electronics

MDPI

Article

Design of Vessel Data Lakehouse with Big Data and AI Analysis Technology for Vessel Monitoring System

Sun Park [1,*], Chan-Su Yang [2,*] and JongWon Kim [1]

[1] AI Graduate School, Gwangju Institute of Science and Technology, Gwangju 61005, Republic of Korea
[2] Marine Security and Safety Research Center, Korea Institute of Ocean Science & Technology, Busan 49111, Republic of Korea
* Correspondence: sunpark@gist.ac.kr (S.P.); yangcs@kiost.ac.kr (C.-S.Y.)

Abstract: The amount of data in the maritime domain is rapidly increasing due to the increase in devices that can collect marine information, such as sensors, buoys, ships, and satellites. Maritime data is growing at an unprecedented rate, with terabytes of marine data being collected every month and petabytes of data already being made public. Heterogeneous marine data collected through various devices can be used in various fields such as environmental protection, defect prediction, transportation route optimization, and energy efficiency. However, it is difficult to manage vessel related data due to high heterogeneity of such marine big data. Additionally, due to the high heterogeneity of these data sources and some of the challenges associated with big data, such applications are still underdeveloped and fragmented. In this paper, we propose the Vessel Data Lakehouse architecture consisting of the Vessel Data Lake layer that can handle marine big data, the Vessel Data Warehouse layer that supports marine big data processing and AI, and the Vessel Application Services layer that supports marine application services. Our proposed a Vessel Data Lakehouse that can efficiently manage heterogeneous vessel related data. It can be integrated and managed at low cost by structuring various types of heterogeneous data using an open source-based big data framework. In addition, various types of vessel big data stored in the Data Lakehouse can be directly utilized in various types of vessel analysis services. In this paper, we present an actual use case of a vessel analysis service in a Vessel Data Lakehouse by using AIS data in Busan area.

Keywords: vessel monitoring system; data lakehouse; big data; AI analysis

check for updates

Citation: Park, S.; Yang, C.-S.; Kim, J. Design of Vessel Data Lakehouse with Big Data and AI Analysis Technology for Vessel Monitoring System. *Electronics* **2023**, *12*, 1943. https://doi.org/10.3390/electronics12081943

Academic Editor: Ping-Feng Pai

Received: 12 January 2023
Revised: 10 April 2023
Accepted: 10 April 2023
Published: 20 April 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Big data is an enormous amount of data that is difficult to collect, store, analyze, and process using legacy application software. Big data technology is showing efficiency by processing big data into a form that users can understand and utilize. The concept of a data lake has emerged to efficiently store, process, and protect big data. Data lakes have the advantage of being cheaper than legacy databases. Data lakes provide a view of raw data that can be used by analytics technologies independent of traditional data storage or systems of record. However, data lakes require ongoing maintenance and a plan for how data is used and accessed. Without ongoing data lake maintenance, data management is difficult and expensive. There is also a risk of inaccessible junk data. This inaccessible data lake is called a data swamp. To solve this problem, the concept of a data lakehouse has emerged. A data lakehouse is the implementation of data structures and data management functions similar to a data warehouse on the low-cost storage used in a data lake [1].

Data Lakehouse is a new method to analytics structure that aims to combine traditional Data Lakes and Data Warehouses to serve different analytics needs. Data Lakehouse allows structured queries and enhanced analytics to run on best structured data for a given purpose while hiding the system complexity to users. Data Lakehouse alleviates common issues with Data Warehouse and Data Lake while allowing you to use the benefits of both

structures. This architecture can use structured, semi-structured and unstructured data which supports streaming workloads, machine learning and business intelligence. Data Lakehouse can be utilized as a foundation for constructing entirely novel systems and fusing the Data Lakes and the Data Warehouses by using platforms and frameworks [2].

A VMS (Vessel Monitoring System) is a generic word describing systems utilized in commercial fishing to help fisheries regulators track and monitor the fishing activity of ships. It is a core role of MCS (Monitoring Control and Surveillance) at national levels. VMS systems are utilized to enhance the operation and sustainability of the vessel environment by ensuring good fishing habits and preventing illegal fishing to protect and improve the fishermen's livelihood [3].

72% of the planet is covered by seas, oceans and other marine regions, 95% of which is hardly explored. The marine area is one of the most used economic areas by mankind. In other words, the marine area carries out economic activities through fishing, tourism, transportation, and logistics, and is used as a renewable energy resource such as wind and tidal power. For this reason, the maritime industry is an important and strategic industry and is continuously growing. Additionally, the marine domain has recently begun to provide a large, diverse and heterogeneous data. Marine domain data is growing at a rapid rate. Marine terabytes data is collected every month, and marine petabytes data is used in public already. Big data from heterogeneous sources such as satellites, buoys, ship, and sensors can be used as material for applications for environmental protection, security, error prediction, transportation route optimization and energy production. However, marine big data has a problem of high heterogeneity of data sources, which the marine solutions are still fragmented and underdeveloped [4,5].

In this paper, we propose a Vessel Data Lakehouse with Big Data and AI analysis technology for Vessel Monitoring System that can efficiently manage various heterogeneous vessel related data. The advantages of the proposed method in this paper are as follows. First, by using the Vessel Data Lake, it is possible to store vessel related big data in various formats at low cost. By placing the Vessel Data Lakehouse layer on top of the Data Lake, heterogeneous vessel related big data can be managed and controlled. Various types of vessel related big data stored in the Data Lake can be directly used for various types of analysis services.

The remainder of the paper is organized as follows. Section 2 describes the related studies on Data Lakehouse and VMS with vessel applications, and Section 3 presents the proposed the Vessel Data Lakehouse for Vessel Monitoring System. Then Section 4 shows the experiment results, and Section 5 concludes the paper.

2. Related Work

2.1. Data Lakehouse

Orescanin and Hlupic proposed a high-level Data Lakehouse structure consisting of an extraction layer, a Data Lake layer, and a Data Warehouse layer. The extraction layer uses data integration (ETL) tools or streaming tools such as MQ (Message Queue) frameworks (i.e., Kafka and Spark) to ingest data from data sources into the Data Lakehouse. The Data Lake layer is the takeoff/phase area, which is a temporary place where source data is prepared for the next processing. In this area, data is stored until a certain number of batch loads or time segments have passed, after which it is deleted. In the Data Warehouse part, three layers are defined: the manual entry layer, the base layer, and the performance and analytics layer. The manual entry layer maintains attributes and dimensions that do not exist in the resource system. The base layer is the part of the Data Warehouse that has data loaded and transformed directly from resource systems. The performance and analytics layer hold additional tables for analytics purposes based on the data aggregated from the base layer [2]. Armbrust et al. argue replacing from Data Warehouse to Lakehouse, since Lakehouse is based on open direct-access data formats and supports machine learning and data science. They mentioned the Lakehouse's advantage is a combination of key strengths of Data Lakes and Data Warehouses, which low-cost storage in an open format that can

be accessed from a variety of systems in the former, with the strong management and optimization capabilities of the latter. They also propose a Lakehouse implementation that holds system store data in a low-cost object store using Apache Parquet and Delta Lake [6]. Begoli et al. proposed the concept of a Data Lakehouse in the domain of biomedical research and health data analysis, which it was implemented using Apache Spark for data processing and Delta Lake for data lake management. They also proposed a Lakehouse data intake scenario to process the heterogeneous and complex structure of the Lakehouse and the data in it. They noted that many projects using Lakehouse require mature, empirical researches and specific implementations [7]. The authors of this paper proposed the concept of Marine Data Lakehouse Architecture for managing maritime analytics application as a previous study. We designed Marine Data Lakehouse Architecture to consist of Data Management & Governance layer and Maritime Analytics Services layer. The Data Management & Governance layer performs data storage management, data preprocessing, a collection of processes, roles, policies, standards, and metrics. The Maritime Analytics Services layer performs marine data analysis services and visualization services [8]. Harby and Zhlkernine presented a comparative review of existing data Warehouse, Data Lake and Data Lakehouse technologies, highlighting their strengths and weaknesses. They also propose the concept of the necessity and necessary functions of the Lakehouse architecture, which has recently attracted a lot of attention in the big data management research community [9]. Kumar and Li separated storage and computers by using the Databricks Lakehouse platform for ingestion, connecting to upstream databases and storing the data in AWS S3 buckets. They showed that because they created a Databricks cluster similar in size to their Redshift cluster and configured it to dynamically scale up/down over the duration of a query as needed, queries run—11× faster on Databricks, but at half the cost [10].

The Data Lakehouse is not only a new concept, but it is still a conceptual construct. For this reason, nowadays, these Data Lakehouse concepts are spreading in special domain areas [2,6,7,9,10]. In this paper, a Vessel Data Lakehouse architecture is designed by adopting the Data Lakehouse concept to maritime applications. In this paper, the detailed implementation of Vessel Data Lakehouse at the module level based on actual ocean observation data and open source for the marine domain, rather than Lakehouse function descriptions, is unveiled for the first time.

2.2. VMS and Vessel Applications

Hery et al. designed the website to predict tuna fishing location using Naive Bayes and SVM based on Indonesia sea VMS data. Their homepage consists of five sub-menus: Get Data, Data Processing, Data Visualization, Analysis, and Prediction Visualization [11]. Zhao et al. proposed a hybrid interpolation scheme of trawler fishing track to interpolate missing VMS samples using Cubic Hermite Spline (CHS) and Long-Short Term Memory (LSTM) for Satellite-based VMS Traces [12]. Ahmed et al. proposed a space-time track association algorithm based on marine vessel AIS data for tracking sea ships as the ship's location and movement observations [13]. Beek et al. combined VMS and VIIRS fishing vessel data and used the LLFI package in R to merge the data with the VMS data set to track and identify the anonymous fishing boats detected by VIIRS in Natuna Indonesia [14]. Huang et al. proposed an edge computing-based adaptive trajectory transmission policy (EC-ATT) framework for VMS to enhance communication efficiency. Each ship has edge computing intelligent nodes that collect, process and transmit data. The Edge Computing server is configured to improve collaborative computing between the Edge and the Cloud, which transmits data through the Beidou navigation satellite [15].

Li et al. proposed a framework to integrate fishing vessel data from AIS (Automatic Information System) and VBD (VIIRS boat detection) data for mapping and analyzing fishery strength in the northern South China Sea, which the regional features and rules of fishing strength in typical seasons (i.e., February, April, September, and November) in the northern South China Sea in 2018 were systematically analyzed [16]. Souza et al. proposed a method to recognize fishing activity from S-AIS data for three major fishing

gear types: trawl, longline and purse seine. A Hidden Markov Model (HMM) was created using the vessel speed as an observed variable for a trawler. They designed a data mining (DM) method for long-liners. They also implemented a multilayer filtering strategy based on vessel speed and operating time for purse seines [17]. Alba et al. proposed an AIS (Automatic Identification System) localization monitoring application using a mobile phone. They searched for a way to create and implement a monitoring application for an AIS using a mobile phone. It is a monitoring system to check the location and movement of a ship or a ship in the surrounding area in a mobile and ubiquitous networking environment [18]. Prasad et al. proposed a new route extraction algorithm that effectively characterizes global route behavior and captures seasonal trends by processing historical AIS data (running on tens of gigabytes of daily data from more than 100,000 ships at sea). The proposed method is performed by formulating a sigmoid based turning waypoint recognition that does not rely only on changes in the ship's course derived from AIS messages [19].

