas in the coast of Portugal (or in other locations around the World, as long as the data needed to perform such assessment is available at the BigDataOcean platform).

The assessment of wave power potential in a specific location is based upon three correlated components:

- **Preliminary resource assessment**: based only on the resource availability at that location;
- **Final resource assessment**: takes into account a specific technology for wave energy converters (defined by the user)
- **Preliminary environment impact assessment**: considers the effects that a wave farm would have in the selected location, by correlating the selected location with protected areas near it and vessels activities (e.g. fishing or vessels common routes)

These three pilot components will dictate the wave power conversion potential of the selected location while allowing the user to select different ratios that will result in different studies performed and different visualisation outputs. As an example, the user can select only the preliminary resource assessment without considering equipment and environment impact or can give the same importance to all three components. The three components use different datasets and have different algorithm needs. At this stage, the envisaged data/algorithm needs of each component can be explained as follows.

The preliminary resource assessment, illustrated in Figure 3-6, consists of the evaluation of the wave power potential taking into account only the wave conditions at that particular location. The main output of this study is then the available wave power for conversion. In addition to the different data processing algorithms shown in Figure 3-4, this assessment also needs to employ mathematical formulas in order to calculate the maximum available power. These will be coded using Python language and use as input the wave conditions datasets. Results will be visualised using 2D histograms, graphs and other user-friendly tools.

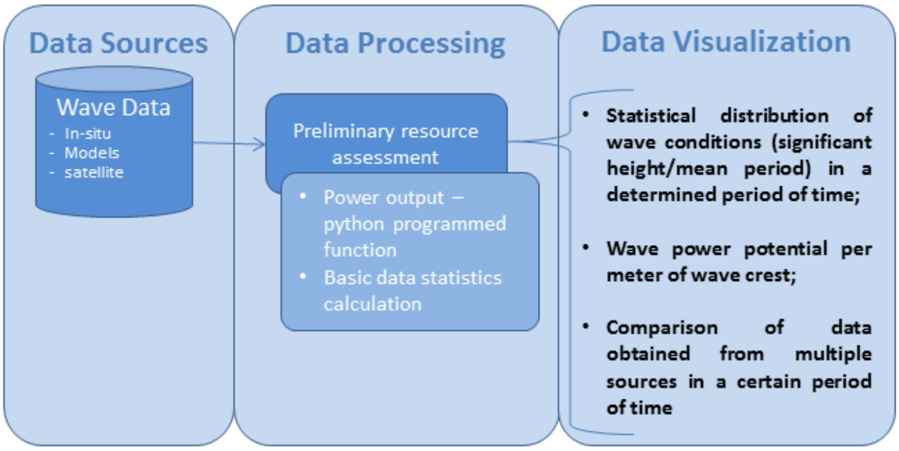

Data Sources
Wave Data
- In-situ
- Models
- satellite
Data Processing
Preliminary resource assessment
• Power output – python programmed function
• Basic data statistics calculation
Data Visualization
• Statistical distribution of wave conditions (significant height/mean period) in a determined period of time;
• Wave power potential per meter of wave crest;
• Comparison of data obtained from multiple sources in a certain period of time

Figure 3-6: Preliminary resource assessment methodology

The preliminary resource assessment briefly explained in the last paragraph allows having a first idea of the potential that one location has for wave power conversion. However, in order to have a more real impression of the possibility to install a wave farm in that location, it is necessary to take into consideration the technology (the wave power converters) that will can be installed there. Taking this into account, a more complete resource assessment study can also be performed in the pilot, considering the technology specificities. In the example that will be completely designed in the BigDataOcean platform, several technologies will be considered and the user will have the possibility to verify which technology is more suitable for a particular location. In addition to this, the user will also have the possibility to upload the characteristics of new technologies and verify their applicability to the area under study. Figure 3-7 shows the used methodology and main outputs of the complete resource assessment. Results obtained from the complete resource assessment study will be shown to the user in different formats, including visual tools (such as histograms and graphs) and text reports, depending on the output. As an example, the user should be able to perform a comparison of different technologies and the results should consider the most adequate tools to illustrate the differences.

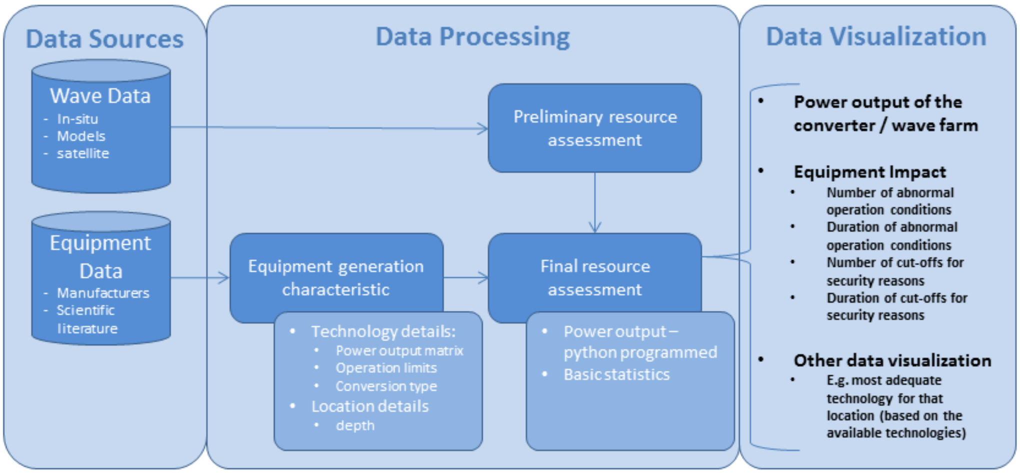

Data Sources
Wave Data
- In-situ
- Models
- satellite
Equipment Data
- Manufacturers
- Scientific literature
Data Processing
Preliminary resource assessment
Equipment generation characteristic
Final resource assessment
• Technology details:
• Power output matrix
• Operation limits
• Conversion type
• Location details
• depth
• Power output – python programmed
• Basic statistics
Data Visualization
• Power output of the converter / wave farm
• Equipment Impact
• Number of abnormal operation conditions
• Duration of abnormal operation conditions
• Number of cut-offs for security reasons
• Duration of cut-offs for security reasons
• Other data visualization
• E.g. most adequate technology for that location (based on the available technologies)

Figure 3-7: Complete resource assessment methodology

Complementing the two already illustrated components of the pilot, the user can also verify the adequacy of a location for deployment of wave farms, taking into account its impact in the region, considering both the environmental and the socioeconomic contexts. In order to verify this impact, the pilot will use additional data sources such as the location of nearby protected areas (if there are any) both offshore and inland and will also use data coming from other pilots in the BigDataOcean platform containing the vessels most common paths in the area, density of vessels seen in that area for a certain period of time and port locations, in order to access the possible impact on maritime activities such as fishing. This preliminary environmental impact assessment is depicted in Figure 3-8. As outputs of this component the user can mainly see a statistical analysis showing the possible impact that the deployment of a wave farm would have in that location considering the already mentioned contexts.

By considering the three pilot components described above, the user can reach a **final wave power potential assessment**. Using the results obtained from the three components, the potential will be evaluated and graded, according to some stablished criteria. As already mentioned, the user can change the grading ratios of each one of the three components, thus performing different studies. The different KPI's for each one of the three components will be fully clear to the user and whenever possible, the user can also change the value of these KPI's (e.g. consider less or more important the existence of a protected area nearby the location under study for the preliminary environment impact assessment).

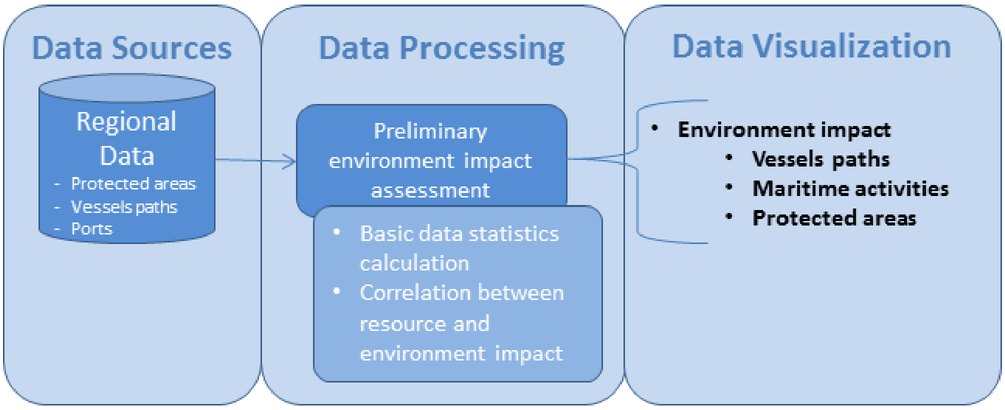


Figure 3-8: Preliminary environment impact assessment methodology

The pilot aims not only to allow the user to study one specific zone considering the three pilot components but also to perform **additional services**. Such services are built mainly based on using different visualisation of the pilot related data. These additional services will be designed during the pilot execution and some examples already considered include:

- Automatic verification of the best technology to apply to one specific location (based on the technologies whose characteristics are uploaded in the BDO platform);
- Automatic verification of the best location to deploy wave farms when considering larger regions and one specific technology (e.g. calculate the best location in the whole Portuguese Coast);
- Presentation of the best pairs of location / technology for larger regions (e.g. in the whole Portuguese Coast, indicate what technologies are more adequate for several locations).
- Quickly extrapolation to other geographic areas around the world (by just uploading wave characteristics in those locations using well defined criteria and file formats)

3.3.2 Pilot Methodology

3.3.2.1 Data sources and data acquisition process

In order to perform a correct assessment of wave power potential, considering multiple technologies and verifying the effects that these have in the environment, it is necessary to take into consideration multiple data sources, as seen in the last section. These data sources are essential for the pilot and a correct understanding of their composition is needed to correctly integrate them into the BigDataOcean platform.

The main component of the wave energy pilot is related to the assessment of the wave energetic potential at a specific location. Therefore, as expected, most of the needed data are related to ocean and wave conditions. These data can be mostly accessed through the Copernicus Marine platform[10] and are available in netCDF format[11] , which already contains some metadata and uses standards such

[10] http://marine.copernicus.eu/

[11] https://www.unidata.ucar.edu/software/netcdf/

as the CF-1.0 standard used for climate and forecast metadata[12]. Other data sources such as the output of numerical models for wave forecast are being considered for the pilot but these are also available using netCDF format and following the same standards as the data obtained from Copernicus.

In order to perform a preliminary study on the equipment operating conditions, which will dictate the impact on the equipment caused by the operation conditions, and on the environment, additional data sources need to be considered. These are mostly related to the equipment operating characteristics, which include operating parameters and power outputs, and to the environmental and socioeconomic context of the area under study. One goal of the pilot is to perform the study in a pilot area but easily extrapolate results for other geographies, which means that additional datasets particular to the areas under consideration are needed. Essentially, these include information about protected areas and sea vessel activities (such as common cargo vessel routes, fishing areas, etc.). Some of these necessary datasets are already available inside the consortium and others can be accessed from open source databases.

The identified data sources provide datasets using ftp services. This means that most datasets do not need to be duplicated in the BigDataOcean platform. However, the decision on duplicating data or using external datasets depends on the different processing needs and will be evaluated at a further stage of the pilot. After the data acquisition from the multiple identified sources, a process of data cleaning and pre-processing will allow improving the confidence in the used data and consequently on the pilot results. In the following sections, these different processes will be explained in detail.

3.3.2.2 Raw Data pre-processing / cleaning

Schema definition of raw data

The pilot is focused on three different correlated components (power potential assessment, environmental impact, equipment impact) that have different data needs. However, all data sources need to be correlated in order to perform a full scale evaluation of the wave energy potential of a certain location, considering different technologies and their impact in the environment. Bearing this in mind, the identified datasets are organised as follows.

- **Ocean and wave characteristics** datasets: contain information obtained either from in-situ measurements, numerical models for wave conditions forecast or satellite data. In addition, other meteorological characteristics such as wind conditions are also considered. The different variables foreseen to be needed for the pilot are:
    - *time*: date and time of the measurement or modelling process;
    - *latitude;*
    - *longitude;*
    - *depth:* depth at which the measurement is performed;
    - *sea_surface_wave_from_direction:* contains the mean/instantaneous direction from where sea waves are coming;
    - *sea_surface_wave_to_direction:* contains the mean/instantaneous direction of sea waves;
    - *sea_surface_wave_from_direction_at_variance_spectral_density_maximum:* direction of most energetic waves;

[12] http://cfconventions.org

- *sea_surface_wave_significant_height:* contains the significant wave height calculated for the time period under analysis (e.g. 1 hour);
- *sea_surface_wave_maximum_height:* the greatest trough to crest distance measured during the observation period;
- *sea_surface_wave_zero_upcrossing_period (or mean_wave_period):* defined as the average of the zero-crossing periods obtained from the frequency moments of the variance wave spectrum;
- *sea_surface_wave_period_at_variance_spectral_density_maximum:* period of the most energetic waves in the total wave spectrum at a specific location
- *sea_surface_height:* contains the sea surface height above mean sea level over a period of time;
- *eastward_sea_water_velocity;*
- *northward_sea_water_velocity;*
- *sea_water_speed;*
- *wind_from_direction;*
- *wind_speed;*

Besides the ocean and wave characteristics described above, in order to perform preliminary environment and equipment impacts assessment, different other data sources are identified as critical for the pilot success. To evaluate the impact on the environment, which is focused on the socioeconomic impact in the area considered for wave farm deployment, a density map of the vessels paths (based on historical data) in the selected area and the location of the nearest ports are needed. In addition, the location of the nearest protected areas is also needed, in order to define the viability of the specified location. These required datasets are available either inside the consortium or already identified in open-source platforms. Aiming at determining the possible impact on the equipment operation, different equipment characteristics are needed. Among others, power matrix output (maximum power output considering the wave characteristics) and maximum operating points (in terms of wave height and period) should be considered.

3.3.2.3 Handling inconsistent entries and data cleaning

Data collected from the different data sources can include inconsistent entries. Thus, it is necessary to implement a strategy for data cleaning. NetCDF files contain quality flags for all included variables which enable checking the data quality and decide upon which data entries should be excluded from further processing. In addition to this, some files contain additional variables which are not foreseen to be used in the scope of the pilot. The data cleaning process should also remove these variables in order to improve data processing operations.

As mentioned above, Data Cleaning includes mainly two aspects: removal of bad data entries and non-used variables. However, once the pilot considers data from multiple sources, it is also necessary to perform a validation of data entries in order to mitigate data inconsistencies. In order to do so, the different variables used in the pilot that are available at multiple sources need to be compared and their values should be coherent. As one example, it does not make sense that the wave direction is coming from land; this would mean that the variables *sea_surface_wave_from_direction* and *sea_surface_wave_to_direction* are swapped. This data validation process will enable a high confidence in the pilot results, from the user perspective.

Detecting Outliers from in-situ measurements

In order to perform a wave energy potential assessment, three different kinds of datasets will be used: in-situ measurements, satellite data and forecast models outputs. In order to have sound results, it is necessary to consider a strategy for outliers removal, which are particularly common in in-situ measurements. Thus, the datasets will be filtered using an algorithm for outlier removal based on data statistics.

3.3.2.4 Data enrichment

The different datasets used in the pilot are coming from multiple sources. Even if most datasets are in netCDF, the standard leaves space for different interpretations which means that different files might represent different realities when variables are analysed. Take as example the variable time. One netCDF file can consider time as days whilst other considers it as seconds. In addition to this, the reference time can be different which means that the times in the file will have different meanings. Thus, it is necessary to perform a homogenisation of these datasets. This process of data enrichment will allow having a single time reference which facilitates the processing needs of the pilot.

3.3.2.5 Processing / Knowledge extraction

The most important dataset, which is the core of the pilot analysis, is a composition of pairs of mean wave height and period. After the data enrichment process which allows a single time reference for all datasets, and outliers removal, these pairs of mean height/periods need to be processed using simple statistics in order to check which pairs are more common. This will allow having a statistical distribution of wave conditions for the different areas under study, which is the main processing result needed to implement the different services offered by the pilot.

3.3.2.6 Feature Selection / Clustering

Datasets obtained from in-situ measurements are related to the particular location where the measurement instrument is located but datasets obtained from numerical models usually cover a large area (e.g. the whole Portuguese Atlantic Coast). Depending on the application desired by the user (e.g. if the user wants to verify the wave energy potential at a determined location) it is important to apply clustering algorithms, based on the location/area considered. Thus, a clustering technique based on different latitude and longitude specific clusters datasets is required. Additionally, the verification of impacts on environment and equipment should also use this clustering based on location. The application of clustering algorithms will then decrease the computation power required for the pilot services.

