



UiT The Arctic University of Norway

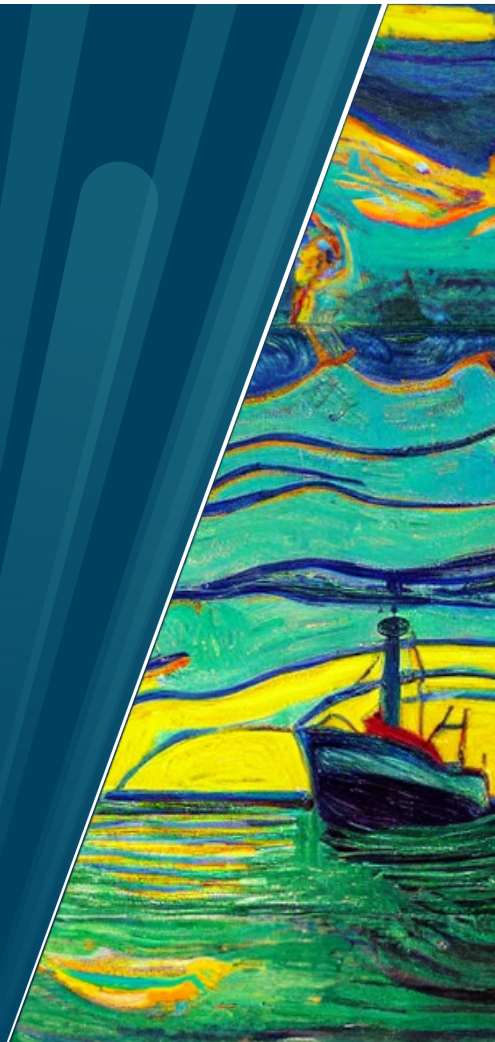
Faculty of Science and Technology

Department of Computer Science

**Dutkat - A Privacy-Preserving System for Automatic Catch
Documentation and Illegal Activity Detection in the Fishing Industry**

Tor-Arne Schmidt Nordmo

A dissertation for the degree of Philosophiae Doctor – February 2023



Acknowledgments

I would like to express my deepest gratitude to my supervisors, Dag Johansen, Håvard Dagenborg, and Michael Riegler, for their consistent support, direction, and motivation throughout the completion of this thesis. Their invaluable contributions and knowledge have played a critical role in shaping my research. Specifically, I would like to acknowledge Dag, who has supported me throughout the Ph.D. journey, and the significant impact that Michael has had in saving my Ph.D. with his expert guidance, availability, unwavering support, and useful discussions.

I would also like to thank my fellow Ph.D. colleagues, Aril Ovesen, Enrico Tedeschi, Aakash Sharma, and Bjørn Aslak Juliussen for their support and for providing a stimulating and collaborative environment. Their contributions to my research have been invaluable. Specifically, Aril for being a dear friend for all these years and for joining me on my Ph.D. journey. Also, to André Pedersen, whose invaluable knowledge, friendship and solidarity for being Ph.D. candidates, albeit at different universities.

I would like to express my sincere appreciation to my parents, Anne-Stina and Torbjørn Nordmo, my siblings, and my grandparents for their love, support and encouragement throughout my academic journey. Their unwavering belief in me has been my source of inspiration.

I would also like to thank all the co-authors and participants who have taken part in my research, without them this thesis would not have been possible.

Thank you all for your invaluable contributions.

Abstract

United Nations' Sustainable Development Goal 14 aims to conserve and sustainably use the oceans and their resources for the benefit of people and the planet. This includes protecting marine ecosystems, preventing pollution, and overfishing, and increasing scientific understanding of the oceans. Achieving this goal will help ensure the health and well-being of marine life and the millions of people who rely on the oceans for their livelihoods. In order to ensure sustainable fishing practices, it is important to have a system in place for automatic catch documentation.

This thesis presents our research on the design and development of Dutkat, a privacy-preserving, edge-based system for catch documentation and detection of illegal activities in the fishing industry. Utilising machine learning techniques, Dutkat can analyse large amounts of data and identify patterns that may indicate illegal activities such as overfishing or illegal discard of catch. Additionally, the system can assist in catch documentation by automating the process of identifying and counting fish species, thus reducing potential human error and increasing efficiency. Specifically, our research has consisted of the development of various components of the Dutkat system, evaluation through experimentation, exploration of existing data, and organization of machine learning competitions. We have also implemented it from a compliance-by-design perspective to ensure that the system is in compliance with data protection laws and regulations such as GDPR. Our goal with Dutkat is to promote sustainable fishing practices, which aligns with the Sustainable Development Goal 14, while simultaneously protecting the privacy and rights of fishing crews.

Contents

Acknowledgments	i
Abstract	iii
Acronyms	xiii
1 Introduction	1
1.1 Background and Motivation	2
1.2 Thesis Statement	5
1.3 Scope and Limitations	7
1.4 Research Methodology	8
1.5 Research Context	10
1.6 Contributions	11
1.6.1 Publication I	12
1.6.2 Publication II	14
1.6.3 Publication III	14
1.6.4 Publication IV	15
1.6.5 Publication V	16
1.6.6 Publication VI	16
1.6.7 Publication VII	17
1.6.8 Publication VIII	17
1.6.9 Author Legend	18
1.6.10 Additional Contributions	18

Contents

1.7	Outline	20
2	Background and Related Work	23
2.1	Illegal, Unregulated and Unreported Fishing	23
2.1.1	Problems	26
2.1.2	Proposed Solutions	29
2.2	Summary and The Way Forward	32
2.3	Brief Overview of Machine Learning Relevant to the Thesis	32
2.3.1	Deep Learning	35
2.3.2	Anomaly Detection	37
2.3.3	Datasets Relating to the Fishing Industry	38
2.4	Related Work	41
2.4.1	Distributed Edge-based Systems	41
2.4.2	Machine Learning	42
2.4.3	Summary of Related Work	45
2.5	Ethical considerations	46
2.6	Summary	47
3	Requirement Analysis	49
3.1	Functional Requirements	50
3.2	Non-Functional Requirements	52
3.2.1	Compliance-by-Design	52
3.2.2	Confidentiality	53
3.2.3	Integrity	53
3.2.4	Availability	53
3.3	Proposed Architecture	54
3.4	Summary	55
4	The Dutkat System	57
4.1	Edge Computational Node	58
4.1.1	Distributed Computation	60

4.1.2	Distributed File System	60
4.2	Mainland Backend Subsystem	60
4.3	Data	61
4.3.1	Automatic Identification System	62
4.3.2	Sales Notes and Trip Logs	63
4.3.3	Satellite Imagery	63
4.3.4	Njord: a Fishing Trawler Video Dataset	65
4.4	Analysis	67
4.4.1	The need for Edge Computation for Analysis in the Fishing Industry	67
4.4.2	Detection of Events and Anomalies in Multimodal Time-Series Data	69
4.4.3	Detection of Slipping Events	75
4.5	Competitions	84
4.5.1	The FishAI Competition	85
4.5.2	MediaEval Benchmark - The NjordVid Task	86
4.5.3	Lessons Learned From The Competitions	88
4.6	Summary	89
5	Discussion	91
5.1	Contributions to Objectives	91
5.1.1	Sub-objective 1	91
5.1.2	Sub-objective 2	92
5.1.3	Sub-objective 3	92
5.1.4	Sub-objective 4	93
5.1.5	Main Objective	94
5.2	Traceability	95
5.3	Generalisability	96
5.4	Digitalisation and sustainable fishing	96
5.5	Legal and Ethical Considerations	97

Contents

5.6	Summary	99
6	Concluding Remarks	101
6.1	Conclusion	101
6.2	Future Work	103
	References	104
A	List of Papers	123
A.1	Author Legend	123
A.2	Paper I: Dutkat: A Multimedia System for Catching Illegal Catchers in a Privacy-Preserving Manner	123
A.3	Paper II: File System Support for Privacy-Preserving Anal- ysis and Forensics in Low-Bandwidth Edge Environments . .	130
A.4	Paper III: Fishing Trawler Event Detection: An important step towards digitization of sustainable fishing	151
A.5	Paper IV: Njord: A Fishing Trawler Dataset	160
A.6	Paper V: Detection of Slipping Events using Multimodal Data	167
A.7	Paper VI: Áika: A Distributed Edge System For AI Inference	170
A.8	Paper VII: FishAI: Sustainable Commercial Fishing	190
A.9	Paper VIII: NjordVid: A Fishing Trawler Video Analytics Task	194

List of Figures

1.1	A Norwegian newspaper article about the fish crime being ranked as one of the biggest threats to welfare state [11]. . . .	3
1.2	Illustration showing relation between contributions and thesis objectives.	12
1.3	Overview of how the objectives relate to the different parts of the system.	13
2.1	Global trends of the world's fish stocks [36].	25
2.2	Discard of dead fish by FV Margiris, the second biggest fishing vessel on earth. Documented by Sea Shepard France [122].	27
2.3	Surveillance feeds on the command bridge of the Hermes trawler.	30
2.4	The components of a Convolutional Neural Network (CNN). .	35
2.5	Global Fishing Watch (GFW) uses their data to map fishing activity around the globe [130].	38
3.1	System architecture that satisfies the requirements outlined above. The edge computational node (1), mainland hub (2), and other data sources (3) used by the mainland hub are shown.	54
4.1	An overview of how the Dutkat system can be applied to data during a fishing trip	58

List of Figures

4.2	An image showing the Automatic Identification System (AIS) path of a vessel (in black) fishing. The data was collected from Kystverket.	62
4.3	An example of a Synthetic Aperture Radar (SAR) image. . .	64
4.4	Sample frames from different videos of the Njord dataset. . .	65
4.5	Sample predictions made by the YOLOv5m model.	66
4.6	top-1 video-level accuracy vs. bitrate plot	68
4.7	Example of pipeline configuration for our Multimodal Event/Anomaly Detection pipeline.	70
4.8	Illustration of the embedding process. Note how the embeddings of the labels are in a different order than the original index.	72
4.9	Pipeline showing how fishing vessels are filtered and ranked. .	76
4.10	Overview of Oil slicks around Svalbard	77
4.11	Illustration of oil slick detection and AIS overlay	82
4.12	Illustration of FishAI participant’s web application solution .	87

List of Tables

2.1	An overview of existing datasets and data sources relevant to the fishing industry.	40
4.1	Comparison of our system against baselines, with standard metrics and Matthews Correlation Coefficient (MCC). Each table corresponds to the results from a specific dataset. Bold indicates the best performance.	74
4.2	Table of Regression and Classification results from the participants of the FishAI competition. Abbreviated metrics are Mean Square Error (MSE), Root-Mean-Squared Error (RMSE), Mean Absolute Error (MAE), MCC. Best results in bold.	86

List of Tables

Acronyms

AI Artificial Intelligence. 2, 6, 11, 13, 32, 44, 46, 47, 60, 98, 102

AIS Automatic Identification System. x, 3, 6, 26, 27, 29, 38–40, 44, 45, 52, 61–63, 75–79, 81, 83, 86, 92, 93, 95

CNN Convolutional Neural Network. ix, 35, 36, 93

CPU Central Processing Unit. 58

CSG Cyber Security Group. 10

DAG Directed Acyclic Graph. 16, 50, 60, 91

DCA Daily Catch Activity. 63

DEP Reports on Departure. 63

ECHR European Convention on Human Rights. 98

EEA European Economic Area. 97

EEZ Exclusive Economic Zone. 44

EU European Union. 97, 98

EW Extra-Wide Swath. 78

FPS Frames Per Second. 66

Acronyms

GDPR General Data Protection Regulation. 97, 98, 102

GFW Global Fishing Watch. ix, 38, 44

IoT Internet of Things. 42

IUU Illegal, Unreported and Unregulated. 5, 6, 20, 23–26, 44, 97, 99

IW Interferometric Wide Swath. 78

MAE Mean Absolute Error. xi, 86

MCC Matthews Correlation Coefficient. xi, 74, 86

MMSI Maritime Mobile Service Identity. 62, 63

MSC Marine Stewardship Council. 27

MSE Mean Square Error. xi, 86

NORA Norwegian Artificial Intelligence Research Consortium. 85

ONR Official Norwegian Report. 26, 28–32, 47

POR Reports on Landing. 63

RMSE Root-Mean-Squared Error. xi, 86

SAR Synthetic Aperture Radar. x, 38–40, 63, 64, 76–79

SGX Software Guard Extensions. 10

SOG Speed Over Ground. 62

TPM Trusted Platform Module. 10, 58

Chapter 1

Introduction

Fish is an important source of protein and other nutrients for many people, and fishing and fish farming provide nutrition to millions of people worldwide. The United Nations' Sustainability Development Goal 14 is to “conserve and sustainably use the oceans, seas, and marine resources for sustainable development” [124]. Overfishing is a major problem facing our oceans today, and it is important to take steps to ensure that future generations will be able to enjoy the bounty of the sea. In Norway, fishing is the second biggest export and accounted for over 14 billion USD in value in 2021 [115]. Therefore, it is clear that commercial fishing needs to be controlled in a competent and sustainable manner. However, there are two major issues in the fishing domain that need to be tackled in the pursuit of resource control. First, there are inaccuracies in fishers' catch documentation, either due to bad estimations of how much they catch or deliberate manipulation of estimates to hide how much they have caught. Second, inspection of fishing vessels happens during landing and, occasionally, at sea by the coast guard, and only affects approximately 4% of the fishing fleet, due to lack of manpower and knowledge of where to send inspectors [79]. Assisting and ensuring that fishers report the correct amount of fish, and bring bycatch to land has to be achieved in order to subdue the large amount of dark numbers in the fishing industry.

Due to the difficulties of controlling what happens on fishing vessels at sea, video surveillance has been presented as a potential solution [15, 79, 95], however the manpower needed to manually inspect all video footage does not scale, nor does it take the fishers' privacy into account. Artificial Intelligence (AI) can facilitate the analysis of such video data, but one needs to consider how these models are to be trained, given the lack of relevant data.

In this thesis, we want to leverage the potential of AI, together with an edge-based distributed system, to investigate if it is possible to develop a privacy-preserving distributed system for automatic documentation of catch and detection of illegal activities in the fisheries domain. We explore a multitude of data sources, both on the fishing vessel (edge) and what is accumulated on land. We work with people with domain knowledge, such as fishers and inspectors, to collect data and distinguish what is needed by the industry.

1.1 Background and Motivation

Global fisheries contribute to food security for millions of people worldwide. It is estimated that the sea produces 17% of the current production of edible meat globally, and that this percentage will increase dramatically as the world's population grows [16]. Capture fisheries play a pivotal role for guaranteeing food security from the sea, but illegal fishing and over-exploitation are widespread problems that negatively impact the sustainability of wild fish stocks. Examples of such illegal activities include fishing vessels that operate without valid licenses, fishing on stocks that are depleted and close to extinction, fishing with illegal gears, and negligence in submitting correct catch data. These problems occur globally, but are perhaps particularly prominent around the poorer coastal countries that cannot launch effective countermeasures to combat such organised illegal activities.



Økokrim: Fiskerikriminalitet en av de største truslene

Økokrim mener fiskerikriminalitet er en av de største truslene mot velferdsstaten i årets trusselvurdering. Nå vil regjeringen ta grep.

Figure 1.1: A Norwegian newspaper article about the fish crime being ranked as one of the biggest threats to welfare state [11].

In some geographical areas, like the northern Atlantic, evidence-based regulatory efforts have been shown to be efficient for maintaining sustainable harvesting of fish populations. A requirement, though, is that such regulations are combined with thorough fishing vessel monitoring, surveillance, and control. This includes, for instance, requiring vessels to have Automatic Identification System (AIS) that tracks position installed, maintain and submit logs to the authorities when fish gear is launched, when catch is landed on deck, and what species and volumes were caught, and where the catch will be delivered for further processing. Fishing vessels are also subject to an inspection at any time and without notification, both while at

sea or when docking, by either the coast guard or other public inspectors.

Though such operational control regimes make cheating difficult, they are far from flawless. Failures to report a catch is one example of how criminals can cheat, another way to cheat can be done by under-reporting of fish quantities landed. The monetary gain for such illegal activities can be relatively profitable; the UN estimates that illegal fishing activities amount to figures in the order of billions USD a year, and involves corruption and other financial crimes, such as large-scale tax evasion and money-laundering [24].

Some argue that digital surveillance technologies can mitigate fishery crime problems. Automatic 24/7 video-surveillance systems and sensors on board the fishing vessels at sea are particularly touted as a technological solution [15, 95]. The Danish Fisheries Authority already equips tenfold fishing vessels from the country's fleet with such equipment, a mutually agreed decision with volunteering fishermen. Norway is also planning mandatory deployment of similar surveillance cameras as one of several remedies to manage, control, and combat illegal fishing activities [79].

At a first glance, such video surveillance might sound like a plausible approach to combat crimes, but it comes with a significant disadvantage with regard to privacy. Suddenly, somebody can be watching every fisherman while working on deck, which can be considered personally invasive. The proportionality law principle is also at stake striking a balance between capturing potential illicit behaviour versus intruding on personal spheres of tenfold thousand of fishermen at work [35]. Workers expect certain guarantees if video surveillance is to be used; workers expect surveillance to be *transparent* with regard to placement of cameras and why they are used, they expect to know who has *access* to the footage and that it is securely stored, and they want to ensure *equality*, i.e., there should not be more data collected of a certain group of people compared to other groups. There are also challenges with implementing video surveillance in a work place, such as adequate coverage over the space of work, while at the same time, not

intruding in areas where one expects privacy [66]. In addition, the analysis of video data in combination with other sensor data and external data is a challenge by itself. Combining different modalities for analysis comes with challenges regarding how to combine them, how to determine the importance of the different input sources, etc. Scale and performance are obvious technical problems considering that fishing vessels in the order of thousands will have to transfer voluminous multimedia data over satellite links or radio-based technologies to mainland operational control centers. Additionally, the tedious and labor-intensive task if this data is to be inspected in real-time by human operators is overwhelming.

1.2 Thesis Statement

We aim to provide a system for handling the issues outlined in Section 2.1 in a privacy-preserving manner. Our goal is then to prevent Illegal, Unreported and Unregulated (IUU) fishing, while at the same time provide automatic documentation of catch for fishers. This has to be done in a way that does not necessitate manual inspection of videos from fishing vessels. However, we need a method for providing evidence that can aid the decision-making process done by inspectors on land. The automatic documentation of catch is an additional incentive for fishers to utilise such a method. Thus, a “digital inspector”, in place of an actual inspector, has to be devised. This digital inspector has to be able to detect illegal activity on the fishing vessel and warn control authorities on land by sending evidence.

The main hypothesis of this thesis is therefore:

It is possible to develop a digital inspector that can be situated on fishing vessels for automatic documentation of catch and privacy-preserving surveillance in cases of fish fraud.

The goal, then, is to determine whether it is possible design and implement a system that uses sensors on fishing vessels to assist in catch

documentation and detect potential illegal activities, combined with data that is being sent and can be processed on land, such as AIS, sales notes, etc.

From this research question, the objectives targeted by this dissertation are as follows (also illustrated in Figure 1.2):

Main Objective: Research and develop a distributed edge-based system prototype for privacy-preserving analysis of sensor data on fishing vessels, that can assist fishers by automatically documenting their catch and detect potential illegal activity on-board the fishing vessel. Anonymised data and local analysis results will then be sent to the mainland to facilitate the decision making for inspectors.