Evmides et al. proposed a new framework for collecting, processing, storing, and analyzing AIS data in real time and algorithms to perform it efficiently and scalable. The proposed framework has been operating in Cyprus for the past few years and has collected and processed approximately 1 billion AIS messages in the Eastern Mediterranean Sea [20]. Liu et al. propose to develop a big data-based multilevel computational framework for extracting the most popular shipping routes. It uses trajectory simplification and density clustering algorithms to generate maritime transportation networks. It also uses the KDE (Kernel Density Estimation) method to visualize transport route heat-maps related to traffic frequency in a specific area. The most popular shipping routes are extracted through a sliding window algorithm performed on the shipping route heat-map [21]. Huang et al. proposed a Fishing Vessels Relationships Discovery (FVRD) system that calculates the interaction time between fishing boats and uses it as a weight to create a relationship network. Their method utilizes a trajectory process model to interpolate trajectories, align time steps, and assess spatial proximity between ships to create companionships for fishing vessels. The model then combines the periods between ongoing partnerships to construct a relationship model over time windows of 1 day, 1 week, 2 weeks and 4 weeks. After creating the relationship model, FVRD calculates important metrics of the relationship model to reveal some important conclusions [22]. Xiao et al. discussed some obvious limitations of existing VTS systems and key design considerations for their conversion to active VTS systems for advanced decision support for surveillance managers and operators. Their framework was elaborated with regard to a layered bottom-up processing flow in data processing, knowledge bases, intelligence services, and HMI visualizers [23]. Tampakis et al. present the structure of the i4sea big data platform for coastal monitoring and fishing vessel activity analysis and demonstrate the operation of some use case pilot scenarios. Their platform uses a lambda architecture to facilitate access to both batch and stream processing in a hybrid method, which have a balance between latency, throughput and fault tolerance [24]. Lytra et al. proposed a scalable data management solution to analyze challenges and requirements related to big marine data applications for multi-segment marine applications that integrate data of different velocity, variety, and volume under an inter-linked, trusted, multilingual engine [4].

3. Vessel Data Lakehouse for Vessel Monitoring System

As shown in Figure 1, Vessel Data Lakehouse consists of Extraction and Ingestion layer, Vessel Data Lake layer, Vessel Data Warehouse Model, and Vessel Application Services. This paper focuses on the implementation of Vessel Data Lakehouse for VMS and the application example of the implemented system. The Extraction and Ingestion layer in Figure 1a extracts vessel-related data from data sources using a push or pull approach based on Message Queuing technology and stores it in the Vessel Data Lake layer of Figure 1b. The Vessel Big Data layer of the Vessel Data Warehouse Model in Figure 1c manages the data in the Vessel Data Lake layer with data loaded and transformed directly from the resource system, analyzes Vessel Big Data, and supports the data to enable AI analysis. The

Vessel AI layer supports AI analysis and prediction for vessel application services. Vessel Application Service in Figure 1d supports Visualization service, Big Data Analysis service, and AI Analysis service for vessels.

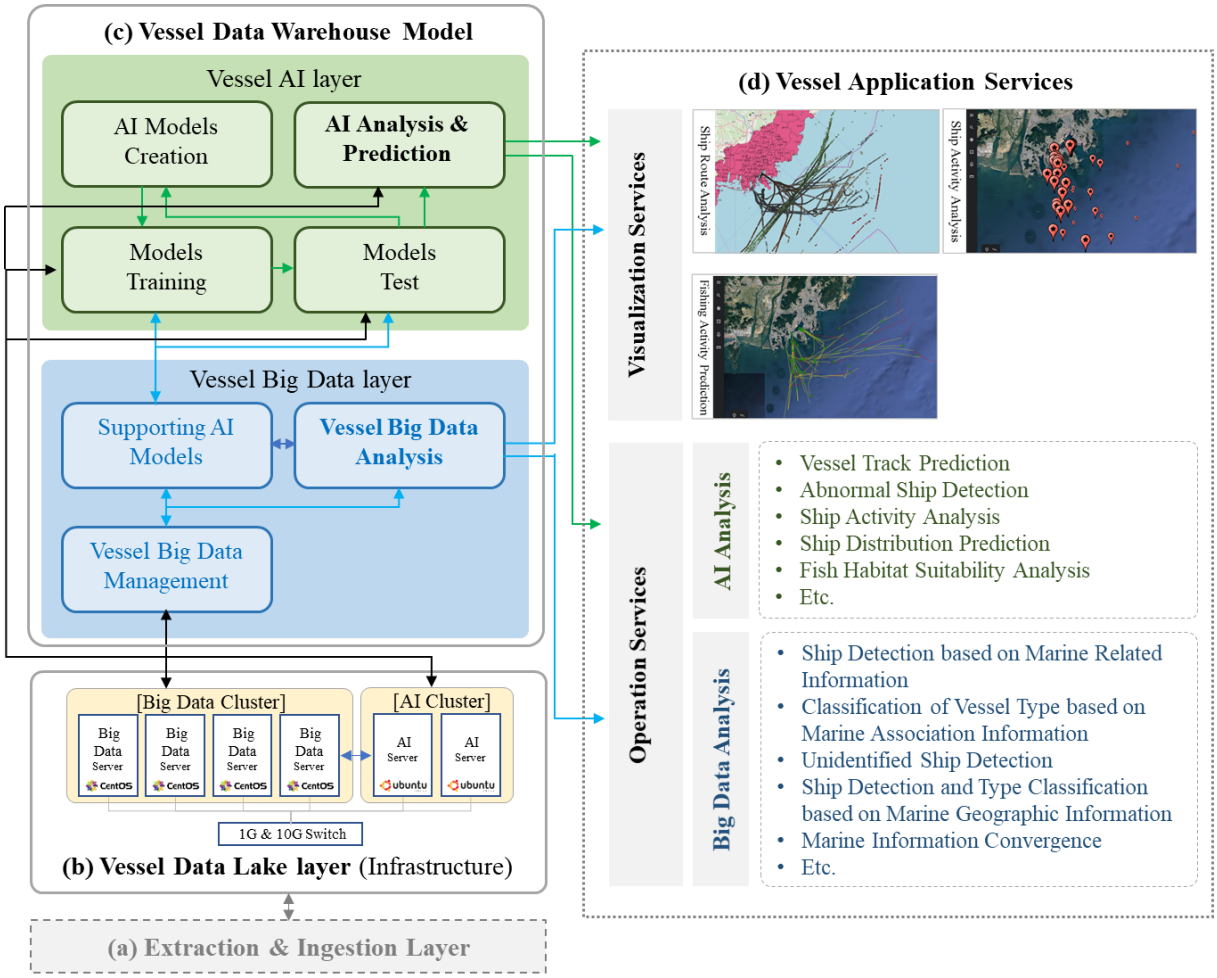


Figure 1. Conceptual Diagram of Vessel Data Lakehouse. (**a**) Extraction & Ingestion Layer; (**b**) Vessel Data Lake layer; (**c**) Vessel Data Warehouse Model; (**d**) Vessel Application Service.

3.1. Extraction and Ingestion Layer

In the Extraction and Ingestion layer, data is imported from an external source in the format specified by the Vessel Big Data Management module of the Vessel Big Data layer in Figure 1c and stored in the Vessel Data Lake layer in Figure 1b. In this paper, vessel-related data received from the Korea Institute of Ocean Science and Technology Maritime Safety Research Center are stored in the Vessel Data Lake by using the Extraction and Ingestion layer. The ETL (Extract Transform Load) function was implemented by using Python to load the vessel-related csv file to the Data Lake. Table 1 shows the size and number of rows for each type of vessel related data used in this paper.

Table 1. Kind of Vessel-related Data import to Marine Data Lake.

Type of Vessel-Related Data		Size	Number of Rows
AIS (Automatic Information System) [5]	(a) staticais	1.3 MB	13,856
	(b) daynamicais	31.0 GB	354,857,410
(c) V-Pass (Vessel-Pass) [25]		2.8 GB	35,053,969
(d) VBD (VIIRS boat detection) [16]		10.3 GB	9,709,207
Observation data	(e) Disaster Prevention Weather Observatory (hour) [26]: Aws_1hr_2019123123	49.5 KB	714
	(f) Disaster Prevention Weather Observatory (minute) [26]: Aws_min_201912310000	59.7 KB	714
	(g) tide station [27]: Khoa_79980120191216	100.3 KB	1441
	(h) synoptic weather station [26]: Khoa_busan20191216	12.3 KB	145
	(i) Main route marine observation buoy [27]: Khoa_ieodo20191216	172.3 KB	1441
	(j) Marine observation buoys in major sea areas [27]: Khoa_jjea20191216	6.1 KB	48
	(k) wave observation buoy [26]: Khoa_sf_0001200191216	96.3 KB	1441
	(l) sea fog observatory [27]: Kma_utc2019121622475	1.3 KB	24
	(m) marine science base [27]: Shk60_202001050002	15 KB	59
	(n) Marine weather observation buoy [26]: Vbko60_20200106110022194	75 Byte	1

3.2. Vessel Data Lake Layer

In this subsection, a Hadoop [28] cluster-based Vessel Data Lake was implemented to enable storage and processing of large volume vessel data in the Vessel Data Lake layer. Apache Hadoop is open source software for reliable and scalable distributed computing. The Apache Hadoop software library is a framework that allows distributed processing of large data sets across clusters of computers using a simple programming model. Figure 2 shows a conceptual diagram of the software stack of the Vessel Data Lake based on Hadoop cluster. Hadoop HDFS (Hadoop Distributed File System) is a file system that stores large files of tens of terabytes or petabytes or more in distributed servers and enables fast processing of the stored data. Hadoop YARN (Yet Another Resource Negotiator) manages numerous tasks in clusters composed of dozens or more nodes, which it manages distributed resources such as resources (i.e., CPU, RAM) to be used for specific tasks. Hadoop MapReduce is a data processing model designed to process large amounts of data in a distributed/parallel computing environment. When large data is received, it divides the data into blocks of a specific size and executes Map Task and Reduce Task for each block. The hardware cluster of Vessel Data Lake consists of Vessel Big Data cluster and Vessel AI cluster. The Vessel Big Data cluster is a Hadoop-based Data Lake hardware node which it consists of 1 master node and 3 slave nodes, and detailed specifications are shown in Table 2. The Vessel AI cluster supports big data analytics and AI analytics associated with GPUs. Each node has a 1G network card and a 10G network card. The 10G network card is used for internal data movement, and the 1G network card is used for external control of nodes.

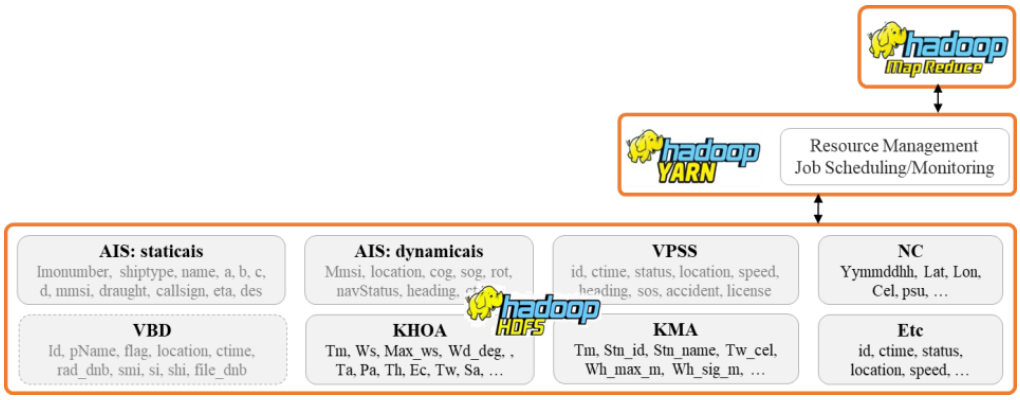


Figure 2. Conceptual Diagram of Hadoop-based Vessel Data Lake (Software stack).