3.4 Processing patterns for Vessel Fault Prediction

3.4.1 Introduction

This section analyses the processing patterns to be developed for the Vessel Fault prediction and Maintenance Recommendations pilot led by ANEK Lines and FOINIKAS. This pilot concerns the operational tools that are currently being used as part of the process that provides essential information for properly handling fault prediction and maintenance recommendations on various types of vessels. Through this pilot, data from a number of sensors that constantly collect operational and performance data on every critical aspect (e.g. engines' strain, fuel consumption) as well as inspection and defects reports will be used to formulate a knowledge base that each stakeholder exploits towards the effort of being proactive rather than reactive, towards controlling unpredicted damages and/or mechanical failures, in order to avoid unnecessary costs. Therefore, the overall objective of this pilot is the same for both ANEK Lines and FOINIKAS. Nevertheless, the two pilot partners will plan, execute and analyse the maintenance recommendations use case under different requirements and conditions, different existing infrastructure as well as different results usage. Therefore, a different data processing chain will be employed for each company. The main different maintenance conditions and requirements between these two companies include:

- ANEK Lines operates passenger vessels, and due to seasonality a yearly scheduled maintenance plan reduces unplanned repairs. Moreover, unplanned repairs during season (especially in summer) have a huge financial and reputational impact. On the other hand, the tankers operated by the FOINIKAS shipping company are travelling non-stop during all year and every immobilisation of the vessel has important financial impact.
- The current location of the tanker during an urgent maintenance request is a significant factor affecting the total cost and time-to-complete of an unpredicted damage. Tankers operate around the globe, so unplanned repairs are riskier as not all ports have the required expertise. Passenger vessels on the other hand are geographically restricted, but face also the same issue if operated within an area with few shipyards.
- Tankers (FOINIKAS) use different ships with different machinery and are exposed to different weather and sea conditions than the passenger vessels (ANEK LINES)
- FOINIKAS and ANEK LINES have different systems and utilise different workflows for data management.

This diversity between the two pilot partners will significantly contribute to better validate and enhance the BigDataOcean platform and Fault prediction and Maintenance recommendations use case, as well as expand the stakeholders of this use case. The following sections will present the processing chain of each company as well as the overall Big Data Ocean approach.

3.4.2 ANEK LINES scope, databases and raw input data

ANEK Lines is the largest passenger shipping company in Greece. It operates passenger ferries, mainly on Piraeus-Crete and Adriatic Sea lines. ANEK LINES services are offered to clients through a strategically dispersed network of offices located in most European countries with Greece's mainland and Islands providing a broad sales network for the domestic market.

ANEK Lines maintains a large number of IT systems and databases regarding the daily operations of the vessels operations. The data that will be exploited in the context of the vessel maintenance pilot are coming from the following systems:

EPOS (Operation System)

ANEK EPOS system includes data for ANEK's passenger vessels that include vessel data related to course as well as fuel and trip related data. More specifically it contains:

Vessel data

idShip	The identifier of the ship
Lat	The latitude of its position
Lon	The longitude of its position
Speed	The current speed
Direction	The direction the ship is facing
tGPS	The timestap

Vessel trip data

idTrip	The identifier of the trip
idShip	The identifier of the vessel
idTerminalStart	The starting terminal of the trip
idHistoryStart	An identifier for the start of the trip
idTerminalStop	The final terminal of the trip
idHistoryStop	An identifier for the end of the trip
Fuel	Current fuel levels
FuelCons	Fuel consumed during the trip
Diesel	Current diesel levels
DieselCons	Diesel consumed during the trip
Water	Current water level
WaterCons	Water consumed during the trip
Lubricant	Current lubricant levels
LubricantCons	Lubricants consumed during the trip
Passengers	The number of passengers during this trip
Cars	The number of cars during this trip
Trucks	The number of trucks during this trip

M/E maintenance schedule and Spare-parts Planned Maintenance System (PMS)

The Main Engine maintenance schedule and the Plan Maintenance System are being used by ANEK's marine engineers and operations managers to cost effectively manage their vessels maintenance tasks. PMS enables ANEK to plan, monitor, record and implement complex maintenance tasks, including compliance reporting and maintenance monitoring. Crew on board the vessel can use the system to record when tasks have been completed, attach comprehensive history, notes, request new parts, or maintain stock (if required).

The PMS database includes a large number of metadata related to the maintenance of the ship including the identifier of the maintenance task, the main components / equipment under review, the category of the action (inspection, repair, etc.), the current status of the maintenance of the task, the current status of the component / equipment, the responsible people and the dates of the check, inspection or repair.

3.4.3 FOINIKAS Shipping scope, databases and raw input data

FOINIKAS Shipping Company is an independent Greek tanker owner, active in the shipment of petroleum products. FOINIKAS has a large list of customers and performs various trading services around the world. The company responds to market demands for fixing on either a spot or period (time charter) or consecutives voyages. The company's objectives are to transport crude oil and oil products as well as other liquid cargoes in bulk safely, efficiently and in an environmental friendly way with respect to its employees and the community. FOINIKAS ensures the highest level of readiness of its vessel utilising a planned maintenance and survey system, which is closely monitored by the company's superintendents/class society and is strictly in accordance with International Legislation. This high level of readiness ensures high standard of service, high degree of utilisation, trading flexibility, overall safety and reliability.

The shipping industry business landscape has evolved significantly over the past decade due the global economy, technology advancements and regulatory pressures. Following this evolution FOINIKAS' tankers are equipped with state-of-the-art systems and databases. The data that will be exploited in the context of the vessel maintenance pilot are coming from the following systems:

Defects Reporting Database

The defects reporting database includes all communication and details about defects in ship components and equipment and their status. More specifically the metadata of this database include:

- Date Created
- Due date
- Vessel
- Description of the defect
- Category of the defect
- Priority
- The responsible who took action by
- Follow up by
- The list of actions taken for this defect
- General remarks about the defect

PMS (Planned Maintenance System) Database

The Planned Maintenance System allows FOINIKAS to carry out maintenance in intervals according to manufacturers and class requirements. The maintenance, primarily supervised by the on board personnel, is then credited towards inspections required by periodic surveys.

- *The code of the task*

- *The compoment under examination*
- *Description of the scheduled task*
- *The Job Code*
- *Category*
- *Status*
- *Due Date*
- *Start Date*
- *Completion Date*
- *Last Done*
- *Duration*
- *Monitor Type*
- *Cause of the defect*
- *Period*
- *Unit*
- *Counter*
- *The related Class requirement*
- *The related International Safety Management Code (ISM) requirement*
- *Defect*
- *Responsible*
- *Department*
- *Priority*
- *Created on*
- *Approved by*
- *Approved on*
- *Confirm Start*
- *Notes*
- *Reviewed by*
- *Reviewed on*

WEB SIGNALS Database

The Web Signals database contains a vast amount of information related to the tankers trip and stops, including engine speed, consumptions, distance travelled, loading performance, discharging performance, weather condition, lubricants. More specifically, a number of metadata included are:

- *Geographical coordinates, latitude and longitude of the position of the vessel.*
- *Condition of the sea and the swell*
- *Direction and force of the wind.*
- *Steaming time and distance run at slow speed*
- *Steaming time and distance run at full speed*
- *Time changes from setting the clock forward or backward with +(plus) for forward movements and – (minus) for backward movements*
- *Average engine load indicator*
- *Average speed for the distance covered from the previous noon or from fullaway*
- *Average number of revolutions per minute or indication of the propeller*
- *The remaining distance from the noon position to the destination port*

- *Estimated date and time of arrival at the destination port followed by the speed under which the estimation has been made*
- *Consumption of fuel and diesel oil exclusively for the operation of the main engine.*
- *Consumption of fuel and diesel oil in the electric generators*
- *Consumption of fuel and diesel oil in the boilers to provide steam for the heating of fuel*
- *Consumption of water in tons for the ordinary daily needs of the vessel excluding consumptions for tank washing, cargo heating*
- *Duration of tank washing operations with cold water in hours and minutes and consumption of fuel oil, diesel oil and water for the tank washing operations.*
- *Duration of tank washing operations with hot water in hours and minutes and consumption of fuel oil, diesel oil and water for the tank washing operations.*
- *Duration of inert gas system operation in hours and minutes. Consumptions in FUEL OIL DIESEL OIL and Water, only for Inert gas system operation.*
- *Duration of Tank ventilation system operation for GAS-FREE in hours and minutes, consumptions in FUEL OIL, DIESEL OIL and WATER, (if consumed), only for the operation of the tank ventilation system.*
- *Quantity of ballast handled (ballasting, deballasting and duration of ballast handling in hours, minutes and consumptions of bunkers and water only for ballast handling.*
- *Quantity of cargo transferred from one tank to another for various reasons such as blending of cargoes, cargo heating or other reasons*
- *Cargo heating duration, in hours and minutes. Consumptions of FUEL OIL, DIESEL OIL and WATER only for cargo heating reasons cargo temperature, and quantity of cargo heated.*
- *Remaining on board bunkers, Lubricants and Water. FUEL OIL, DIESEL OIL and WATER*
- *Water production of evaporators in tons*
- *Quantity of Lubricants in litres replenished in the stern tube tank for the last 24 hrs.*
- *Commencement and termination for stoppages or slow-down.*
- *In case of stoppage, the duration and consumptions of the operations specified in and not separately for the duration of normal steaming and for the duration of stoppage.*

3.4.4 Big Data Ocean approach

As indicated in the previous sections, this pilot involves the management of data of great diversity in terms of data acquisition and processing. The overall Big Data Ocean approach for the processing chain of vessel fault and maintenance recommendations data is presented in the following figure.

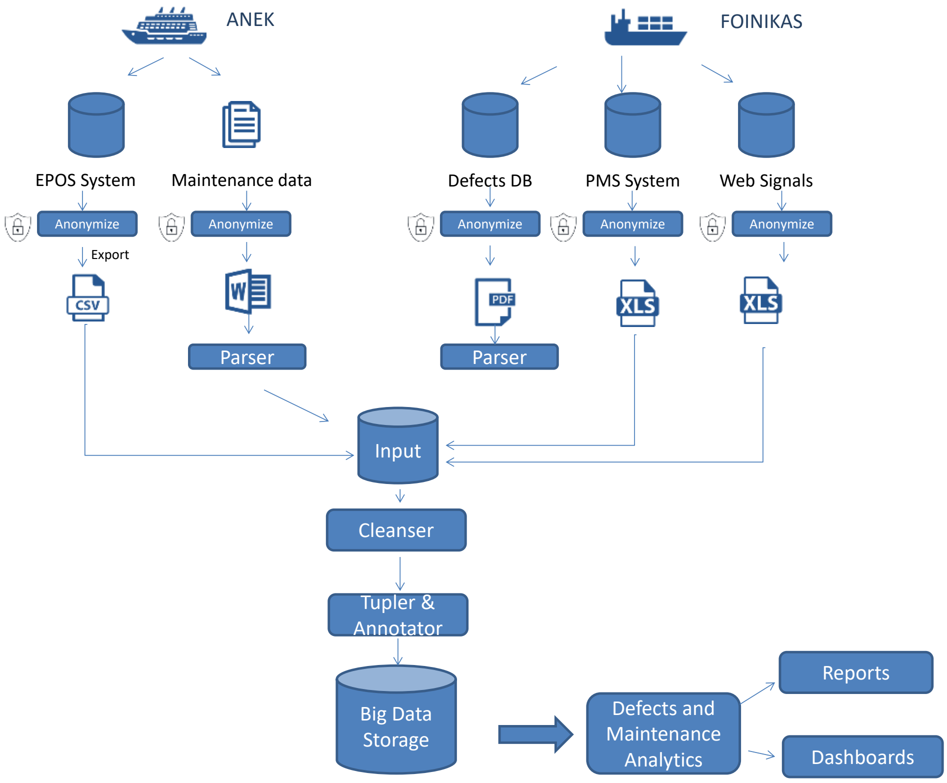


Figure 3-9: Vessel Fault Prediction

More specifically the Big Data Ocean approach for this pilot involves the following phases:

Data Acquisition

The data acquisition phase involves harvesting data regarding the vessels trip characteristics as well as maintenance schedule and status. Real-time information about the vessel and its trip is gathered by the ships sensors and on board-real time systems. As the internet connectivity is limited during a ship trip (especially on long trips around the globe), the data is being send at specific intervals or when arriving at a port in batches. The data is thereafter gathered centrally from the IT systems and databases of each company (EPOS, PMS, WEB SIGNALS). In the case of ANEK lines defects and maintenance information has to be harvested manually due to the lack of a fully electronic end-to-end workflow of defects and fault reporting

Pre-processing and anonymisation

Pre-processing of the data involves selecting and cleansing the data in order to avoid out-of-range values or invalid data, impossible data combinations and missing values. Analysing data that has not been carefully pre-processed can produce misleading results. Moreover, pre-processing involves the enhancement of the representation and quality of data. The companies (ANEK, FOINIKAS) are responsible for pre-processing the data and making the appropriate filtering and queries that will result to the datasets used as input to the BDO platform. At this phase irrelevant and redundant information present or noisy and unreliable data is omitted in order to improve the quality of the further analysis.

Moreover, at this phase both companies anonymise the datasets by stripping or obfuscating sensitive information, including ship names and personal information of the crew and staff, for privacy issues.

Curation

At this phase the raw, anonymised and cleansed input is harmonised in a common format through the tupler (tabular view). Non-machine processable documents (word and pdf – reports for the defects) are being processed with special text parsers that will convert the documents in to machine-processable and querable formats. Thereafter the raw input is being semantically enriched to be aligned with the common BDO vocabulary and schema to allow further processing capabilities by the consumption applications (queries, visualisation, and analytics).

Storage

At the final phase, semantically enriched datasets are stored within the main storage database of the Big Data Ocean Platform.

Usage

The enriched datasets stored within the BDO storage can thereafter be utilised by the query, visualisation and analytic services of the BDO platform. More specifically, users will be able to perform fault and maintenance analytics in order to either create reports that would be beneficial for the operational manager, engineers and other stakeholders as well as real-time dashboards with the ability to create email notifications. The analysis will significantly enhance the maintenance schedule planning and will feed a flexible, scientific model for maintenance & repair recommendations through BigDataOcean analytics. Through sharing and combining data, models and analyses through entire fleets (or operations, or geographical areas) interesting patterns are expected to come up, aiding further the maintenance planning and recommendation procedures. Proactiveness will reduce the resources needed for unplanned maintenance and the related machinery stock needed. The results of the analysis will pave the way to the commercialisation of enriched data and maintenance recommendations model in more ship types and other industries.

3.5 Other processing patterns for Maritime Applications

The current section serves mainly as a placeholder for the follow up version of the current deliverable. The current version includes the processing patterns for each of the four pilots in the project. Nevertheless, sooner than later the need may arise (stemming probably from other stakeholders) for additional processing patterns to be designed and documented based upon the identified needs. This section serves for the documentation of these patterns in the future revision of this deliverable.

4 Algorithms

4.1 Knowledge extraction and Business Intelligence

The current section defines the algorithms that will facilitate the proper execution flow of the four project pilots at first, and of future services as the project evolves, describing their purpose and referencing their software implementations, if available. The current section also leaves room for additional mathematical algorithms which may also comprise custom implementations, if such needs are identified during the project lifecycle. The documentation of the current section capitalises upon the maritime data stakeholders needs identified within the context of Task 2.2 and documented in deliverable D2.1, the user stories collected within the context of T4.2 and documented in deliverable D4.1, as well as upon the initial architectural decisions taken within the context of Task 4.4 and documented in deliverable D4.2.

Towards this end, given that the most prominent solution with regards to the big data processing framework to be adopted and deployed in the context of the project is Spark, the set of algorithms to be deployed and supported mainly revolve around Spark's machine learning (ML) library, namely MLib[13]. Thus, sub-chapters have been created according to the categorisation of the available algorithms based upon their functionality. At this point, three main categories have been identified:

1. **Basic Statistics**, which includes common statistics algorithms (min, max, mean, correlations, hypothesis testing etc.).
2. **Machine Learning**, which includes common machine learning algorithms associated with Classification (binary and multi-class) and Regression (including linear and non-linear models), Clustering (k-means, Gaussian mixtures etc.), Collaborative Filtering.
3. **Featurisation**, which includes mainly algorithms for Dimensionality reduction, Feature extraction, Transformation and Selection.