Sub-objective 1: Develop an edge-based subsystem that can assist in automatic documentation and detect fish fraud on the fishing vessel. This requires a system that can process multiple sources of data using resource-intensive AI algorithms.

Sub-objective 2: Develop a subsystem on the mainland which can leverage data that is related to fishing and data being sent from the fishing vessels. This data includes AIS, catch sales notes, satellite data, etc. This system will need to combine and correlate these data sources with data sent from the fishing vessels to be utilised in a decision-making procedure which decides which vessels that should be scheduled for inspection.

Sub-objective 3: Relevant data on both the fishing vessel and mainland needs to be evaluated and potentially acquired. We therefore need to work closely with e.g. fishers and fishing sales organisations in order to fully understand what data is most useful to analyse in order to assist in automatic documentation of catch and detection of IUU fishing.

Sub-objective 4: Research and develop analytical approaches for handling the multitude of varied sources of data in the fishing domain. This includes both evaluating which existing methods can be used on data from the fishing domain, and developing new algorithms to deal with e.g. the lack of data or unusual scenarios without analogous data from other domains, etc.

1.3 Scope and Limitations

Based on the objectives outlined in Section 1.2, the scope of this thesis is on researching and providing the foundation for a system which leverages multiple data sources to both assist fishers' catch documentation and detect possible fish fraud when it occurs. This needs to be done in a privacy-preserving fashion to uphold the privacy rights of the crew and minimise the intrusion in the workspace. We explore similarities between scenarios from the fishing domain and scenarios from other domains to determine whether we can utilise already existing methods or have to develop new approaches.

We focus mostly on the problem of action recognition and event detection for the purposes of detecting illegal actions in the fishing domain, as this seems to be the harder problem to tackle. However, in certain works we discuss how catch analysis can be accomplished, e.g. through utilising existing fish datasets for detecting species and estimating biomass. For fish fraud detection, data is more scarce/non-existent, publicly at least. Therefore, we need to explore the data already available that can be used, and we investigate and collect new data that can be used for our purposes.

We limit the scope of the thesis by mostly focusing on the sub-tasks and their corresponding subsystems separately. As such, a final prototype that joins all of the sub-systems is future work. Furthermore, we limit the application of the system to large-scale fishing trawlers that already have the

computational power or the space for such a system on-board, though the power requirements are still limited to commercial hardware requirements.

1.4 Research Methodology

The Association for Computing Machinery (ACM) Education Board approved and endorsed a report from the Task Force on the Core of Computer Science, for release in 1989 [23]. The task force puts forward a novel intellectual framework that determines the criteria, discipline, and norms of computing and the basis upon which the computing curricula can be based. Computing is defined as a intersection between applied mathematics, science, and engineering. All of these three processes are essential in the discipline. Computer science combines the processes into the paradigms of (i) theory, (ii) abstraction, and (iii) design (specific).

In this dissertation, our work is related to these topics in several ways. In the below section, we will describe each process and discuss how our dissertation covers these topics.

Theory Theory is rooted in mathematical aspects and involves the development of valid theory. The report identifies the following theory steps as (i) characterise objects of study (definition), (ii) hypothesise possible relationships among them (theorem), (iii) determine whether the relationships are true (proof), and (iv) interpret results.

For the theoretical part, we apply image processing on 2D geometries in both image and video data, natural language processing on textual data, and apply tabular data analysis as well. These modalities are at times merged for use in filtering results or need to be weighted against each other, which requires knowledge about their importance.

Abstraction Abstraction process is used for modeling and is directly related to the experimental scientific method. The report describes the ab-

straction process through the following steps: (i) form a hypothesis, (ii) conduct a model and make a prediction, (iii) design an experiment and collect data, and (iv) analyse results.

In our work, we have performed several experiments to verify our hypotheses. Some experiments demonstrate that a system design choice is valid, and others illustrate the viability of a specific machine learning-based model and its generalisability to domains outside of fishing. We explore a multitude of different data sources to evaluate what data is most useful depending on the task. The different sub-systems have requirements that are tested against to ensure that they are upheld.

Design The report describes the design into four steps: (i) state requirements, (ii) state specifications, (iii) design and implement the system, and (iv) test the system.

In order to develop a system for automatic catch analysis and privacy-preserving detection of illegal activities in the fishing domain, we have had to design and implement multiple components of the overarching system. Designing and testing these components required domain knowledge with regard to intercommunication between these components, what data is needed, and how it needs to be handled.

In addition to the research methodology outlined above, there are also several research methods specific to machine learning and AI [63] that we will apply in our research. These methods include data collection and pre-processing, feature selection and engineering, model selection and tuning, and evaluation of model performance. For data collection and preprocessing, we will gather data from fishing vessels or data which is federated to the mainland. Feature selection and engineering will involve identifying the most important features for our machine learning models, as well as creating new features that may improve performance. Model selection and tuning will involve experimenting with various machine learning models and their

hyperparameters to identify the best model for the task. Finally, we will evaluate our models' performance using various metrics, such as precision, recall, and F1 score, if applicable. Through the application of these research methods, we aim to develop a privacy-preserving, edge-based system that can help promote sustainable fishing practices while protecting the rights and privacy of fishing crews.

1.5 Research Context

The research of this dissertation was conducted in the context of the Cyber Security Group (CSG) and the Corporo Sano research group. Part of the research has also been conducted as part of the BBChain project in collaboration with the University of Stavanger. The CSG focuses on investigating fundamental systems problems in practical application domains, such as soccer, healthcare, and recently, fishing.

The fishing domain and use of distributed AI solutions represents an area the group is familiar with. Over three decades ago, StormCast [48, 59, 60, 61], probably the first sensor network prototype ever with distributed expert-systems built for trawlers and coast guard ships connected through satellite communication links, was built. StormCast predicted suddenly erupting hazardous weather situations at sea over large areas. A mainland public version of StormCast has been continuously operational since 1993¹.

In more recent years, there have been multiple works related to managing large quantities of personal digital data [47], such as video data, and how to handle privacy [62]. Runtimes using Trusted Platform Modules (TPMs) have also been developed using Software Guard Extensions (SGX) [46] and TrustZone [98]. TPMs allow for application isolation and attestation using special hardware. This can be used to increase data privacy and confidentiality.

¹<https://weather.cs.uit.no/>

In 2021, the Dutkat project started as a larger multi-disciplinary project with computer scientists and legal scholars, with a goal of building a system for enforcing sustainable fishing. We chose a compliance-by-design approach to ensure that the privacy and data rights of the fishers represent a focus from the beginning. There are different people working on different components and aspects required for the system to function, such as the storage solution [91] and the computational framework [3].

1.6 Contributions

The summarised main contributions of the thesis are:

- Design of Dutkat, an edge-based distributed system for privacy-preserving catch documentation and detection of illegal activities.
- Development and prototyping of components of the Dutkat system and evaluate if they are needed through experimentation.
- Exploration of existing data relevant to the thesis and collection of new data from unique sources that was release openly.
- Organisation of multiple AI competitions for finding solutions that lead to sustainable fishing and privacy preservation of fishing crews.
- Design of components of the Dutkat system through a compliance-by-design perspective, and discussions of relevant legal requirements.

This dissertation is a collection of the publications below. We will give brief summaries of them. Furthermore, additional publications which are not directly relevant to the dissertation are also listed. A legend of the full names corresponding to the initials is given at the bottom.

In Figure 1.2, we illustrate how the main publications relate to the objectives described in Section 1.2.

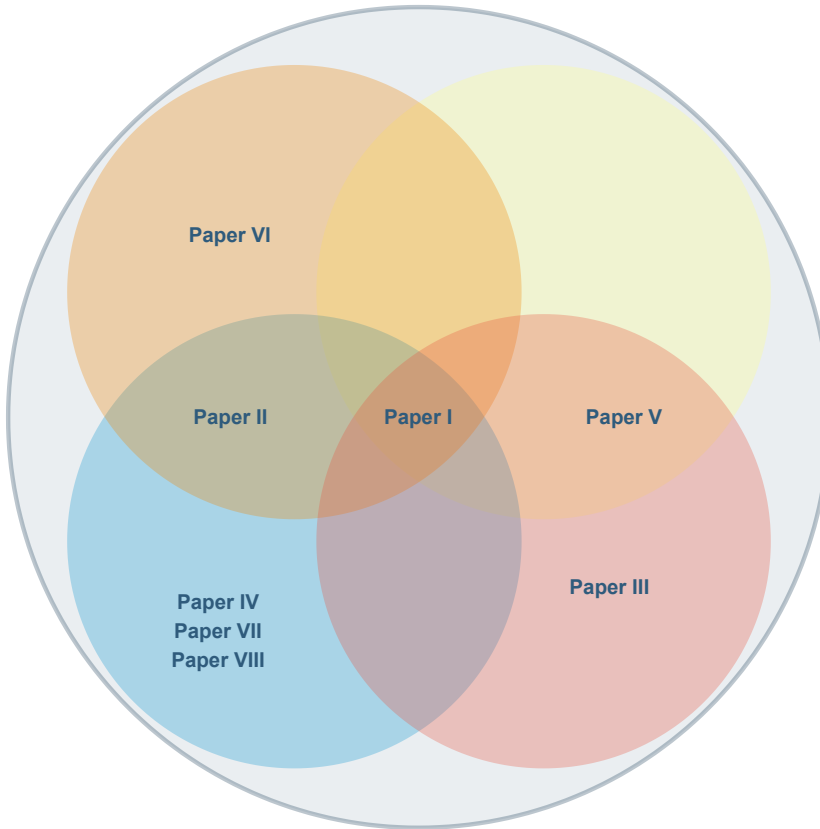
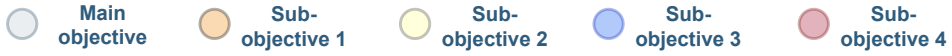


Figure 1.2: Illustration showing relation between contributions and thesis objectives.

1.6.1 Publication I

Tor-Arne S. Nordmo et al. “Dutkat: A Multimedia System for Catching Illegal Catchers in a Privacy-Preserving Manner”. In: *Proceedings of the 2021 Workshop on Intelligent Cross-Data Analysis and Retrieval*. ICDAR '21. Taipei, Taiwan: Association for Computing Machinery, 2021, pp. 57–61. ISBN: 9781450385299

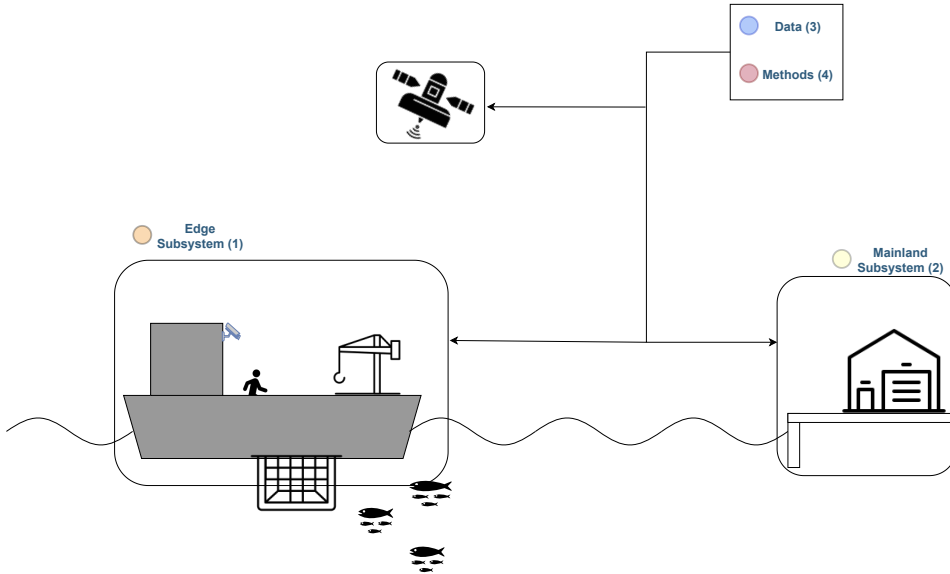


Figure 1.3: Overview of how the objectives relate to the different parts of the system. The objectives are color-coded in the same fashion as in Figure 1.2. The edge subsystem is to be situated on the fishing vessel and the mainland subsystem on the mainland. The Data and Methods sub-objectives refer to or utilise data generated on the fishing vessel or via satellites, or is data that is federated to the mainland.

This paper introduces the design of the Dutkat system and its generalised distributed edge-based system. It also describes the problem of fraud in the fishing industry, proposed solutions, and issues with these solutions. The Dutkat system is presented as a privacy-preserving solution that utilises AI to detect illegal activity locally on the fishing vessel. It is also described as a system for automatic documentation of catch. As this was the preliminary design of the Dutkat system, it contributes to all the objectives of the thesis.

Author contributions (initials)

Conceptualisation: T-A.S.N., D.J., M.A.R.; **Data collection:** T-A.S.N., M.A.R., **Methods, data analysis and interpretation:** T-A.S.N., M.A.R., **Drafting:** T-A.S.N., M.A.R., A.B.O., H.D.J, P.H., D.J., **Critical revision:** T-A.S.N., M.A.R., A.B.O., H.D.J, P.H., D.J.

1.6.2 Publication II

Aril Bernhard Ovesen et al. “File System Support for Privacy-Preserving Analysis and Forensics in Low-Bandwidth Edge Environments”. In: *Information* 12.10 (2021). ISSN: 2078-2489. DOI: 10.3390/info12100430. URL: <https://www.mdpi.com/2078-2489/12/10/430>

This work introduces Dorvu, a geo-distributed file system with support for fine-grained access control policies and software modules. This distributed file system is a component of the Dutkat edge subsystem. In the paper, we also demonstrate the infeasibility of sending video data over satellite for real-time analysis, even when reducing resolution and frame rate. This paper contributes to the first sub-objective of the thesis.

Author contributions (initials)

Conceptualisation: A.B.O, D.J.; **Data collection:** A.B.O., T-A.S.N., **Methods, data analysis and interpretation:** A.B.O., T-A.S.N., **Drafting:** A.B.O., T-A.S.N., M.A.R., H.D.J, P.H., D.J., **Critical revision:** T-A.S.N., M.A.R., A.B.O., H.D.J, P.H., D.J.

1.6.3 Publication III

Tor-Arne Schmidt Nordmo et al. “Fishing Trawler Event Detection: An important step towards digitization of sustainable fishing”. In: *2023 International Conference on Applied Artificial Intelligence (ICAPAI)*. 2023, pp. 1–8

This work presents a machine learning-based pipeline for real-time unsupervised anomaly detection on multimodal data streams. The method consists of three steps: (i) feature extraction, (ii) an optional embedding layer, and (iii) a moving average-based anomaly detection method. We

evaluate our method on three distinct, well-known datasets with labels, and an additional unlabeled dataset (by manual inspection). The method outperforms simple baselines on all labeled datasets. This paper contributes to the fourth sub-objective of the thesis.

Author contributions (initials)

Conceptualisation: T-A.S.N., M.A.R.; **Data collection:** T-A.S.N., **Methods, data analysis and interpretation:** T-A.S.N., **Drafting:** T-A.S.N., M.A.R., A.B.O., H.D.J, P.H., D.J., **Critical revision:** T-A.S.N., M.A.R., A.B.O., H.D.J, P.H., D.J.

1.6.4 Publication IV

Tor-Arne Schmidt Nordmo et al. “Njord: A Fishing Trawler Dataset”. In: MMSys ’22. Athlone, Ireland: Association for Computing Machinery, 2022, pp. 1–6. ISBN: 9781450384346. DOI: 10.1145/3458305.3463373. URL: <https://doi.org/10.1145/3458305.3463373>

This paper presents the Njord dataset, a dataset consisting of surveillance videos from an off-shore fishing trawler. 71 of the 198 videos are richly annotated with both bounding box and classification labels. In the paper we also provide a baseline analysis and discuss possible research questions Njord could help answer. This paper contributes to the third sub-objective of the thesis.

Author contributions (initials)

Conceptualisation: T-A.S.N., D.J., M.A.R.; **Data collection:** T-A.S.N., A.B.O., B.A.J., S.A.H., V.T., **Methods, data analysis and interpretation:** T-A.S.N., S.A.H., V.T., **Drafting:** T-A.S.N., S.A.H., V.T., A.B.O., H.D.J, P.H., M.A.R., D.J., **Critical revision:** T-A.S.N., S.A.H., V.T., A.B.O., H.D.J, P.H., M.A.R., D.J.

1.6.5 Publication V

T. S. Nordmo et al. “Detection of Commercial Fishing-related Slipping Events using Multimodal Data”. In: *2022 IEEE International Symposium on Multimedia (ISM)*. Los Alamitos, CA, USA: IEEE Computer Society, Dec. 2022, pp. 155–156. DOI: 10.1109/ISM55400.2022.00032. URL: <https://doi.ieeecomputersociety.org/10.1109/ISM55400.2022.00032>

This work proposes a method for combining and analysing multiple sources of data to detect slipping events, i.e., deliberate release of dead or dying fish. We utilise positional data from fishing vessels, sales notes, oil slick detection, and vessel detection from satellite imagery. The paper also presents the legal aspects of slipping events. This paper contributes to the second, third and fourth sub-objectives of the thesis.

Author contributions (initials)

Conceptualisation: T-A.S.N., D.J.; **Data collection:** T-A.S.N., M.M.E., **Methods, data analysis and interpretation:** T-A.S.N., M.M.E., **Drafting:** T-A.S.N., M.M.E., B.A.J., M.A.R., D.J., **Critical revision:** T-A.S.N., M.M.E., B.A.J., M.A.R., D.J.

1.6.6 Publication VI

Joakim Aalstad Alslie et al. “Aika: A Distributed Edge System for AI Inference”. In: *Big Data and Cognitive Computing 6.2* (2022). ISSN: 2504-2289. DOI: 10.3390/bdcc6020068. URL: <https://www.mdpi.com/2504-2289/6/2/68>

This paper presents Áika, a robust distributed machine learning inference system that can be placed onboard fishing vessels. It supports a Directed Acyclic Graph (DAG) computational structure composed of agents

and controllers, while being tolerant to failures. The system is evaluated against agent crashes and tested on distributed workloads. This paper contributes to the first sub-objective of the thesis.

Author contributions (initials)

Conceptualisation: J.A.A., D.J., A.B.O., T-A.S.N.; **Data collection:** J.A.A., **Methods, data analysis and interpretation:** J.A.A., **Drafting:** J.A.A., A.B.O., T-A.S.N., H.D.J, P.H., M.A.R., D.J., **Critical revision:** J.A.A., A.B.O., T-A.S.N., H.D.J, P.H., M.A.R., D.J.

1.6.7 Publication VII

Tor-Arne Schmidt Nordmo et al. “FishAI: Sustainable Commercial Fishing”. In: *Nordic Machine Intelligence 2.2* (2022), pp. 1–3

This short paper presents the FishAI challenge, the second challenge of the Nordic AI Meet. It consists of three different tasks that are to be solved based on multiple public datasets that are provided. The paper gives an overview of the data and tasks. This paper contributes to the third sub-objective of the thesis.

Author contributions (initials)

Conceptualisation: T-A.S.N., M.A.R.; **Data collection:** T-A.S.N., **Methods, data analysis and interpretation:** T-A.S.N., M.A.R., **Drafting:** T-A.S.N., O.K., S.O.K., B.H., S.A.H., H.D.J, P.H., M.A.R., D.J., **Critical revision:** T-A.S.N., O.K., S.O.K., B.H., S.A.H., H.D.J, P.H., M.A.R., D.J.