Table 2. Hardware Specifications of Vessel Data Lake.

Cluster		CPU	RAM	SSD	NIC	GPU	OS
Vessel Big Data cluster	master node	Intel i9-7940x	64 GB	2 TB	1Gb/1Gb	-	CentOS 7.7
	slaver node 01	Intel i9-7940x	64 GB	2 TB	1Gb/1Gb	-	CentOS 7.7
	slaver node 02	Intel i9-7940x	64 GB	2 TB	1Gb/1Gb	-	CentOS 7.7
	slaver node 03	Intel i9-7940x	64 GB	2 TB	1Gb/1Gb	-	CentOS 7.7
Vessel AI cluster	ai node 01	Intel i9-7940x	64 GB	2 TB	1Gb/1Gb	2080 * 2	Ubuntu 18.04
	ai node 02	Intel i9-7940x	64 GB	2 TB	1Gb/1Gb	2080 * 2	Ubuntu 18.04

3.3. Vessel Data Warehouse Model

Vessel Data Warehouse Model consists of Vessel Big Data layer and Vessel AI layer. The Vessel Big Data layer transforms the source data and loads it into the Data Lake to enable vessel big data analysis or vessel AI analysis. Vessel AI layer supports basic models and analysis tools for vessel AI analysis.

3.3.1. Vessel Big Data Layer

The Vessel Big Data layer consists of Vessel Big Data Management module, Supporting AI Models module, and Vessel Big Data Analysis module. Vessel Big Data Management module performs original data collection, data purification/modeling, data imposing/data export, data insertion/deletion, and data saving/loading for big data analysis. It also transforms source data to enable direct big data processing. Supporting AI Models module performs data preprocessing, missing data processing, categorical data processing, and feature scaling for AI analysis. It also is possible to support data clearing, labeling, storage, convergence analysis for all types of vessel structured and unstructured data. Vessel Big Data Analysis as shown in Figure 1c provides Ship Detection based on Marine Related Information, Classification of Vessel Type based on Marine Association Information, Unidentified Ship Detection, Ship Detection and Type Classification based on Marine Geographic Information, Marine Information Convergence, and others analysis function. Figure 3b shows the functions of the Vessel Big Data layer and the implementation software stack based on open source.

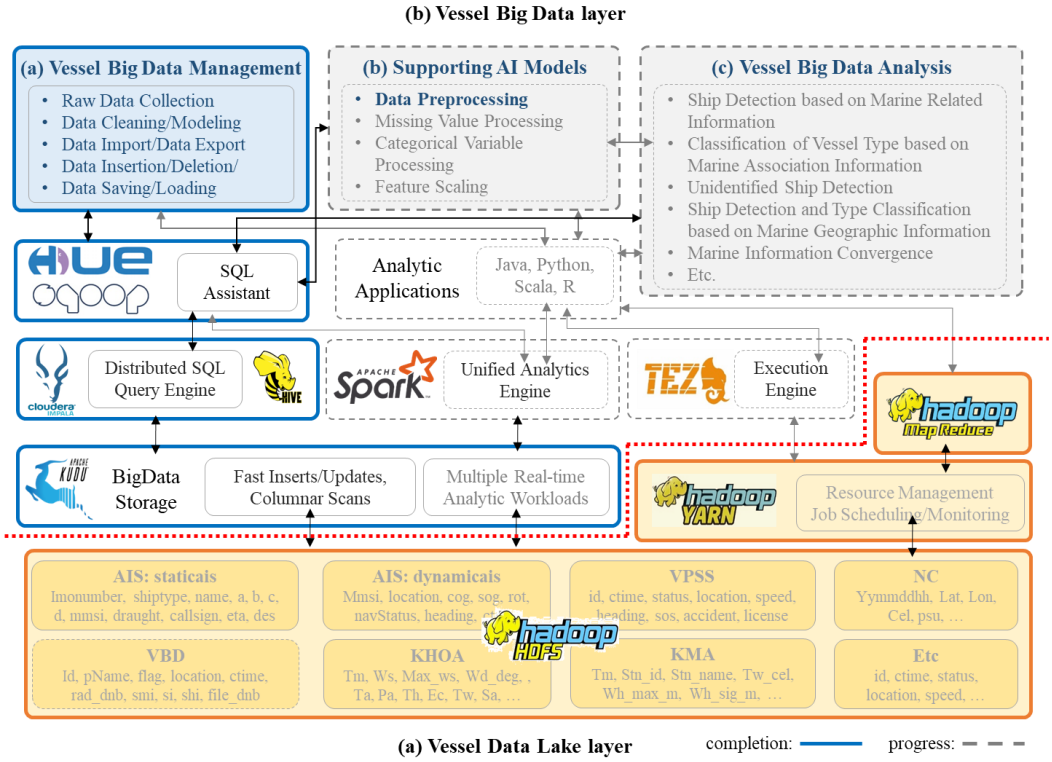


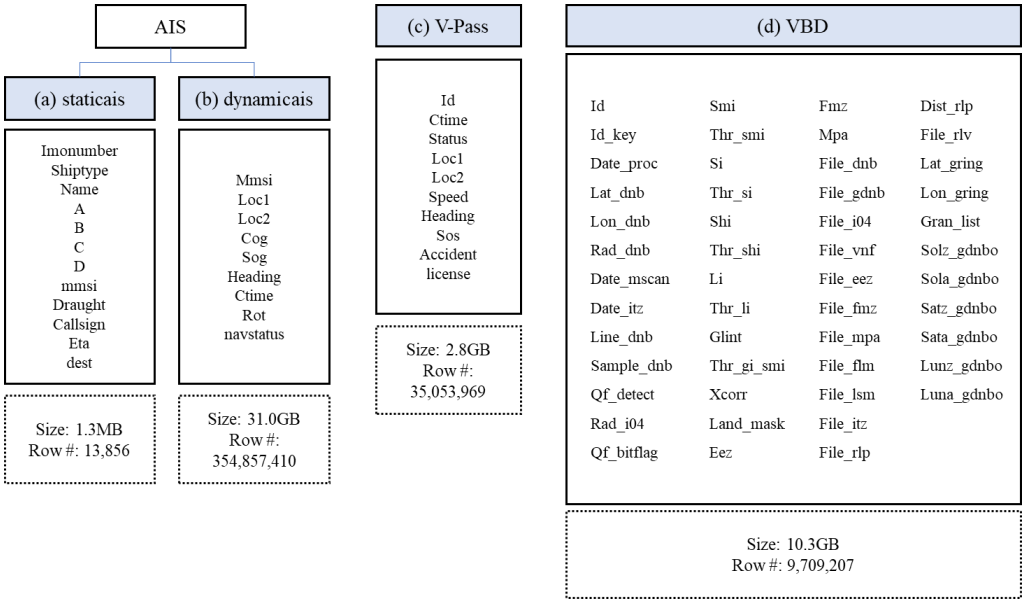
Figure 3. Open Source Software stack of Vessel Big Data layer.

The software stack of the Vessel Big Data layer is composed of the following. The Hue [29] of the Vessel Big Data layer in Figure 3b is used to process the functions of the Vessel Big Data Management module or the Support AI Models module with SQL on the dashboard. Hue is an open-source SQL Assistant for databases and data warehouses which supported by dashboards form. In this paper, we use Impala [30] and Kudu [31] in Figure 3b to build a Data Warehouse on the Hadoop file system (e.g., Data Lake), which it handles the functions of the Vessel Big Data Management module or the Support AI Models module. Apache Impala is a query processing engine. Apache Kudu is an open-source distributed data storage engine that makes it easy to do fast analysis on fast-changing data. Unlike many other columnar storage, Kudu provides a primary key which enabling millisecond-level random access. Since Kudu supports both OLAP (online analytical processing) and OLTP (online transaction processing) queries, which the structure of the big data analysis system can be simplified. The Analytic Application module in Figure 3b supports tools to program and implement functions of each module that cannot be processed with SQL. It supports programming languages such as Java, Python, Scala, and R that can program each function based on the Spark [32] module and the TEZ [33] module. Apache Spark is a unified computing open source engine and set of libraries for processing data in parallel in a clustered environment. Spark supports Python, Java, Scala, and R, and provides a wide range of libraries from SQL to streaming and machine learning. Apache TEZ is a MapReduce alternative data processing framework that runs on top of Hadoop Yarn. TEZ saves the processing results of the Map phase in memory and directly transfers them to the Reduce phase to improve speed by reducing IO overhead.

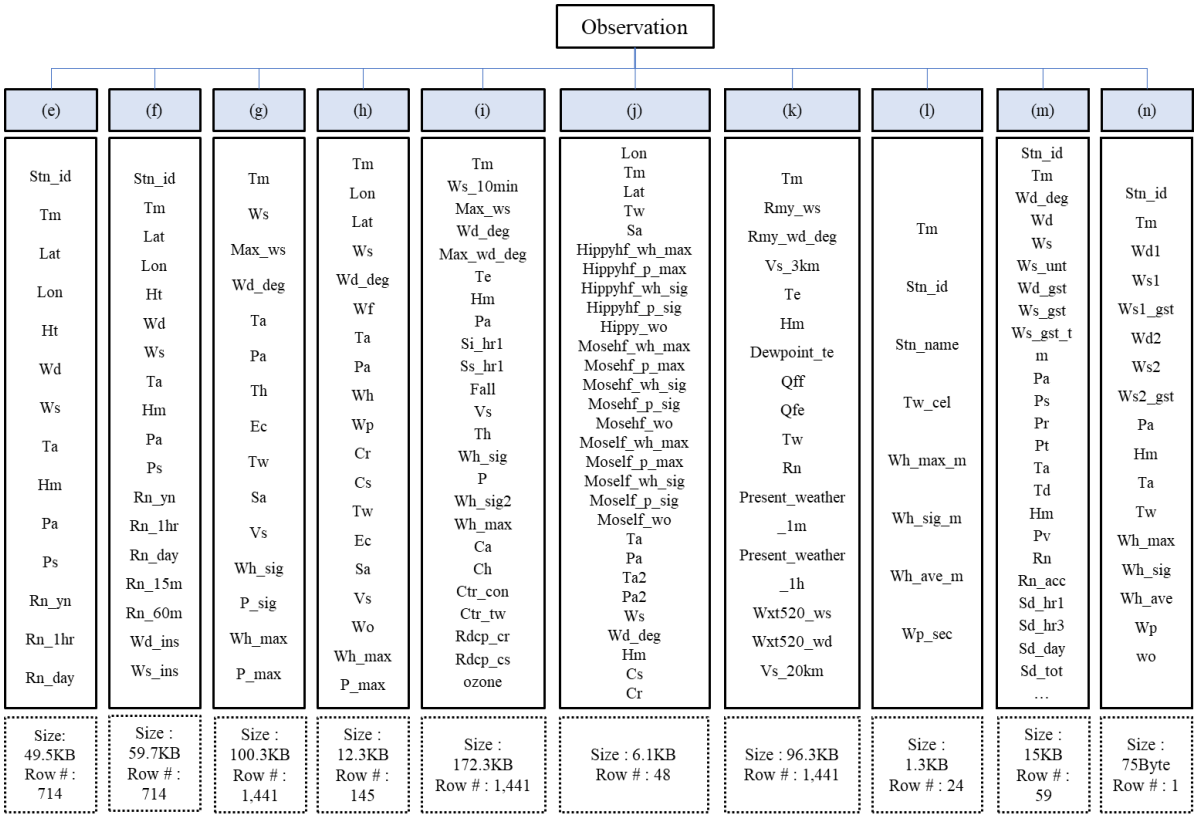
- Building a Vessel Data Lakehouse

This subsection shows how to build a Data Lakehouse in Data Lake with vessel-related data from Table 1 using the Vessel Big Data layer. Figure 4 shows the schema for building AIS, V-Pass, VBD, and Observation data in Table 1 into a Data Lakehouse. Figure 5 describes the meaning of the field names in the schema of Figure 4. A table is created in the schema format defined in Figure 4 by using Impala SQL in Vessel Big Data layer. There are two ways to import the original csv file data into a Data Lakehouse table: using Impala SQL

and using the Import Python module of the Extraction and Ingestion layer. In this paper, Impala SQL is used as an import method. Figure 6 shows the tables of Data Lakehouse by using Impala SQL command of "show tables;". In Figure 6, a table with a different name than Table 1 shows an intermediate table created for analysis. Figure 7 shows the contents of the AIS Static table using the "SELECT * FROM ais.staticais" command in Impala SQL.