4.1.1 Basic Statistics

Basic statistics algorithms can be classified in five (5) main subcategories:

1. Summary statistics
2. Correlations
3. Stratified Sampling
4. Hypothesis Testing
5. Random Data Generation

4.1.1.1 Summary Statistics

Summary Statistics Description

In descriptive statistics, summary statistics are used to summarise a set of observations, in order to communicate the largest amount of information as simply as possible. Summary statistics summarise

[13] https://spark.apache.org/docs/latest/ml-guide.html

and provide information about the data set available and tells something about the values in it. This includes where the average lies and whether the data are skewed. Summary statistics mainly (yet not exclusively) include mean, median, mode, minimum value, maximum value, range, and standard deviation and fall into three main sub-categories:

1. Measures of location (also called central tendency).
2. Measures of spread.

Measures of Location

Measures of location give an insight on where the data are centred at, or where a trend lies. Measures of location include:

- **Mean** (also called the arithmetic mean or average). If you're trying to find the mean in statistics, what you are looking for most of the time is the average of a data set (the Arithmetic Mean). The arithmetic mean is the average of a set of data.
- **Geometric mean** (used for interest rates and other types of growth). The geometric mean is a type of average, usually used for growth rates, like population growth or interest rates. While the arithmetic mean adds items, the geometric mean multiplies items. Also, you can only get the geometric mean for positive numbers.
- **Trimmed Mean** (the mean with outliers excluded). A trimmed mean (sometimes called a truncated mean) is similar to a mean, but it trims any outliers. Outliers can affect the mean (especially if there are just one or two very large values), so a trimmed mean can often be a better fit for data sets with erratic high or low values or for extremely skewed distributions.
- **Median** (the middle of a data set). The median is a number in statistics that tells you where the middle of a data set is. It's used for many real-life situations, like Bankruptcy law, where you can only claim bankruptcy if you are below the median income in their state.

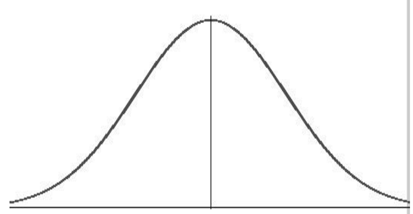
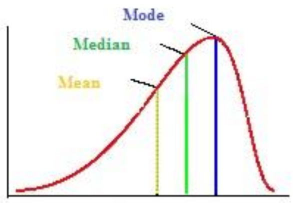
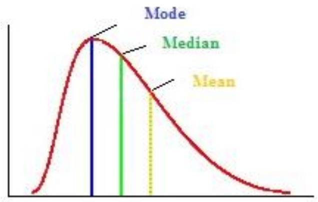
Measures of Spread

Measures of spread give an insight on how spread out or varied your data set is. For example, test scores that are in the 60-90 range might be expected while scores in the 20-70 range might indicate a problem. Range isn't the only measure of spread though. Measures of spread include:

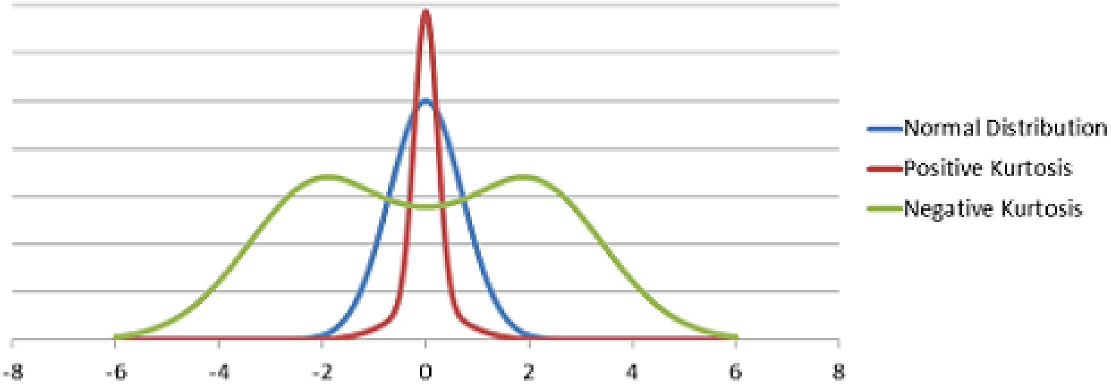
- **Range** (how spread out your data is). The range in statistics is a measure of spread: it's the difference between the highest value and the lowest value in a data set. The same two steps are used whether you are dealing with positive or negative numbers or time (i.e. seconds or minutes).
- **Interquartile range** (where the "middle fifty" percent of your data is). The interquartile range is a measure of where the "middle fifty" is in a data set. Where a range is a measure of where the beginning and end are in a set, an interquartile range is a measure of where the bulk of the values lie. That's why it's preferred over many other measures of spread (i.e. the average or median) when reporting things like school performance or SAT scores. The interquartile range formula is the first quartile subtracted from the third quartile: IQR = Q3 – Q1.
- **Quartiles** (boundaries for the lowest, middle and upper quarters of data. Quartiles in statistics are values that divide your data into quarters. However, quartiles aren't shaped like pizza slices; Instead they divide your data into four segments according to where the numbers fall on the number line. The four quarters that divide a data set into quartiles are: 1) The lowest 25% of

numbers. 2) The next lowest 25% of numbers (up to the median). 3) The second highest 25% of numbers (above the median). 4) The highest 25% of numbers.

- **Skewness** (does your data have mainly low, or mainly high values?). A distribution is skewed if one tail is longer than another. These distributions are sometimes called asymmetric or asymmetrical distributions as they don't show any kind of symmetry. Symmetry means that one half of the distribution is a mirror image of the other half. The normal distribution is a symmetric distribution with no skew. The tails are exactly the same. A left-skewed distribution has a long left tail. Left-skewed distributions are also called negatively-skewed distributions. That's because there is a long tail in the negative direction on the number line. The mean is also to the left of the peak. A right-skewed distribution has a long right tail. Right-skewed distributions are also called positive-skew distributions. That's because there is a long tail in the positive direction on the number line. The mean is also to the right of the peak.

		
Normal Curve	Left-Skewed Curve	Right-Skewed Curve

- **Kurtosis** (a measure of "peakedness"). Kurtosis tells you how "peaked" your graph is, or how high the graph is around the mean. It's also the fourth moment in statistics. A positive value means that you have too little data in your tails. A negative value means that you have too much data in your tail. This heaviness or lightness in the tails means that your data looks more peaked (or less peaked).



Summary Statistics Software Implementation

Software implementation of most of the aforementioned summary statistics (min, max, mean, count, variance, quartiles) is already available through the machine learning library of SPARK and is available here[14]. Implementation is available in Scala, Java and Python. Nevertheless, additional open source implementations in other programming languages are also available in other statistics libraries, including for example Java-ML, R and Weka. In addition, custom software libraries can also be developed in the case of specific requirements not met by the open source libraries available.

[14] https://spark.apache.org/docs/latest/mllib-statistics.html#summary-statistics

Summary Statistics Usability in BigDataOcean

Summary statistics are usually used to summarise a set of observations. The calculated values can be used to get a simple description of the data as quickly and simply as possible. Furthermore, the summary statistics are usually parts of more complex algorithms.

4.1.1.2 Correlations

Correlations Description[15]

Correlation is a statistical technique that can show whether and how strongly pairs of variables are related. For example, height and weight are related; taller people tend to be heavier than shorter people. The relationship isn't perfect. People of the same height vary in weight, and you can easily think of two people you know where the shorter one is heavier than the taller one. Although this correlation is fairly obvious your data may contain unsuspected correlations. You may also suspect there are correlations, but don't know which are the strongest. An intelligent correlation analysis can lead to a greater understanding of your data. Like all statistical techniques, correlation is only appropriate for certain kinds of data. Correlation works for quantifiable data in which numbers are meaningful, usually quantities of some sort. It cannot be used for purely categorical data, such as gender, brands purchased, or favourite colour.

Pearson r correlation

Pearson r correlation is the most widely used correlation statistic to measure the degree of the relationship between linearly related variables. For example, in the stock market, if we want to measure how two stocks are related to each other, Pearson r correlation is used to measure the degree of relationship between the two.

The main result of the correlation is called the correlation coefficient (or "r"). It ranges from -1.0 to +1.0. The closer r is to +1 or -1, the more closely the two variables are related. If r is close to 0, it means there is no relationship between the variables. If r is positive, it means that as one variable gets larger the other gets larger. If r is negative it means that as one gets larger, the other gets smaller (often called an "inverse" correlation).

While correlation coefficients are normally reported as r = (a value between -1 and +1), squaring them makes then easier to understand. The square of the coefficient (or r square) is equal to the percent of the variation in one variable that is related to the variation in the other. After squaring r, ignore the decimal point. An r of .5 means 25% of the variation is related (.5 squared =.25). An r value of .7 means 49% of the variance is related (.7 squared = .49).

The following formula is used to calculate the Pearson r correlation:

$$r = \frac{N \sum xy - \sum (x)(y)}{\sqrt{N \sum x^2 - \sum (x^2)][N \sum y^2 - \sum (y^2)]}}$$

For the Pearson r correlation, both variables should be normally distributed (normally distributed variables have a bell-shaped curve). Other assumptions include linearity and homoscedasticity.

[15] http://www.statisticssolutions.com/correlation-pearson-kendall-spearman/

Linearity assumes a straight-line relationship between each of the variables in the analysis and homoscedasticity assumes that data is normally distributed about the regression line.

Spearman rank correlation

Spearman rank correlation is a non-parametric test that is used to measure the degree of association between two variables. It was developed by Spearman, thus it is called the Spearman rank correlation. Spearman rank correlation test does not assume any assumptions about the distribution of the data and is the appropriate correlation analysis when the variables are measured on a scale that is at least ordinal. The following formula is used to calculate the Spearman rank correlation:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

Spearman rank correlation test does not make any assumptions about the distribution. The assumptions of Spearman rho correlation are that data must be at least ordinal and scores on one variable must be monotonically related to the other variable.

Kendall rank correlation

Kendall rank correlation is a non-parametric test that measures the strength of dependence between two variables. If we consider two samples, a and b, where each sample size is n, we know that the total number of pairings with a b is n(n-1)/2. The following formula is used to calculate the value of Kendall rank correlation:

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n(n-1)}$$

Correlations Software Implementation

Software implementation of Pearson and Spearman correlation algorithms is already available through the machine learning library of SPARK and is available here[16]. Implementation is available in Scala, Java and Python. Nevertheless, additional open source implementations in other programming languages are also available in other statistics libraries, including for example Java-ML, R and Weka. In addition, a custom software library e.g. for Kendall rank correlation can also be developed in the case of specific requirements not met by the open source libraries available.

Correlations Usability in BigDataOcean

Correlations can be used to determine if there is a relation between two variables and how strongly related these two variables are. This can lead to a greater understanding of the data and give useful insight about the data.

For example, dataset with locations of protected areas and vessels activities (e.g. fishing or vessels common routes) on the areas could provide useful insight if there are any correlation between the environmental impact and any of the vessels activities.

[16] https://spark.apache.org/docs/latest/mllib-statistics.html#correlations

4.1.1.3 Sampling

Sampling Description

Samples are parts of a population. For example, you might have a list of information on 100 people out of 10,000 people. You can use that list to make some assumptions about the entire population's behaviour. Unfortunately, it's not quite that simple. When you do stats, your sample size must be optimal — not too large or too small. Then once you've decided on a sample size you must use a sound technique for actually drawing the sample from the population. There are two main areas:

- Probability Sampling uses randomisation to select sample members. The probability of each member being chosen for the sample is known, although the odds do not have to be equal.
- Non-probability sampling uses non-random techniques (i.e. the judgment of the researcher). This is where you can't calculate the odds of any particular item, person or thing being included in your sample.

The most common techniques you'll likely encounter in statistics include taking a sample with and without replacement. Specific techniques include:

Bernoulli sampling

Bernoulli sampling is where independent Bernoulli trials on population elements determine whether the element becomes part of the sample. All population elements have an equal probability of being included in each selection of a single sample. The sample sizes in Bernoulli samples follow a binomial distribution. Poisson sampling is less common. Each population member being sampled is given an independent Bernoulli trial to determine if the element is included in the sample.

Cluster Sampling

Cluster sampling divides the population into groups (clusters). A random sample is then selected from the clusters. It's used when researchers don't know the individuals in a population but they do know which groups are in a population.

Systematic Sampling

In systematic sampling elements are selected for a sample from an ordered sampling frame. A sampling frame is just a list of participants that you want to get a sample from. One type of systematic sampling is the equal-probability method where an element is selected from a list and then every kth element is selected using the equation k = N\n where n is the sample size and N is the size of the population.

Simple Random Sample (SRS)

SRS is where a Simple Random Sample is chosen completely randomly so that each element has the same probability of being chosen as any other element and each subset of elements has the same probability of being chosen as any other subset of k elements.

Stratified Sampling

In stratified sampling, each subpopulation is sampled independently. The population is first divided into homogeneous subgroups before getting the sample. Each population member only belongs to one group. Simple random or systematic sampling is applied within each group to choose the sample. Stratified Randomisation is a sub-type of stratified sampling used in clinical trials. Patients are divided into strata and then randomised with permuted block randomisation.

Sampling Software Implementation

Software implementation of stratified sampling is already available through the machine learning library of SPARK and is available here[17]. Implementation is available in Scala, Java and Python. Nevertheless, additional open source implementations in other programming languages are also available in other statistics libraries, including for example Java-ML, R and Weka. In addition, custom software libraries e.g. for the other sampling techniques aforementioned can also be developed in the case of specific requirements not met by the open source libraries available.

Sampling Usability in BigDataOcean

Sampling is the process of selecting units from a population of interest so that by studying the sample we may fairly generalise our results back to the population from which they were chosen.

In the context of BigDataOcean for example, a dataset is available with vessels activities containing multiple features and one of them is the vessel type. We could extract a sample with a particular structure, for example our sample to contain 30% container ships, 30% bulkers and 40% fishing ships.

4.1.1.4 Hypothesis Testing

Hypothesis Testing Description

A hypothesis is an educated guess about something in the world around you. It should be testable, either by experiment or observation. If you are going to propose a hypothesis, it's customary to write a statement. Your statement will look like this: "If I (do this to an independent variable), then (this will happen to the dependent variable)." For example: If I (give patients counselling in addition to medication) then (their overall depression scale will decrease). A good hypothesis statement should: 1) Include an "if" and "then" statement (according to the University of California). 2) Include both the independent and dependent variables. 3) Be testable by experiment, survey or other scientifically sound technique. 4) Be based on information in prior research (either yours or someone else's). 5) Have design criteria (for engineering or programming projects).

Hypothesis testing in statistics is a way for you to test the results of a survey or experiment to see if you have meaningful results. You're basically testing whether your results are valid by figuring out the odds that your results have happened by chance. If your results may have happened by chance, the experiment won't be repeatable and so has little use.

Common test statistics include:

- One-sample tests
- Two-sample tests
- Paired tests
- Z-tests
- T-Tests
- Chi-squared tests
- F-tests

One-sample tests

One-sample tests are appropriate when a sample is being compared to the population from a hypothesis. The population characteristics are known from theory or are calculated from the population.

Two-sample tests

[17] https://spark.apache.org/docs/latest/mllib-statistics.html#stratified-sampling

Two-sample tests are appropriate for comparing two samples, typically experimental and control samples from a scientifically controlled experiment.

Paired tests

Paired tests are appropriate for comparing two samples where it is impossible to control important variables. Rather than comparing two sets, members are paired between samples so the difference between the members becomes the sample. Typically, the mean of the differences is then compared to zero. The common example scenario for when a paired difference test is appropriate is when a single set of test subjects has something applied to them and the test is intended to check for an effect.

Z-tests

Z-tests are appropriate for comparing means under stringent conditions regarding normality and a known standard deviation.

T-Tests

A t-test is appropriate for comparing means under relaxed conditions (less is assumed).

Chi-squared tests

Chi-squared tests use the same calculations and the same probability distribution for different applications:

- **Chi-squared tests for variance** are used to determine whether a normal population has a specified variance. The null hypothesis is that it does.
- **Chi-squared tests of independence** are used for deciding whether two variables are associated or are independent. The variables are categorical rather than numeric. It can be used to decide whether left-handedness is correlated with libertarian politics (or not). The null hypothesis is that the variables are independent. The numbers used in the calculation are the observed and expected frequencies of occurrence (from contingency tables).
- **Chi-squared goodness of fit tests** are used to determine the adequacy of curves fit to data. The null hypothesis is that the curve fit is adequate. It is common to determine curve shapes to minimise the mean square error, so it is appropriate that the goodness-of-fit calculation sums the squared errors.