1.6.8 Publication VIII

Tor-Arne S. Nordmo et al. “NjordVid: A Fishing Trawler Video Analytics Task”. In: *Proceedings of CEUR Multimedia Benchmark Workshop (MediaEval)*. 2022

This short paper presents the NjordVid task of the MediaEval Benchmark. It consists of two different tasks that are to be solved based on the Njord dataset described in paper IV. The paper gives an overview of the data and tasks. This paper contributes to the third sub-objective of the thesis.

Author contributions (initials)

Conceptualisation: T-A.S.N., M.A.R.; **Data collection:** T-A.S.N., **Methods, data analysis and interpretation:** T-A.S.N., **Drafting:** T-A.S.N., A.B.O., H.D.J, M.A.R., D.J., **Critical revision:** T-A.S.N., A.B.O., H.D.J, M.A.R., D.J.

1.6.9 Author Legend

T-A.S.N.: Tor-Arne S. Nordmo, J.A.A.: Joakim A. Alslie, M.E.E.: Martine E. Espeseth, B.H.: Birte Hansen, P.H.: Pål Halvorsen, S.A.H.: Steven A. Hicks, D.J.: Dag Johansen, H.D.J.: Håvard D. Johansen, B.A.J.: Bjørn A. Juliussen, O.K.: Ove Kvalsvik, S.O.K.: Svein O. Kvalsund, A.B.O.: Aril B. Ovesen, M.A.R.: Michael A. Riegler, V.T.: Vajira Thambawita

1.6.10 Additional Contributions

In addition to the main paper contributions outlined above, there have been several additional contributions that have emerged throughout the research of the thesis. These contributions are not directly related to the main focus of the thesis, but are connected due to the methods applied and the knowledge gained during the research process. These additional contributions demonstrate my ability to extend my acquired knowledge and apply it outside of the specific area of my Ph.D. project.

- Enrico Tedeschi et al. “Predicting Transaction Latency with Deep Learning in Proof-of-Work Blockchains”. In: *2019 IEEE International Conference on Big Data (Big Data)*. 2019, pp. 4223–4231. DOI: 10.1109/BigData47090.2019.9006228

In this paper, we present a deep learning model for predicting whether a given incoming transaction will be included in a block in the Bitcoin blockchain. This can prevent users of Bitcoin from overpaying their fee to the miner.

- Enrico Tedeschi et al. “On Optimizing Transaction Fees in Bitcoin Using AI: Investigation on Miners Inclusion Pattern”. In: *ACM Trans. Internet Technol.* (Mar. 2022). ISSN: 1533-5399. DOI: 10.1145/3528669. URL: <https://doi.org/10.1145/3528669>

This paper further develops ideas from [117], and engineers new features that are related to revenue and fairness in the Bitcoin blockchain. By assuming that miners are rational, i.e. want to maximise their profits, we develop new features that allow our model to achieve an accuracy of 91%.

- André Pedersen et al. *Hybrid guiding: A multi-resolution refinement approach for semantic segmentation of gigapixel histopathological images*. 2022. DOI: 10.48550/ARXIV.2112.03455. URL: <https://arxiv.org/abs/2112.03455>

In this work, we propose a cascaded convolutional neural network design for semantic segmentation of gigapixel histopathological images. It consists of a patch-wise method, followed by a refinement network. Patches are sampled in a hierarchical manner to ensure the model has a balanced view of the data.

- Tor-Arne Schmidt Nordmo et al. “Arctic HARE: A Machine Learning-based System for Performance Analysis of Cross-country Skiers”. In: *Proceedings of the 29th ACM International Conference on Multimedia Modeling*. MMM '23. Bergen, Norway: Association for Computing Machinery, 2022

In this paper, we present Arctic HARE, a machine learning-based skiing-technique training system on the edge. On-body sensors are compared to video-based approaches for classifying ski techniques. We achieve higher than 96% accuracy.

1.7 Outline

This dissertation is structured as follows:

Chapter 2 In this chapter, we give an overview of the problems and proposed solutions with regard to IUU fishing and verification in the Norwegian fishing industry. We also give a brief overview of relevant machine learning concepts and present related work in the areas of distributed edge-based systems and machine learning.

Chapter 3 In this chapter, we present the functional and non-functional requirements for our system, including requirements for both the edge computational node and the mainland hub. We discuss the different data sources that need to be considered and how this data needs to be processed, stored, and displayed, taking into account the unique requirements of each component of the system.

Chapter 4 We describe Dutkat, the edge-based system for privacy-preserving catch documentation and detection of illegal activities. We describe the components of the system and the data sources we have considered in our research. Then, we present analytical methods and results we

have utilised/developed. We also present the different competitions that have been organised as part of the research of this thesis.

Chapter 5 Here, we discuss our contributions to each objective. Then, we discuss traceability, how the system is generalisable to multiple use-cases, lessons learned throughout our research, and ethical and legal considerations.

Chapter 6 Finally, the concluding chapter of the thesis provides a summary of the key findings and contributions of the research and also presents directions for future work and potential areas for improvement.

Chapter 1. Introduction

Chapter 2

Background and Related Work

In this chapter, we will provide an overview of the problems covered in this thesis and provide a literature review to discuss other approaches and solutions to the sub-problems that arise in IUU fishing or similar scenarios. The first section presents the issues of IUU fishing and specifically weaknesses within sustainable fishing in Norway, followed by proposed solutions. We discuss potential problems of these solutions that need to be addressed. Then, we give an overview of the field of machine learning, going more deeply into relevant methods that are applied in our use case. Finally, we discuss related work in the areas of distributed edge-based systems and how machine learning has been applied in the fishing domain or similar scenarios previously.

2.1 Illegal, Unregulated and Unreported Fishing

IUU Fishing is a global problem that affects the marine ecosystems and a crucial food source for humans. Fish account for almost 20 percent of the total animal protein intake [36]. As can be seen in Figure 2.1, a large

proportion of exploited fish stocks has been classified as overfished for the past two decades, and the trend is rising. IUU fishing can have serious negative impacts on fish stocks, ecosystems, and the people who depend on them. It often takes place in areas that are already overfished, and can undermine efforts to rebuild fish stocks and promote sustainable fisheries. IUU fishing also undermines the rule of law, and creates unfair competition for those who fish legally and operate their businesses within the bounds of the law [74].

IUU fishing consists of three parts [37]:

Illegal Fishing which consists of fishing in the jurisdiction of a State without permission or violation of national or international obligations within the given region.

Unreported Fishing refers to fishing that goes unreported or misreported to the relevant national authority.

Unregulated Fishing is when conservation and management regulations are not taken into consideration or explicitly contravened by a fishing vessel.

There are a number of reasons why IUU fishing takes place. In some cases, it is the result of poor governance and weak enforcement of fisheries laws. In other cases, it is deliberate and organised crime, driven by the profit motive. IUU fishing often takes advantage of the fact that it is difficult to monitor and regulate fishing activity in the open ocean. Therefore, it is hard to detect and prosecute those engaged in IUU fishing. IUU fishing is a serious problem that needs to be addressed urgently. It threatens the sustainability of fish stocks, the livelihoods of those who depend on fishing, and the rule of law. Strong action is needed to tackle this problem, and to ensure that fishing is managed in a way that is sustainable and consistent with the rules and regulations set by regional, national or international authorities [73, 97].

2.1. Illegal, Unregulated and Unreported Fishing

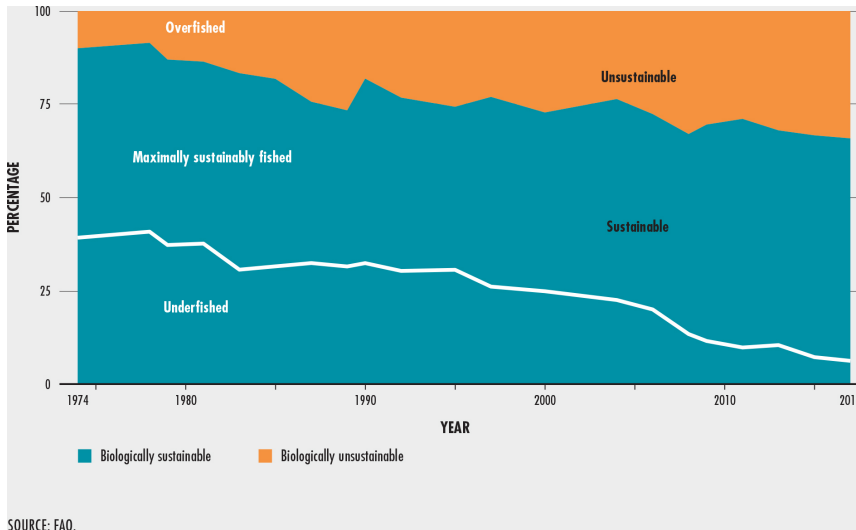


Figure 2.1: Global trends of the world's fish stocks [36].

It is very difficult to regulate the fishing industry, because detecting, locating, and apprehending boats performing such acts on the high seas is challenging without implementing new monitoring and surveillance solutions [37]. The economic incentives of IUU fishing, combined with the low probability of being caught, make it extremely difficult to effectively combat IUU fishing. This exploitation raises important issues of fairness and equity relating to the sustainable management of world fisheries. However, the problem of IUU fishing is complex, and there is no easy solution. A number of measures have been proposed to address it. These include strengthening fisheries management and enforcement, improving data collection and monitoring, and increasing international cooperation.

The aims of UN Sustainable Development Goal 14 is to help preserve and sustainably use the world's oceans, seas, and marine resources. This is important because the oceans play a vital role in the Earth's ecosystems and support the livelihoods of billions of people. The oceans are also a major source of food and energy, and are a key part of the global economy. Conserving and sustainably using the world's oceans, seas, and marine resources

is essential to protecting the environment and ensuring the long-term health of the planet [124]. Specifically, sub-goal 4 is relevant for IUU fishing, where they aspire to effectively regulate the harvesting of fish and end overfishing. IUU fishing should be ended and science-based management plans should be put in place to restore fish stocks to sustainable yields.

This dissertation focuses on the problems outlined in the Official Norwegian Report (ONR), *Framtidens Fiskerikontroll (The Future of Sustainable Fisheries)* from 2019 [79]. These problems mainly consist of the lack of verifiable data, weaknesses in laws pertaining to fisheries, and difficulty in organisation of control and inspections. The ONR also presents some solutions to these problems, focusing on technological and legal solutions. We will go over some of these problems and solutions.

2.1.1 Problems

Verifiable Data

According to the ONR, lack of verifiable data with regard to capture, processing, and storage is the greatest challenge for enforcement of control in the fishing industry. Today, fishers in Norway are required to log and submit certain data during fishing and landing of the catch. The types of data depend on the size of the vessel, AIS is, for example, not required for fishing vessels under 15 meters in length. An overarching issue with the data that is required to be submitted by fishers is that it is often self-reported based on manual inputs. This issue is specifically prominent with smaller vessels that do not have the ability to e.g. weigh the fish while at sea, and the production vessels, which catch can be transferred to by other vessels [79].

There have been multiple cases of cheating data in the fishing industry. Larger fishing vessels are required to report their location via AIS [71], but due to the fact that catch is reported manually, the correlation between the path of the vessel and where they claim they have fished can be manufac-

2.1. Illegal, Unregulated and Unreported Fishing

tured [56]. As part of the “Cod offensive 2022”, the Norwegian Directorate of Fisheries detected discrepancies between the AIS and the sales notes of three fishing vessels of approximately 10 meters in length. The fishing vessels had reported to have fished outside of the 12 nautical mile area around the coast of Norway, but they had fished inside instead. The incentive for this is assumed to be related to Marine Stewardship Council (MSC) certification, where MSC-certified fish is more desirable by buyers [29].



Figure 2.2: Discard of dead fish by FV Margiris, the second biggest fishing vessel on earth. Documented by Sea Shepard France [122].

Another event that is difficult to verify, and thus prevent, is discard of catch. Section 15 of the Norwegian act relating to the management of wild living marine resources requires that all catches of fish shall be landed, with a few exemptions. Fishing vessels may only discard fish if it is viable to continue living. Despite this, there have been multiple cases of discard has been detected by the coast guard. In February 2022, over 100,000 dead fish were discarded by the second biggest fishing vessel outside of France (see Figure 2.2). A representative of the vessel’s owners claimed it was due to a net break, however Sea Shepard France believe it was an intentional discard

due to undesired type of fish [122]. Problems like these could be detected with an onboard system for catch documentation and surveillance.

Legal Aspects

The ONR also describes current challenges with regard to laws pertaining to the fishing industry. The laws are either described as partially unclear or convoluted with regard to multiple layers of special considerations that need to be taken into account. Due to this difficulty, it can even be tedious to figure out how big the quota of a given vessel can be. Certain fishers also get exemptions, which lead to a perception of unequal treatment.

Specifically with cod caught in the winter season, control authorities note that there are strong indicators for weight of the catch being under-reported. This is due to the fact that there are difference between fishing vessels with regard to whether they can process the fish before it is brought to land. Some fishers claim there is an unfair advantage in having the ability to process the fish at sea. The legislation allows for a “dynamic factor” to recalculate the weight of the catch. This “dynamic factor” can vary depending on the fisher, often resulting in under-reporting [79].

Organisation

Due to the lack of verifiable data, physical inspections and control have been seen as the only option for uncovering potential illegal activities. This is extremely limiting and requires immense resources. The ONR report shows that few serious infractions are detected, and the consequences are not disincentivising enough. The lack of verifiable data combined with occasionally low quality of sales organisations’ internal data and a lack of tools for sharing this data between authorities and sales organisations make organisation of sustainable fishing very difficult. For instance, there does not exist any singular national register of fishing quotas. Each sales organisation has needed to develop their own solutions. Quota calculations can also be

2.1. Illegal, Unregulated and Unreported Fishing

complicated and therefore require manual recalculations and updates [79]. In the few years since 2019, national access to parts of catch and activity data from the fishing fleet has been provided.

2.1.2 Proposed Solutions

The solutions proposed in the ONR entail a goal to realise an automatic catch documentation system that can provide verifiable catch data.

Surveillance

In the ONR they propose to increase digitalisation and openness in the fishing industry. They want to make it easier to share data between private and public sources and introduce sensors on board the fishing vessels. Video surveillance is discussed in the ONR as one of the potential solutions for verifying the actions taken on the fishing vessel. This could prevent, for instance, the discard of unwanted catch (e.g. due to undesirable size of catch or bycatch) and has been tested for in for example Scotland, Canada, USA, and Denmark. Video surveillance could also serve as alternative to having an observer onboard. Video cameras can either be used for documenting what is happening on deck or documenting the production process of cutting and sorting the fish. Some larger fishing vessels have already introduced surveillance systems, like what can be seen in Figure 2.3. However, multiple potential issues arise from such a solution.

Introducing video cameras aboard fishing vessels will impede on the fishers' privacy. The employers will be able to see everything that the workers are doing, and this can lead to a feeling of being constantly watched. This can be very stressful for some workers, and can lead to a decrease in productivity. Additionally, if the footage from the cameras is leaked, it could embarrass or humiliate the workers who are caught on camera. Another issue with video data is transferring it to land. AIS data is already transmitted using satellite communication, however the ONR (and our own



Figure 2.3: Surveillance feeds on the command bridge of the Hermes trawler.

research [91]) conclude that it is not feasible to send the video data. Manually collecting video data from boats at landing time and parsing the data is extremely resource intensive. This also does not allow for inspectors to handle illegal cases as soon as possible.

Legal Measures

Here, we will provide a brief overview of some of the legal measures proposed in the ONR. Multiple requirements in the current legal language depend on who is on board the fishing vessel when actions are taken. However, currently there is no standardised way for authorities to easily get information about the crew of a specific ship. It has been proposed that crew lists should be mandatory and available, both on board and on land. Verifying

2.1. Illegal, Unregulated and Unreported Fishing

the identities of the crew is another challenge that needs to be addressed. In the ONR, biometric authentication via smartphones are presented as a solution, and it has already been applied in other industries like in construction. The identification solution can also be used to verify which crew member actually performed an action or submitted a report [79].

Currently, there are no requirements for boats under 13 meters in length, which represent between 10 and 15 per cent of the fishing fleet, to report movement, activity, or catch. This means that control authorities essentially have no tools for detecting misreported catch when it is sold. It is possible to include these fishing vessels into the existing reporting system, however changes would need to be introduced, e.g., vessels that fish close to the port where they will deliver the catch (usually smaller vessels) do not have to report the docking time two hours before docking, this exception could be removed [79]. However, there have been some changes to the regulations since 2019, and it has been decided that, from 2024, all fishing vessels that sell fish will have to report their movement and activity [39].

In order to realise verifiable data from fishing vessels, the ONR proposes to make it mandatory to introduce technology for verifying catch. The lack of such technology currently makes it difficult to detect deviations from reported catch and the actual catch numbers. Multiple solutions have been proposed: certified scales, devices for counting the number of fish, or measuring the volume. These different solutions would need to be assessed in order to be able to recommend the most suitable one, and different solutions are more apt for certain fishing vessels depending on their size, age, and the type of fish they catch [79].

The CatchID Program

The CatchID program is a new initiative by the Norwegian Directorate of Fisheries that aims to identify, investigate, and get fishers to use new technologies for automatic documentation of catch. The technology will

also provide better data for authorities, reducing the chance of incorrect reporting, and enabling better control of landings. The overall objective of modernising the resource registration in the fishing industry is to improve the sustainability and competitiveness of the industry and ensure the responsible management of marine resources. It was created as a response to the ONR and is currently in its early stages [54].

2.2 Summary and The Way Forward

Serious problems in the fishing industry have been identified. A lack of verifiable data, holes/ambiguities in the legal framework, and sub-optimal organisation of data sharing solutions between sales teams and inspection authorities. The technologies that have been proposed as solutions to these problems range from integrating surveillance systems on all fishing vessels, automatically detecting, recognising, and logging catch using video cameras and other sensors on-board or in trawler nets and identification of crew via biometrics.

In this thesis, we design and develop several components of a privacy-preserving system that can perform catch analysis and detection of suspicious events. In order to do this, we utilise the technologies, such as machine learning, that are described in the next sections.

2.3 Brief Overview of Machine Learning Relevant to the Thesis

Machine learning is a subfield of AI that deals with the design and development of algorithms that can learn from and make predictions on data. These algorithms are used in a variety of applications, such as recommender systems, image classification, and speech recognition. Machine learning algorithms are often categorised into three main types: supervised learning, un-

2.3. Brief Overview of Machine Learning Relevant to the Thesis

supervised learning, and reinforcement learning, and two sub-/intermediate types: Semi-supervised learning and self-supervised learning.

Supervised Learning In supervised learning, the algorithm learns from the training data that is labeled with the correct answers. The goal is to learn a function that can map the input data to the correct output labels. This function can then be used to make predictions on new, unseen data. Supervised learning algorithms can be divided into two main groups: regression and classification. Regression algorithms are used when the output labels are continuous values. For example, you could use a regression algorithm to predict the price of a house based on its size, number of bedrooms, and location. Classification algorithms are used when the output labels are discrete values. For example, you could use a classification algorithm to predict whether an email is spam or not. There are many different supervised learning algorithms, and the choice of algorithm depends on the nature of the data and the task that you are trying to solve [119].