(**a**) Schema of AIS, V-Pass and VBD;



(**b**) Schema of Marine Observation data

Figure 4. Data Lakehouse schema for AIS, V-Pass, VBD, and Observation data.

(a) staticais		(b) dynamicais	
Field Name	Description	Field Name	Description
imonumber	IMO number	mmsi	MMSI
shiptype	Type of ship	loc1	location
name	name	loc2	location
A	dimension	cog	course over ground
B	dimension	sog	speed over ground
C	dimension	heading	Heading
D	dimension	ctime	time
mmsi	Maritime Mobile Service Identity (MMSI)	rot	rate of turn
draught	ships draught	navstatus	navigational status
callsign	Call sign		
eta	estimated time of arrival		
dest	destination		

(**a**) Description of field names of AIS schema;

(c) V-Pass	
Field Name	Description
ctime	time
status	status
loc1	location
loc2	location
speed	speed
heading	heading
sog	speed over ground
accident	type of accident
license	buisnessman

(**b**) Description of field names of V-Pass schema;

(d) VBD (VIIRS boat detection)			
Field Name	Description	Field Name	Description
id	integer VBD ID.	EEZ	Exclusive Economic Zone for VBD pixel.
id_Key	Unique VBD ID.	FMZ	Fishery Management Zone for VBD pixel
Date_Proc	Date/time of VBD processing.	MPA	Marine Protected Area for VBD pixel
Lat_DNB	VBD pixel latitude from VIIRS DNB geolocation file	File_DNB	VIIRS DNB HDF5 file
Lon_DNB	VBD pixel longitude from VIIRS DNB geolocation file	File_GDNB	VIIRS DNB geolocation HDF5 file
Rad_DNB	Radiance of VBD pixel in VIIRS DNB band	File_I04	VIIRS I04 file
Date_Mscan	VBD pixel date-time at mid-point of DNB scan reported	File_VNF	VIIRS Nightfire file used to cross-match VBD pixel
Date_LTZ	VBD pixel date-time at mid-point	File_EEZ	Exclusive Economic Zone reference vector file
Line_DNB	Line number of VBD pixel in VIIRS DNB band	File_FMZ	Fishery Management Zone reference vector file
Sample_DNB	Sample number of VBD pixel in VIIRS DNB band	File_MPA	Marine Protected Area reference vector file
Rad_I04	Radiance of VBD pixel in VIIRS I4 band (3.7 um)	File_FLM	Flare mask reference file
QF_Detect	Integer quality flag for VBD pixel	File_LSM	Land-sea mask reference file.
QF_Bitflag	Quality flag for VBD algorithm.	File_LTZ	Local time zone reference file
SMI	Spike Median Index value for VBD pixel	File_RLP	Recurring light points vector file.
Thr_SMI	Spike Median Index threhsold for VBD pixel.	Dist_RLP	this is the distance to the nearest recurring light point.
SI	Sharpness Index value for VBD pixel	File_RLV	Recurring light polygon vector file.
Thr_SI	Sharpness Index threhsold for VBD pixel.	Lat_Gring	Latitude values as a series of semi-colon separated points
SHI	Spike Height Index value for VBD pixel	Lon_Gring	Longitude values
Thr_SHI	Spike Height Index threhsold for VBD pixel.	Gran_List	Granule names as a semi-colon separated list
LI	Lunar Illuminance value for VBD pixel	SOLZ_GDNBO	Solar zenith angle angle relative to the VBD pixel measured
Thr_LI	Lunar Illuminance threshold.	SOLA_GDNBO	Solar azimuth angle angle
Glint	Probability of having lunar glint impacting DNB data. Values range from 0-1.	SATZ_GDNBO	Satellite zenith angle relative to the VBD pixel measured
Thr_Gl_SMI	Spike Median Index threshold inside probable glint ellipse.	SATA_GDNBO	Satellite azimuth angle relative to the VBD pixel measured
Xcorr	DNB vs I-band local cross-correlation value.	LUNZ_GDNBO	Lunar zenith angle angle
Land_Mask	Land-sea mask flag. Land=0. Water=3. Near-shore=1,2.	LUNA_GDNBO	Lunar azimuth angle angle

(**c**) Description of field names of VBX schema

(e) AWS hour	
Field Name	Description
STN_ID	AWS ID
TM	Observation time (year month date time)
LAT	Latitude (deg)
LON	Longitude (deg)
HT	Altitude (m)
WD	10-minute average wind direction (0.1 deg)
WS	1 minute average wind speed (0.1 m/s)
TA	1 minute average temperature (0.1 C)
HM	1 minute average humidity (0.1%)
PA	1 minute average local pressure (0.1 hPa)
PS	1 minute mean sea level pressure (0.1 hPa)
RN_YN	Precipitation detection (0: no precipitation)
RN_1HR	Hourly cumulative precipitation (0.1 mm)
RN_DAY	Cumulative daily precipitation (0.1 mm)

(f) AWS minute	
Field Name	Description
STN_ID	AWS ID
TM	Observation time (year month date hour minute)
LAT	Latitude (deg)
LON	Longitude (deg)
HT	Altitude (m)
WD	1 minute average wind direction (0.1 deg)
WS	1 minute average wind speed (0.1 m/s)
TA	1 minute average temperature (0.1 C)
HM	1 minute average humidity (0.1%)
PA	1 minute average local pressure (0.1 hPa)
PS	1 minute mean sea level pressure (0.1 hPa)
RN_YN	Precipitation detection (0: no precipitation)
RN_1HR	Hourly cumulative precipitation (0.1 mm)
RN_DAY	Cumulative daily precipitation (0.1 mm)
RN_15M	15-minute transit cumulative precipitation (0.1 mm)
RN_60M	60-minute transit cumulative precipitation (0.1 mm)
WD_INS	Daily maximum wind direction (0.1 deg)
WS_INS	Maximum charging speed per day (0.1 m/s)

(l) KMA	
Field Name	Description
STN_ID	Site number
TM	Observation time (year month date hour minute)
WD1	Wind direction 1 (deg)
WS1	Wind speed 1 (m/s)
WS1_GST	GUST wind speed 1 (m/s)
WD2	Wind direction 2 (deg)
WS2	Wind speed 2 (m/s)
WS2_GST	GUST wind speed 2 (m/s)
PA	Local pressure (hPa)
HM	humidity(%)
TA	air temperature (C)
TW	water temperature (C)
WH_MAX	Maximum wave height (m)
WH_SIG	Significant wave height (m)
WH_AVE	Average wave height (m)
WP	wave cycle (sec)
WO	wave direction (deg)

(**d**) Description of field names of AWS and KMA schema

Figure 5. *Cont.*

(g, h, I, j, k, m, n) KHOA			
Field Name	Description	Field Name	Description
STN_ID	Observation Office	CH_MIN	Lowest Ceiling (100m)
TM	Observation time (year month date hour minute)	CT	Cloud shape (statistical table)
WD_DEG	wind direction (deg)	CT_TOP	Upper Cloud Formation (GTS)
WD	Wind direction (32 directions)	CT_MID	Intermediate Cloud Formation (GTS)
WS	wind speed	CT_LOW	Low Tier Cloud (GTS)
WS_UNT	wind speed unit, 0 : 0.1 m/s, 1 : 1 knots	VS	Visibility (10m)
WD_GST	GUST wind direction (10deg)	SS_HR1	sunshine (0.1hr)
WS_GST	GUST wind speed (1m/s)	SI_HR1	Insolation (0.01MJ/m2)
WS_GST_TM	GUST Time (Hour Minutes)	TG	Normal temperature (0.1C)
PA	Local pressure (0.1hPa)	ST_GD	ground condition
PS	Sea level pressure (0.1 hPa)	TS	ground temperature (0.1C)
PR	Air pressure change amount (0.1hPa)	TE_005	0.05m underground temperature (0.1C)
PT	Atmospheric pressure change trend	TE_01	0.1m underground temperature (0.1C)
TA	Temperature (0.1C)	TE_02	0.2m underground temperature (0.1C)
TD	Dew point (0.1C)	TE_03	0.3m underground temperature (0.1C)
HM	Relative humidity (1%)	TE_05	0.5m steel pipe underground temperature (0.1C)
PV	Vapor pressure (0.1 hPa)	TE_10	1.0m steel pipe underground temperature (0.1C)
RN	Hourly precipitation (0.1mm)	TE_15	1.5m steel pipe underground temperature (0.1C)
RN_ACC	Precipitation accumulation period	TE_30	3.0m iron pipe underground temperature (0.1C)
SD_HR1	1 hour new snow (0.1cm)	TE_50	5.0m steel pipe underground temperature (0.1C)
SD_HR3	3 hours of new snow (0.1cm)	WH	Wave height (0.1m)
SD_DAY	One cumulative new snow (0.1cm)	RN_INT	Maximum precipitation intensity (0.1mm)
SD_TOT	Snow cover (0.1 cm)	RN_DAY	Daily precipitation (statistical table)
WC	Current diary (GTS)	ST_SEA	spongy state
WP	Past Diary (GTS)	BF	Beaufart maximum wind power
WW	Prize number (domestic)	RN_JUN	Daily precipitation (GTS)
CA_TOT	All Clouds (1/10)	IR	Precipitation data included
CA_MID	Lower Middle Clouds (1/10)	IX	Inclusion of weather data

(**e**) Description of field names of KHOA schema

Figure 5. The meaning description of field names in the schema of Figure 4.