F-tests

F-tests (analysis of variance, ANOVA) are commonly used when deciding whether groupings of data by category are meaningful. If the variance of test scores of the left-handed in a class is much smaller than the variance of the whole class, then it may be useful to study lefties as a group. The null hypothesis is that two variances are the same – so the proposed grouping is not meaningful.

<u>Hypothesis Testing Software Implementation</u>

Software implementation of Pearson's chi-squared tests for goodness of fit and independence is already available through the machine learning library of SPARK and is available here[18]. The input data types determine whether the goodness of fit or the independence test is conducted. Implementation is available in Scala, Java and Python. Additionally, spark.mllib provides a 1-sample, 2-sided implementation of the Kolmogorov-Smirnov (KS) test for equality of probability distributions. By

[18] https://spark.apache.org/docs/latest/mllib-statistics.html#hypothesis-testing

providing the name of a theoretical distribution (currently solely supported for the normal distribution) and its parameters, or a function to calculate the cumulative distribution according to a given theoretical distribution, the user can test the null hypothesis that their sample is drawn from that distribution. Nevertheless, additional open source implementations in other programming languages are also available in other statistics libraries, including for example Java-ML, R and Weka. In addition, custom software libraries e.g. for the other hypothesis testing techniques aforementioned can also be developed in the case of specific requirements not met by the open source libraries available.

Hypothesis Testing Usability in BigDataOcean

Hypothesis testing refers to statistical procedures used to accept or reject a statistical hypothesis. If the result is statistically significant then the statistical hypothesis can be accepted else it should be rejected.

In the context of BigDataOcean we would like to solve the problem of the number of vessels arriving on a specific port for each weekday. Based on our hypothesis on the number of vessels arriving on the port for each of the week days and the observations we have, running Chi-squared tests will give us insight if the hypothesis should be accepted or rejected.

4.1.1.5 Random Data Generation

Random Data Generation Description

When discussing single numbers, a random number is one that is drawn from a set of possible values, each of which is equally probable. In statistics, this is called a uniform distribution, because the distribution of probabilities for each number is uniform (i.e., the same) across the range of possible values. For example, a good (unloaded) dice has the probability 1/6 of rolling a one, 1/6 of rolling a two and so on. Hence, the probability of each of the six numbers coming up is exactly the same, so we say any roll of our dice has a uniform distribution. When discussing a sequence of random numbers, each number drawn must be statistically independent of the others. This means that drawing one value doesn't make that value less likely to occur again. Random data generation is useful for randomised algorithms, prototyping, and performance testing.

Random Data Generation Software Implementation

spark.mllib, the machine learning library of SPARK, supports generating random RDDs with i.i.d. values drawn from a given distribution: uniform, standard normal, or Poisson. Details about the specific software implementation can be found here[19].

4.1.2 Machine Learning

Machine Learning algorithms comprise the second major category of algorithms to be examined and included within the context of BigDataOcean. This category of algorithms includes common machine learning algorithms associated with Classification (binary and multi-class) and Regression (including linear and non-linear models), Clustering (k-means, Gaussian mixtures etc.), Collaborative Filtering.

[19] https://spark.apache.org/docs/latest/mllib-statistics.html#random-data-generation

4.1.2.1 Classification & Regression

Classification & Regression Description

In machine learning and statistics, **classification** is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. An example would be assigning a given email into "spam" or "non-spam" classes or assigning a diagnosis to a given patient as described by observed characteristics of the patient (gender, blood pressure, presence or absence of certain symptoms, etc.). Classification is an example of pattern recognition. In the terminology of machine learning, classification is considered an instance of supervised learning, i.e. learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering, and involves grouping data into categories based on some measure of inherent similarity or distance.

An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm that maps input data to a category.

Terminology across fields is quite varied. In statistics, where classification is often done with logistic regression or a similar procedure, the properties of observations are termed explanatory variables (or independent variables, regressors, etc.), and the categories to be predicted are known as outcomes, which are considered to be possible values of the dependent variable. In machine learning, the observations are often known as instances, the explanatory variables are termed features (grouped into a feature vector), and the possible categories to be predicted are classes. Other fields may use different terminology: e.g. in community ecology, the term "classification" normally refers to cluster analysis, i.e. a type of unsupervised learning, rather than the supervised learning described in this article.

In statistical modelling, **regression analysis** is a statistical process for estimating the relationships among variables. It includes many techniques for modelling and analysing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors'). More specifically, regression analysis helps one understand how the typical value of the dependent variable (or 'criterion variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed. Most commonly, regression analysis estimates the conditional expectation of the dependent variable given the independent variables – that is, the average value of the dependent variable when the independent variables are fixed. Less commonly, the focus is on a quantile, or other location parameter of the conditional distribution of the dependent variable given the independent variables. In all cases, the estimation target is a function of the independent variables called the regression function. In regression analysis, it is also of interest to characterise the variation of the dependent variable around the regression function which can be described by a probability distribution. A related but distinct approach is necessary condition analysis (NCA), which estimates the maximum (rather than average) value of the dependent variable for a given value of the independent variable (ceiling line rather than central line) in order to identify what value of the independent variable is necessary but not sufficient for a given value of the dependent variable.

Regression analysis is widely used for prediction and forecasting, where its use has substantial overlap with the field of machine learning. Regression analysis is also used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships. In restricted circumstances, regression analysis can be used to infer causal relationships between the independent and dependent variables. However, this can lead to illusions or false relationships, so caution is advisable; for example, correlation does not imply causation.

Many techniques for carrying out regression analysis have been developed. Familiar methods such as linear regression and ordinary least squares regression are parametric, in that the regression function is defined in terms of a finite number of unknown parameters that are estimated from the data. Nonparametric regression refers to techniques that allow the regression function to lie in a specified set of functions, which may be infinite-dimensional.

The most common algorithms supporting the resolution of problems associated with classification and regression include[20]:

Linear Regression

Linear Regression is one of the most widely known modelling technique. Linear regression is usually among the first few topics which people pick while learning predictive modelling. In this technique, the dependent variable is continuous, independent variable(s) can be continuous or discrete, and nature of regression line is linear. Linear Regression establishes a relationship between dependent variable (Y) and one or more independent variables (X) using a best fit straight line (also known as regression line). It is represented by an equation Y=a+b*X + e, where a is intercept, b is slope of the line and e is error term. This equation can be used to predict the value of target variable based on given predictor variable(s). The difference between simple linear regression and multiple linear regression is that, multiple linear regression has (>1) independent variables, whereas simple linear regression has only 1 independent variable.

Bayesian linear regression

In statistics, Bayesian linear regression is an approach to linear regression in which the statistical analysis is undertaken within the context of Bayesian inference. When the regression model has errors that have a normal distribution, and if a particular form of prior distribution is assumed, explicit results are available for the posterior probability distributions of the model's parameters.

Ordinary Least Squares

In statistics, ordinary least squares (OLS) or linear least squares is a method for estimating the unknown parameters in a linear regression model, with the goal of minimising the sum of the squares of the differences between the observed responses (values of the variable being predicted) in the given dataset and those predicted by a linear function of a set of explanatory variables. Visually this is seen as the sum of the squared vertical distances between each data point in the set and the corresponding point on the regression line – the smaller the differences, the better the model fits the data. The resulting estimator can be expressed by a simple formula, especially in the case of a single regressor on the right-hand side.

Logistic Regression

[20] https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/

Logistic regression is used to find the probability of event=Success and event=Failure. We should use logistic regression when the dependent variable is binary (0/ 1, True/ False, Yes/ No) in nature. Here the value of Y ranges from 0 to 1 and it can be represented by following equation:

$$odds = p/\,(1-p) \;=\; probability\ of\ event\ occurrence\ /\ probability\ of\ not\ event\ occurrence$$

$$ln(odds) \;=\; ln(p/(1-p))$$

$$logit(p) \;=\; ln(p/(1-p)) \;=\; b0 + b1X1 + b2X2 + b3X3.... + bkXk$$

Above, p is the probability of presence of the characteristic of interest. A question that you should ask here is "why have we used log in the equation?". Since we are working here with a binomial distribution (dependent variable), we need to choose a link function which is best suited for this distribution. And, it is logit function. In the equation above, the parameters are chosen to maximise the likelihood of observing the sample values rather than minimising the sum of squared errors (like in ordinary regression).

Polynomial Regression

A regression equation is a polynomial regression equation if the power of independent variable is more than 1. The equation below represents a polynomial equation: y=a+b*x^2. In this regression technique, the best fit line is not a straight line. It is rather a curve that fits into the data points. While there might be a temptation to fit a higher degree polynomial to get lower error, this can result in over-fitting. Always plot the relationships to see the fit and focus on making sure that the curve fits the nature of the problem.

Stepwise Regression

This form of regression is used when we deal with multiple independent variables. In this technique, the selection of independent variables is done with the help of an automatic process, which involves no human intervention. This feat is achieved by observing statistical values like R-square, t-stats and AIC metric to discern significant variables. Stepwise regression basically fits the regression model by adding/dropping co-variates one at a time based on a specified criterion. Some of the most commonly used Stepwise regression methods are: 1) Standard stepwise regression does two things. It adds and removes predictors as needed for each step. 2) Forward selection starts with most significant predictor in the model and adds variable for each step. 3) Backward elimination starts with all predictors in the model and removes the least significant variable for each step. The aim of this modelling technique is to maximise the prediction power with minimum number of predictor variables. It is one of the method to handle higher dimensionality of data set.

Ridge Regression

A variation of the Ordinary Least Squares technique, Ridge Regression is a technique used when the data suffers from multicollinearity (independent variables are highly correlated). In multicollinearity, even though the least squares estimates (OLS) are unbiased, their variances are large which deviates the observed value far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. In a linear equation, prediction errors can be decomposed into two sub components. First is due to the biased and second is due to the variance. Prediction error can occur due to any one of these two or both components. Here, we'll discuss about the error caused due to variance. Ridge regression solves the multicollinearity problem through shrinkage parameter λ (lambda).

Lasso Regression

A variation of the Ordinary Least Squares technique, and similar to Ridge Regression, Lasso (Least Absolute Shrinkage and Selection Operator) also penalises the absolute size of the regression coefficients. In addition, it is capable of reducing the variability and improving the accuracy of linear regression models. Lasso regression differs from ridge regression in a way that it uses absolute values in the penalty function, instead of squares. This leads to penalising (or equivalently constraining the sum of the absolute values of the estimates) values which causes some of the parameter estimates to turn out exactly zero. Larger the penalty applied, further the estimates get shrunk towards absolute zero. This results to variable selection out of given n variables.

ElasticNet Regression

ElasticNet is also a variation of the Ordinary Least Squares technique, and is actually hybrid of Lasso and Ridge Regression techniques. It is trained with L1 and L2 prior as regulariser. Elastic-net is useful when there are multiple features which are correlated. Lasso is likely to pick one of these at random, while elastic-net is likely to pick both. A practical advantage of trading-off between Lasso and Ridge is that, it allows Elastic-Net to inherit some of Ridge’s stability under rotation.

Decision Trees

Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modelling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data (but the resulting classification tree can be an input for decision making). Decision trees used in data mining are of two main types: 1) Classification tree analysis, which is used when the predicted outcome is the class to which the data belongs. 2) Regression tree analysis which is used when the predicted outcome can be considered a real number (e.g. the price of a house, or a patient's length of stay in a hospital). The term Classification And Regression Tree (CART) analysis is an umbrella term used to refer to both of the above procedures. Trees used for regression and trees used for classification have some similarities - but also some differences, such as the procedure used to determine where to split.

Gradient Tree Boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalises them by allowing optimisation of an arbitrary differentiable loss function. Gradient boosting is typically used with decision trees (especially CART trees) of a fixed size as base learners. For this special case Friedman proposes a modification to gradient boosting method which improves the quality of fit of each base learner.

Random Forest

Bootstrap aggregating, also called bagging, is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. Although it is usually applied to decision tree methods, it can be used with any type of method. Bagging is a special case of the model averaging approach. Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Support Vector Machines

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyse data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. When data are not labeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The clustering algorithm which provides an improvement to the support vector machines is called support vector clustering[21] and is often used in industrial applications either when data are not labeled or when only some data are labeled as a preprocessing for a classification pass.

Least squares support vector machines (LS-SVM)

Least squares support vector machines (LS-SVM) are least squares versions of support vector machines (SVM), which are a set of related supervised learning methods that analyse data and recognise patterns, and which are used for classification and regression analysis. In this version one finds the solution by solving a set of linear equations instead of a convex quadratic programming (QP) problem for classical SVMs. Least squares SVM classifiers, were proposed by Suykens and Vandewalle[22]. LS-SVMs are a class of kernel-based learning methods.

Naive Bayes

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes has been studied extensively since the 1950s. It was introduced under a different name into the

[21] Ben-Hur, Asa, Horn, David, Siegelmann, Hava, and Vapnik, Vladimir; "Support vector clustering" (2001) Journal of Machine Learning Research, 2: 125–137.

[22] Suykens, J.A.K.; Vandewalle, J. (1999) "Least squares support vector machine classifiers", *Neural Processing Letters*, 9 (3), 293-300.

text retrieval community in the early 1960s, and remains a popular (baseline) method for text categorisation, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including support vector machines. It also finds application in automatic medical diagnosis. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers. In the statistics and computer science literature, Naive Bayes models are known under a variety of names, including simple Bayes and independence Bayes. All these names reference the use of Bayes' theorem in the classifier's decision rule, but naive Bayes is not (necessarily) a Bayesian method.

Gaussian naive Bayes

A variation of naive Bayes, used when dealing with continuous data, making a typical assumption that the continuous values associated with each class are distributed according to a Gaussian distribution.

Multinomial naive Bayes

A variation of naive Bayes, implementing the naive Bayes algorithm for multinomially distributed data. With a multinomial event model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial.

Bernoulli naive Bayes

A variation of naive Bayes, Bernoulli Naive Bayes is applied to data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued variable. In the multivariate Bernoulli event model, features are independent booleans (binary variables) describing inputs.

k-Nearest Neighbours (k-NN)

In pattern recognition, the k-nearest neighbours algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression

In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbour.

In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbours.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

Both for classification and regression, it can be useful to assign weight to the contributions of the neighbours, so that the nearer neighbours contribute more to the average than the more distant ones. The neighbours are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data. The algorithm is not to be confused with k-means, another popular machine learning technique.

Perceptron

In machine learning, the perceptron is an algorithm for supervised learning of binary classifiers (functions that can decide whether an input, represented by a vector of numbers, belongs to some specific class or not). It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector. The algorithm allows for online learning, in that it processes elements in the training set one at a time. The perceptron is a linear classifier, therefore it will never get to the state with all the input vectors classified correctly if the training set D is not linearly separable, i.e. if the positive examples cannot be separated from the negative examples by a hyperplane. In this case, no "approximate" solution will be gradually approached under the standard learning algorithm, but instead learning will fail completely. Hence, if linear separability of the training set is not known a priori, one of the training variants below should be used. But if the training set is linearly separable, then the perceptron is guaranteed to converge, and there is an upper bound on the number of times the perceptron will adjust its weights during the training.

Multiclass Perceptron

Like most other techniques for training linear classifiers, the perceptron generalises naturally to multiclass classification. Here, the input x and the output y are drawn from arbitrary sets. A feature representation function f(x,y) maps each possible input/output pair to a finite-dimensional real-valued feature vector. As before, the feature vector is multiplied by a weight vector w, but now the resulting score is used to choose among many possible outputs.

The following table summarises the list of analysed algorithms, highlighting the main problem types that these algorithms aim at resolving.