Unsupervised Learning Unsupervised machine learning algorithms are a category of machine learning algorithms that are used to find patterns in data. They do not require a labeled dataset, which makes them very powerful when it comes to exploring large and complex datasets [49]. There are many different types of unsupervised machine learning algorithms, but some of the most popular ones include clustering algorithms and dimensionality reduction algorithms. Clustering algorithms are used to group data points together based on their similarity, while dimensionality reduction algorithms are used to reduce the number of features in a dataset while still preserving the important information [119].

Reinforcement Learning Reinforcement learning is a computational approach to learning where agents take actions in an environment in

order to maximise some notion of cumulative reward. The key difference between reinforcement learning and other types of machine learning is that reinforcement learning is not concerned with predicting labels or output values, but rather with predicting which action will lead to the most reward in the future. A typical reinforcement learning scenario involves an agent that is placed in some kind of environment and must learn to interact with that environment in order to achieve some goal. For example, in a game of chess, the goal might be to win the game, while in a robotics task the goal might be to navigate to a specific location. The agent will receive some kind of feedback after each action that indicates how successful that action was in terms of achieving the goal. Based on this feedback, the agent will learn which actions are likely to lead to success and will adjust its behaviour accordingly [100].

Semi-supervised Learning Semi-supervised learning is a machine learning approach that combines both labeled and unlabeled data to train a model. It is usually used when there is not enough labeled data available to train a model with traditional supervised learning methods. Semi-supervised learning algorithms typically make use of both the class labels and the relationships between data points to learn a model. This approach can often improve the accuracy of the resulting model compared to traditional supervised learning methods [14].

Self-supervised Learning Self-supervised learning is a neural network training technique where the model is trained using a task that is automatically generated from the data. The aim is to learn a representation of the data that can be used for downstream tasks such as classification. Common pretext tasks used to train self-supervised models include image rotation, predicting the next frame in a video, and predicting the order of a sequence [67].

2.3. Brief Overview of Machine Learning Relevant to the Thesis

Our goal of assisting catch documentation and detecting abnormal events can be more effectively achieved by leveraging machine learning to automatically identify patterns in the data. Machine learning allows us to make better use of the data by spotting trends that would be difficult for humans to discern. Additionally, machine learning can help us prioritise which data is most important to focus on, and can do so more accurately and quickly than humans.

2.3.1 Deep Learning

Deep Learning is a subset of machine learning based on artificial neural networks. Neural networks are a type of machine learning algorithm that are used to simulate the workings of the human brain. Deep learning is a type of neural network that is composed of multiple layers. The term “deep” refers to the number of layers in the network. Deep learning is used for a variety of tasks, including image recognition, natural language processing, and predictive modeling [10].

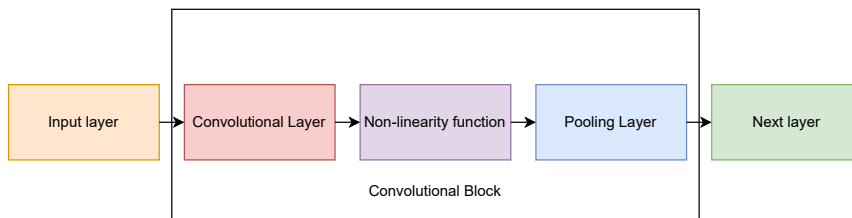


Figure 2.4: The components of a Convolutional Neural Network (CNN), explained below. The next layer can consist of e.g. another convolutional block or a feed-forward layer.

A Convolutional Neural Network (CNN) is a type of artificial neural network that is used to process data with a grid-like topology. The network is made up of a series of layers, each of which consists of a set of nodes (or neurons) that are connected to each other. The nodes in the first layer are connected to the nodes in the second layer, and so on. The nodes in each layer are connected to the nodes in the previous and next layers in a way that

preserves the spatial relationship between the nodes. An illustration of the key components can be seen in Figure 2.4. The name “convolutional neural network” comes from the fact that the nodes in the first layer are connected to the nodes in the second layer in a way that is similar to the way that the coefficients in a convolutional filter are connected to the input signal. The convolutional filter is a mathematical operation that is used to find patterns in data. The nonlinearity layer is used to introduce nonlinearity into the input data. This is done by applying a nonlinear function to the output of the previous layer. The most common nonlinear function is the rectified linear unit (ReLU), which sets all negative values to zero. The pooling layer is used to reduce the dimensionality of the input data by downsampling. This is done by applying a pooling operation to the output of the previous layer. The most common pooling operation is max pooling, which takes the maximum value from each region of the input. In CNNs used for classification, multiple convolutional blocks are used, followed by a fully-connected neural network that performs the classification on the features extracted by the convolutional part of the network. The CNN is a powerful tool for analysing data because it is able to take advantage of the spatial relationship between the data points. This is especially useful for image data, where the pixels in an image are typically arranged in a grid. The CNN is able to learn to recognise patterns in the data that are not apparent to the human eye [10].

We utilise CNNs in multiple components of the system we propose in this thesis. For example, in fish detection and recognition, the goal of this task is to identify the fishes present in the video frame, and to classify them to facilitate the automatic catch documentation. Another example would be human detection and activity recognition, where we want to detect whether a human is in the frame of a surveillance camera, for storage saving purposes, and classify human actions from the video to detect abnormal activity. Human actions are generally highly variable, and are difficult to

2.3. Brief Overview of Machine Learning Relevant to the Thesis

generalise from a few annotated examples, particularly considering there are no publicly available datasets like this for fishing vessels.

2.3.2 Anomaly Detection

Machine learning is increasingly being used for anomaly detection as it can learn to identify patterns in data that are indicative of a problem. This is especially useful in situations where it is difficult to identify anomalies using traditional methods. For example, machine learning can be used to detect fraudulent financial transactions, or to identify unusual patterns of behavior that could indicate a security threat. Anomaly detection is a challenging problem, as there can be many causes of anomalies and they can be difficult to spot. However, machine learning is proving to be a powerful tool for tackling this problem. Most novel methods are deep learning-based, however these approaches require a lot of data [13].

One example is the use of a deep autoencoder for detecting credit card fraud; the autoencoder is trained on a dataset of normal transactions, and then applied to a dataset of known fraud cases in order to identify new fraud [103]. Other examples include the use of deep learning for detecting intrusions in computer networks [126], and for identifying anomalies in medical images [65].

There are a few examples of unsupervised anomaly detection methods that can be used in order to find outliers in data sets. One example is the isolation forest algorithm, which works by isolating individual observations in the data set and then measuring the anomaly score for each observation [75]. In time series data, anomalies can arise in multiple ways, such as literal outliers that deviate sufficiently from the distribution, or changing points that shift the distribution. Anomaly detection on time series also depends on whether the time series is univariate or multivariate [12].

2.3.3 Datasets Relating to the Fishing Industry

Datasets related to the fishing industry can be difficult to obtain due to fishing vessels being far out at sea. However, some do exist. Global Fishing Watch (GFW) is a company that aims to provide open-access and transparency with regard to human activity at sea. They have several public datasets that focus on AIS and general information related to sea vessels, such as locations of anchorages, bathymetry data, distances from any point to the closest port, etc. This data can be useful, when combined with other data, to detect abnormal behaviour, like Park et al. detecting illegal fishing in collaboration with GFW [93].

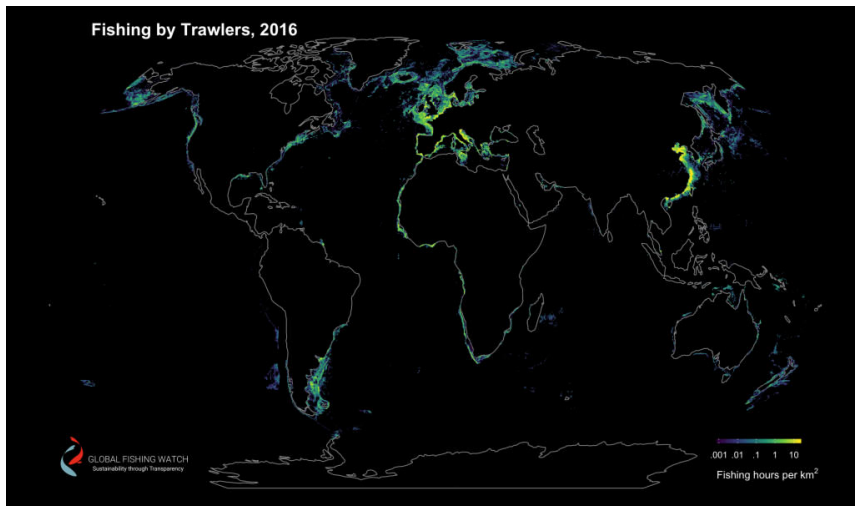


Figure 2.5: GFW uses their data to map fishing activity around the globe [130].

In Norway, there are also multiple public data sources provided by the Norwegian Directorate of Fisheries¹ and Kystverket². AIS positional data, activity data, and sales notes are publicly available, and with these data sources it is possible to gain insight into the fishing industry.

There are also several datasets for ship detection based on Synthetic

¹<https://www.fiskeridir.no/Tall-og-analyse/AApne-data>

²<https://kystdatahuset.no>

2.3. Brief Overview of Machine Learning Relevant to the Thesis

Aperture Radar (SAR) and optical satellite imagery [101]. SAR is used because it can penetrate clouds, making it useful for ship detection. Ships can be difficult to spot with traditional optical imagery because they are often small and can be obscured by clouds, however optical datasets also exist.

In Table 2.1, we present a brief overview of some of the most cited datasets relevant to the fishing industry.

We see multiple fish image datasets, some of which can be useful for automatic catch documentation, though many of them consist of underwater images which, while being useful for certain types of catch documentation [81], only covers a minority of use-cases. Analysing the catch when it is on board a fishing vessel can potentially be more useful, and therefore requires datasets of fish on the conveyor belt on board fishing vessels. The reason for this is due to generalisation failure, which is what occurs when a model trained on a specific datasets cannot generalise to data that is outside the distribution of the training data [6, 19]. Hence, a model trained on a dataset which contains only underwater images of fish, might not generalise to images of fish on the fishing vessel.

We also see datasets that focus on ship detection, based mainly on satellite imagery. This can, e.g., be used to detect fishing vessels that have turned their AIS off or who fish outside their jurisdiction.

There are also multiple data sources that are relevant to the fishing industry, such as the data provided by government sources. However, many of these sources are lacking in data quality, and have not been prepared for a machine learning-based use-case.

To conclude, some datasets related to the fishing industry exist, but nothing specific to what is actually happening on the fishing vessel. This is sorely needed, as it is the first data that can be used to understand the daily work of fishermen, and how many fish are caught, and what are the effects of this activity on the environment. Therefore, we have created a dataset

Chapter 2. Background and Related Work

Table 2.1: An overview of existing datasets and data sources relevant to the fishing industry. The overview consists of fish image datasets, positional and satellite imagery-based datasets, and the dataset that we have curated for this research that represents a hitherto unexplored frontier of the fishing domain.

Dataset	Content	Size
Fish image datasets		
Labeled Fishes in the Wild [18]	Images of fish, invertebrates, and seabed	4,092 images [†]
NorFisk [17]	Images of saithe and salmonids from fish farms	12,514 images [†]
A Large-Scale Fish Dataset [123]	Images of fish from supermarket	1,000 images ^{†◊}
Fish-Pak [104]	Images of fish from fish farms in Pakistan	915 images [†]
Positional and satellite imagery-based datasets		
Anonymised AIS training data [5]	AIS data for detecting fishing events	approx. 3GB
FUSAR-Ship [52]	SAR satellite image data with AIS data of vessels in images	5,000 SAR images, corresponding AIS data
SAR-Ship-Dataset [129]	SAR satellite image data of ships	210 SAR images, 59,535 ships [†]
HRSID [52]	SAR satellite image data of ships	5,604 SAR images, 16,951 ships [†]
SSDD [136]	SAR images with ships annotated	15 large-scale images [†]
Boat Re-ID dataset [113]	Images of boats in bay	5,523 images of a total of 107 boats ^{†◊}
Ships in Satellite Imagery [76]	Optical satellite images of ships from Google Earth	1,061 images [†]
MASATI-v2 [44]	Optical satellite images of ships, land, coasts, and sea	7,389 images [†]
Our dataset		
Njord [90]	Surveillance videos from fishing trawler	198 10-minute videos ^{†♣}

[†]Including ground truth segmentation masks or bounding boxes

[◊]Originally fewer images, expanded with augmentation

[♣]71 videos are annotated with bounding boxes and classification labels

based on surveillance videos from a fishing trawler. We will describe this dataset further in Section 4.3.4.

2.4 Related Work

Here we will give a brief overview of related work, focusing on distributed edge-based systems and related machine learning-based approaches.

2.4.1 Distributed Edge-based Systems

In certain use-cases, placing computational resources closer to the edge, i.e. closer to where the data that is to be analysed is being generated, can be beneficial. In systems with many devices on the edge, a centralised architecture for handling the processing of the data being generated simply does not scale. To add onto that, sending data over low-connectivity/high latency connections continuously is often not feasible for certain types of applications [26]. Systems with a distributed edge-based design have been described and used in other use-cases.

Fitwi et al. [40] describe a system for masking private information in video frames from surveillance cameras by doing detection and filtering on the edge. In their paper, they discuss the advantages and disadvantages of a cloud-based surveillance system vs. an edge-based approach. They argue that some pre-processing (i.e., motion detection for filtering of scenes) can happen on the edge, but that deep learning-based object detection methods are too computationally intensive to perform on the edge. They introduce a chaos-based encryption scheme that is used on the data before it is sent to a server for processing. There, they perform window and people detection and face recognition.

D'souza et al. [25] describe a similar system that uses object detection for surveillance camera video streams, and whitelists classes of objects that should *not* be censored. They claim that censoring everything, and whitelisting certain objects and revealing them using object detection leads to better privacy. This also reduces bandwidth requirements, as less information needs to be sent over the network. Our system consists of many

similar components that allows us to preserve privacy. However, a major difference between Dutkat and these two works is our relative greedy use of bandwidth.

Vasisht et al. [127] present FarmBeats, an Internet of Things (IoT) platform for data-driven agriculture. Their platform is designed for long-term, large-scale deployment in low-connectivity conditions. FarmBeats leverages Wi-Fi and TV White Spaces to allow for high bandwidth connections between the farmer’s home internet and the IoT base station, and the base station and the sensor devices. They also describe how drones are utilised as a main data source, using path planning, image stitching, and compression to optimise battery life and amount of useful sensor data.

All of these approaches utilise a distributed edge-based design for different goals; offloading some of the computation, maintaining privacy, or making a system robust in low-connectivity environments. For our system, we need all of these properties of distributed edge-based systems to provide robust automatic catch documentation in a privacy-preserving manner that can scale.

2.4.2 Machine Learning

Given the objectives described in the previous chapter and the problems within the fishing domain outlined in Section 2.1, catch documentation needs robust tools for automatically detecting fish, recognising their species, and logging their size/biomass. Secondly, we examine at how positional and satellite imagery can be used to detect suspicious fishing vessels. Thirdly, we also need to investigate approaches for detecting abnormal activity to determine whether something illegal is occurring. Finally, we discuss some considerations regarding multimodal data, because our system needs to be able to perform catch analysis and detect abnormal events based on multiple data sources, both at sea and on land.

Using machine learning to analyse fish automatically can be challenging,

depending on your goals. All image-based machine learning techniques rely on training data that includes examples of a wide range of environmental factors, such as lighting, water clarity, etc. With overlapping fish, segmentation can be very challenging. In their article, Alsmadi et al. [4] investigate several datasets, the data needs of deep learning approaches, and how fish classification has historically been accomplished using conventional machine learning techniques. Fish weight assessment from photographs can be challenging due to the lack of a 3D view and the fact that fish density varies depending on species, feed, age, etc. A LinkNet-34 segmentation network is used by Konovalov et al. [68] to automatically segment fish in photos. They then contrast utilising a convolutional network to directly estimate weight via regression with mathematical models for calculating fish weight based on length, height, and/or segmented area. Using stereoscopic images, Garcia et al. [45] estimate fish measurements. They demonstrate how, in the situation of overlapping fish, segmentation is facilitated by stereo imagery. After segmenting the fish with Mask RCNN [50], they perform morphological operations to determine its length. This might then be incorporated into a mathematical weight estimation model. Sokolova et al. [111] also utilise Mask RCNN to detect catch in trawling operations. This can be used to determine catch composition and quanta, and also ascertain whether any unwanted catch has been caught.

With regard to the catch documentation, there are also multiple solutions from the industry that are based on machine learning methods. For example, Sintef, as part of their SMARTFISHH2020 project, have researched and developed multiple catch analysis and fish monitoring methods [38, 108, 110], including detecting fish discard via CCTV [41]. They also have a new project called EveryFish that is focused on preventing overfishing that will start in 2023 [109]. The company team.fish are also utilising video analysis for real-time monitoring of catch, to facilitate sustainable fishing [27]. BioSort and Cermaq are developing a system for monitoring the health of

fish in fish farms. The system is able to recognise individual fish and track growth and whether they have caught parasites [55]. Shinkei Systems has also developed a system that can replace the current manual system for processing fish after it has been caught. They utilise AI-based methods to automatically euthanise fish in a quicker and more humane way than humans can [106].

There have been a number of studies on illegal vessel detection using AIS positional and satellite data. Shahir et al. [105] describe an approach to detecting and tracking dark fishing operations by vessels (i.e. fishing vessels that have turned off their AIS in order to fish illegally). The approach is based on profiling and ranking fishing vessels by analysing their routine operations over extended time periods to uncover abnormal activity patterns associated with dark fishing. Kurekin et al. [70] for monitoring non-cooperative vessels in the Ghanaian Exclusive Economic Zone (EEZ) uses data from the synthetic aperture radars on Sentinel-1 and the Multi Spectral Imager on Sentinel-2. The detection algorithms have a high success rate with 91% of registered vessels being matched to a satellite detection. The satellite data also yielded estimates for length and width of vessels that matched the distribution found in the area. Park et al. [94] and Milios et al. [78] also describe methods for detecting vessels fishing in illegal waters using multimodal approaches. Welch et al. [132] utilise the GFW AIS dataset, described in Section 2.3.3, to detect AIS disabling hot spots in commercial fisheries. Disabling hot spots were located near the EEZ of Argentina and West African nations and in the Northwest Pacific, all regions of IUU concern. Disabling was highest near transshipment hot spots and near EEZ boundaries, particularly contested ones.

In Rodriguez-Moreno et al. [99], the authors describe the challenges and state-of-the-art methods for performing activity recognition from video. They explain that one needs to first recognise the relevant areas that contain humans in the frames, and then a variety of techniques can be applied.

Hand-crafted feature extraction has been the traditional approach, while in more recent times, deep learning-based solutions have been preferred. In the more specific area of recognising suspicious activity, the task becomes more difficult. In the Tripathy et al. [121] survey, the authors show usage of object detection, tracking, and activity recognition in different use-cases. However, in all these cases it is relatively simple to find examples of suspicious activity on video, as the settings are more conventional. In the case of the fishing industry, it is more difficult due to the lack of data. In a paper by van Essen et al. [33], they propose to detect discard of fish by estimating the weight of fish detected on the conveyor belt. However, we also want to detect other illegal events, such as when fish nets are destroyed on purpose, or when fish are transferred to other boats. To do this, we assert that surveillance cameras are also required to monitor crew activity.