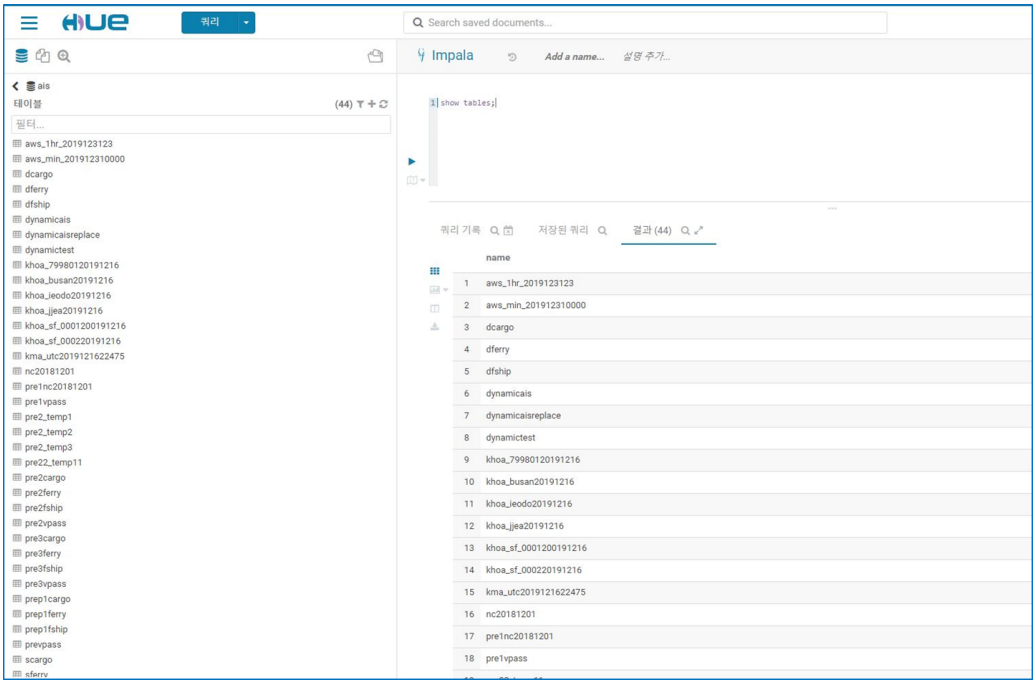
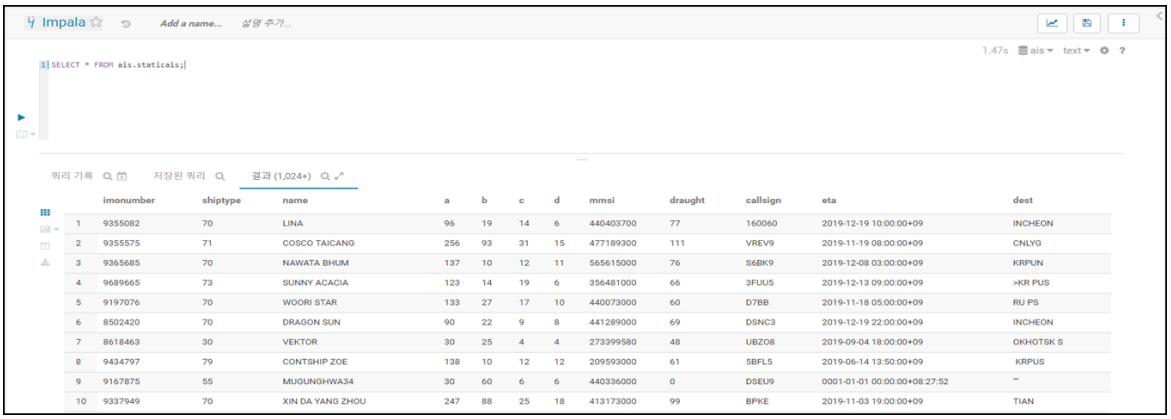


Figure 6. Data Lakehouse tables for AIS, V-Pass, VBD, and Observation data.



	imonumber	shiptype	name	a	b	c	d	mmsi	draught	callsign	eta	dest
1	9355082	70	LINA	96	19	14	6	440403700	77	160060	2019-12-19 10:00:00+09	INCHEON
2	9355575	71	COSCO TAICANG	256	93	31	15	477189300	111	VREV9	2019-11-19 08:00:00+09	CNLYG
3	9365685	70	NAWATA BHUM	137	10	12	11	565615000	76	S6BK9	2019-12-08 03:00:00+09	KRPUN
4	9689665	73	SUNNY ACACIA	123	14	19	6	356481000	66	3FUU5	2019-12-13 09:00:00+09	>KR PUS
5	9197076	70	WOORI STAR	133	27	17	10	440073000	60	D7BB	2019-11-18 05:00:00+09	RU PS
6	8502420	70	DRAGON SUN	90	22	9	8	441289000	69	DSNC3	2019-12-19 22:00:00+09	INCHEON
7	8618463	30	VEKTOR	30	25	4	4	273399580	48	UBZO8	2019-09-04 18:00:00+09	OKHOTSK S
8	9434797	79	CONTSHIP ZOE	138	10	12	12	209593000	61	5BFL5	2019-06-14 13:50:00+09	KRPUS
9	9167875	55	MUGUNGHWA34	30	60	6	6	440336000	0	DSEU9	0001-01-01 00:00:00+08:27:52	–
10	9337949	70	XIN DA YANG ZHOU	247	88	25	18	413173000	99	BPKE	2019-11-03 19:00:00+09	TIAN

Figure 7. Contents of the AIS Static table.

3.3.2. Vessel AI (Artificial Intelligence) Layer

The Vessel AI layer in Figure 8 consists of the Vessel AI model module, ML (Machine Learning)/DL (Deep Learning) Algorithm module, and AI Framework module. The Vessel AI model module in Figure 8a consists of 5 basic models for ship AI analysis and other extended models. In the Vessel AI model module, an analysis model is created, trained, and tested based on a model suitable for each AI analysis purpose of vessel-related data stored in the data layer, and then AI analysis or prediction is performed. This module supports basic models such as Vessel Track Prediction Model, Abnormal Ship Detection Model, Ship Activity Analysis Model, Ship Distribution Prediction Model and Fish Habitat Suitability Analysis Model. If a special analysis other than the basic model is required, an extended model can be created by selecting an appropriate algorithm from the ML/DL Algorithm module. The ML/DL algorithm module in Figure 8b supports the machine learning algorithm or deep learning algorithm used in the Vessel AI Model module. This algorithm module provides HMM, Association Rule, K-means, Decision Tree, Random Forest, CNN (Convolutional Neural Network), and RNN (Recurrent Neural Network) as basic algorithms. If an extended algorithm is required, it is supported by the AI framework in Figure 8c. The AI framework module provides Anaconda, an integrated development environment for Python, and TensorFlow, PyTorch, and Keras, ML development frameworks.

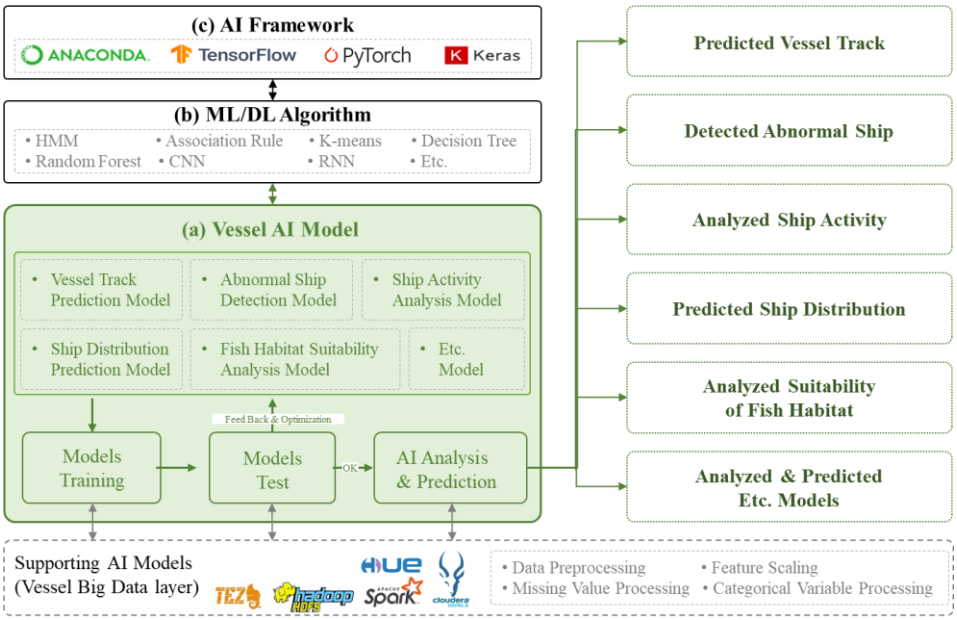


Figure 8. Conceptual Diagram of Vessel AI layer. (**a**) Vessel AI Model; (**b**) ML/DL Algorithm; (**c**) AI Framework.

3.4. Vessel Application Services

Vessel Application Services in Figure 1d consists of Visualization Services and Operation Services. Operation Services provides Vessel Big Data Analysis results of the Vessel Big Data layer and AI Analysis results of the Vessel AI layer. Visualization Services provides visualization of Big Data Analysis results and AI Analysis results. Since this paper focused on building a Vessel Data Lakehouse, only a few Vessel Application Services were implemented as use cases. The next subsection presents a method of vessel distribution and activity intensity using Vessel Big Data layer and a method of predicting fishing activity using Vessel AI layer.

3.4.1. Identification of Distribution of Ship Types

This subsection shows how to identify the distribution by ship type based on AIS in Busan and visualize the activity intensity of each ship. The calculating of distribution location by ship type consists of three steps. In the first step, only the data of the Busan area is extracted from the AIS data, and the extracted data is preprocessed. The second step calculates the activity intensity of each ship. The last step is to visualize the activity intensity of the ship on the map. Table 3 shows the characteristics of the AIS data built in the Vessel Data Lakehouse.

Table 3. Characteristics of AIS Data.

AIS				**Location**	
ship	ship type	ship number	period	latitude	longitude
cargo	70–79	9058	2018-12-01 – 2019-12-19	33°–38°	124°–132°
tanker	80–89	3104			
passenger	60–69	133			
fishing	30	458			
other		1103			

In the preprocessing step, Impala SQL is used to extract AIS data within the range of Busan in Figure 9. Table 4a shows the number of data for each ship type in Busan area. Table 4b shows the number of data extracted from the data in Table 4a at 2-min intervals. From the data in Table 4a, MMSI (Maritime Mobile Service Identity) in AIS is used as a key, clustered at daily intervals, 10 SOGs (Speed Over Ground) in AIS are extracted from daily data, and the COG (Course Over Ground) in AIS is normalized by using Equation (1). Classify the category labels as follows: 0 is a fishing ship, 1 is a non-fishing ship, 2 is a ferry, and 3 is a cargo. Figure 10 shows the preprocessing results of AIS data for fishing ships.

$$ncog = \frac{\sum_{k=1}^{n} |cog_{i+1} - cog_i|}{n} \quad (1)$$

here, *ncog* (i.e., *scog*) is a normalization of *cog* in AIS, *cog* in AIS is a course over ground.

Table 4. Preprocessing of AIS Data.