Algorithm	Variation of	Main Problem Type
Linear Regression	-	Binary Classification, Multiclass Classification, Regression
Bayesian linear regression	Linear Regression	Binary Classification, Multiclass Classification, Regression
Ordinary Least Squares	-	Regression
Logistic Regression	-	Binary Classification, Multiclass Classification, Regression
Polynomial Regression	-	Regression
Stepwise Regression	-	Regression
Ridge Regression	Ordinary Least Squares	Regression

Lasso Regression	Ordinary Least Squares	Regression
ElasticNet Regression	Ordinary Least Squares	Regression
Decision Trees	-	Binary Classification, Multiclass Classification, Regression
Gradient Tree Boosting	Decision Trees	Binary Classification, Multiclass Classification, Regression
Random Forest	Decision Trees	Binary Classification, Multiclass Classification, Regression
Support Vector Machines	-	Binary Classification, Multiclass Classification, Regression
Least squares support vector machines (LS-SVM)	Support Vector Machines	Binary Classification, Multiclass Classification, Regression
Naive Bayes	-	Binary Classification, Multiclass Classification
Gaussian naive Bayes	Naive Bayes	Binary Classification, Multiclass Classification
Multinomial naive Bayes	Naive Bayes	Binary Classification, Multiclass Classification
Bernoulli Naive Bayes	Naive Bayes	Binary Classification, Multiclass Classification
k-Nearest Neighbours (k-NN)	-	Binary Classification, Multiclass Classification, Regression
Perceptron		Binary Classification
Multiclass Perceptron	Perceptron	Multiclass Classification, Regression

Table 4-1: List of classification and regression algorithms

Classification & Regression Software Implementation

Software implementation of the majority of the classification and regression algorithms aforementioned is already available through the machine learning library of SPARK and is available here[23]. Nevertheless, additional open source implementations in other programming languages (including for example Python) are also available in other statistics libraries, including for example Java-ML, R and Weka. In addition, custom software libraries e.g. for the other classification and regression techniques aforementioned can also be developed in the case of specific requirements not met by the open source libraries available.

Classification & Regression Usability in BigDataOcean

[23] https://spark.apache.org/docs/latest/mllib-classification-regression.html

Classification and regression algorithms can have multiple applications on BigDataOcean project. To list only some of them the following examples are applicable:

- Linear regression could be used to check if we can predict the value of a feature provided that we know the value of another feature and we suspect that there is relationship between them. For example, we could apply linear regression with a regularisation method to determine if we can predict the value of sea water speed having the wind speed as the only input.
- If the prediction of value is based on a high dimensional dataset, the Lasso regression method with specified weights indicating the importance of each variable could indicate the correlation of each variable on the predicted value. One example of this could be the wave energy potential could be predicted and a high dimensional dataset containing ocean and wave characteristics.
- Due to their rule-based architecture, decision trees can detect efficiently events from data without noise. Using streaming data from sensors could easily detect events and the possibility an alert to be raised.

4.1.2.2 Collaborative Filtering (Recommendation)

Collaborative Filtering (Recommendation) Description

In machine learning, collaborative filtering is the method of making automatic predictions or filtering about the interests of a user by collecting preferences, activities and behaviours of many users. Collaborative filtering is one of the techniques used by recommender systems. Recommender systems are systems that are using machine learning algorithms from the field of artificial intelligence in order to provide product or service recommendations to the users. These systems are used to help users find new items based on the information about the user or the recommended item and play an important role in decision-making, helping users to maximise profits or minimise risks. Recommender systems and especially systems designed for collaborative filtering have become increasingly popular the latest years among large vendors and are utilised in a variety of areas and industries.

Collaborative filtering methods are based on collecting and analysing a large amount of information on users' behaviours, activities or preferences and predicting what users will like based on their similarity to other users. A key advantage of the collaborative filtering approach is that it does not rely on machine analysable content and therefore it is capable of accurately recommending complex items without requiring an "understanding" of the item itself. The concept of collaborative filtering is based on the assumption that users who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past. So, the underlying assumption of the collaborative filtering approach is that if a person A has the same opinion as a person B on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person.

A model is built for the user's behaviour based on explicit and implicit forms of data collection. Explicit data collection includes user's preferences through personal rating, voting or ranking of items and implicit data collection includes analysing and tracking user's behaviour like search terms, viewing or usage of the items. The comparison of the collected data to similar or dissimilar data collected from other users is involved in the calculation of the list of recommended items for the user.

Collaborative filtering relies only on the past user behaviour such as previous items viewed, used or rated without the need of explicit profiles for the items.

The two primary areas of collaborative filtering are the neighbourhood methods and latent factor models. Neighbourhood methods are centred on computing the relationships between items or, alternatively, between users. The item-oriented approach evaluates a user's preference for an item based on ratings of "neighbouring" items by the same user. A product's neighbours are other products that tend to get similar ratings when rated by the same user.

Latent factor models are an alternative approach that tries to explain the ratings by characterising both items and users on a number of factors inferred from the ratings patterns. Some of the most successful realisations of latent factor models are based on matrix factorisation. In its basic form, matrix factorisation characterises both items and users by vectors of factors inferred from item rating patterns. High correspondence between item and user factors leads to a recommendation. These methods have become popular in recent years by combining good scalability with predictive accuracy. In addition, they offer much flexibility for modelling various real-life situations.

Matrix factorisation models map both users and items to a joint latent factor space of dimensionality f, such that user-item interactions are modeled as inner products in that space. Accordingly, each item i is associated with a vector $q_i \in R^f$ and each user u is associated with a vector $p_u \in R^f$. For a given item i, the elements of q_i measure the extent to which the item possesses those factors, positive or negative. For a given user u, the elements of p_u measure the extent of interest the user has in items that are high on the corresponding factors, again, positive or negative. The resulting dot product, $q_i^T p_u$, captures the interaction between user u and item i the user's overall interest in the item's characteristics.

To learn the factor vectors (p_u and q_i), the system minimises the regularised squared error on the set of known ratings:

$$\min_{q^{*},p^{*}} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\| q_i \|^2 + \| p_u \|^2)$$

where κ is the set of the (u,i) pairs for which r_{ui} is known (the training set).

The system learns the model by fitting the previously observed ratings. However, the goal is to generalise those previous ratings in a way that predicts future, unknown ratings. Thus, the system should avoid overfitting the observed data by regularising the learned parameters, whose magnitudes are penalised. The constant λ controls the extent of regularisation and is usually determined by cross-validation.

Alternating Least Squares

Collaborative filtering aims to fill the missing entries of the user-item association matrix and learn the latent factors described above to predict the missing entries. For this purpose, the algorithm of alternating least squares (ALS) is used. That is because since both q_i and p_u are unknown the equation is not convex. However, if we fix one of the unknowns, the optimisation problem becomes quadratic and can be solved optimally. Thus, ALS techniques rotate between fixing the q_i 's and fixing the p_u's. When all p_u's are fixed, the system re-computes the q_i's by solving a least-squares problem, and vice versa. This ensures that each step decreases equation until convergence.

The reason why the ALS is preferable is that it makes use of system's parallelisation since the system can compute each q_i independently of the other item factors and also can compute each p_u

independently of the other user factors. This gives rise to potentially massive parallelisation of the algorithm.

Collaborative Filtering (Recommendation) Software Implementation

Software implementation of collaborative filtering is already available through the machine learning library of SPARK and is available here[24]. Implementation is available in Scala, Java and Python. Nevertheless, additional open source implementations in other programming languages are also available in other statistics libraries, including for example R and Weka. In addition, custom software libraries e.g. for the other sampling techniques aforementioned can also be developed in the case of specific requirements not met by the open source libraries available.

4.1.2.3 Clustering

Clustering Description

Cluster analysis or clustering is a multivariate method which aims to classify a set of objects on the basis of a set of measured variables into a number of different groups (clusters) such that similar subjects are placed in the same group (called cluster). In other words, clustering is an exploratory data analysis tool which aims at sorting different objects into groups in a way that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise. If plotted geometrically the objects within the clusters will be close together, while the distance between clusters will be farther apart. Cluster analysis is also referred as segmentation analysis or taxonomy analysis. It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics.

However, cluster analysis itself is not one specific algorithm nor an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimisation that involves trial and failure. It is often necessary to modify data preprocessing and model parameters until the result achieves the desired properties. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them.

The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. The data used in cluster analysis can be interval, ordinal or categorical. However, having a mixture of different types of variable will make the analysis more complicated. This is because in cluster analysis you need to have some way of measuring the distance between observations and the type of measure used will depend on what type of data you have. In general, the following questions must be addressed:

- How do we measure similarity
- How do we form clusters
- How many clusters do we form

[24] https://spark.apache.org/docs/latest/mllib-collaborative-filtering.html

There are a number of different methods that can be used to carry out a cluster analysis and these methods can be classified as follows:

- Hierachical methods
    - Agglomerative methods: Each object start in their own separate cluster; the two most similar clusters are then combined and this is done repeatedly until all objects are in one cluster. At the end, the optimum number of clusters is chosen out of all cluster solutions.
    - Divisive methods: All objects start in the same cluster and the reverse strategy from Agglomerative method is applied until every object is in their own separate cluster.
- Non-hierarchical methods also known as k-means clustering methods. In these methods the desired number of clusters in specified and the most appropriate solution is chosen.

K-means

K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. K-means clustering is a method of vector quantisation, originally from signal processing, that is popular for cluster analysis in data mining and is one of the most commonly used clustering algorithms that clusters the data points into a predefined number of clusters.

K-means clustering aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. The main idea is to define *k* centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate *k* new centroids as barycentre of the clusters resulting from the previous step. After we have these *k* new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the *k* centroids change their location step by step until no more changes are done. In other words, centroids do not move any more.

Finally, this algorithm aims at minimising an *objective function*, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

,

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the *n* data points from their respective cluster centres.

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centres. The k-means algorithm can be run multiple times to reduce this effect.

Gaussian mixture

Gaussian mixture models (GMM) are often used for data clustering. A mixture model is a probabilistic model for representing the presence of subpopulations within an overall population, without requiring that an observed data set should identify the sub-population to which an individual observation belongs. Formally a mixture model corresponds to the mixture distribution that represents the probability distribution of observations in the overall population. However, while problems associated with "mixture distributions" relate to deriving the properties of the overall population from those of the sub-populations, "mixture models" are used to make statistical inferences about the properties of the sub-populations given only observations on the pooled population, without sub-population identity information.

A Gaussian Mixture Model represents a composite distribution whereby points are drawn from one of k Gaussian sub-distributions, each with its own probability. So, a Gaussian mixture can be described a probabilistic model in which all points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters.

To induce the maximum-likelihood model the expectation-maximisation algorithm is used. The expectation-maximisation algorithm is an iterative method to find maximum likelihood or maximum a posteriori estimates of parameters in statistical model that depend on unobserved latent variables. This is achieved by alternating between performing an expectation step which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximisation step, which computes parameters maximising the expected log-likelihood found on the expectation step. These parameter-estimates are then used to determine the distribution of the latent variables in the next expectation step.

Power iteration clustering (PIC)

Power iteration clustering is a simple and scalable clustering method that finds a very low-dimensional data embedding using truncated power iteration on a normalised pair-wise similarity matrix of the data points. It computes a pseudo-eigenvector of the normalised affinity matrix of the graph via power iteration and uses it to cluster vertices. More specific, it uses power iteration to find a vector that is a linear combination of the *k* eigenvectors corresponding to the *k* smallest eigenvalues $\lambda_1, . . ., \lambda_k$ of the normalised Laplacian matrix $D^{-1}L = I - D^{-1}W$, where λ_k is assumed to be significantly greater than λ_{k+1}, and all λ_i for i in {2, . . . , k} are assumed to be sufficiently close to λ_1. The vector found by power iteration is viewed as a 1-dimensional embedding of the data points and is used to cluster data points into *k* clusters, in a manner similar to spectral clustering.

Latent Dirichlet allocation (LDA)

Latent Dirichlet allocation (LDA) is a generative probabilistic model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics. More specific topics correspond to cluster centres, and documents correspond to examples (rows) in a dataset. Topics and documents both exist in a feature space, where feature vectors are vectors of word counts (bag of words). Rather than estimating a clustering using a traditional distance, LDA uses a function based on a statistical model of how text documents are generated.

Bisecting k-means

Bisecting k-means is like a combination of k-Means and hierarchical clustering. Instead of partitioning the data into *k* clusters in each iteration, Bisecting k-means splits one cluster into two sub clusters at each bisecting step (by using k-means) until *k* clusters are obtained. As Bisecting k-means is based on k-means, it keeps the merits of k-means and also has some advantages over k-means.

First, Bisecting k-means is more efficient when *k* is large. For the k-means algorithm, the computation involves every data point of the data set and *k* centroids. On the other hand, in each Bisecting step of Bisecting k-means, only the data points of one cluster and two centroids are involved in the computation. Thus, the computation time is reduced. Secondly, Bisecting k-means produce clusters of similar sizes, while k-means is known to produce clusters of widely different sizes. Finally, bisecting K-means has a time complexity which is linear in the number of observations. If the number of clusters is large and if refinement is not used, then bisecting k-means is even more efficient than the regular K-means algorithm.

Streaming k-means

Streaming k-means is the online algorithm for k-means. An online algorithm is one that can process its input piece-by-piece in a serial fashion, i.e., in the order that the input is fed to the algorithm, without having the entire input available from the start. When data arrive in a stream, we may want to estimate clusters dynamically, updating them as new data arrive. The computations are performed iteratively, with data arriving during the computation is single observations or in batches and typically an intermediate result is calculated based on the existing data and then recalculated as new data arrive. For each batch of new data, all points are assigned to the nearest cluster, new cluster centres are computed and each cluster is updated with a controlled decay of the estimates as parameter.

Clustering Software Implementation

Software implementation of the aforementioned clustering algorithms are already available through the machine learning library of SPARK and is available here[25]. Nevertheless, additional open source implementations in other programming languages (including for example Python) are also available in other statistics libraries, including for example Java-ML, R and Weka. In addition, custom software libraries e.g. for the other classification and regression techniques aforementioned can also be developed in the case of specific requirements not met by the open source libraries available.

Clustering Usability in BigDataOcean

Cluster methods described can be used in any scenario that requires grouping of data into a number of groups or the categorisation of data based on selected attribute.

[25] https://spark.apache.org/docs/latest/mllib-clustering.html

4.1.3 Featurisation

Featurisation comprise the third major category of algorithms to be examined and included within the context of BigDataOcean. This category of algorithms includes mainly algorithms for Dimensionality reduction, Feature Extraction, Transformation and Selection.

4.1.3.1 Dimensionality Reduction

Dimensionality Reduction Description

The recent explosion of data set size, in number of records and attributes, has triggered the development of a number of big data platforms as well as parallel data analytics algorithms. At the same time though, it has pushed for usage of data dimensionality reduction procedures. In machine learning and statistics, **dimensionality reduction** or **dimension reduction** is the process of reducing the number of random variables under consideration. It can be used to extract latent features from raw and noisy features or compress data while maintaining the structure.

Large amounts of data might sometimes produce worse performances in data analytics applications. As most data mining and analytics algorithms are column-wise implemented, which makes them slower and slower on a growing number of data columns. The purpose of dimensionality reduction is to reduce the number of columns in the data set and lose the smallest amount of information possible at the same time.

Singular value decomposition (SVD)

Singular value decomposition (SVD) factorises of a real or complex matrix is the factorisation of A into the product of three matrices A = UDV^T where:

- U is an orthonormal matrix, whose columns are called left singular vectors,
- D is a diagonal matrix with non-negative diagonals in descending order, whose diagonals are called singular values,
- V is an orthonormal matrix, whose columns are called right singular vectors.

Singular value decomposition is the generalisation of the eigendecomposition of a positive semidefinite normal matrix (for example, a symmetric matrix with positive eigenvalues) to any matrix via an extension of the polar decomposition. Singular value decomposition provides a convenient way for breaking a matrix, which perhaps contains some data we are interested in, into simpler, meaningful pieces.

Principal component analysis (PCA)

Principal component analysis (PCA) is probably the most popular multivariate statistical technique and it is used by almost all scientific disciplines. PCA is used abundantly in all forms of analysis and has found application in fields such as face recognition and image compression. PCA is a common technique for finding patterns in data of high dimension because it is a simple, non-parametric method of extracting relevant information from confusing data sets. The other main advantage of PCA is that once you have found these patterns in the data you compress the data, in example by reducing the number of dimensions, without much loss of information. So, with minimal additional effort PCA provides a roadmap for how to reduce a complex data set to a lower dimension to reveal the sometimes hidden, simplified dynamics that often underlie it. It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences.

PCA analyses a data table representing observations described by several dependent variables, which are, in general, inter-correlated. Its goal is to extract the important information from the data table and to express this information as a set of new orthogonal variables called principal components. PCA converts a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables by using an orthogonal transformation.

To achieve that it computes the first coordinate with the largest variance possible making sure that the succeeding coordinate in turn will have the largest variance possible. PCA also represents the pattern of similarity of the observations and the variables by displaying them as points in maps.

PCA is used widely in dimensionality reduction.

Self-Organising Map (SOM)

The Self-Organising Map (SOM), also known as Kohonen network, is one of the most popular artificial neural network models. The Self-Organising Map is based on unsupervised learning, which means that no human intervention is needed during the learning and that little needs to be known about the characteristics of the input data. The SOM could be used for clustering data without knowing the class memberships of the input data. It is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretised representation of the input space of the training samples, called a map, and is therefore a method to do dimensionality reduction.