According to Baltruvisaitis et al. [9], there are five major challenges of multimodal data analysis in machine learning. Two of these challenges, representation and fusion, are highly relevant to our use-case. The machine learning pipeline in the system has to utilise complementarity in e.g. the AIS and sales notes data to predict some anomalous activities, and requires prediction from different models to form a final judgement on whether a fishing vessel should be inspected or not.

2.4.3 Summary of Related Work

Section 2.4 has discussed the related work on distributed edge-based systems and machine learning related to the fishing industry and activity recognition. It discusses the need for such concepts in relation to the focus of our thesis, as well as the challenges associated with them. The section describes various approaches that have been considered to address these challenges, including the use of sensors, machine learning, and edge-based computing.

Our system needs to handle multimodal input and analyse it using machine learning, all while in an edge environment. Low connectivity, privacy

preservation and scaling the system to handle thousands of fishing vessels make distributed edge-based systems a useful design to utilise. Fish detection and recognition is a well-researched problem, however the other events that we need to analyse on the fishing vessel are less so. Detecting and recognising suspicious activity on-board is a more difficult problem, given that video data from fishing vessels is non-existent (with the exception of our dataset), thus we also need to look into unsupervised approaches for detecting such events.

2.5 Ethical considerations

The introduction of surveillance in the workplace, particularly on fishing vessels, raises privacy issues for workers. While surveillance can be used to improve safety and monitor productivity, it can also be used to invade workers' privacy. In addition, the use of surveillance cameras on fishing vessels can raise concerns about the security of the footage and who has access to it. Therefore, our system follows a compliance-by-design approach to privacy, by ensuring that minimal, processed data is sent to the mainland and the raw data is stored securely onboard.

Inspection of fishing vessels is the process of checking fishing vessels for compliance with catch, environmental, and other regulations. Inspectors may board vessels to conduct inspections, and they may also inspect vessels from the shore. Thus essentially creating a guilty-until-proven-innocent environment in the fishing industry. In order to avoid this, data that is generated on the vessel needs to be encrypted, and only be accessible given a plausible cause and a legal warrant. Our system can supply evidence via verifiable data to inform inspectors of which vessels to inspect.

AI is revolutionising the way we live and work, but if we are to apply it in the fishing domain (or any other domain), it is increasingly important to consider the ethical and privacy implications of its use. AI algorithms

are trained on vast amounts of potentially personal data, and the decisions made by these systems can have a profound impact on individuals and society as a whole. As a result, it is crucial to consider the privacy of the data that is used to train and operate AI systems. Furthermore, the use of AI raises important ethical questions around fairness, accountability, and transparency. For instance, decisions made by AI systems must be impartial and unbiased, taking into account all relevant factors and avoiding discrimination on the basis of race, gender, or any other protected characteristic. Additionally, it is essential to ensure that these systems are transparent, so that individuals can understand why decisions have been made and have the ability to contest them. AI also raises important issues around accountability, particularly when decisions have a significant impact on people's lives. In light of these considerations, it is important to ensure that AI systems are designed, developed, and deployed in a manner that respects privacy, ethics, and human rights [7, 107].

2.6 Summary

In this chapter, we have discussed issues regarding sustainable fishing, specifically the problem of overfishing. We have also presented some of the solutions proposed in the ONR, and problems related to these solutions. Then, we presented machine learning datasets relevant to the fishing industry, specifically those related to fishery monitoring and catch analysis. There is a need for more data on what is happening on fishing vessels, in order to improve our understanding of how illegal activity occurs on fishing vessels. Finally, we discussed related work regarding distributed edge-based systems and state-of-the-art machine learning solutions. The distributed edge-based system is a model we choose to adhere to in our system design, in order to make it capable of performing analyses both on the edge and on the mainland, all in a privacy-preserving manner.

Chapter 2. Background and Related Work

In the next chapters, we will first present a requirement analysis of our system, then we will describe and discuss the design of our system and the components we have developed. We will also present the analytical approaches we have used and the data we have worked on.

Chapter 3

Requirement Analysis

In this chapter, we present the functional and non-functional requirements of our system. Our system, Dutkat, is a privacy-preserving, multimodal system for automatic catch documentation and suspicious activity detection. It consists of two major components; the edge computational node placed on fishing vessels and the mainland hub. Data is stored securely on the fishing vessel and can only be accessed when evidence of illegal activity has been processed and sent to authorities. Analysis of multiple data sources have to be performed both on the fishing vessel and on land.

In Norway, inspection authorities on mainland currently perform inspections of trawlers when they dock at harbours to sell their catch. The selection of vessels and fish buyers to inspect, among the many, is primarily based on pre-fixed scheduling and on historical data, e.g., sales notes. With Dutkat, we aim to notify the authorities of suspicious activity so that targeted inspections can be based on flagging of specific vessels. In this way, we can replace the current randomised inspection regime with an evidence-based scheduling regime that can help authorities focus their limited control resources.

Given the background and related work presented in the previous chapter, we are aware of the problems within the fishing domain with regard

to enforcing sustainable fishing practices and we have seen some proposed solutions. Our system takes a compliance-by-design approach by focusing on the privacy of the fishers, while at the same time allowing analysis of the catch and detection of suspicious behaviour on-board.

Requirements engineering is a process in which required services for a system and the constraints which the system operates under are specified. Requirements are descriptions that state the system service and/or constraints generated during the requirement engineering process [112].

3.1 Functional Requirements

Functional requirements are defined as statements of services that the system should provide, how the system should react to particular inputs, how the system should behave in particular situations or what the system should not do [112]. In this section, we outline the functional requirements for the system.

- **Edge Computational Node**
 - **DAG Computational Model** The system must support a general distributed DAG computation model composed of up to multiple pipelines that run concurrently. The distributed DAG model must support machine learning inference tasks to run within the DAG through pre-trained machine learning models. The system must also support that machine learning inference may be executed over multiple nodes.
 - **Modular Redundancy** Our system on-board a fishing vessel needs modular redundancy to ensure that the system can continue to function even if one or more of its component parts fail. This is because the system needs to be able to continue to monitor the

3.1. Functional Requirements

vessel and its surroundings for potential threats or illicit activities, even if part of the system is damaged or malfunctioning.

- **Data Collection** The data collected by the edge-based surveillance system on fishing vessels is used to monitor the activities of the vessels, automatically document the catch, and to identify potential illegal fishing activities. Video cameras and various other sensors are possible data sources.
- **Multimodal Analysis and Anomaly Detection** The system should be able to automatically document the catch and detect anomalies on-board the vessel by analysing data from the different data sources.
- **Communication with external sources** The results of the analysis of the data generated on the edge has to be transmitted to a central hub. The system must also support communication from external resources, such that the system can acknowledge that it is still up and running. In addition, the system must be able to provide its current state to an external source. The system should also be able to provide a history of registered failures as well.
- **Fine-grained Access Control of Storage** The storage system on the edge component requires fine-grained access control to ensure that only authorised personnel can access the system and view the footage. The access control helps to protect the privacy of individuals who are captured on the footage and to prevent the misuse of the system. The data can be removed periodically if the processing of it has finished and did not result in the detection of any anomalies.
- **Failure Recovery** The system must have a recovery scheme that enables it to recover from failures to the best extent possible.

This includes software failures that can happen internally in the system and hardware failures.

- **Mainland Hub**

- **Data Federation** The hub component of the system must collect/receive data from multiple sources, including the edge computation nodes. Data that can be collected from public sources include AIS, weather, and satellite imagery data.
- **Multimodal Analysis** This component also needs to analyse the data from different sources, and combine the results with the processed data sent from the fishing vessels. The analysis should be able to detect vessels which are acting suspiciously.
- **Decision Making Support** It should be possible for users of the system, i.e., control authorities, to interface with the mainland hub and easily view the data from multiple sources.

3.2 Non-Functional Requirements

Non-functional requirements are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process and constraints imposed by standards. They are often applied to the system as a whole rather than individual system features and services [112]. In this section, we will outline the non-functional requirements related to the system.

3.2.1 Compliance-by-Design

The system needs to be designed in a way that ensures compliance with GDPR standards and related legal standards from the outset. This includes taking into account the principles of data minimisation, data accuracy, data security, and data accountability to enforce the privacy of the data subjects.

3.2.2 Confidentiality

The Dutkat system will function as a distributed system that allows data processing on the fishing vessel itself. For video surveillance data to be kept confidential, it must stay on the vessel while it is out at sea. If an external source requests data from the system, it should only return metadata, such as anonymised results from the machine learning pipeline on the vessel, failure logs, the system's state, in order to preserve the confidentiality of those aboard the fishing vessel.

3.2.3 Integrity

It is important to guarantee data integrity in an edge environment with potential bad actors, because data integrity ensures that the data stored in the system is accurate and complete. Data integrity can be compromised by malicious actors who may try to alter or delete data. To protect data integrity, we can use cryptographic methods to verify the authenticity and integrity of data.

3.2.4 Availability

The Dutkat system ensures availability, specifically it allows for continuous analysis on the vessel, by using an edge computational node that is present on the fishing vessel. This node is designed to continuously gather and analyse data from various sensors and other sources, without being dependent on a connection to the mainland hub. This is particularly important when the vessel is in remote areas or far out at sea, where connection to the mainland hub is not available. By having an edge node on the vessel, it becomes possible to continuously analyse data, even in these situations, and then send the collected and processed data over to the mainland at next opportunity.

3.3 Proposed Architecture

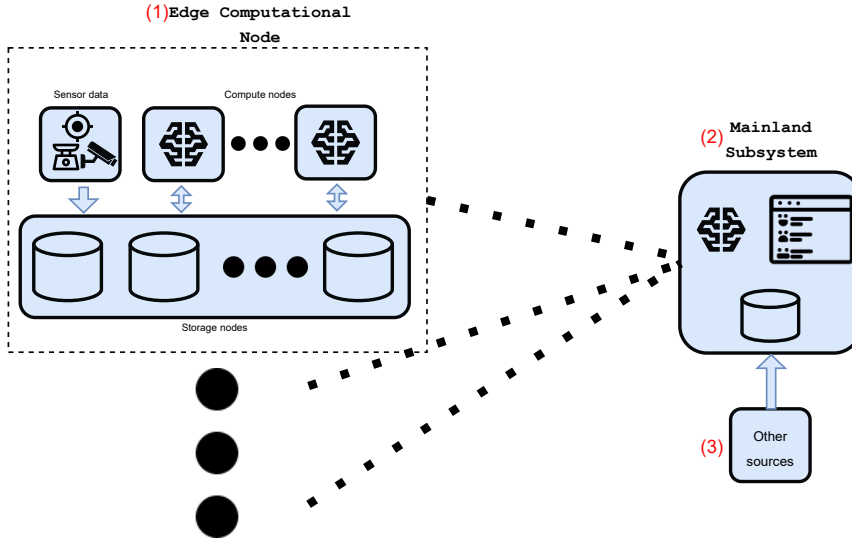


Figure 3.1: System architecture that satisfies the requirements outlined above. The edge computational node (1), mainland hub (2), and other data sources (3) used by the mainland hub are shown.

The proposed architecture that satisfies the requirements presented above is illustrated in Figure 3.1. The edge computational node consists of three major components: the sensors, the compute nodes, and the storage nodes. Note that the compute and storage nodes might physically be the same node.

The mainland hub should collect processed data from the edge computational nodes on each fishing vessel, along with other data, to facilitate decision-making support for inspection authorities. The results should be presented in a readable fashion and should be saved so that they can be used as historical data for future inferences.

The other sources that are utilised in the mainland hub's decision-making can e.g. be historical data such as sales notes and trip logs, weather data, and satellite imagery data.

3.4 Summary

In this chapter, we have presented the functional and non-functional requirements for our system. The edge computational node and the mainland hub have different requirements due to the different data sources that need to be taken into account and how this data needs to be processed, stored, and displayed.

With the system requirements already defined, the next chapter will describe the system components, but will focus on the analytical methods we have developed and explored. We will also describe the different data sources we have utilised in our analysis.

Chapter 3. Requirement Analysis

Chapter 4

The Dutkat System

This chapter details the Dutkat system, the data sources analysed or generated in this thesis, and the analytical methods developed for the thesis. Basically, the Dutkat system consists of two major subsystems, the edge computational node that is to be present on the fishing vessel, and the mainland hub that will collect and analyse data and provide decision-making support for control authorities. Analysis of the different data sources requires various methods due to their different modalities.

As was previously stated in Section 1.5, the Dutkat project is a larger effort with multiple people working on different aspects. Specifically, the distributed computation system presented in Section 4.1.1 and the distributed file system in Section 4.1.2 consist of major contributions from others. Therefore, we will, in this chapter, briefly present the Dutkat system and its components, but focus on the data sources and the analysis approaches we have explored and developed.

In Figure 4.1, we can see an overview of the Dutkat system, from departure from port to landing and reporting of catch. We can also observe the different analysis methods and data sources used for the different parts of the trip.

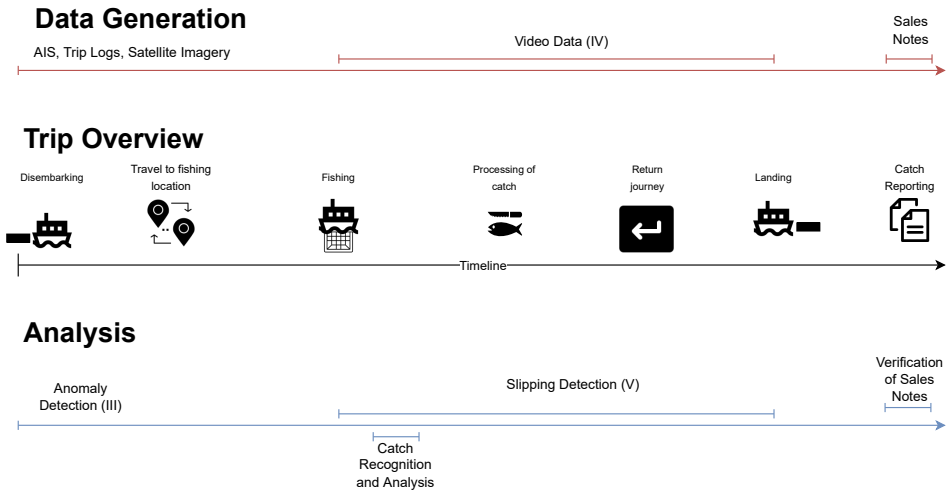


Figure 4.1: An overview of how the Dutkat system can be applied to data during a fishing trip. Different analytical methods are applied to the different parts of the trip in order to perform catch analysis and detection of suspicious events. Components with Roman numerals correspond to our research papers.

4.1 Edge Computational Node

Dutkat introduces multiple nodes for computation and data storage on fishing vessels in Norwegian waters as part of the edge subsystem. The edge subsystem, also called the edge computational node, is responsible for providing the system with the functionality to handle sensitive data while addressing concerns of physical tampering and unauthorised access on-board. Some tamper resistance techniques can be applied during circuit board design [133], while tamper detection methods typically involve dedicated sensors that can trigger alerts and responses [131]. Software integrity can be enforced through dedicated hardware such as TPMs, or through a Central Processing Unit (CPU) with integrity guarantees inherent in its architecture, such as those provided by ARM TrustZone or Intel SGX [133, 135].

The video and other sensor data will be stored locally on the edge computational node. The storage solution requires a fine-grained access-control

scheme to maintain the privacy of the crew of the vessel. Several video surveillance systems propose privacy models that provide a definition of information that is considered sensitive, which is detected through computer vision and subsequently removed from recordings before entering persistent storage [133]. There are multiple approaches to removal, such as obfuscation, replacing the sensitive data with a blank area, abstraction, and encryption.

Video recordings with encrypted areas can be stored in multiple privacy levels, with a system that presents different information to different users, based on their security clearance. In practice, this involves storing multiple versions of the same video recording, with certain features removed or modified. Every version must be encrypted using different encryption keys. The lowest clearance levels may only retrieve metadata or derived information, while the highest clearance level may retrieve the least modified version of a video, or a video enhanced with additional information [102].

We have worked with governing bodies to properly identify requirements and needs and ensure that all data captured on edge devices can only be used in a legally authorised forensics and control context. This way, no privacy-sensitive data is transmitted out of an edge node as a rule, only the analysis based on the data that might indicate that certain events potentially occurred. Simultaneously, all ground truth data is kept for potential use governed by regulations and existing law with regard to examination of private data. Consider this as a 24/7 life-logger continuously analysed, but only manually accessed in exceptional cases for forensic purposes. Data that is not used within a given timeframe will be deleted. We leave the timeframe open as a configuration option, to be determined by potential future regulatory requirements.

4.1.1 Distributed Computation

In order to facilitate analysis of data from multiple sources, all on the fishing vessel, which will also be robust and resilient, we need a system that can distribute tasks over multiple nodes in a fault-tolerant fashion. Therefore, we have developed the Áika system, which is a robust system for executing distributed AI applications on the edge. Áika is a fault-tolerant system that supports a DAG computational model, often exhibited by AI models. A key property of Áika is how it remains active and performs continuous analysis of data during various component failures. The system is designed to be both tolerant of faults and able to detect and monitor them [3].

4.1.2 Distributed File System

Given the distributed nature of the computational system, a distributed file system is required to securely store and handle access to the files from different computational sub-nodes on a given fishing vessel. Therefore, we introduce the Dorvu storage system, which provides confidentiality of the generated data using fine-grained access control. Depending on how, e.g., the video data is processed, it necessitates different levels of access control, so the raw output is handled differently from the processed video data that has potentially been anonymised. Dorvu is implemented as a FUSE [42] application, which allows remote mirroring of content. When the files are generated, they have an associated configuration file which specifies which groups/individuals should have access to which version of a given datum [91].

4.2 Mainland Backend Subsystem

Multiple data sources already transmit data to the mainland due to legal requirements. Catch notes are collected from sales organisations, owned

by fishers, which facilitate the sales process and ensure fair prices. AIS is continuously transmitted by fishing vessels (of a certain size), while other activity logs such as sales notes and trip logs are submitted at landing.

The centralised server hub is resource-rich and can use existing public cloud file systems supporting efficient, reliable, and centralised storage of multimedia data, sensor data, and machine learning results from the edge nodes. The edge nodes are less resource rich, compared to resources available on the mainland, and are physically located on active fishing vessels. That being said, larger vessel which account for a major part of the catches performed in a year, are often equipped with computers for monitoring and processing. When along the coast and near the shore, communication options include cellular networks and radio networks, but when more distant and offshore, satellite communication is the main option.

The analysis carried out on mainland primarily aims to identify suspicious activity. Next, this is used to schedule targeted manual inspections. This analysis can potentially be done by receiving a single video I-frame or just a binary signal identifying potential suspicious activity at that specific vessel. Deciding whether a given vessel should be inspected will also be chosen based on analysis of sales notes and positional data.

4.3 Data

In this section, we will describe the different data sources we have used in our research. For the analysis that can be carried out on the mainland hub, we federate data from multiple sources, including AIS positional data, activity data such as trip logs, and satellite imagery data. With regard to the edge computational node, we have focused on analysis of video surveillance data. However, there existed no such dataset at the time that could be used to train a model for analysis of what happens on the fishing vessel, so we will present our dataset that fills this gap and which was developed as part of this work.