	(a) Number of Row	**(b) Extraction Data (Per 2 Min)**	**(c) Extraction Data (Per 1 Day)**
Fishing	17,853,172	131,390	1761
Ferry	8,236,992	143,757	2803
Cargo	56,377,013	292,082	5273

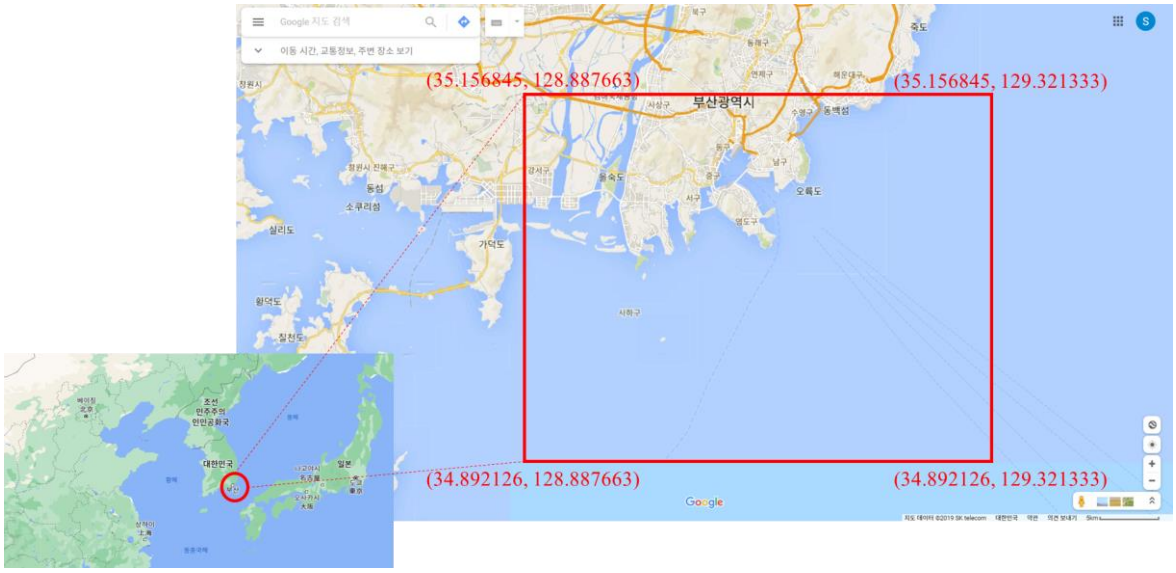


Figure 9. AIS Data Extraction Range.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	mmsi	start-ctime	end-ctime	start-longi	start-latitu	end-longi	end-latitu	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	sum of co	category
2	273828800	2019-06-05 4:43	2019-06-05 5:08	129.28	35.1625	129.2399	35.12067	7.5	7.4	7.7	7.5	7.6	7.6	7.6	7.7	7.8	7.4	0.132	1
3	273828800	2019-06-05 6:30	2019-06-05 6:46	129.068	35.00471	129.0452	34.9946	6	5.8	5.3	4.8	4.6	4.6	5	5.9	6	6	0.667	0
4	273828800	2019-07-12 11:21	2019-07-12 11:57	129.1223	34.91319	129.1302	34.92053	6.1	4.8	4.3	0.7	0.6	0.8	1.1	0.5	1.1	0.5	1.233	0
5	273828800	2019-07-16 20:17	2019-07-16 20:32	129.0127	35.03444	129.0421	35.01871	9.1	9.5	8.8	8.4	9	9	8.9	8.4	8.4	8.3	1.09	1
6	273314740	2019-03-15 11:51	2019-03-15 12:07	129.0197	35.01268	129.0296	34.965	9.3	10.5	11.2	11.4	11.5	11.6	10.8	10.9	11	11.1	0.176	1
7	273314740	2019-04-06 12:17	2019-04-06 12:31	129.0284	35.02004	129.0729	34.99685	9.3	9.8	10.6	11	11.2	11.1	11.2	11.6	11.7	11.8	0.688	1
8	273314740	2019-11-25 15:31	2019-11-25 15:46	129.2853	35.07913	129.2537	35.06126	7.6	7.4	7.9	8.1	8.2	7.8	7.5	7.9	7.7	7.3	0.26	1
9	273314740	2019-11-25 17:58	2019-11-25 18:47	129.0355	34.93876	129.0184	34.91966	6.4	5.2	4.3	4.2	4.1	3.8	4.4	1.6	1.9	1.7	4.926	0
10	273314740	2019-11-26 4:11	2019-11-26 4:27	129.149	34.99341	129.1042	34.96632	9.2	10.6	10.8	10.7	10.7	11.1	10.8	11.2	11.3	11.3	0.196	1
11	273314740	2019-11-26 0:02	2019-11-26 0:16	129.0775	34.95263	129.0663	34.94337	3.3	3.4	3.3	3.3	3.1	3.3	3.3	3.3	3.4	3.5	0.231	0
1753	440126860	2019-09-01 13:51	2019-09-01 14:05	129.2317	35.16724	129.2368	35.16716	2.4	1.8	3.4	3.1	2.5	1.7	1.5	1.4	1.1	1	2.791	0
1754	440126860	2019-09-02 11:15	2019-09-02 12:06	128.9678	35.01485	128.9758	35.01752	2.6	1.3	0.5	0.2	0.5	0.5	0.2	0.1	0.7	2.3	3.536	0
1755	412336831	2019-05-20 7:38	2019-05-20 7:58	129.002	34.89319	129.0705	34.92463	10.6	11.2	11.3	11.7	11.5	11.6	11.7	11.7	11.5	11.5	0.572	1
1756	412336831	2019-05-20 9:13	2019-05-20 9:28	129.0847	35.01431	129.069	35.02442	5.2	3.3	3.2	3.8	4.2	4.2	4	4.1	4.4	4.4	0.481	0
1757	412336831	2019-05-22 14:15	2019-05-22 14:31	129.4014	35.20586	129.4599	35.24717	13.5	13.5	13.6	13.6	13.6	13.6	13.6	13.6	13.6	13.7	0.074	1
1758	412336831	2019-07-21 9:55	2019-07-21 10:12	128.9376	34.99214	128.9982	34.99872	10.4	10.4	10.3	10.4	10.4	10.7	10.3	10.2	10.4	10.6	0.198	1
1759	412336831	2019-07-21 10:35	2019-07-21 11:26	129.0554	35.00475	129.0905	35.00532	5.7	4.5	5.8	6	0.1	0.5	0.2	0.1	0.9	0.5	13.38	0
1760	412336831	2019-07-22 0:21	2019-07-22 16:33	129.0468	35.03148	129.0471	35.03137	0.1	0.1	0.3	0.1	0	0.1	0.2	0.1	0	0.1	0	0
1761	412336831	2019-07-23 6:55	2019-07-23 7:10	129.045	35.02209	129.0889	35.01064	7.6	8	8.3	9	9.1	9.6	10.3	10.4	10.5	10.5	0.784	1
1762	412336831	2019-07-23 0:01	2019-07-23 0:16	129.047	35.03123	129.047	35.03111	0.1	0.1	0.1	0.3	0.2	0	0	0.4	0.2	0.1	0	0

Figure 10. Preprocessing results of AIS data for fishing ships.

In the second step, Equation (2) is used to calculate the cluster strength for each MMSI. The higher the cluster intensity, the higher the vessel's activity. Figure 11 shows the calculation result of cluster strength by MMSI.

$$SC = ns \times (ascog \times 0.2 + scog \times 0.5 + asog \times 0.3) \quad (2)$$

here, SC is a strength of clusters, ns is a number of sampling of MMSI, $ascog$ is an average of sum of abs of cog, $scog$ is a normalization of cog, and $asog$ is an average of sog.

mmsi	start_ctime	end_ctime	start_location	end_location	ns	ascog	ssog	asog	SC
441713000	2019-05-09 11:36:47	2019-12-04 23:57:08	(129.035415649414,35.022148132324 2)	(129.451705932617,35.1212501525879)	14	122.91	83.19	5.94	951.47
273899000	2019-04-27 04:58:05	2019-05-25 17:28:58	(129.506851196289,35.229305267334)	(129.00373840332,35.0804634094238)	9	183.63	65.74	7.30	646.11

mmsi location displayed on map

mmsi activity intensity displayed on the map

Figure 11. Calculation result of cluster strength by MMSI.

In the last step, using the data in Figure 11, it is visualized and displayed on Google Earth as shown in Figure 12.

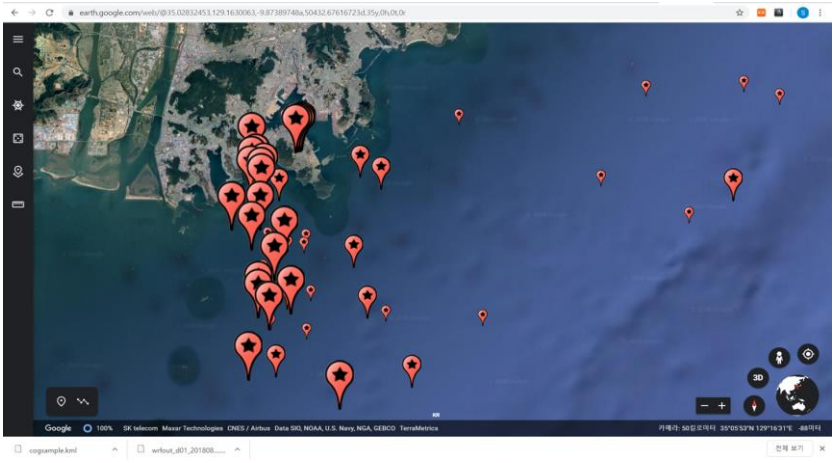


Figure 12. Visualization result of ship's activity intensity on Google Earth.

The Algorithm 1 for identification of distribution of ship types is as follows.

Algorithm 1. ShipActivity(A)

Input: the AIS data set A, cog in AIS is a course over ground, ns is a number of sampling of mmsi, SC is a strength of clusters, ascog is an average of sum of abs of cog, scog is a normalization of cog, and asog is an average of sog
Output: the preprocessing data set T1, the calculated cluster strength data set T2
Method:
01: T1 ← ***Preprocessing***(A);
02: T2 ← ***Clusterstrength***(T1);
03: ***Visualization***(T2);

Preprocessing(A)
04: for $i \leftarrow 1$ to n do
05: if 33 <= $A_{i,\text{latutude}}$ <= 38 and 124 <= $A_{i,\text{longitude}}$ <= 132
06: then $\text{Temp1}_i \leftarrow A_i$
07: end
08: $\text{Temp2}_j \leftarrow kmeans(\text{Temp1}_{i,\text{mmis}})$
09: extract $\text{Temp2}_j \leftarrow$ 2-minute intervals Temp2_j
10: extract $\text{Temp2}_j \leftarrow$ #10 sogs Temp2_j
11: for j ← 1 to m do
12: $\text{Temp3}_j \leftarrow \frac{\sum_{k=1}^{n} |cog_{k+1} - cog_k|}{n}$
13: end
14: for $l \leftarrow 1$ to p do
15: $\text{Temp4}_l \leftarrow \text{classify}(\text{Temp3}_{j,\text{type}})$
16: end
17: Return Temp4_l

Clusterstrength(T1)
18: for $i \leftarrow 1$ to n do
19: $\text{Temp5}_i \leftarrow ns \times (ascog \times 0.2 + scog \times 0.5 + asog \times 0.3)$
20: end
21: Return Temp5_i

Visualization(T2)
22: for $i \leftarrow 1$ to n do
23: $display(\text{T2}_{i,\text{mmsi}})$;
24: $display(\text{T2}_{i,SC})$
25: end

In line 4 to 16, preprocessing phase extract 10 sogs (speed over ground) in AIS are extracted from daily data, then calculate a normalization of *cog*. In line 18 to 21, calculating of cluster strength phase compute the cluster strength for each MMSI. In line 22 to 25, visualization phase, the cluster strength, which is the activity intensity based on the ship's MMSI, is displayed on the google Earth.