Self-organising maps differ from other artificial neural networks as they apply competitive learning as opposed to error-correction learning (such as backpropagation with gradient descent), and in the sense that they use a neighbourhood function to preserve the topological properties of the input space. It provides a topology preserving mapping from the high dimensional space to map units. Map units, or neurons, usually form a two-dimensional lattice and thus the mapping is a mapping from high dimensional space onto a plane. The property of topology preserving means that the mapping preserves the relative distance between the points. Points that are near each other in the input space are mapped to nearby map units in the SOM.

The SOM can thus serve as a cluster analysing tool of high-dimensional data. Also, the SOM has the capability to generalise. Generalisation capability means that the network can recognise or characterise inputs it has never encountered before. A new input is assimilated with the map unit it is mapped to. The SOM can be used to detect features inherent to the problem and thus has also been called SOFM, the Self-Organising Feature Map.

Dimensionality Reduction Software Implementation

Software implementation of the aforementioned dimensionality reduction algorithms, except from the Self-Organising Map, are already available through the machine learning library of SPARK and is available here[26]. Nevertheless, additional open source implementations in other programming

[26] https://spark.apache.org/docs/latest/mllib-dimensionality-reduction.html

languages (including for example Python) are also available in other statistics libraries, including for example Java-ML, R and Weka which include also an implementation of Self-Organising Map algorithm. In addition, custom software libraries e.g. for the other classification and regression techniques aforementioned can also be developed in the case of specific requirements not met by the open source libraries available.

Dimensionality Reduction Usability in BigDataOcean

BigDataOcean will probably deal with a lot datasets that are quite large. However, in large datasets the possibility that highly correlated subsets of variable exist is very high. The accuracy and reliability of a classification or prediction model will suffer if we include highly correlated variables or variables that are unrelated to the outcome of interest because of over fitting.

In that case PCA can be used, selecting a subset of variables together with a low complexity method for classification or regression. Applying PCA multiple times will make sure that the optimal number of selected variable is reached.

To achieve high-dimensional data exploration Self-Organising Map should be used. The most common usage of Self-Organising Map is the grouping of observations which leads to a mapping of similar behaviour that can be easily visualised and give useful insights.

4.1.3.2 Feature Extraction

Feature Extraction Description

Feature extraction methods construct combinations of input variables and practically offer a new -reduced- set of features by transforming the original ones. The desired task can be performed with sufficient accuracy by employing this reduced representation instead of the complete initial dataset. Especially in classification and regression algorithms, a carefully extracted feature set has been proven to increase the learning rates and overall performance in many cases. It is worth mentioning there are simpler algorithms for dimensionality reduction, belonging to the 'feature selection' category, whose purpose is to find a subset of the initial set of features by removing the most redundant or irrelevant ones.

Term frequency-inverse document frequency (TF-IDF)

Tf-idf stands for term frequency-inverse document frequency, and the tf-idf weight is a weight often used in information retrieval and text mining and user modelling. The statistical measure used in order to evaluate the importance of a word to a document in a collection or corpus is the tf-idf weight. The number of times a word appears in the document increases proportionally the tf-idf value, but the tf-idf value is often offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. Nowadays, tf-idf is one of the most popular term-weighting schemes.

Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. One of the simplest ranking functions is computed by summing the tf-idf for each query term; many more sophisticated ranking functions are variants of this simple model.

This algorithm is useful when you have a document set, particularly a large one, which needs to be categorised. It is especially nifty because you don't need to train a model ahead of time and it will automatically account for differences in lengths of documents.

Word2Vec

Word2vec is a particularly computationally-efficient predictive model for learning word embeddings from raw text. Word2vec consists of a group of related models, which are shallow, two-layer neural networks. These models are the Continuous Bag-of-Words model (CBOW) and the Skip-Gram model. Algorithmically, these models are similar, except that CBOW predicts target words (e.g. 'mat') from source context words ('the cat sits on the'), while the skip-gram does the inverse and predicts source context-words from the target words. This inversion might seem like an arbitrary choice, but statistically it has the effect that CBOW smooths over a lot of the distributional information (by treating an entire context as one observation). For the most part, this turns out to be a useful thing for smaller datasets. However, skip-gram treats each context-target pair as a new observation, and this tends to do better when we have larger datasets.

The purpose and usefulness of Word2vec is to group the vectors of similar words together in vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space. That is, it detects similarities mathematically. Word2vec creates vectors that are distributed numerical representations of word features, features such as the context of individual words. It does so without human intervention.

4.1.3.3 Feature Selection

In machine learning and statistics, feature selection, also known as variable selection or attribute selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. It is the automatic selection of attributes in your data (such as columns in tabular data) that are most relevant to the predictive modelling problem you are working on.

The objective of feature selection is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data.

Chi-Squared feature selection

Chi Square Test is used in statistics to test the independence of two events. Given dataset about two events, we can get the observed count O and the expected count E. Chi Square Score measures how much the expected counts E and observed count O derivate from each other.

In feature selection, the two events are occurrence of the feature and occurrence of the class. In other words, we want to test whether the occurrence of a specific feature and the occurrence of a specific class are independent. If the two events are dependent, we can use the occurrence of the feature to predict the occurrence of the class. We aim to select the features, of which the occurrence is highly dependent on the occurrence of the class.

When the two events are independent, the observed count is close to the expected count, thus a small chi square score. High scores on X^{22} indicate that the null hypothesis of independence should be rejected and thus that the occurrence of the term and class are dependent. If they are dependent then we select the feature for the text classification. In other words, the higher value of the X^2 score, the more likelihood the feature is correlated with the class, thus it should be selected for model training.

Recursive feature elimination (RFE)

Recursive feature elimination algorithm firstly fits the model to all of the features. Each feature is then ranked according to its importance to the model. Let S be a sequence of ordered numbers representing the number of features to be kept ($S_1 > S_2 > S_3$...). At each iteration of feature selection algorithm, the S_i top raked features are kept, the model is refit and the accuracy is assessed. The value of S_i with the best accuracy is assessed and the top S_i features are used to fit the final model.

Feature Extraction and Selection Software Implementation

Software implementation of the aforementioned feature extraction and selection algorithms, are already available through the machine learning library of SPARK and Spark-sklearn library which are available here[27]. Nevertheless, additional open source implementations in other programming languages (including for example Python) are also available in other statistics libraries, including for example Java-ML, R and Weka. In addition, custom software libraries e.g. for the other classification and regression techniques aforementioned can also be developed in the case of specific requirements not met by the open source libraries available.

Feature Extraction and Selection Usability in BigDataOcean

Term frequency-inverse document frequency can be used when scoring or ranking a documents' relevance to any word when it is needed. One common use of the algorithm is the event detection based on the measure of the word in several documents. On the other hand, Word2vec is a useful method to identify any inconsistency in the given values of a datasets. For example, it identifies any word that does not fit in a list of words. Finally, recursive feature elimination could be used to eliminate feature with useless information from a dataset.

4.1.4 List of algorithms

The following table summarises the list of algorithms that will be made available through the BigDataOcean platform to its end users.

Algorithm	Variation of	Main Problem Type
Mean		Summary statistics (measures of location)
Geometric mean	Mean	Summary statistics (measures of location)
Trimmed mean	Mean	Summary statistics (measures of location)
Median		Summary statistics (measures of location)
Range		Summary statistics (measures of spread)

[27] https://spark.apache.org/docs/latest/mllib-feature-extraction.html

Interquartile range	Range	Summary statistics (measures of spread)
Quartiles	-	Summary statistics (measures of spread)
Skewed	-	Summary statistics (measures of spread)
Kurtosis	-	Summary statistics (measures of spread)
Pearson r correlation		Correlation
Spearman rank correlation	-	Correlation
Kendall rank correlation	-	Correlation
Bernoulli sampling	-	Sampling
Cluster sampling	-	Sampling
Systematic sampling	-	Sampling
Simple Random Sample	-	Sampling
Stratified Sampling	-	Sampling
One-sample tests	-	Hypothesis Testing
Two-sample tests	-	Hypothesis Testing
Paired tests	-	Hypothesis Testing
Z-tests	-	Hypothesis Testing
T-tests	-	Hypothesis Testing
Chi-squared tests	-	Hypothesis Testing
F-tests	-	Hypothesis Testing
Random data Generator	-	Random data generation
Linear Regression	-	Binary Classification, Multiclass Classification, Regression
Bayesian linear regression	Linear Regression	Binary Classification, Multiclass Classification, Regression
Ordinary Least Squares	-	Regression
Logistic Regression	-	Binary Classification, Multiclass Classification, Regression

Polynomial Regression	-	Regression
Stepwise Regression	-	Regression
Ridge Regression	Ordinary Least Squares	Regression
Lasso Regression	Ordinary Least Squares	Regression
ElasticNet Regression	Ordinary Least Squares	Regression
Decision Trees	-	Binary Classification, Multiclass Classification, Regression
Gradient Tree Boosting	Decision Trees	Binary Classification, Multiclass Classification, Regression
Random Forest	Decision Trees	Binary Classification, Multiclass Classification, Regression
Support Vector Machines	-	Binary Classification, Multiclass Classification, Regression
Least squares support vector machines (LS-SVM)	Support Vector Machines	Binary Classification, Multiclass Classification, Regression
Naive Bayes	-	Binary Classification, Multiclass Classification
Gaussian naive Bayes	Naive Bayes	Binary Classification, Multiclass Classification
Multinomial naive Bayes	Naive Bayes	Binary Classification, Multiclass Classification
Bernoulli Naive Bayes	Naive Bayes	Binary Classification, Multiclass Classification
k-Nearest Neighbours (k-NN)	-	Binary Classification, Multiclass Classification, Regression
Perceptron	-	Binary Classification
Multiclass Perceptron	Perceptron	Multiclass Classification, Regression
Collaborative filtering	-	Recommendation
K-means	-	Clustering, Classification
Gaussian mixture	-	Clustering, Classification
Power iteration clustering (PIC)	-	Clustering, Classification
Latent Dirichlet allocation (LDA)	-	Clustering, Classification
Bisecting k-means	K-means	Clustering, Classification

Streaming k-means	K-means	Clustering, Classification
Singular value decomposition (SVD)	-	Dimensionality reduction
Principal component analysis (PCA)	-	Dimensionality reduction
Self-Organising Map (SOM)	-	Dimensionality reduction
Term frequency-inverse document frequency (TF-IDF)	-	Feature extraction
Word2Vec	-	Feature extraction
Chi-Squared feature selection	-	Feature Selection
Recursive feature elimination (RFE)	-	Feature Selection

Table 4-2: List of available algorithms

The following table maps the desired functions per pilot, to the list of algorithms that will be made available through the BigDataOcean platform to the end users.

Pilot	Desired Function / Purpose Description	Facilitating Algorithm
Mare Protection	Get oil spill seasonal statistics on ship routes regarding historical runs	Mean or Geometric mean, Range or Quartiles
Mare Protection	Detect correlation between vessels position and oil spill trajectory	Pearson r, Spearman rank or Kendall rank correlation
Mare Protection	Perform basic quality control on in-situ data from various sources available on BigDataOcean platform	Range or Interquartile range
Mare Protection	Clustering of meteorological and oceanographic data coming from different sources, like in-situ measurements (from fixed stations, Argos, vessels, etc.) or forecasting models. For instance, group parameters into wind, waves, currents, sea temperature, salinity or other	K-means or Streaming k-means, Gaussian mixture models (GMM) or Power iteration clustering (PIC)
Mare Protection	Get subset of meteorological and oceanographic data: decide and cut the temporal period and the geographical area of interest in order to save downloading time and resources	Recursive feature elimination (RFE)
Wave Energy	Verify the applicability of different wave energy converter technologies to a sea	Summary Statistics: mean, range, interquartile range

	location – through the evaluation of historical wave conditions and checking the survivability characteristics of the technology	
Wave Energy	Determine wave characteristics at specific locations (which are not located in the grid points of numerical models)	Regression: Linear regression
Wave Energy	Evaluation of similarities between different locations (in terms of wave potential)	Clustering: K-means
Wave Energy	Evaluation of similarities between different locations (in terms of wave potential)	Clustering: K-means
Anomaly Detection	Detect correlations between vessel positional data and other data sources	Pearson r correlation, Spearman rank correlation, or Kendall rank correlation
Anomaly Detection	Perform analytics on specific vessel types	Systematic sampling, or Stratified sampling
Anomaly Detection	Identify route similarities among vessels performing the same itinerary	K-means, Bisecting k-means, Gaussian mixture models, or Density-based spatial clustering (DBSCAN)
Predictive Maintenance	Search for correlations between the strain of a vessel's main engine and the sea or weather conditions.	Pearson r correlation, Spearman rank correlation, Kendall rank correlation
Predictive Maintenance	Predict future maintenance needs, based both on vessel's and external parameters.	Linear Regression, Bayesian linear regression, Ordinary Least Squares, Logistic Regression, Polynomial Regression, Decision Trees, Random Forest
Predictive Maintenance	Classify the different vessel routes in terms of main engine's strain, based on external parameters.	Decision Trees, Random Forest, Support Vector Machines, Naive Bayes, Multiclass Perceptron
Predictive Maintenance	Perform hypothesis testing about vessels or routes.	One-sample tests, Two-sample tests, Paired tests, Z-tests, T-tests, Chi-squared tests, F-tests

Table 4-3: Mapping pilot needs to algorithms available

The following table maps the list of algorithms that will be made available through the BigDataOcean platform to the end users, to the desired functions to be supported by the project pilots. Ultimately, these algorithms will be able to support additional services, both within the context of the needs of the project pilots, as well as needs of external stakeholders wishing to implement new or exploit existing services offered through the BigDataOcean platform.

Name	Applicable Pilots	Purpose
Summary statistics: (Mean, Geometric mean, Range, Quartiles, Interquartile range)	• Mare Protection • Wave Energy	• Get oil spill seasonal statistics on ship routes regarding historical runs. • Perform basic quality control on in-situ data from various sources available on BigDataOcean platform • Verify the applicability of different wave energy converter technologies to a sea location – through the evaluation of historical wave conditions and checking the survivability characteristics of the technology.
Pearson r correlation	• Mare protection • Anomaly detection • Predictive maintenance	• Detect correlation between vessels position and oil spill trajectory • Detect correlations between vessel positional data and other data sources • Search for correlations between the strain of a vessel's main engine and the sea or weather conditions.
Spearman rank correlation	• Mare protection • Anomaly detection • Predictive maintenance	• Detect correlation between vessels position and oil spill trajectory • Detect correlations between vessel positional data and other data sources • Search for correlations between the strain of a vessel's main engine and the sea or weather conditions.
Kendal rank correlation	• Mare protection • Anomaly detection • Predictive maintenance	• Detect correlation between vessels position and oil spill trajectory • Detect correlations between vessel positional data and other data sources • Search for correlations between the strain of a vessel's main engine and the sea or weather conditions.
K-means	• Mare protection • Wave Energy • Anomaly detection	• Clustering of meteorological and oceanographic data coming from different sources, like in-situ measurements (from fixed stations, Argos, vessels, etc.) or forecasting models. For instance, group parameters into wind, waves, currents, sea temperature, salinity or other

		• Evaluation of similarities between different locations (in terms of wave potential) • Evaluation of similarities between different locations (in terms of wave potential) • Identify route similarities among vessels performing the same itinerary
Streaming k-means	• Mare protection	• Clustering of meteorological and oceanographic data coming from different sources, like in-situ measurements (from fixed stations, Argos, vessels, etc.) or forecasting models. For instance, group parameters into wind, waves, currents, sea temperature, salinity or other
Bisecting k-means	• Anomaly detection	• Identify route similarities among vessels performing the same itinerary
Gaussian mixture models	• Mare protection • Anomaly detection	• Clustering of meteorological and oceanographic data coming from different sources, like in-situ measurements (from fixed stations, Argos, vessels, etc.) or forecasting models. For instance, group parameters into wind, waves, currents, sea temperature, salinity or other • Identify route similarities among vessels performing the same itinerary
Power iteration clustering (PIC)	• Mare protection	• Clustering of meteorological and oceanographic data coming from different sources, like in-situ measurements (from fixed stations, Argos, vessels, etc.) or forecasting models. For instance, group parameters into wind, waves, currents, sea temperature, salinity or other
Recursive feature elimination (RFE)	• Mare protection	• Get subset of meteorological and oceanographic data: decide and cut the temporal period and the geographical area of interest in order to save downloading time and resources
Linear regression	• Wave Energy • Predictive Maintenance	• Determine wave characteristics at specific locations (which are not located in the grid points of numerical models) • Predict future maintenance needs, based both on vessel's and external parameters.
Bayesian linear regression	• Predictive Maintenance	• Predict future maintenance needs, based both on vessel's and external parameters.