4.3.1 Automatic Identification System

The Automatic Identification System (AIS) is positional data from all Norwegian fishing vessels above 15 meters in length. The frequency of AIS messages changes based on the speed of the vessels and can be anywhere from every 10 seconds to every 30 minutes. The variation of the message frequency is due to vessel movement, i.e., vessels that are stationary for longer periods send fewer positional updates. The AIS data contains Maritime Mobile Service Identity (MMSI) for a given vessel, its latitude and longitude, its heading, and its Speed Over Ground (SOG). An example of an AIS path is illustrated in Figure 4.3.

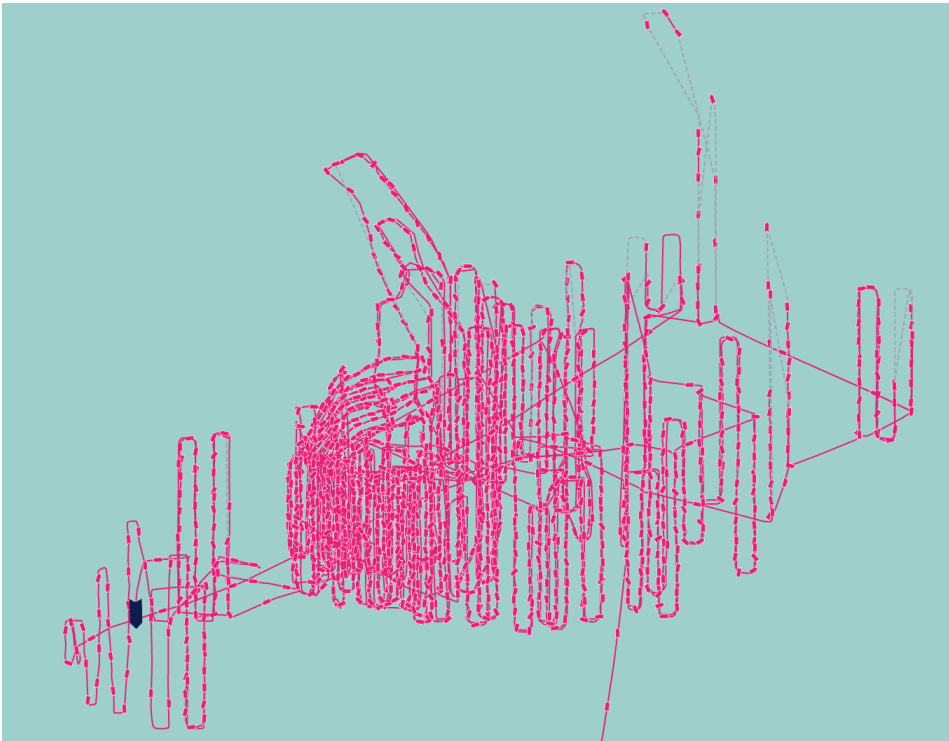


Figure 4.2: An image showing the AIS path of a vessel (in black) fishing. The data was collected from Kystverket.

4.3.2 Sales Notes and Trip Logs

The sales notes consist of information about the catch that is manually logged during landing, e.g., when it was caught, where it was caught, what equipment was used, the species distribution of the catch etc. There are approximately 130 data fields and around one million notes are produced each year by the fishing fleet in Norway. Manual inspection of these sales notes are currently the predominant method for determining if fraud potentially has occurred.

There are multiple trip logs that depict the activities of a given vessel while at sea.

Daily Catch Activity (DCA) contains start and stop times for fishing activities, gear used, species caught, and roundweight of the catch.

Reports on Departure (DEP) contains port of departure and information regarding the vessel's, MMSI, engine power, and length.

Reports on Landing (POR) similar to DEP, but contains the port of return.

In order to correlate sales notes with AIS, we also need to correlate with trip logs to find which MMSI corresponds with the vessel IDs used in the sales notes. These trip logs are lower quality than the sales notes, and often lack detailed information about fishing locations.

4.3.3 Satellite Imagery

Using satellite imagery for maritime surveillance for e.g. ice monitoring, ship monitoring, and oil pollution monitoring are common applications. Common satellite imagery modalities include optical imagery based on visible light, and SAR where the satellite transmits microwaves that are reflected and bounced back to the satellite.



Figure 4.3: An example of a SAR image. This image shows the island of Tromsø. Looking at the bright points in the sea surrounding the island, it is clear how vessels can be easily detected using SAR with a cross-polarisation modality. This image was created from Sentinel-1 data using Sentinel Hub API ¹.

SAR sensors have been used for several years for oil detection at sea. Oil slicks dampen the roughness of the sea that is generated by the wind, creating a contrast between the slick and the surrounding sea, which the SAR sensor detects [134]. SAR is independent of daylight or weather conditions, and can observe through clouds making it very suitable in the Arctic Ocean region.

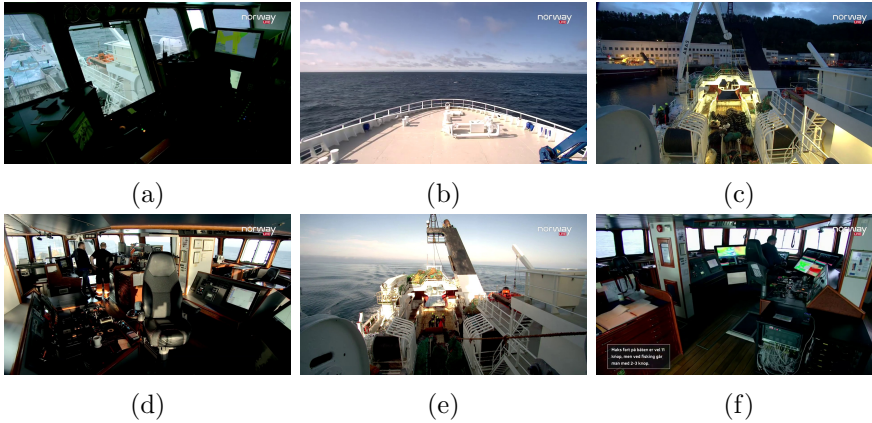


Figure 4.4: Sample frames from different videos of the Njord dataset. (a) - A view from the bridge looking down at the deck, (b) - A view from the front of the vessel, (c) - A view of the deck as the trawler moves from port, (d) - A view of the bridge, (e) - A view of workers on deck, and (f) - A view of the bridge from another angle with an overlay.

4.3.4 Njord: a Fishing Trawler Video Dataset

A sufficiently large, modern fishing vessel is infused with high-tech digital technologies. The bridge of a trawler operating in, for instance, the Arctic Ocean contains numerous terminals visualising geographical position and other vessels in the vicinity, weather conditions and predictions, fish finder sonar data and others. Video streams from the deck and production line under deck are also frequently displayed so that the officer in-charge has real-time information when making operational decisions. Accidents in this industry are not an exception and are an important problem to consider [77, 128]. However, video data from fishing vessels is not publicly available, which makes insight into procedures and events happening on fishing vessels impossible. Therefore, we, with video data provided by Hermes², have created a unique dataset, called Njord, that portrays what occurs on fishing trawlers.

The dataset contains 71 annotated videos and 127 videos that are not

²hermesas.no

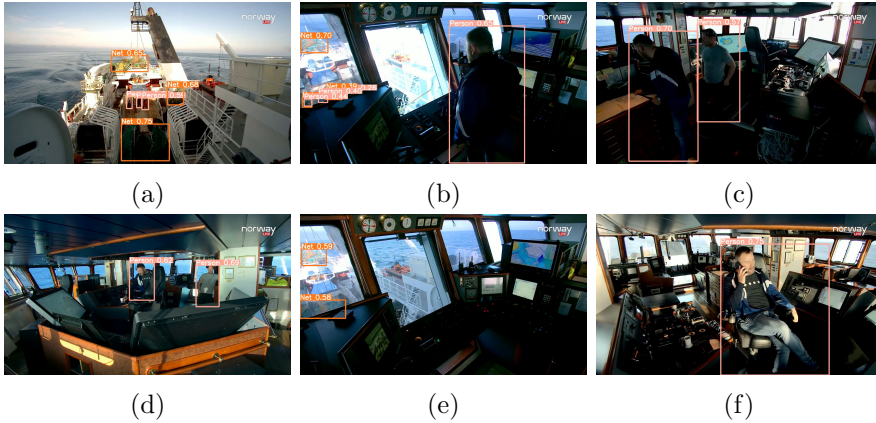


Figure 4.5: Sample predictions on the Njord dataset made by the YOLOv5m model. (a) - we see several workers on deck where the model is able to detect the workers in addition to the nets, (b) - The captain overlooking the workers on deck for which both the captain and workers are detected, (c) - A view from the trawler bridge with two workers detected by the model, (d) - A view of the bridge with a detected worker and captain, (e) - A view of the deck where the model detects the nets, and (f) - The captain sitting on the bridge which is detected by the model.

annotated from live-streams that aired in 2019. The dataset is also meant to be updated over time with more videos and additional data sources [90]. The dataset was used for the MediaEval 2023 NjordVid task, which we present in more detail in Section 4.5.2.

The videos in the dataset are approximately 10 minutes in length each, with a frame rate of 25 Frames Per Second (FPS). There are eight fixed-camera scenes as well as a manually-operated camera, which can be changed on a fixed schedule or by the captain, resulting in scenes with varying durations. Labels have been created manually using Labelbox [72], including bounding boxes for people, other boats, nets, and fish, as well as temporal annotations.

4.4 Analysis

In this section, we will present the analytical approaches/results we have developed during our research. The first two apply to the edge computational node, and the third applies on the mainland hub, based on the data that is analysed.

4.4.1 The need for Edge Computation for Analysis in the Fishing Industry

We have argued in the previous chapters and in Section 4.1, that analysis of the data generated on the fishing vessel should be performed on the fishing vessel. However, there is another reason for why we require the processing to happen on the edge, and that is satellite communication. Satellite communication is expensive, low-bandwidth, and unreliable. Therefore, we evaluate the feasibility of performing machine learning-based inference, specifically activity recognition on video data, on the mainland. We evaluate the throughput of a centralised system handling the machine learning workloads by the bitrate required over the satellite connection to perform inference in real-time. We compare this required bitrate to the optimal and average bitrate which a commercial satellite router can offer. We compress the video data to different degrees by reducing framerate and/or reducing the resolution. We aim to determine how the compression affects the top-1 accuracy of the chosen machine learning model. The average bandwidth of the satellite connection is at ≈ 35 kbps, and the optimal bandwidth, according to the router documentation sheets [57, 118], is at 176 kbps.

In our experiments, we utilise a 18-layer R(2+1)D network, introduced by Tran et al. [120], which was pretrained on the Kinetics-400 dataset [64]. We fine-tune it on the HMDB51 dataset [69], training different models on resized videos and/or videos with reduced frame rate. We train on 16-frame clips, as was done in the pre-training, and we sample these clips

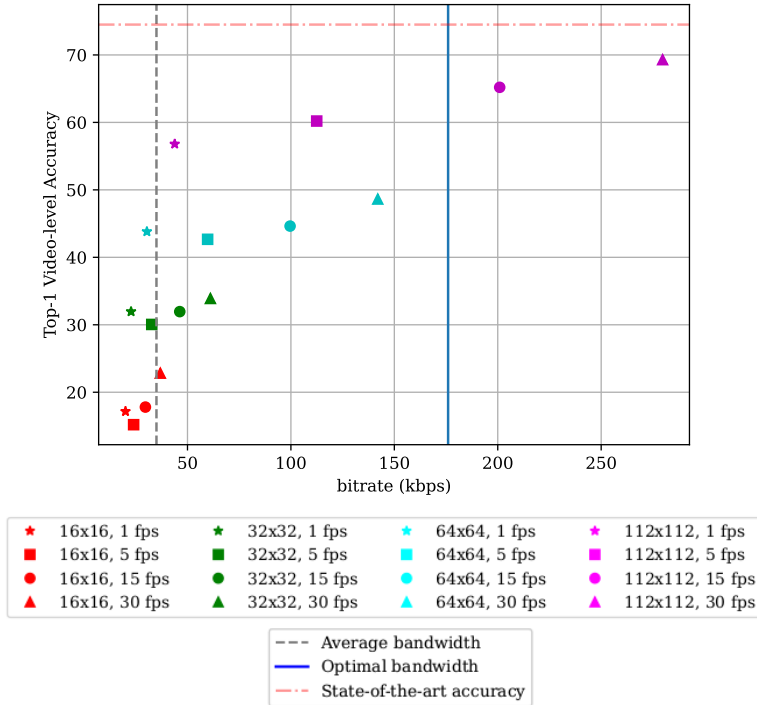


Figure 4.6: Plot over top-1 video-level accuracy (y-axis) vs. bitrate required to perform real-time inference (x-axis), including vertical lines indicating average and optimal bandwidth possible on our satellite connection. The different colors represent different resolutions of the video data used in fine-tuning the machine learning model, and the shapes are different frame rates. The horizontal line represents the top-1 video-level accuracy achieved by fine-tuning on HMDB51 in [120].

using temporal jittering. If the frame rate is too low, then we repeat a frame to fill the clip. The video-level accuracy is calculated by taking the average prediction of 20 different clips from the same video, then we select the top-1 result. The bandwidth required for the videos after resising and reducing frame rate is calculated by applying the reduction in resolution and frame rate on the entire HMDB51 dataset and averaging the bitrate.

Per Figure 4.6, we observe that reducing the resolution results in a dramatic loss in accuracy. However, lowering the framerate reduces the

accuracy to a lesser degree. The highest accuracy we achieved that required a lower bandwidth than the average bandwidth is at 43.81%, which is much lower than the highest accuracy of 69.3%.

Based on our experiments, performing the machine learning-based inference locally on the fishing vessel is the only feasible option currently. One of the advantages of our system is that we can perform a semantic compression of the data produced on the vessel and send it to the mainland. This allows us to only send useful information and also protect the privacy of the fishing crew [91].

4.4.2 Detection of Events and Anomalies in Multimodal Time-Series Data

Given that the data that will be analysed on the fishing vessel is multimodal, i.e. data collected from multiple sources in various forms, detecting anomalies can be challenging. We want to utilise all sources in our detection of abnormal events, and analysing these sources requires different methods. However, these data sources do not have labelled information regarding anomalous events. Therefore, we have applied an unsupervised approach to detecting anomalous events. In the fishing vessel use-case, labeled data for suspicious activities on the fishing vessel is non-existent. Thus, we have to leverage data from different domains. Our approach presented in this section was developed before the Njord dataset was collected and annotated.

We have designed a real-time unsupervised event change/anomaly detection system that can detect such events based on multiple datastreams. The pipeline design consists of three parts: (1) feature extraction, (2) an optional embedding layer, and (3) a moving average-based anomaly detection method.

The pipeline enables analysis of multimodal inputs, such as datasets based on multiple cameras/sensors, or combinations of anomaly detection techniques. We emphasise that, depending on the intended context, our

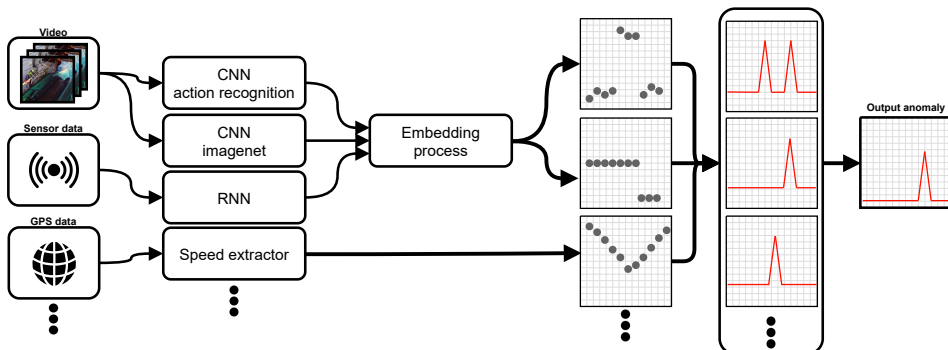


Figure 4.7: Example of pipeline configuration. Notice how multiple, multimodal data streams can be processed in different ways and run through the anomaly detection algorithm, before being combined. The embedding process is also dependent on the feature extractor, and thus is not always necessary.

architecture may be modified with various inputs, feature extraction techniques, and embedding procedures because it is modular. Figure 1 shows an illustration of how our use-case might be configured, with several inputs, various pretrained feature extraction techniques, and a final combination of the results. Anomaly detection outputs can be combined in a variety of ways, such as with straightforward boolean OR/AND operations or more complex weighting schemes.

Feature Extraction

The feature extraction step of our pipeline can be based on a general machine learning model like a pretrained neural network, which we apply. Such a model does not have to be trained on the use-case. For example, in this work we use an action recognition network trained on tasks not relevant for anomaly detection on a fishing trawler. We chose this model due to the fact that it is not well known which actions happen on a commercial fishing vessel and a more general action recognition dataset might be best to generalise to our case. The specific predictions do not matter in isolation, only the change in predictions over time.

To perform action recognition, we utilise an 18-layer R(2+1)D network, introduced by [120]. The network was pretrained on the Kinetics-400 dataset [64], which consists of 400 action classes. The output from this network functions as a simplistic feature extraction, where the predicted class of a subsequence of the video, is a data point in the transformed time-series.

An important aspect to note is that we chose this particular pretrained network due to our Dutkat use-case where we are attempting to identify unexpected and potentially illicit actions of the fishermen. However, due to the lack of data of such actions, we are particularly interested in changes of actions which the pretrained network predicts. These actions will often not be correct, but changes between different actions will most likely reflect a change in the actual actions as well.

Embedding

The output vector from the pretrained neural network is ordered alphabetically. That is, the indices of the output vector are ordered by the alphabetical order of the corresponding labels. Thus, values that are close together in the output vector will not represent any realistic ordinal relationship. In our experiments we observed that this can cause the anomaly detection method to misclassify certain data points due to the output value changing dramatically, while the underlying labels are semantically similar. For example, the output value might change from 112 to 353, which corresponds to indices that are sorted alphabetically, but the underlying labels corresponding to these values could be, for instance, “eating chips” and “tasting food”, which are semantically close.