3.4.2. Predicting Fishing Activity

This subsection shows how to predict the fishing activity using LSTM and visualize the fishing activity of each ship in Google Earth. Fishing activity was predicted using the data in Figure 10, which is the result of the preprocessing algorithm in Section 3.4.1. The fishery activity was predicted by designing the input and output with the ship speed of 10 intervals and *ncog* (normalization of course over ground) in the data in Figure 10 suitable for the LSTM (Long Short-Term Memory) algorithm. Figure 13 shows the results of fishing activities using LSTM designed for input and output. The red line represents the fishing activity and the yellow line represents the sailing of the vessel. A green triangle indicates the current ship's position.

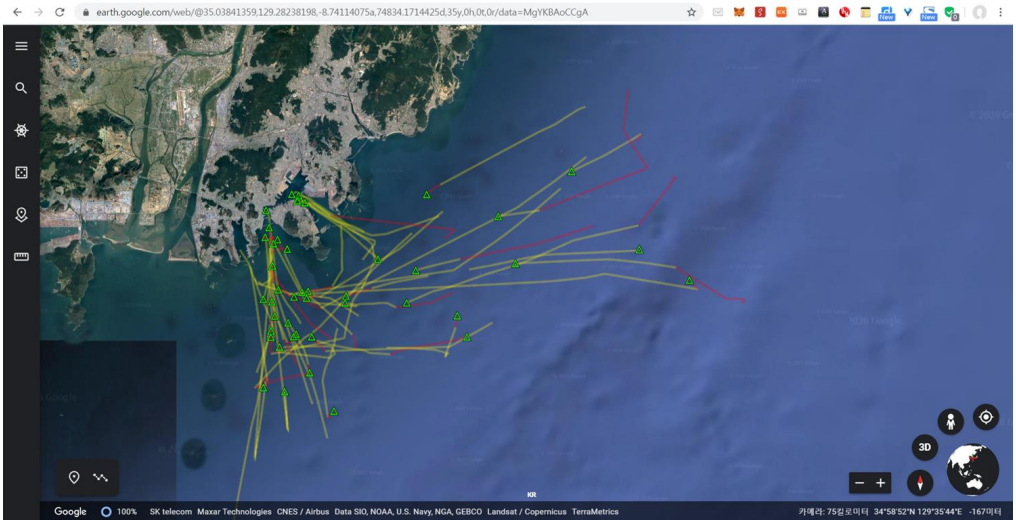


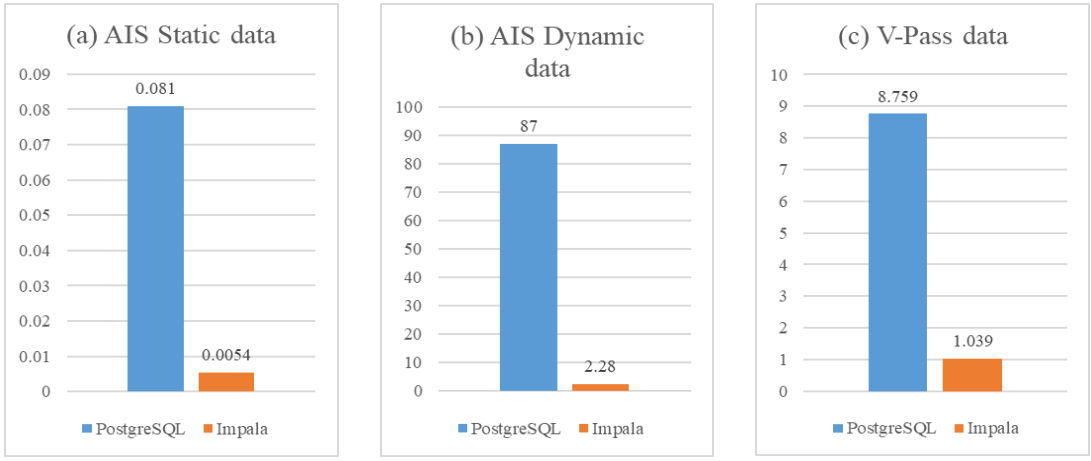
Figure 13. Visualization result of Fishing Activity Prediction on Google Earth.

4. Experimental Results

In this section, the experimental results are presented to Data Lakehouse and analysis results of the implemented system.

4.1. Data Lakehouse Performance Evaluation

In order to measure the performance of the Data Lakehouse, this subsection compared the query processing results of PostgreSQL, a relational database, and Impala, the query engine of the Data Lake. The Marine Data Lake data in Table 1 were used to evaluate the performance of the Data Lakehouse. Figure 14 shows the comparison result of query processing between PostgreSQL and Impala SQL. The result measure is the time it takes to process the count query, which counts the number of rows. The shorter the query processing time, the better the performance. Query processing for the same data between a Data Lake cluster with 4 nodes and PostgreSQL with 1 node was evaluated. The hardware specifications of the Data Lake node and the PostgreSQL node are configured. The evaluation query is "SELECT count(*) FROM table-name".



(**a**) Query response time of AIS static data (**b**) Query response time of AIS dynamic data (**c**) Query response time of V-Pass data

Figure 14. Query Response Time (i.e., second) Comparison Results between PostgreSQL and Impala.

Figure 14a compares and evaluates the AIS Static data, which is the size of 1.3 MB and has 13,856 rows. PostgreSQL query processing time for AIS static data is 0.081 s and Impala query processing time is 0.0054 s, impala is 15.06 times faster than PostgreSQL. Impala's query response rate for AIS static data is about 88.14% faster than PostgreSQL's. Figure 14b compares and evaluates the AIS Dynamic data, which is the size of 31 GB and has 354,857,410 rows. PostgresSQL query processing time for AIS Dynamic data is 87 s and Impala query processing time is 2.28 s, impala is 38.20 times faster than PostgresSQL. Impala's query response rate for AIS Dynamic data is about 93.33% faster than PostgreSQL's. Figure 14c compares and evaluates the V-Pass data, which is the size of 2.8 GB and has 35,053,969 rows. PostgresSQL query processing time for AIS Dynamic data is 8.759 s and Impala query processing time is 1.039 s, impala is 8.43 times faster than PostgresSQL. Impala's query response rate for V-Pass data is about 97.38% faster than PostgreSQL's.

4.2. Marine Analysis Performance Evaluation

This subsection evaluates the performance of marine analysis of Vessel AI layer for fishing activity prediction and fishing vessel type forecasting. The data in Table 4c preprocessed in Section 3.4.1 is used for training and testing the prediction model. 9872 rows of data are used for training and testing in a 70:30 ratios. Decision Trees (DT), Random Forest (RF), LSTM (Long Short-Term Memory), and HMM (Hidden Markov Model) algorithms are used for predictive models. Since this paper is focused on the data point of view of building a Lakehouse using actual maritime observation data, the algorithms of the prediction model used the basic models provided by TensorFlow [34] and Keras [35] without tuning. Data processing within the predictive model used Pandas [36]. The input and output parts of each predictive model were modified to fit the preprocessed data. The type of ship is predicted using the speed as an input value for each prediction model. Fishing activity prediction forecasts only one type of fishing ship with speed and *ncog* as input values to each prediction model.

Figure 15 shows the procedure of Fishing Activity and Ship Type Prediction. Figure 16 shows the results of comparison of the accuracy rate of fishing activity and vessel type prediction. Figure 16a shows the vessel type prediction results of data with mixed ships such as fishing ship, cargo, and ferry. In Figure 16a, we compared the prediction accuracy of vessel classification for four prediction models: DT, RF, LSTM, and HMM. The prediction accuracy of vessel classification of RF is approximately 1.10% higher than that of LSTM, 10.97% higher that of DT, 18.11% higher than that of HMM. Figure 16b shows the prediction results of fishing activities from only fishing ship data. In Figure 16b, we compared the prediction accuracy of fishing activity for four pre-diction models: DT, RF, LSTM, and

HMM. The prediction accuracy of fishing activity of LSTM is approximately 0% higher than that of RF, 2.8% higher that of DT, 16.8% higher than that of HMM. In Figure 16, it can be seen that prediction from somewhat classified data shows better results than prediction from a mixture of different types of data.

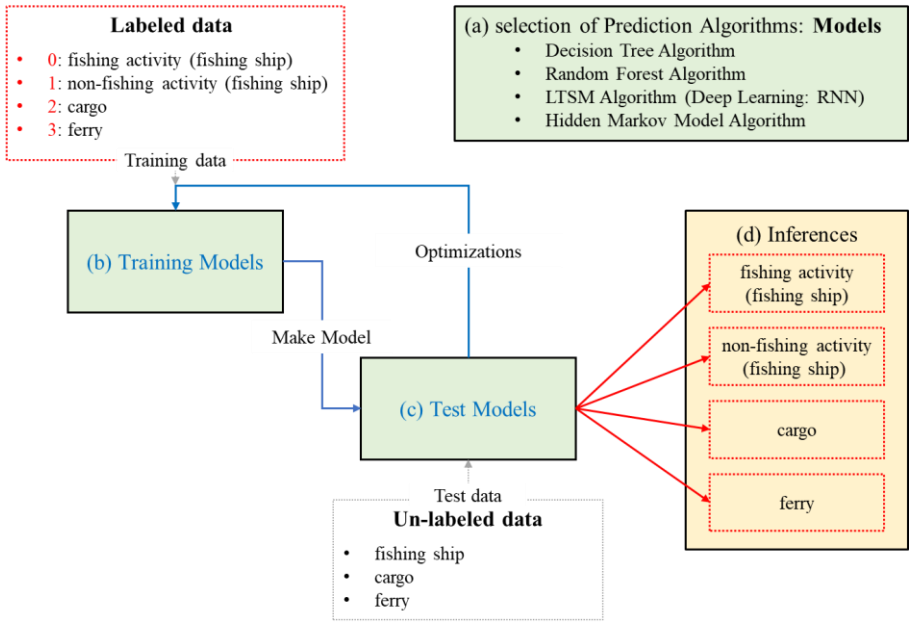
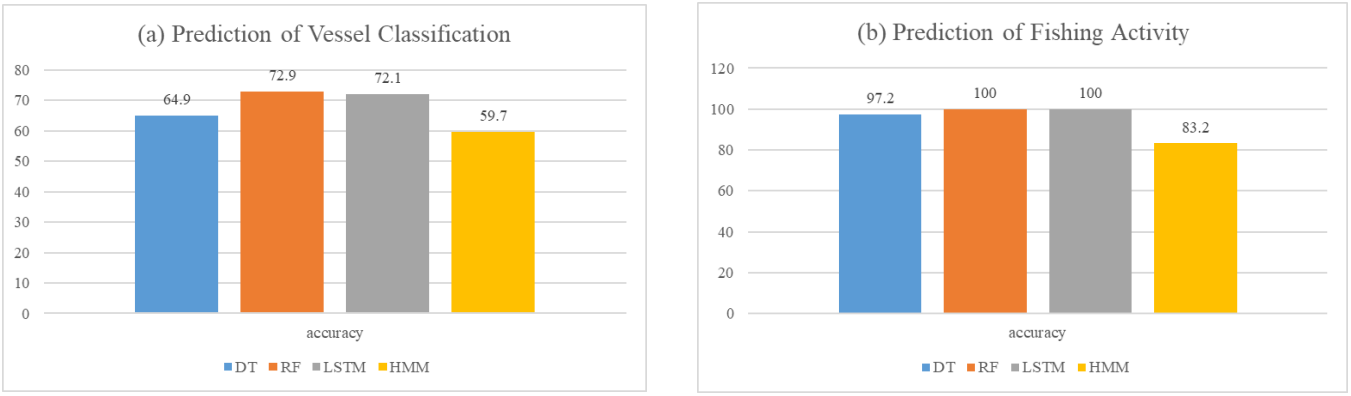


Figure 15. Procedures of Fishing Activity and Ship Type Prediction. (**a**) Selection models phase; (**b**) Training model phase; (**c**) Test models phase; (**d**) Inferences.