Systematic sampling	• Anomaly detection	• Perform analytics on specific vessel types
Stratified sampling	• Anomaly detection	• Perform analytics on specific vessel types
Density-based spatial clustering (DBSCAN)	• Anomaly detection	• Identify route similarities among vessels performing the same itinerary
Ordinary Least Squares	• Predictive Maintenance	• Predict future maintenance needs, based both on vessel's and external parameters.
Logistic Regression	• Predictive Maintenance	• Predict future maintenance needs, based both on vessel's and external parameters.
Polynomial Regression	• Predictive Maintenance	• Predict future maintenance needs, based both on vessel's and external parameters.
Decision Trees	• Predictive Maintenance	• Predict future maintenance needs, based both on vessel's and external parameters. • Classify the different vessel routes in terms of main engine's strain, based on external parameters.
Random forest	• Predictive Maintenance	• Predict future maintenance needs, based both on vessel's and external parameters. • Classify the different vessel routes in terms of main engine's strain, based on external parameters.
Support Vector machines	• Predictive Maintenance	• Classify the different vessel routes in terms of main engine's strain, based on external parameters.
Naive Bayes	• Predictive Maintenance	• Classify the different vessel routes in terms of main engine's strain, based on external parameters.
Multiclass Perceptron	• Predictive Maintenance	• Classify the different vessel routes in terms of main engine's strain, based on external parameters.
One-sample tests	• Predictive Maintenance	• Perform hypothesis testing about vessels or routes.
Two-sample tests	• Predictive Maintenance	• Perform hypothesis testing about vessels or routes.
Paired tests	• Predictive Maintenance	• Perform hypothesis testing about vessels or routes.
Z-tests	• Predictive Maintenance	• Perform hypothesis testing about vessels or routes.
T-tests	• Predictive Maintenance	• Perform hypothesis testing about vessels or routes.
Chi-squared tests	• Predictive Maintenance	• Perform hypothesis testing about vessels or routes.
F-tests	• Predictive Maintenance	• Perform hypothesis testing about vessels or routes.

Table 4-4: Mapping algorithms available to pilot needs

4.2 Usage Analytics

In this section, we examine the different approaches that can be followed in order to record the usage of various tools and assets of the BigDataOcean platform, analyse the usage activity, provide useful statistics and extract typical usage patterns. The purpose of this activity is to better understand the needs of the users of the platform, anticipate their actions, detect possible pain points and provide proposed activities, recommendations, or potential alternatives.

Usage Analytics can be split into three different categories that we will examine separately in this chapter, alongside with metrics and possible third party solutions suitable for each of these categories:

- **Platform Usage Analytics**: Typical platform analytics that measure the size of the platform's user base, their demographics and their high-level behaviour.
- **Data Usage Analytics**: Analytics that focus on what data users search for on the platform, as well as specific dataset usage analytics within BigDataOcean services.
- **Service Usage Analytics**: Analytics about how different BigDataOcean services are deployed and consumed.

4.2.1 Platform Usage Analytics

This type of Usage Analytics is the most high-level, generic type of usage analytics that is commonly used among websites and web applications. The following metrics are typically measured as part of platform usage analytics reports:

Metric	Description
Page Views	The number of times a page was viewed.
Visits/Sessions	A visit is an interaction, by an individual, with a website consisting of one or more requests for a page. If an individual has not taken another action (typically additional page views) on the site within a specified time period, the visit session will terminate.
Unique Visitors	The number of inferred individual people (filtered for spiders and robots), within a designated reporting timeframe, with activity consisting of one or more visits to a site. Each individual is counted only once in the unique visitor measure for the reporting period.
New Visitor	The number of Unique Visitors with activity including a first-ever visit to a site during a reporting period.
Repeat Visitor	The number of Unique Visitors with activity consisting of two or more Visits to a site during a reporting period.
Return Visitor	The number of Unique Visitors with activity consisting of a visit to a site during a reporting period and where the Unique Visitor also visited the site prior to the reporting period.
Entry Page	The first page of a visit.
Landing Page	A page intended to identify the beginning of the user experience resulting from a defined marketing effort.
Exit Page	The last page on a site accessed during a visit, signifying the end of a visit/session.
Visit Duration	The length of time in a session. Calculation is typically the timestamp of the last activity in the session minus the timestamp of the first activity of the session.

Referrer	The referrer is the page URL that originally generated the request for the current page view or object.
Internal Referrer	The internal referrer is a page URL that is internal to the website or a web-property within the website as defined by the user.
External Referrer	The external referrer is a page URL where the traffic is external or outside of the website or a web-property defined by the user.
Search Referrer	The search referrer is an internal or external referrer for which the URL has been generated by a search function.
Visit Referrer	The visit referrer is the first referrer in a session, whether internal, external or null.
Original Referrer	The original referrer is the first referrer in a visitor's first session, whether internal, external or null.
Click-through	Number of times a link was clicked by a visitor.
Click-through Rate/Ratio	The number of click-throughs for a specific link divided by the number of times that link was viewed.
Page Views per Visit	The number of page views in a reporting period divided by number of visits in the same reporting period.
Page Exit Ratio	Number of exits from a page divided by total number of page views of that page.
Single-Page Visits	Visits that consist of one page regardless of the number of times the page was viewed.
Single Page View Visits (Bounces)	Visits that consist of one page-view.
Bounce Rate	Single page view visits divided by entry pages.
Event	Any logged or recorded action that has a specific date and time assigned to it by either the browser or server.
Conversion	A visitor completing a target action.

Table 4-5: Commonly used Platform Analytics Metrics

Monitoring activity using such standardised metrics means rather than recording more specific information, such as specific actions by each user, leads to less privacy concerns, since users are typically hesitant to use tools that record their actions in a way that they can track them back to them, unless this is absolutely necessary.

Since these metrics are commonly used across the industry, there are several available tools that help measure them, generate reports and visualisations and provide notifications to the website / platform administrator(s). The most prominent alternatives are presented in the table below:

Name	Description	Pricing
Google Analytics	Google Analytics is the de-facto industry standard in website usage analytics, with real-time analytics and easy integration.	Free
Mixpanel	Mixpanel provides advanced usage analytics tools such as customer segmentation and conversion prediction mechanisms.	Free (limited), 99 USD / month
Woopra	Woopra goes beyond usage analytics, as it creates activity profiles for every user.	Free (up to 30K requests / month)

Adobe Analytics	Adobe Analytics is a business-oriented analytics solution that is used by industry leaders such as Microsoft and Facebook.	29 USD / month + (as part of Adobe Cloud)
HeapAnalytics	HeapAnalytics provides advanced features such as direct SQL access to analytics data.	Free (up to 5K sessions / month)

Table 4-6: Website Usage Analytics Solutions

Between the various products that provide Website Usage Analytics, Google Analytics is the most compelling option, as it provides customisable analytics, the ability to set and measure custom goals, real-time monitoring, very easy to installation and excellent integration with other Google products, such as Google Ads.

4.2.2 Data Usage Analytics

Acquiring, exposing and promoting high quality, complete and up-to-date datasets is of vital importance for the BigDataOcean platform, as its success greatly depends on the quality and coverage of the provided data.

Contrary to platform analytics, out-of-the box solutions for data usage analytics cannot be easily developed or exploited, since the definition of what a dataset is and when it's used is non-trivial, usually depending on the application. In traditional dataset repositories, website analytics tools may be sufficient for this purpose, since dataset access is provided through some dedicated interface, through which end users can preview, review and get access to the dataset. By measuring page views on these resources, meaningful insights may be extracted regarding dataset usage and popularity.

However, in the case of applications that provide services on top of the available data, such as BigDataOcean, these high-level statistics may be misleading. For instance, an oceanographic database in BigDataOcean might be rarely searched for, explored or rated, and thus its apparent usage would seem to be quite low. This same database, however, could be used in many analyses as additional input, or information from that database could assist the generation of popular visualisations and reports. This effectively means that some lower-level metrics must be defined and measured by the platform itself. The most indicative metrics for data usage analytics include:

Metric	Definition	Provided by
Dataset page views	How many times a dataset preview page was viewed	Platform analytics tools
Dataset unique views	The number of unique users who viewed the dataset preview page	Platform analytics tools
Analyses data	Data volume that was used for all analyses performed from a particular dataset	Analytics
Queried data	Data volume that was returned by queries from a particular dataset	Query Builder
Visualised data	Data volume that was used to generate visualisations from a particular dataset	Visualisations
Average rating	The average of all ratings for a particular dataset	Dataset metadata
Number of comments	The number of comments in total under a particular dataset	Dataset metadata
Number of related requests	Number of times a dataset request was created, based on this dataset	On Demand Data and Services

Table 4-7: Data Usage Analytics Metrics

The fact that some of this metrics are provided by BigDataOcean tools instead of a third part solution means that there is some more effort for the tools' developers, but the benefit of advanced analytics on data usage in the platform, both in direct as well as in indirect scenarios, is of high value and cannot be ignored. The aggregated data usage metrics could be analysed themselves, in order to detect patterns regarding e.g. the datasets or dataset types that are typically combined together or used by the same users, using existing algorithms such as decision trees. The result of this process is that the platform could either predict or provide suggestions on what data an individual client might use in the future, based on their history or the history of others.

4.2.3 Service Usage Analytics

Service Usage Analytics will help the platform's administrators understand several aspects of how its different services are used by its clients, including:

1. Measure which services are more relevant to their needs and more popular.
2. Find out in which ways they are trying to use these services (different configurations, parts of each services that are most used).
3. Understand workflows that span across multiple services and BigDataOcean tools.
4. Potentially understand problems the clients have with using this services in order to improve them in the future.

As in the case of Data Usage analytics, since the provided services depend on the structure of the BigDataOcean platform, there are no external tools that can easily be used to extract metrics that will meaningfully answer the points mentioned above. These metrics have to be defined having the BigDataOcean services and toolset in mind, in order to be able to extract meaningful results.

The metrics that will be used in order to extract usage analytics for the BigDataOcean services are mentioned in the table below. A similar analysis of these metrics as the one mentioned in 4.2.2 could take place in order to enable the system to detect usage patterns and provide recommendations and predictions.

Metric	Definition	Provided by
Service uses	The number of times each particular service was used	Platform analytics tools, Service metadata
Service unique users	The number of unique users that consumed a particular service	Platform analytics tools, Service metadata
Query complexity	The complexity of the queries that have been defined using the Query Builder (scale of 1-5)	Query Builder
Average #datasets combined in queries	Average number of datasets combined in a query that was generated using the Query Builder	Query Builder
Number of algorithm usages	Number of times an algorithm was used to define an analysis	Analytics
Algorithm configuration statistics	Statistics measuring what configurations were used for analyses that were based on the same algorithm	Analytics

Number of visualisation type uses	Number of times a visualisation type was selected to create a visualisation	Visualisations
#Avg. embedded queries	Average number of queries embedded in a report	Report Builder
#Avg. embedded analyses	Average number of analyses results embedded in a report	Report Builder

5 Conclusions

The scope of D3.2 was to document the preliminary efforts undertaken within the context of Tasks T3.2 – Multi-Source Big Data Harmonisation and Processing Patterns for Maritime Applications, and T3.3 – Knowledge Extraction, Business Intelligence and Usage Analytics Algorithms. Towards this end, the scope of the current deliverable was threefold: Firstly, it aimed at defining and documenting the services which will provide the means for collecting and harmonising multi source big maritime data. In this context, the maritime information sources identified and documented in deliverable D2.1, as well as the big data semantic vocabularies analysed and documented in deliverable D3.1, were analysed to design the services which will allow stakeholders to reference and use metadata shared by multiple sources and data providers with big maritime data. Secondly, the services were defined that will address the maritime data stakeholders needs, as identified and documented in deliverable D2.1, and will provide the means for processing multi source big maritime data as per the processing patterns defined by the project pilot partners. Thirdly, the list of algorithms that will facilitate the proper execution flow of the four project pilots at first, and of future services as the project evolves was defined taking into consideration again the maritime data stakeholders needs identified and documented in deliverable D2.1, the user stories collected and documented in deliverable D4.1, as well as the architectural decisions documented in deliverable D4.2.

The delivery of the BigDataOcean service related to harmonisation, knowledge extraction, business intelligence and usage analytics is a living process that will last until M19. The second and final version of the current deliverable, namely D3.3, will integrate additional identified functional requirements, mainly originating from the BigDataOcean pilots that will be translated to updates, customisations or modification of the services accordingly.

References

[1] B. Santos, K. Lunday, Maritime Domain Awareness-International involvment to promote maritime security and safety, Proc. Mar. Saf. Secur. Counc. Coast Guard J. Saf. Secur. Sea. (2009).

[2] e-Navigation Strategic Action Plan, US Committee on the Marine Transportation System, 1200 New Jersey Avenue, SE, 2012.

[3] L. Millefiori, D. Zissis, L. Cazzanti, G. Arcieri, Computational Maritime Situational Awareness Techniques for Unsupervised Port Area, NATO UNCLASSIFIED REPORTS, SCIENCE AND TECHNOLOGY ORGANISATION CENTRE FOR MARITIME RESEARCH AND EXPERIMENTATION, La Spezia, Italy, 2016.

[4] J. Roy, M. Davenport, Categorisation of Maritime Anomalies for Notification and Alerting Purpose, in: NATO Work. Data Fusion Anom. Detect. Marit. Situational Aware., 2009.

[5] T. Bedford, R.M. Cooke, Probabilistic risk analysis : foundations and methods, Cambridge University Press, 2001.

[6] J.R.W. Merrick, J.R. Van Dorp, Speaking the Truth in Maritime Risk Assessment, (n.d.).

[7] J.-F. Balmat, F. Lafont, R. Maifret, N. Pessel, A decision-making system to maritime risk assessment, Ocean Eng. 38 (2011) 171–176. doi:10.1016/j.oceaneng.2010.10.012.

[8] A.G. Eleye-Datubo, A. Wall, J. Wang, Marine and Offshore Safety Assessment by Incorporative Risk Modeling in a Fuzzy-Bayesian Network of an Induced Mass Assignment Paradigm, Risk Anal. 28 (2008) 95–112.doi:10.1111/j.1539-6924.2008.01004.x.

[9] Zaili Yang, S. Bonsall, Jin Wang, Fuzzy Rule-Based Bayesian Reasoning Approach for Prioritization of Failures in FMEA, IEEE Trans. Reliab. 57 (2008) 517–528. doi:10.1109/TR.2008.928208.

[10] G. Pallotta, M. Vespe, K. Bryan, Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction, Entropy. 15 (2013) 2218–2245. doi:10.3390/e15062218.

[11] B. Ristic, B. La Scala, M. Morelande, N. Gordon, Statistical analysis of motion patterns in AIS Data: Anomaly detection and motion prediction, (2008) 1–7.

[12] S. Ricci, C. Marinacci, L. Rizzetto, The Modelling Support to Maritime Terminals Sea operation: The Case Study of Post Messina, 2014.

[13] J. Wade, R. Cloutier, B. Huijbrechts, M. Velikova, S. Michels, R. Scheepens, Metis1: An Integrated Reference Architecture for Addressing Uncertainty in Decision-support Systems, Procedia Comput. Sci. (2015) 476–485.

[14] S. Ray, A. Brown, N. Koudas, R. Blanco, A. Goel, Parallel in-memory trajectory-based spatiotemporal topological join, in: n.d.: pp. 361–370.

[15] A. Dahlbom, L. Niklasson, Trajectory clustering for coastal surveillance, (2007) 1–8. doi:10.1109/ICIF.2007.4408114.

[16] S.C. David Lindsay, Effective Probability Forecasting for Time Series Data Using Standard Machine Learning Techniques, in: S. Singh, M. Singh, C. Apte, P. Perner (Eds.), Pattern Recognit. Data Min., Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. doi:10.1007/11551188.

[17] G.K.D. de Vries, M. van Someren, Machine learning for vessel trajectories using compression, alignments and domain knowledge, Expert Syst. Appl. 39 (2012) 13426–13439.

[18] K. Kowalska, L. Peel, Maritime anomaly detection using Gaussian Process active learning, (n.d.) 1164–1171.

[19] R. Laxhammar, G. Falkman, E. Sviestins, Anomaly detection in sea traffic - A comparison of the Gaussian Mixture Model and the Kernel Density Estimator, (n.d.) 756–763.