The embedding process is illustrated in Figure 4.8. The textual labels that correspond to the output vector indices are first embedded via a *sent2vec* model, devised and implemented by [92], that was trained on a Wikipedia corpus. It is an efficient unsupervised algorithm that can be used to create vector representations of sentences and phrases, and has been

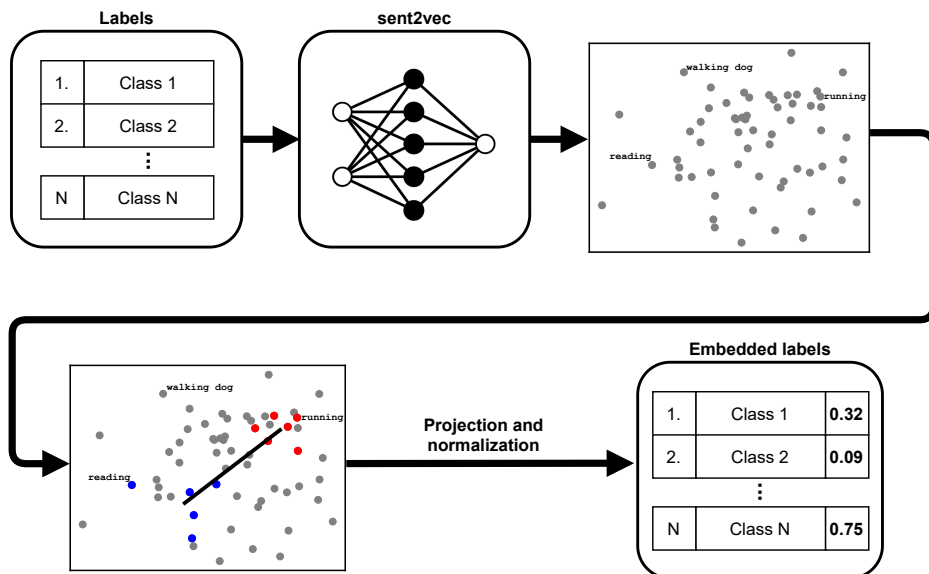


Figure 4.8: Illustration of the embedding process. Note how the embeddings of the labels are in a different order than the original index.

shown to outperform existing algorithms when used for tasks such as semantic similarity, semantic search, and text classification. Then, x , ten in our case, high-intensity and low-intensity activities were chosen from the labels to function as “support-vectors”, analogous to how a support vector machine works. A line is then drawn between the means of the low-intensity support-vectors and the high-intensity support-vectors. This line represents an intensity axis. Every point is then projected onto this line, and they are then max-min scaled to values between 0 and 1.

The axis used is based on the assumption that anomalies in videos containing people only arise when a drastic change in intensity occurs. We conjecture that such changes are context dependent and might change depending on the dataset on which the method is to be used.

Anomaly detection method

Most anomaly detection methods evaluate if a data point is part of the dataset distribution, and if not, labels it as an anomaly. We, however, are interested in an unexpected transition in the values of a time-series from single or multiple data streams, where the values are still part of the distribution. These are known as changing points [12].

Since we are interested in changing points, i.e., points in a time-series where the sequence changes for an interval of time, we chose a moving average approach. This approach compares the median of the previous n points to the current point in a time-series. If the current point is outside of the historical interquartile range, it is deemed an anomaly.

Evaluation

We evaluate our pipeline on three labeled video datasets from different domains, and an unlabeled version of the Njord dataset presented in section 4.3.4. We use a pretrained activity recognition model trained on the Kinetics-400 dataset [64] for feature extraction.

The *Multiple cameras fall* (Falling) dataset by [8] consists of 24 videos with eight cameras filming the scenarios. Of these, 23 depict a person performing several activities, before falling onto a mattress or chair. The remaining video does not contain a fall. The *Real-world Anomaly Detection in Surveillance Videos* (AV) dataset, created by [114], consists of 1,900 videos containing for example actions like abuse, arrests, arson, assaults, and accidents. The Soccernet-v2 dataset ([21, 22]) consists of 500 broadcast soccer games with 300,000 temporal annotations including action and camera labels. Events such as, e.g., fouls can be useful to determine whether anomalies can be detected using our pipeline.

All datasets except the Falling dataset were analysed using a simple linear pipeline which consists of the input, a pretrained action recognition

network, an embedding layer, and the anomaly detection algorithm. The Falling dataset contains multiple data streams capturing the same event, therefore the configuration is slightly different. Multiple data sources, corresponding with the different camera angles, are fed into the same pre-trained network and embedding layer, before being the processed data is run through the anomaly detection algorithm separately. These detection streams are then combined naively using a simple boolean-OR combination, i.e., at the current timestep, if an anomaly is detected in any of the streams, it is regarded as an anomaly in the combined output.

Falling Dataset					
Method	Recall	Precision	F1-score	Accuracy	MCC
Uniform	0.49	0.49	0.46	0.50	0.009
Constant 1	0.50	0.38	0.43	0.23	N/A
Constant 0	0.50	0.12	0.19	0.77	N/A
Ours	0.76	0.82	0.79	0.86	0.57
AV Dataset					
Method	Recall	Precision	F1-score	Accuracy	MCC
Uniform	0.50	0.50	0.34	0.50	-0.003
Constant 1	0.50	0.01	0.01	0.01	N/A
Constant 0	0.50	0.49	0.50	0.99	N/A
Ours	0.58	0.57	0.57	0.99	0.15
Soccernet-v2 Dataset					
Method	Recall	Precision	F1-score	Accuracy	MCC
Uniform	0.50	0.50	0.43	0.50	-0.001
Constant 1	0.50	0.07	0.12	0.14	N/A
Constant 0	0.50	0.43	0.45	0.86	N/A
Ours	0.50	0.82	0.48	0.87	0.10

Table 4.1: Comparison of our system against baselines, with standard metrics and Matthews Correlation Coefficient (MCC). Each table corresponds to the results from a specific dataset. Bold indicates the best performance.

From this analysis, we can see that the system is performing well on the task of unsupervised anomaly detection. Specifically, we can observe that multiple data streams appear to lead to better results (Falling vs the other

two). It is apparent that the type of classifier that produces the input for the anomaly detector is important. The basic action recognition model used was best suited for the falling detection whereas the event in the other two datasets were not specifically part of the actions.

Using more specific models most probably would lead to better results, for example, a soccer or crime event-specific model for the respective dataset (or even combination of models capturing different aspects of the stream) [89]. In [89] we perform a deeper analysis of the AV and Soccernet datasets, examine the effect interval range has on the results, and manually evaluate the method on the unlabeled Njord dataset.

4.4.3 Detection of Slipping Events

“Slipping”, or the deliberate release, of dead or dying fish is a potentially illegal act, depending on the jurisdiction and the fish species being released. The reason why slipping can be illegal is because it makes management of fishing catches difficult. Essentially, a fishing vessel has a quota that allows them to fish a certain amount in order to keep the fishing industry sustainable, but if fish are released dead, and it is not logged, this can lead to massive overfishing. The economic incentive of illegally releasing caught fish is driven by the potential for a larger catch. By releasing a smaller fish and keeping only the larger ones, fishermen can extract a higher economic return from their catches.

In collaboration with KSAT³, we have developed a method for detecting fishing-related slipping events. Our approach combines satellite imagery, AIS, and fishing vessel activity logs such as catch notes and trip logs. We utilise these data sources in order to detect possible suspects that could have release fish, based on whether they have fished in the area, what type of fish they were catching, and whether they turned off their AIS [83].

³<https://www.ksat.no>

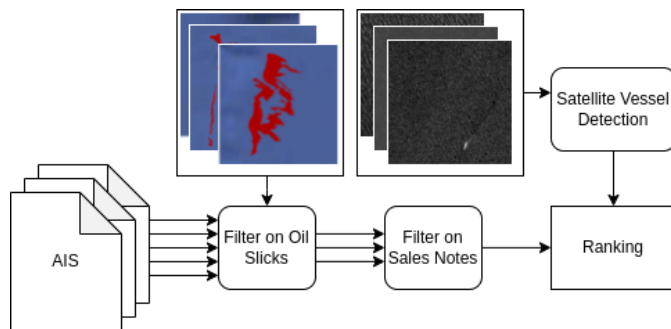


Figure 4.9: Pipeline showing how fishing vessels are filtered and ranked. The AIS positional data is filtered using the oil slicks and the sales notes to ensure that only vessels that have fished in the oil slick areas with pelagic gear. The vessel detection done to check whether they have turned off their AIS, in which case they are more suspect.

Oil Slick Detection

The oil slick detection and classifications used in this study were extracted from SAR images using semi-manual methods integrated in the KSAT analysis tool. The oil shape, contrast, and contextual information about ship tracks and positions are used when labelling the oil slick in order to mitigate the risk of false detection. False detection are look-alikes that could be e.g. low wind areas, algae, rain-cells, or ship-wakes that all have a dark signature similar to actual oil slicks.

The detection were provided by KSAT from a request by Norwegian Petroleum Directorate (NPD) in relation to a seepage (i.e. naturally occurring leaks of oil through sediments on the seafloor [125]) study in the Northern Barents Sea. Only the detection related to vessel activities are used and illustrated in this study. These detections are most likely related to fishing activities (fish oil), but could also be related to natural seeps either triggered by natural causes or through the fishing activities that are in direct contact with the seafloor. Additionally, these detections can also be wastewater or other pollution type from the vessels. Biogenic slicks, as fish oil, also dampens the sea roughness (capillary and short-gravity waves)

and appears dark in SAR images (see, e.g., [43]). 320 detections related to vessel activities from 2016-2021 are available for this study, but only the detections (119 detection) from 2021 are used, due to the AIS availability during that year. The detection, and what they look like in SAR imagery, are depicted in Figure 4.10.

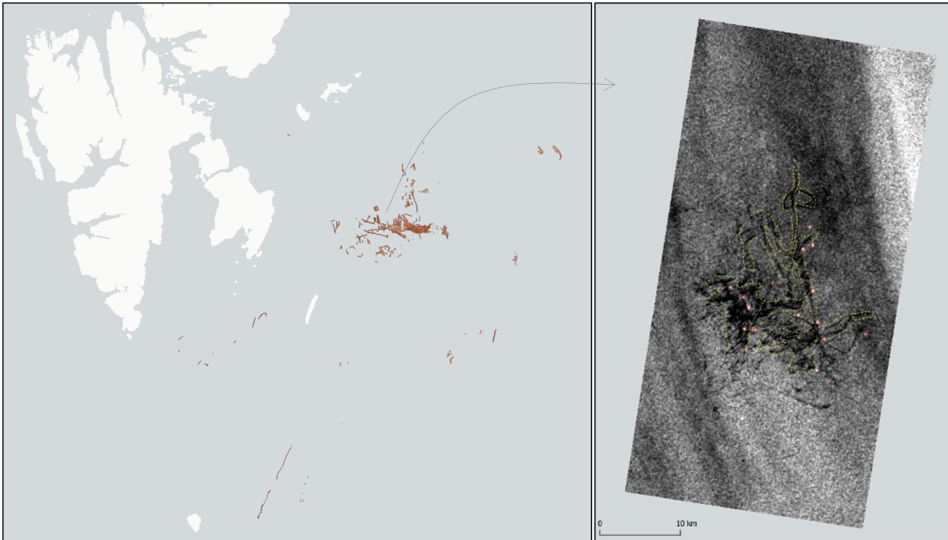


Figure 4.10: An overview of the detection south and east of Svalbard, and how oil slicks appear in SAR imagery. Contains modified Copernicus Sentinel data from 2021, processed by KSAT.

SAR sensors are in sun-synchronous polar orbits and acquire data in morning and/or afternoon (local time) over the Barents Sea, which occurs almost every day due to the high latitude.

Vessel Detection

Due to the illegality of purposely slipping, we conjecture there is a possibility of fishing vessels turning off their AIS systems during such events. Therefore, we specifically consider AIS paths that suddenly stop and correlate with satellite imagery to detect “dark” vessels that have done so. Fishing boats going dark is a well-known phenomenon when illegal commercial fishing activities are unfolding.

The satellite data we analyse is downloaded using the `eo-learn` framework. We have collected SAR data from Sentinel-1’s Interferometric Wide Swath (IW) and Extra-Wide Swath (EW) modes, with VV/VH and HH/HV polarisations, respectively. Patches are collected in $T \times 256 \times 256 \times C$ arrays, where T represents the different times the 256×256 patch has been sampled, and C is the number of polarisations (in our case, two). The IW and EW patches have a spatial resolution of $20m \times 22m$ per pixel and $40m \times 40m$ per pixel, respectively. The EW patches are upsampled to 20×20 using nearest neighbor interpolation to preserve pixel value information.

After downloading the patches, we had 139 IW patches and 204 EW patches, of varying temporal resolution. There are fewer of the IW patches due to the different coverage of the different modes [31]. The Sentinel-1 constellation consists of two satellites that have a repeat cycle of six days, but only one satellite is currently active making the repeat cycle 12 days [32]. However, both Sentinel-1b and Sentinel-1a were active for the data used in this study. In the Arctic, the repeat frequency is less than one day due to the orbits of the satellites [30]. We extract patches in the interval from one day prior to and until when the corresponding oil slick was detected.

We pretrain a Yolov5 [58] network using the SAR Ship dataset by Wang et al [129]. This dataset consists of SAR data from Gaofen-3 and Sentinel-1, both C-band sensors, resulting in similar SAR signatures for both vessel and oil. However, we only utilise the subset of images with the lowest resolutions (10×10 (GF-3 FSII), 20×22 (Sentinel-1 IW), and 25×25 (GF-3 QPSII)), due to us only having access to Sentinel-1 IW/EW data. This network is used for IW data.

For the EW data, the resolution is too coarse to apply the vessel detection network, so we utilise an adaptive thresholding on patches that are suspected to contain a boat. The threshold is dependent on the mean of the entire patch. This is determined by both the changes in time within the patches and the AIS paths, which we discuss further below. We apply

these methods on all respective patches and find the ones containing vessels. Both the Yolov5 network and the adaptive thresholding are applied on the respective patches on both channels individually. Then, a union of the sets of detection in both channels is done. The thresholding algorithm is assumed to be less accurate for the HH-polarised patch, due to ocean clutter, but the SAR Ship dataset contains both co- and dual-polarisation images, so the Yolov5 network is assumed to work well with both polarisation combinations.

In order to correlate the SAR-based vessel detection with the AIS data we interpolate between AIS points if needed, using the position, heading, and speed of the vessel. Similar to the vessel detection, we examine boats that are close to the oil slicks in the interval one day before to and including when the oil slicks were detected. Biogenic slicks, like fish oil, usually forms thin films on the surface (one molecule thick), and might not be detected the next day as a results of higher wind ($> 5\text{m/s}$) and weathering processes (e.g., mixing with water, evaporation) [2].

Filtering based on AIS, Sales Notes and Trip Logs

We utilise sales notes for filtering relevant vessels, i.e., vessels which use purse seine gear and catch pelagic fish. We focus on these types of fish, because these are rich in oil and thus release it more readily. Therefore, a release of dead pelagic fish has a higher chance of being detectable with satellite imagery. The sales notes consist of information about the catch that is manually logged during landing, e.g., when it was caught, where it was caught, what equipment was used, and the species distribution of the catch.

We focus on sales notes where different varieties of purse seines have been used, e.g., purse seines with/without lights and single/two boat seines. Accounting for the gear above, the remaining sales notes contain catches of typical pelagic fish: *clupea harengus*, *scomber scombrus*, *micromesistius*

poutassou, and *mallotus villosus*, as well as fish that can appear in pelagic zones occasionally: *pollachius virens*, *melanogrammus aeglefinus*, and *gadus morhua*.

Ranking algorithm and Evaluation

Due to the difficulty of this task, there exist few verified cases of slipping occurring. Therefore, we do not have any examples that can be utilised to weight the criteria that are applied in the ranking algorithm below. Ranking is used when multiple boats are potential suspects for a given oil slick. The ranking algorithm is a simple multi-criteria decision making algorithm that calculates a weighted sum for each vessel based on multiple criteria: time spent close to the oil slick, weighted by the distance from the oil slick polygon, if the self-reported fishing location is within an extended version of the same oil slick polygon, and whether they are determined to have turned off their AIS. These are calculated for each vessel in 1 below.

Algorithm 1 Calculating vessels' criteria for ranking

```

1: procedure CALCULATECRITERIA( $S, o$ )  ▷ calculate criteria for vessels
    $x$  in  $S$  around oil slick  $o$ 
2:    $A \leftarrow \text{zeros}(|S| \times 3)$   ▷ empty matrix  $A$ ,  $3 = \#$ criteria to compare
3:   for all  $x \in S$  do
4:      $i \leftarrow x.\text{idx}$ 
5:      $A_{i0} \leftarrow o.\text{contains}((x.\text{fishLoc})$   ▷ Check if the vessels fishing
       location is within the bounds of the oil slick.
6:      $A_{i1} \leftarrow \text{timeAroundO}(x.\text{ais}, o)$ 
7:      $A_{i2} \leftarrow \text{AISturnedOff}(x)$ 
8:   end for
9:   return  $A$ 
10: end procedure

```

The `timeAroundO` function calculates the time vessel x spent around an oil spill by first utilising the sub-sequence of AIS positions that start at the point where a vessel enters the extended polygon of the oil spill. Then, points are added by interpolation, so that the frequency of the points is at

two minutes (which is the highest frequency we have in our AIS dataset). Then the total amount of time until the vessel leaves the extended area is divided by the average distance from the actual (unextended) polygon. This causes boats that are close to the oil slick for a long period of time to be ranked higher. The `AISturnedOff` function checks whether an AIS path is not transmitted for more than 30 minutes, then looks at the satellite imagery to determine whether a given vessel still is within the area of an oil slick. This might indicate that the given vessel has turned off their AIS.

Using our approach we are able to filter potential suspects that might have caused an fish oil slick due to deliberate release of deceased fish. We start with the AIS data of a total of 1,794 vessels that have travelled in the given time frame. Then we filter based on the oil slicks by enlarging them by 5 kilometers in all directions (to account for potential drift of the oil slick, and inaccuracy of the AIS). We correlate the AIS data of all boats which have intersected with an oil slick within the 1-day time frame before and including the time of the oil slick detection. This reduces the potential suspects down to 132. Then the list is reduced to only 18 by excluding the fishing vessels that have not used purse seines or similar equipment in the area.

From the 303 patches resulting in 45,039 satellite imagery patches over time, only 38 of them contained a vessel. 16 can be associated with one of the 18 suspected vessels above. The remaining 22 do not correspond with any paths of the suspected vessels, so they might possibly be international vessels. This caveat is discussed further in the next subsection. The three suspected vessels that are not detected in any patches do not have paths that intersect with the patches in the time frame they were acquired.

Finally, we rank boats if there are multiple possible suspects around a given oil slick. We look at individual oil slicks, and the boats surrounding it within the time frame. If multiple boats have been close to the oil slick, we first look at whether the boat has fished there based on the sales notes.

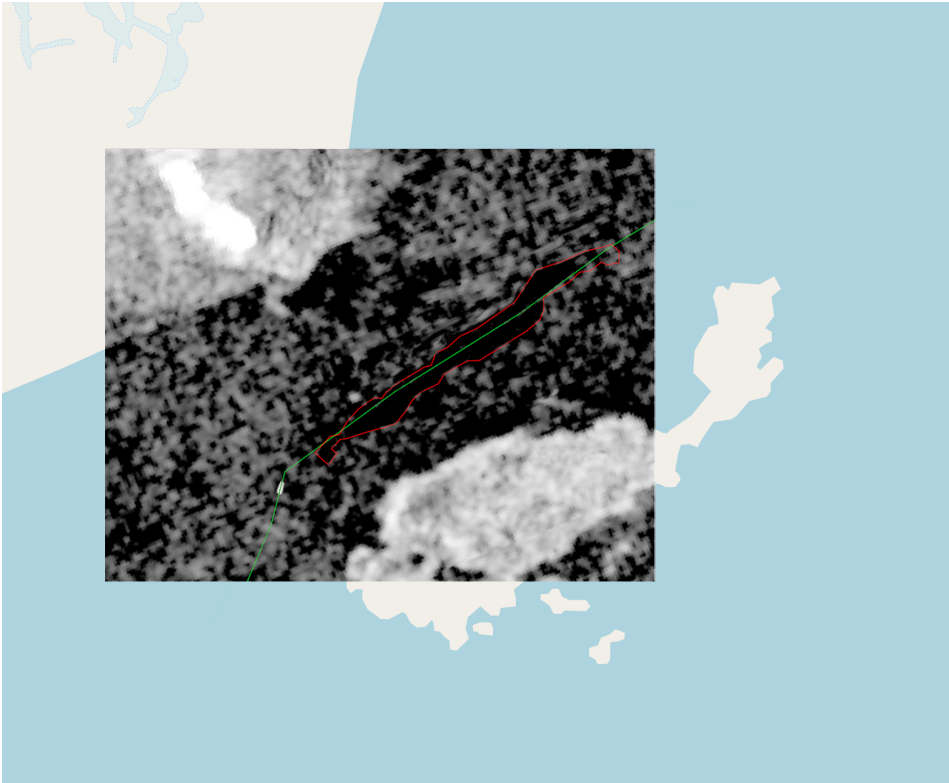


Figure 4.11: Illustration showing oil slick detection (darker area with red outline), with overlapping satellite imagery (EW mode) of a fishing vessel and its corresponding AIS path (green) between Edge Island and Half Moon Island at Spitzbergen. Contains modified Copernicus Sentinel data from 2021.