(**a**) Results of vessel classification prediction; (**b**) Results of fishing activity prediction.

Figure 16. Comparison of Accuracy in Prediction Fishing Activities and Vessel Types.

5. Conclusions

Various challenges are currently affecting the development of large-scale marine data services, limiting users' ability to use the full potential of this data ecosystem. From a technical point of view, these challenges are mainly related to the big data nature and high level of heterogeneity of marine data sources. In this paper, we designed and implemented the architecture of Vessel Data Lakehouse, which can efficiently manage various types (i.e., heterogeneous) of vessel-related data. The proposed Vessel Data Lakehouse consists of Extraction and Ingestion layer that can collect and store data, Vessel Data Lake layer that can handle marine big data, Vessel Data Warehouse Model that supports marine big data processing and AI, and Vessel Application Services that supports marine application services. The Extraction and Ingestion layer extracts vessel-related data from data sources and stores it in the Vessel Data Lake layer of Data Warehouse Model. The Vessel Data Lake layer constructed a Data Lake for AIS, VPSS, VBD, Observation data based on Apache

Hadoop. The Vessel AI layer of the Data Warehouse Model supports AI analysis and prediction for vessel application services. Vessel Application Service supports Visualization service, Big Data Analysis service, and AI Analysis service for vessels. In this paper, a use case of constructing a Vessel Data Lakehouse using actual vessel-related data and a use case of analyzing vessel distribution and fishing activities with Vessel Application Service were shown, respectively. As a result of the experiment from about 34 GB of data of AIS and VPSS, the Data Lakehouse showed 92.95% higher average query response rate than the relational database, demonstrating the efficiency of the proposed Data Lakehouse. Since the Data Lakehouse in this paper focuses on structured AIS data or observational time series data, it is still insufficient for processing large-scale ocean image data. We plan to expand our current Data Lakehouse using Delta Lake [37] and satellite imageries (i.e., satellite AIS data, satellite SAR data, satellite EO/IR data) to handle large amounts of image data in future work.

Author Contributions: Conceptualization, S.P. and C.-S.Y.; methodology, S.P.; software, S.P.; validation, S.P., C.-S.Y. and J.K.; formal analysis, S.P.; investigation, S.P.; resources, C.-S.Y.; data curation, S.P.; writing—original draft preparation, S.P.; writing—review and editing, C.-S.Y. and J.K.; visualization, S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01842, Artificial Intelligence Graduate School Program (GIST)). This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-02068, Artificial Intelligence Innovation Hub). This work was supported by the project "Monitoring System of Spilled Oils Using Multiple Remote Sensing Techniques" funded by the Korea Coast Guard.

Data Availability Statement: The data used to support the findings of this study are included within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Data Lakehouse. Available online: https://databricks.com/glossary/data-lakehouse (accessed on 11 January 2023).
2. Orescanin, D.; Hlupic, T. Data Lakehouse—A Novel Step in Analytics Architecture. In Proceedings of the 44th International Convention on Information, Communication and Electronic Technology, Opatija, Croatia, 27 September 2021.
3. Vessel Monitoring System. Available online: https://en.wikipedia.org/wiki/Vessel_monitoring_system (accessed on 11 January 2023).
4. Lytra, I.; Vidal, M.E.; Orlandi, F.; Attard, J. A Big Data Architecture for Managing Oceans of Data and Maritime Applications. In Proceedings of the International Conference on Engineering, Technology and Innovation, Madeira, Portugal, 27 June 2017.
5. Lin, B. Overview of High Performance Computing Power Building for the Big Data of Marine Forecasting. In Proceedings of the 2020 International Conference on Big Data and Informatization Education (ICBDIE), Zhangjiajie, China, 23 April 2020.
6. Armbrust, M.; Ghodsi, A.; Xin, R.; Zaharia, M. Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics. In Proceedings of the 11th Annual Conference on Innovative Data System Research, Online, 11 January 2021.
7. Begoli, E.; Goethert, I.; Knight, K. A Lakehouse Architecture for the Management and Analysis of Heterogeneous Data for Biomedical Research and Mega-biobanks. In Proceedings of the 2021 IEEE International Conference on Big Data, Online, 15 December 2021.
8. Park, S.; Cha, B.R.; Kim, J.W. Designing Marine Data Lakehouse Architecture for Managing Maritime Analytics Application. In Proceedings of the 9th International Conference on Advanced Engineering and ICT-Convergence, Jeju Island, Republic of Korea, 13 July 2022.
9. Harby, A.A.; Zulkernine, F. From Data Warehouse to Lakehouse: A Comprarative Review. In Proceedings of the 2022 IEEE International Conference on Big Data, Osaka, Japan, 17 January 2022.
10. Kumar, D.; Li, S. Separating Storage and Compute with the Databricks Lakehouse Platform. In Proceedings of the 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA), Shenzhen, China, 12 October 2022.
11. Hery, H.; Lukas, S.; Yugopuspito, P.; Murwantara, I.M.; Krisnadi, D. Website Design for Locating Tuna Fishing Spot Using Naïve Bayes and SVM Based on VMS Data on Indonesian Sea. In Proceedings of the 3rd International Seminar on Research of Information Technology and Intelligent System, Yogyakarta, Indonesia, 10 December 2020.

12. Zhao, Z.; Tian, Y.; Hong, F.; Huang, H.; Zhou, S. Trawler Fishing Track Interpolation using LSTM for Satellite-based VMS Traces. In Proceedings of the Global Oceans, U.S. Gulf Coast, Singapore, 5 October 2020.
13. Ahmed, I.; Jun, M.; Ding, Y. A Spatio-Temporal Track Association Algorithm Based on Marine Vessel Automatic Identification System Data. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 20783–20797. [CrossRef]
14. Beek, R.V.; Gaol, J.L.; Agus, S.B. Analysis of Fishing with Led Lights in and around MPA and No Take Zones at Natuna Indonesia through VMS and VIIRS Data. In Proceedings of the IEEE Asia-Pacific Conference on Geoscience, Electronics and Remote Sensing Technology, Jakarta, Indonesia, 7 December 2020.
15. Huang, J.; Wan, J.; Yu, J.; Zhu, F.; Ren, Y. Edge Computing-Based Adaptable Trajectory Transmission Policy for Vessels Monitoring Systems of Marine Fishery. *IEEE Access* **2020**, *7*, 50684–50695. [CrossRef]
16. Li, X.; Xia, Y.; Su, F.; Wu, W.; Zhou, L. AIS and VBD Data Fusion for Marine Fishing Intensity Mapping and Analysis in the Northern Part of the South China Sea. *Int. J. Geo-Inf.* **2021**, *10*, 277. [CrossRef]
17. Souza, E.N.; Boerder, K.; Matwin, S.; Worm, B. Improving Fishing Pattern Detection from Satellite AIS Using Data Mining and Machine Learning. *PLoS ONE* **2016**, *11*, e0163760.
18. Alba, J.M.M.; Dy, G.C.; Virina, N.I.M.; Samonte, M.J.C. Localized Monitoring Mobile Application for Automatic Identification System (AIS) for Sea Vessels. In Proceedings of the IEEE 7th International Conference on Industrial Engineering and Applications, Paris, France, 4 January 2020.
19. Prasad, P.; Vatsal, V.; Chowdhury, R.R. Maritime Vessel Route Extraction and Automatic Information System (AIS) Spoofing Detection. In Proceedings of the 2021 International Conference on Advances in Electrical Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 19 February 2021.
20. Evmides, N.; Odysseos, L.; Michaelides, M.P. An Intelligent Framework for Vessel Traffic Monitoring using AIS Data. In Proceedings of the 23rd IEEE International Conference on Mobile Data Management, Online, 6 June 2022.
21. Liu, R.W.; Liang, M.; Nie, J.; Garg, S.; Zhang, Y.; Xiong, Z. Extraction of Hottest Shipping Routes: From Positioning Data to Intelligent Surveillance. In Proceedings of the IEEE 22nd International Conference on Information Reuse and Integration for Data Science, Las Vegas, NV, USA, 10 August 2021.
22. Huang, H.; Cui, X.; Bi, X.; Liu, C.; Hong, F.; Guo, S. FVRD: Fishing Vessels Relationships Discovery System Through Vessel Trajectory. *IEEE Access* **2020**, *8*, 112530–112538. [CrossRef]
23. Xiao, Z.; Fu, X.; Zhao, L.; Zhag, L.; Teo, T.K.; Li, N.; Zhang, W.; Qin, Z. Next-Generation Vessel Traffic Services Systems—From "Passive" to "Proactive". *IEEE Intell. Transp. Syst. Mag.* **2022**, *15*, 363–377. [CrossRef]
24. Tampakis, P.; Chondrodima, E.; Pikrakis, A.; Theodoridis, Y.; Pristouris, K.; Nakos, H.; Petra, E.; Dalamagas, T.; Kandiros, A.; Markakis, G. Sea Area Monitoring and Analysis of Fishing Vessels Activity: The i4sea Big Data Platform. In Proceedings of the 21st IEEE International Conference on Mobile Data Management, Versailles, France, 30 June 2020.
25. Han, J.R.; KIM, T.H.; Choi, E.Y.; Choi, H.W. A Study on the Mapping of Fishing Activity using V-Pass Data—Focusing on the Southeast Sea of Korea. *J. Korean Assoc. Geogr. Inf. Stud.* **2021**, *24*, 112–125.
26. Weather Data Open Portal. Available online: https://data.kma.go.kr/cmmn/main.do (accessed on 6 January 2023).
27. Ocean Data in Grid Framework. Available online: http://www.khoa.go.kr/oceangrid/khoa/intro.do (accessed on 6 January 2023).
28. Apache Hadoop. Available online: https://hadoop.apache.org/ (accessed on 9 January 2023).
29. Hue. Available online: https://gethue.com/ (accessed on 9 January 2023).
30. Apache Impala. Available online: https://impala.apache.org/ (accessed on 9 January 2023).
31. Apache Kudu. Available online: https://kudu.apache.org/ (accessed on 9 January 2023).
32. Apache Spark. Available online: https://spark.apache.org/ (accessed on 9 January 2023).
33. Apache TEZ. Available online: https://tez.apache.org/ (accessed on 9 January 2023).
34. TensorFlow. Available online: https://tensorflow.org/ (accessed on 9 April 2023).
35. Keras. Available online: https://keras.io/ (accessed on 9 April 2023).
36. Pandas. Available online: https://pandas.pydata.org/ (accessed on 9 April 2023).
37. Delta Lake. Available online: https://delta.io/ (accessed on 29 March 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.