[20] F. Johansson, G. Falkman, Detection of vessel anomalies - a Bayesian network approach, in: 2007 3rd Int. Conf. Intell. Sensors, Sens. Networks Inf., IEEE, 2007: pp. 395–400. doi:10.1109/ISSNIP.2007.4496876.

[21] A. Nicholson, F. Cozman, S. Mascaro, A.E. Nicholso, K.B. Korb, Anomaly detection in vessel tracks using Bayesian networks, Int. J. Approx. Reason. 55 (2014) 84–98.

[22] R.O. Lane, D.A. Nevell, S.D. Hayward, T.W. Beaney, Maritime anomaly detection and threat assessment, (2010) 1–8.

[23] F. Fooladvandi, C. Brax, P. Gustavsson, M. Fredin, Signature-based activity detection based on Bayesian networks acquired from expert knowledge, (2009) 436–443.

[24] N. Bomberger, B. Rhodes, M. Seibert, A. Waxman, Associative Learning of Vessel Motion Patterns for Maritime Situation Awareness, in: 2006 9th Int. Conf. Inf. Fusion, IEEE, 2006: pp. 1–8. doi:10.1109/ICIF.2006.301661.

[25] B.J. Rhodes, N.A. Bomberger, M. Zandipour, Probabilistic associative learning of vessel motion patterns at multiple spatial scales for maritime situation awareness, in: 2007 10th Int. Conf. Inf. Fusion, IEEE, 2007: pp.1–8. doi:10.1109/ICIF.2007.4408127.

[26] S.-B.C. Sang-Jun Han, Kyung-Joong Kim, Evolutionary Learning Program's Behavior in Neural Networks for Anomaly Detection, in: N.R. Pal, N. Kasabov, R.K. Mudi, S. Pal, S.K. Parui (Eds.), Neural Inf. Process., Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. doi:10.1007/b103766.

[27] J.B. Kraiman, S.L. Arouh, M.L. Webb, Automated anomaly detection processor, in: A.F. Sisti, D.A. Trevisani (Eds.), Proc. SPIE 4716, Enabling Technol. Simul. Sci. VI, 128, International Society for Optics and Photonics, 2002: pp. 128–137. doi:10.1117/12.474940.

[28] M. Seibert, B.J. Rhodes, N.A. Bomberger, P.O. Beane, J.J. Sroka, W. Kogel, et al., SeeCoast port surveillance, in: M.J. DeWeert, T.T. Saito, H.L. Guthmuller (Eds.), 2006: p. 62040B. doi:10.1117/12.666980.

[29] M. Morel, J.-P. George, A. Littaye, F. Jangal, A. Napoli, M.-A. Giraud, et al., SCANMARIS Project – Detection of Abnormal Vessel Behaviours, in: NATO Work. Data Fusion Anom. Detect. Marit. Situational Aware. (NATO MSA 2009), La Spezia, Italy, 2009.

[30] C. Griffin, Learning and Prediction for Enhanced Readiness: An ONR Office 31 Program, in: Present. to TTCP MAR AG-8, 2009.

[31] M. Géhant, M., Roy, V., Marmorat, J.-P., and Bordier, A Behaviour Analysis Prototype for Application to Maritime Security, in: NATO Work. Data Fusion Anom. Detect. Marit. Situational Aware. (NATO MSA 2009), La Spezia, Italy, 2009.

[32] DARPA, Fast Connectivity for Coalitions and Agents Project, Fact Sheet, 2005.

[33] J. Tozicka, M. Rovatsos, M. Pechoucek, S. Urban, MALEF: Framework for distributed machine learning and data mining, Int. J. Intell. Inf. Database Syst. 2 (2008) 6. doi:10.1504/IJIIDS.2008.017242.

[34] European Parliament and of the European Council, DIRECTIVE 2009/16/EC, 2009.

[35] L.A. Zadeh, Fuzzy sets, Inf. Control. 8 (1965) 338–353. doi:10.1016/S0019-9958(65)90241-X.

[36] L.A. Zadeh, Making computers think like people, IEEE Spectr. 21 (1984) 26–32. doi:10.1109/MSPEC.1984.6370431.

[37] K. Chatzikokolakis, P. Spapis, A. Kaloxylos, G. Beinas, N. Alonistioti, Spectrum sharing: A coordination framework enabled by fuzzy logic, in: 2015 Int. Conf. Comput. Inf. Telecommun. Syst., IEEE, 2015: pp. 1–5. doi:10.1109/CITS.2015.7297761.

[38] Elliot, A. (1986) Shear diffusion and the spread of oil in the surface layers of the North Sea, Dt. Hydrogr. Z. 39, 113-137

[39] Johansen, O. (1985) Particle in Fluid Model for Simulation of Oil Drift and Spread, Oceanographic Center, SINTEF Group, Trondheim, Norway

[40] Rasmussen, D. (1985) Oil spill modelling a tool for cleanup operations, Oil Spill Conference 243-249

[41] Mellor, G. (1991) User 's Guide for a Three-Dimensional, Primitive Equation, Numerical Ocean Model, Princeton University, Princeton

[42] Mellor, G. L. and Yamada, T. (1982) Development of a turbulence closure model for geophysical fluid problems, Rev. Geophys. Space Phys., 20, 851–875

[43] Stiver, W., Shiu, W., and Mackay, D. (1989) Evaporation times and rates of specific hydrocarbons in oil spills, Environ. Sci. Technol., 23, 101–105

[44] Gundlach, E. R. (1987): Oil holding capacities and removal coefficients for different shoreline types to compute simulate spills in coastal waters, Proc. Oil Spill Conf., 451–457

Annex I: Mare Protection Details

Environmental Input files

HYDRODYNAMIC DATA

VARIABLES: velocity zonal component, velocity meridional component, temperature, and salinity (u,v,T and S).

NUMBER of LEVELS: 15 levels from surface to the bottom, keeping the higher vertical resolution close to the surface. The levels are 0, 5, 10, 15, 20, 30, 50, 80, 150, 300, 650, 1000, 1500, 2500, 3500.

DEPHTS: z-layers.

TEMPORAL RESOLUTION: 1-h resolution (forecast) and 6-h resolution (analysis/historical data)

FORMAT:

- NetCDF CF 1.0 or higher
- No staggered GRID (horizontal velocity and temperature on the same grid)
- All variables in one file per day
- 24-time step in one file
- Time origin 00:00
- Average hourly fields (for example from 8:00 to 9:00, centred at 8:30)
- Variables standard name:

Variable	Long name	Standard name	units
uvel	Velocity Zonal Component	sea_water_x_velocity	m s-1
vvel	Velocity Meridional Component	sea_water_y_velocity	m s-1
potemp	Potential Temperature	sea_water_potential_temperature	degrees C
psal	Practical Salinity	sea_water_salinity	psu

NAMING CONVENTION:

{valid date}_{freq flag}{average flag}-{producer}-{parameter}-{config}-{region}-{bul date}_{product type}-fv{file version}.nc

where

· **valid date** YYYYMMDD is the validity day of the data in the file

· **freq flag** is the frequency of data values in the file (h = hourly, 6h = 6 hourly, d =daily)

· **average flag** is i=instantaneous, m=mean (usually daily)

· **producer** is a short version of the production unit e.g. INGV, HCMR, IFREMER, etc.

· **parameter** is a four-letter code for the parameter or parameter set from Standard BODC (this fields will be empty in the case all the variables are in one file)

· **config** identifies the producing system and configuration e.g. mfs_sys4b.

· **region** is a three-letter code for the region

· **bul date** bYYYYMMDD is the bulletin date the product was produced

· **product type** is a four letters code for the product type and forecast horizon (this is a useful information when one handles with sets of different forecasts bulletin for the same date), e.g. fc01 for the first day of forecast or an01 for the first day of analysis.

· **file version** is xx.yy where xx is the version (00, 01 or 02) and yy is an incremental version number

Example:

20121127_hi-HCMR-OCEAN-POSEIDON-MED-b20121126_FC01-fv02.00.nc

BATHYMETRY: A separate file containing the model bathymetry is provided as the vertical distance below the surface of zero depth

FORMAT:

- NetCDF CF 1.0 or higher
- Positive values downwards
- standard name:

Variable	Long name	Standard Name	Units
depth	Depth below Sea level	Sea_floor_depth_below_sea_level	m

NAMING CONVENTION:

{producer}-{config}-{region}-BATH_fv{file version}.nc

where

· **producer** is a short version of the production unit e.g. INGV, HCMR, IFREMER, etc.

· **config** identifies the producing system and configuration e.g. mfs_sys4b.

· **region** is a three-letter code for the region

· **file version** is xx.y where xx is the version (00, 01 or 02) and yy is an incremental version number

Example:

HCMR-POSEIDON-AEG-BATH_fv02.00.nc

METEOROLOGICAL DATA:

VARIABLES: wind velocity components at 10 m, air temperature (2m) and atmospheric pressure at sea level.

TEMPORAL RESOLUTION: according to the meteorological data providers (no harmonisation is foreseen)

FORMAT:

- NetCDF CF 1.0 or higher

- No staggered GRID (horizontal velocity and temperature on the same grid)
- All variables in one file
- Time origin 00:00
- Average or instantaneous fields according to the meteorological data providers (no harmonisation is foreseen)
- Variables standard name:

Variable	Long name	Standard Name	Units
x_wind10	Wind Speed along East-West Direction at 10 meters above Sea Level	x_wind	m s-1
Y_wind10	Wind Speed along North-South Direction at 10 meters above Sea Level	y_wind	m s-1
tair2m	Air Temperature at 2 meters height above Sea Level	air_temperature	degrees C
pmsl	Air Pressure at Sea Level	air_pressure_at_sea_level	hPa

NAMING CONVENTION: Same naming convention of the hydrodynamic data

WAVES DATA:

VARIABLES: wave height, the wave direction and the wave period.

TEMPORAL RESOLUTION: 1-h resolution (forecast) and 6-h resolution (analysis)

FORMAT:

- NetCDF CF 1.0 or higher
- No staggered GRID (horizontal velocity and temperature on the same grid)
- All variables in one file
- 24 time step in one file
- Time origin 00:00
- Average/Instantaneous hourly fields
- Variables standard name:

Variable	Long name	Standard Name	Units
wsh	Sea Surface Wave Significant Height	sea_surface_wave_significant_height	m
wper	Mean Wave Period	sea_surface_wave_zero_upcrossing_period	s
wdir	Sea Surface Wave Direction	sea_surface_wave_to_direction	degrees

		"to_direction" indicates the direction towards which the velocity vector is headed.	

NAMING CONVENTION: Same naming convention of the hydrodynamic data

Oil spill Input and Output files:

Naming convention:

xxxxyymmdd_hhmmZZ.inp

xxxxyymmdd_hhmmZZ.out

where

xxxx is name of the simulation chosen by the user;

yymmdd is the date of the spill;

hhmm the time of the spill;

ZZ is _F for a forecast and _H for a hindcast.

INPUT FILE STRUCTURE:

The input file structure covers 7 different cases:

- Point source - single spill;
- Point sources - multiple spill;
- Areal source (polygon) - single spill;
- Areal sources (polygons) - multiple spills;
- Areal (polygons) and point sources - multiple spills;
- Areal source (satellite file) – single spill;
- Areal source (satellite file) – multiple spill;

If the latitude/longitude field is substituted with 999, then the oil slick is described by polygon which will be given at the end of the file under the corresponding serial spill number (SPILL_NUM**);**

If the latitude/longitude field is substituted with 888, then the oil slick data is given by satellite file: the file name will be given at the end of the file.

If N_SPILL is higher than 1, then the position, date, time, duration and volume of the N_SPILL oil spills for each oilspill under the corresponding serial spill number (SPILL_NUM**);**

N_SPILL : Number of spills (point or areal source)

SPILL_NUM : Oilspill serial number

LAT LON : Position of the oil slick: latitude (positive for North, negative for South) and longitude (positive for East, negative for West) in degrees (decimal number)

DEPTH : Depth of the oil spill: in m (it is 0 in the case of a surface slick)

YYYY MM DD hhmm : Date and Time start

DURATION : Duration of the spill release in hours

VOLUME : Total amount (volume) of spilled oil in m3

The last part from SPILL_NUM to VOLUME is repeated for each oil spill (N_SPILL times).

XXXX : user selected name of the simulation

DENSITY/API/TYPE : Density of oil (kg/m3) or API number or Type of oil

SIM_TYPE :Type of the requested simulation (FORWARD or BACKWARD)

SIM_LENGTH : Length of the requested simulation in hours

STEP : requested time interval between 2 outputs in hours (decimal number)

GRD_SIZE : Grid size for concentration output reconstruction (m)

OCEAN_MODEL : Ocean forcing requested (three-digit number, see table 1)

WIND_MODEL : Atmospheric forcing requested (three-digit number, see table 2)

WAVE_MODEL : Wave forcing requested (three-digit number, see table 3)

If the latitude/longitude field of an oil spill starts with 999, then the oil slick is described by several points (polygon) in the following lines, under the corresponding serial spill number (SPILL_NUM)

SPILL_NUM : Oilspill serial number (number of oilspill that this polygon refers to)

N : number of points of the polygon that will be given

LAT_P LON_P : first point of the polygon = latitude (positive for North, negative for South) and longitude (positive for East, negative for West) in degrees (decimal number)

LAT_P LON_P : second point of the polygon

LAT_P LON_P : third point of the polygon

If the Lat value of an oil spill starts with 888, then the last line is the name of the satellite file containing the slick data:

name_satellite_file.gml

	Hydrodynamic data	**ID**
HCMR	POSEIDON High resolution Aegean Model	001
HCMR	POSEIDON Mediterranean Model	002

Table 1. Three-digit identification number for hydrodynamic data.

	Meteorological data	**ID**
HCMR	POSEIDON ETA weather forecasting system	101

Table 2. Three-digit identification number for meteorological data.

	Waves data	ID
HCMR	POSEIDON WAM Cycle 4 for the Mediterranean	201
HCMR	POSEIDON WAM Cycle 4 for the Aegean	202

Table 3. Three-digit identification number for waves data.

OUTPUT FILE STRUCTURE:

The predicted output variables of the model contain the position of each particle in the sea (longitude, latitude and depth), the evaporated volume of the initial oil, the emulsified volume, the volume remain on the beach and the oil volume reached the sea floor. The output is derived in an ASCII file format.

The output file contains for each time step the volume of oil on surface, subsurface, on coast and at the bottom.

The file starts with the first 16 lines of the input file ((except for the case of multiple oil spill, in this case the entire input file will be reported), ended by a line containing:

'*** End of Input ***'

At each time step the output contains:

First the coordinates of the center of mass of each spill:

A header is needed, followed by Spill_Num number of lines containing the data:

Spill_num GC_lat GC_lon

"**Spill_Num**"=Oilspill serial number

"**GC_lat**"= latitude of the slick center of mass

"**GC_lon**"= longitude of the slick center of mass

Then a line of global data that refers to the whole spill:

A header is needed, followed by a line containing the data:

Time (hrs), N, %ev, %srf, % em, %disp, % cst, %btm, max_visc, min_visc, dens

"**Time**" = number of hours (in decimal number) after the start time (start time given in the line 2 above)

"N" = number of data point (grid points used to reconstruct the concentration)

"**% ev**" = percentage of evaporated oil (optional)

"**% srf**" = percentage of oil on surface (optional)

"**% em**" = percentage of emulsified oil (optional)

"**% disp**" = percentage of dispersed oil (optional)

"**% cst**" = percentage of oil on coast (optional)

″**% btm**″ = percentage of oil at the bottom (optional)

″**max_visc**″ = updated maximum viscosity of oil

"min_visc"= updated maximum viscosity of oil

″**dens**″ = updated density of oil expressed as kg/m3

The output data may consist of the original parcels in some models or of aggregated parcels in others.

Then another header is necessary, followed by a list of the data for all the parcels or aggregates. Volumes here include the effects of evaporation and emulsification.

'Lat Lon Dpth Status Volume Dens Visc'

"Lat Lon " = latitude and longitude of parcel or aggregated parcels (in decimal degrees)

″**Dpth**″ = depth (position on the vertical, positive value) of the particle in meter

″Status″ = the concentration of oil can be at: 0 (at the bottom, if available), 1 (in the water column, if available), 5 (on sea surface), 10 (on the coast)

"Volume"= oil volume of parcel or aggregated parcels expressed as m3

"Dens" (optional)= density of parcel or aggregated parcels expressed as kg/m3

"Visc"(optional)= viscosity of parcel or aggregated parcels

If the optional field are not provided they are filled with -99 (not available data).