Then, if they have fished close by, we order them based on the time spent there. No vessels were determined to have turned off their AIS. In the end, 18 fishing vessels were determined to have possibly performed a slipping event. 16 of these were detected in satellite imagery. 11 of the 18 fished in the area of a corresponding oil slick.

Discussion

The AIS data only reflects Norwegian boats, which can be an issue due to the areas around Svalbard containing a lot of international fishing traffic.

This is because of the Treaty of Paris that allows vessels from multiple states to fish and hunt in the areas surrounding the Spitzbergen archipelago [80]. This might mean that a non-Norwegian fishing vessel can deliberately have released dead fish resulting in an oil slick which cannot be attributed due to lack of international AIS data. Additionally, there were drops in AIS transmissions. This is most likely due to the fact that our AIS data is potentially only sourced from ground station receivers. Therefore, because many of the oil slicks are far from land, the vessels might transmit their position to AIS satellites which doesn't appear in our AIS dataset.

Regarding the vessel detection using satellite imagery, a very small minority of the patches contain detected vessels. This might be due to the timing of the satellite passing over the area, which mostly occurs around 05:00 UTC (98%), with a few cases happening around 15:00 UTC (2%). Fishing vessels usually operate during the day. In this study, we only considered free satellite data which, as stated previously, has a limited temporal coverage. To make the satellite-based vessel detection a viable component for detecting dark vessels, one would need more frequent data. Therefore, an option is acquiring satellite data from commercial sources (like Radarsat-2, TerraSAR-X, Gaofen-3, ICEYE, and/or COSMO) as well.

An important point to note regarding the sales notes is that only Norwegian boats landing fish in Norway are included. Therefore, if a boat lands in Russia, the UK, or an EU country, they will not be included in the sales notes.

For future work, we would like to investigate additional data sources to deal with the limitations described above. Nevertheless, our proof-of-concept prototype demonstrates the potential for detect slipping events based on the combination of AIS, satellite imagery, and sales notes [83].

4.5 Competitions

During the research related to this thesis, we have organised two competitions/challenges, namely, the “FishAI: Sustainable Commercial Fishing” competition and the “NjordVid: Fishing Trawler Video Analytics Task” challenge, which was a part of MediaEval 2022⁴.

Organising scientific competitions serves several important purposes from a scientific perspective. One key benefit is that it helps to build a community around a new research direction or topic. By bringing together researchers from diverse backgrounds to compete and collaborate on a common challenge, competitions can facilitate the exchange of ideas and foster a sense of community within the field. This can be especially important for emerging areas of research that may not yet have a well-established community of practitioners.

Scientific competitions also promote open science and the sharing of datasets. By making datasets and research materials available to all competitors, competitions encourage transparency and the dissemination of knowledge. This not only allows for more rigorous evaluation and comparison of different approaches, but also enables others to build upon and extend previous work.

In addition, scientific competitions can serve as a way to distribute and evaluate new datasets, such as our Njord dataset. By organising a competition around a specific dataset, researchers can benchmark their approaches and identify the most effective methods for addressing a particular challenge. This can be especially useful for datasets that are too large or complex to be thoroughly analysed by a single research group.

⁴<https://multimediaeval.github.io/editions/2022/>

4.5.1 The FishAI Competition

FishAI was a competition hosted in collaboration with Norwegian Artificial Intelligence Research Consortium (NORA), which focused on developing solutions that can help optimise commercial fishing activities in Norway in a sustainable fashion. Norway is Europe's largest fishing and aquaculture nation, with a fishing zone of 2.1 million square meters and commercial vessels that land fish worth around NOK 20 billion every year. However, the daily migration patterns of fish can be difficult to predict, and fishing vessels often spend a lot of time and fuel searching for fish. The competition organisers believe that smarter use of publicly available data could help improve the sustainability of the fishing industry by reducing unnecessary transport distances and improving catch rates. Participants in the competition are invited to use data to develop solutions that can help fishermen make more informed decisions and optimise their fishing operations [82].

The competition was divided into three tasks:

1. Build a model that can predict which coordinates a vessel should prioritise in order to maximise the likelihood of catching a type of fish of your choosing.
2. Create a report of your analysis that can be read by experienced fishermen; an user-friendly visualisation that a captain can read to make a assessment of where the vessel should search for fish the next day.
3. Make a Sustainable Fishing Plan; a weekly plan that suggests the routes the fisherman should follow to optimise fish caught and fuel consumption.

The evaluation of the first task was based on the metrics in the Table 4.2, and the other two tasks were qualitatively evaluated fishermen and data scientists based on multiple criteria such as presentation, usefulness, correctness. The regression metrics are measured geographic coordinate

Evaluation		Teams				
Type	Metrics	Poseidon	Fishit	FishMaze	Intito	Craig Syms
Regression	MSE	47.91	51.43	75.40	118.88	412.55
	RMSE	6.92	7.17	8.68	10.90	20.31
	MAE	4.96	4.43	6.66	7.66	14.67
Classification	Precision	0.23	0.18	0.30	0.33	0.05
	Recall	0.24	0.28	0.26	0.37	0.07
	F1	0.21	0.21	0.15	0.27	0.06
	MCC	0.21	0.25	0.07	0.30	0.003
Avg. Distance	Kilometers	873.18	778.36	1098.48	1272.97	6552.44

Table 4.2: Table of Regression and Classification results from the participants of the FishAI competition. Abbreviated metrics are Mean Square Error (MSE), Root-Mean-Squared Error (RMSE), Mean Absolute Error (MAE), Matthews Correlation Coefficient (MCC). Best results in bold.

distances. 38 teams registered at the start of the competition, with five submitting results and four writing working notes after the competition was over. Most teams applied random forest or XGBoost approaches for their predictions. For tasks 2 and 3, most teams developed web applications for visualisation of where to catch fish and route planning, such as in Figure 4.12.

The prediction task was difficult, due to the coarseness of the location data in the sales notes, and was commented on by multiple participants. Nevertheless, the participants’ exploration and approaches were useful to determine which sources of data they deemed to be practical. The best performing group included AIS data for their predictions, which shows that combining data sources is necessary in the fishing domain in order to get the full picture.

4.5.2 MediaEval Benchmark - The NjordVid Task

The NjordVid task is part of the MediaEval Multimedia Benchmark 2022. It focused on evaluating algorithms for detecting events in surveillance videos from a fishing trawler. The task consists of two subtasks: detection of events on the boat, such as people moving and fish caught, and preserving

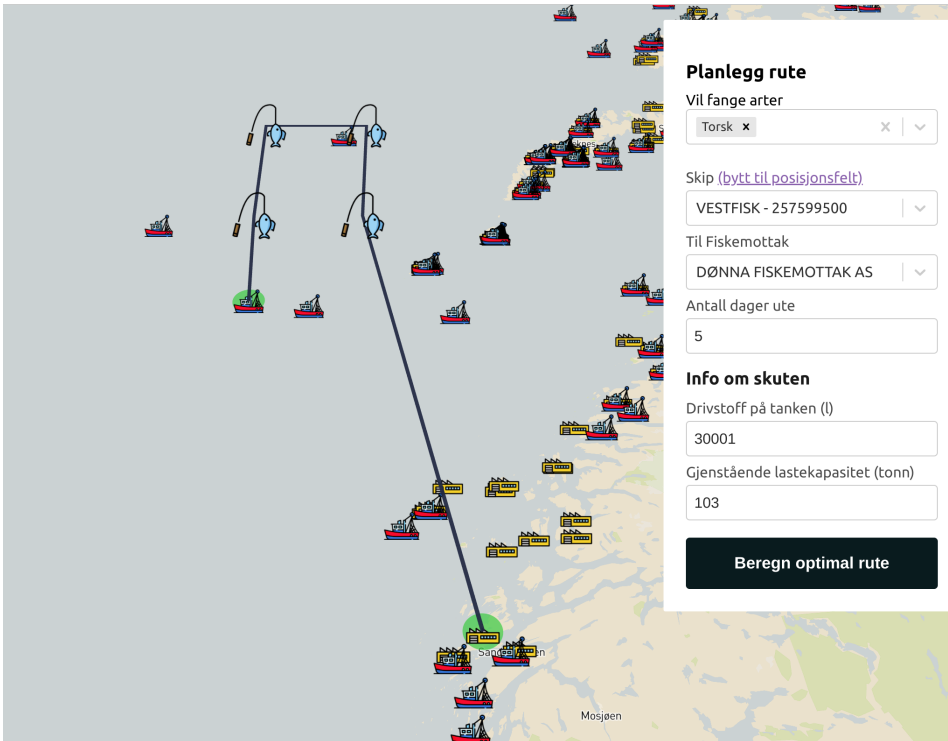


Figure 4.12: This illustrates the web application which shows a potential weekly plan for a fishing crew [20]. This is one of the solutions of the FishAI participants.

the privacy of the people working on the boat. The dataset used is the Njord dataset [90]. The goal of the task is to improve the accuracy of event detection algorithms in this context, and to develop methods for preserving the privacy of the people on the boat without hindering the analysis of the videos. The task is motivated by the need for surveillance on fishing vessels to maintain sustainable practices and prevent fish fraud through privacy-preserving methods [84].

MediaEval allows for two different paper submissions; regular papers that try to solve the tasks and compete for good scores, and Quest for Insight (Q4I) papers that try to get a deeper understanding about the dataset provided. We had one participant with a Q4I paper that looked at gener-

alisation failure and provided a method for detecting distributional shift, and evaluated several object detection models on their ability to generalise. The varying weather and lighting conditions in the Njord dataset lead to generalisation failure if they are not present in the training set.

Privacy-focused tasks and new tasks, such as ours, have historically had difficulty with getting participants in MediaEval. Therefore, when we publish the task again for 2023, we will expand our outreach and publicise the task more locally and through social media. We will also implement some baseline results which might entice potential participants by having something to beat.

4.5.3 Lessons Learned From The Competitions

Hosting competitions was beneficial for the thesis as it allowed for the exploration of existing data sources and brought in different perspectives and skills. It also helped to raise awareness and engage with a wider audience in the fishing domain, which is not well-known. The first competition, focused on optimizing fish catch, demonstrated the challenges posed by limited and coarse data sources in the fishing industry. The participants' exploration and approaches showed that combining data sources is necessary in order to get a full understanding of the situation. The second competition, focused on detecting events in surveillance videos, highlighted the need for privacy-preserving methods to maintain sustainable practices in the fishing industry and prevent fish fraud. It also highlighted the importance of expanding outreach and implementing baseline results. These competitions provided opportunities to engage with a wider audience, demonstrate the benefits of automation and digitization, and encourage collaboration between various stakeholders in the fishing industry.

Based on the results and feedback from the participants, there is potential for hosting more competitions in the future to further advance the project's goals. This could include expanding the scope of the competition

to other areas of the fishing industry, such as improving the management of fishing quotas, using video data from the fish processing pipeline on fishing vessels, or developing new techniques for reducing the environmental impact of fishing operations based on satellite imagery. In terms of the NjordVid task, the next step forward would be to attract more participants by creating some baselines and proposing more sub-tasks.

4.6 Summary

This chapter presented the Dutkat system with its main distributed system components. The edge computational node that is located on fishing vessels and the mainland hub that federates data from the edge and other sources, the data sources considered, and goes into detail on the analytical approaches developed and researched. The analysis focuses on (1) proving that the analysis of the data generated on the edge has to occur on the edge due to satellite communication constraints, (2) the description of an analysis pipeline for detection of anomalies and changing points, and (3) an approach for detecting the illegal slipping events using AIS, satellite imagery, and catch notes. Finally, we present and discuss the scientific challenges that we have organised related to this thesis.

Chapter 4. The Dutkat System

Chapter 5

Discussion

In this chapter, we discuss our contributions in relation to the objectives set in Chapter 1. We also discuss how the Dutkat system can support traceability in the seafood industry, and how it might be utilised in other use-cases outside of the fishing domain. Finally, we discuss lessons learned, and relevant ethical and legal considerations.

5.1 Contributions to Objectives

5.1.1 Sub-objective 1

Sub-objective 1 aims to develop an edge-based system which can process multiple sources of data using potentially resource-intensive AI algorithms that can assist in automatic documentation and detect fish fraud on a fishing vessel.

We have designed the edge computational node, which is a part of the Dutkat system, designed to be located on fishing vessels to provide automatic catch documentation and detect suspicious events if they occur [85]. Several components of the edge computational node have been developed, specifically, the distributed computational system [3] and distributed file system [91]. The first allows us to perform distributed DAG-structured

computations in a fault-tolerant manner, and the second enforces privacy-preserving access control on data collected on the vessel and processed in intermediate and resultant data.

5.1.2 Sub-objective 2

The aim of sub-objective 2 is to develop a subsystem on the mainland that can combine and correlate data related to fishing, such as AIS, catch sales notes, and satellite data, with data sent from fishing vessels in order to decide which vessels should be inspected.

A general mainland hub that can process and offer decision support has been designed in [85]. In our research, we have correlated multiple data sources that are available on the mainland, such as AIS, sales notes, satellite imagery data, etc. to determine whether fishing vessels have performed slipping events [83]. However, this approach can additionally be utilised to detect unreported transfer of catch between vessels, fishing outside their jurisdiction, etc.

5.1.3 Sub-objective 3

For sub-objective 3, data must be evaluated and acquired from both fishing vessels and mainland sources in order to better understand what data is most useful for automatic documentation of catch and detection of illegal fishing. This will require close collaboration with fishers and fishing sales organisations.

We have collaborated with control authorities, such as Norges Råfisklag and the Norwegian Fishing Directorate, to determine what data sources they use in their analysis of fishing vessels and their activities. In the paper presenting the Dutkat system [85], the sources used by control authorities on the mainland were determined to mainly be AIS positional data and historical sales notes data. With the edge computational node introduced, we also discuss approaches for utilising video data for catch analysis and suspicious activity recognition.

The Njord dataset [90], based on surveillance video data supplied by the Hermes Fishing Trawler, is the first dataset, to our knowledge, that depicts what occurs on a fishing vessel.

In our study on detecting slipping events [83], we process and correlate multiple data sources to detect whether a given fishing vessel might have caused a slipping event. We utilise AIS, sales notes, trip logs, and satellite imagery data.

Additionally, in the FishAI [88] competition described in Section 4.5.1, where several public data sources were suggested as useful sources of data (though the participants could employ other datasets). Sales notes, sea surface temperature, sea salinity, and moon phase data are datasets provided with the challenge.

5.1.4 Sub-objective 4

The aim of sub-objective 4 involves evaluating existing methods and developing new algorithms to address the variety of data sources in the fishing domain, specifically for cases where data is lacking or scenarios are unusual.

As stated previously, in the Dutkat paper [85], we briefly discuss existing approaches for fish species recognition and the challenges of detecting suspicious activity in surveillance videos.

Due to the lack of existing labelled data for suspicious activity on fishing vessels, we design an unsupervised pipeline for detecting anomalies/changing points [89]. This pipeline, feature extractors pretrained on general/unrelated data, can combine multiple data modalities to detect changing points. We evaluate the approach on multiple datasets.

We utilise CNNs for generating baselines for the Njord dataset [90] and for ship detection in the slipping paper [83]. However, in the slipping paper, we also apply simple thresholding techniques for ship detection for lower resolution modes of the satellite data.

5.1.5 Main Objective

The main objective of this project aims to develop a distributed edge-based system prototype to allow privacy-preserving analysis of sensor data on fishing vessels. The system will help fishers automatically document their catch and detect potential illegal activity, while anonymising the data and sending the local analysis results to the mainland for further analysis and visualisation for inspectors to make decisions. Currently, only 4% of the fishing fleet of Norway gets inspected, and the majority of these inspections occur during landing. With our system, the coast guard can intervene quickly after an illegal event has occurred on a fishing vessel.

We wanted to achieve this goal through the sub-objectives mentioned above. With the Dutkat paper [85], we designed the system and discussed related work and challenges with developing such a system and with the fishing domain in general. We developed components of the edge computational node [3, 91] and designed a pipeline for utilising data generated from the different data modalities on a fishing vessel to detect anomalies [89]. We contributed with the Njord dataset [90], which gives insight into the procedure and what events that occur on fishing vessels. We also developed an approach for detecting slipping events [83], which is an analysis method to be run on the mainland hub. Finally, we organised two scientific challenges, one for developing decision support for sustainable fishing for fishers [88], and the other for developing privacy-preserving solutions for analysis of video data generated on fishing vessels [86].

In our research, the events on fishing vessels as well as what is needed on the mainland has been explored. A variety of methods have been utilised in order to gain a comprehensive understanding of the fishing domain. By taking this approach, a broader range of insights and perspectives have been gained, which would not have been possible through a more narrow focus. A diverse range of data modalities have also been utilised, including

image and video data from on-board cameras and satellites, time-series AIS positional data, and tabular sales notes data. This has allowed for a more well-rounded analysis and a more complete understanding of what is needed to realise the Dutkat system.

5.2 Traceability

The seafood industry is worth \$164 billion annually [51] and employs over 50 million people around the world [28]. Despite its importance, the industry is plagued by issues of traceability. In particular, it is difficult to track the provenance of seafood from “boat to plate”. This is a major problem because it makes it difficult to ensure that seafood is safe and sustainable.

One way to improve traceability in the seafood industry is to use blockchain technology. Blockchain is a distributed database that allows for transparent and secure record-keeping. This makes it ideal for tracking the provenance of seafood. There are already a number of companies using blockchain to track seafood. Viant, for example, is using blockchain to track yellowfin tuna from Fiji. The company has developed a “digital passport” for each tuna, which includes information on the fish’s origin, catch date, and other important data. This passport is stored on a blockchain, which allows consumers to track the fish from boat to plate. Blockchain-based traceability systems have the potential to transform the seafood industry. They can help ensure that seafood is safe and sustainable. In addition, they can give consumers the peace of mind that comes with knowing the exact origins of their seafood.

However, blockchain technology can only ensure that the data input cannot be tampered with. Verifiable data needs to be input into the blockchain system for traceability to be viable. The Dutkat system can provide this verifiable data at the time of catch, which could then be stored in a blockchain system for traceability. We have previously worked on blockchain research [117, 116], and are looking into applying blockchain for traceability with Nofima¹.

¹<https://nofima.com/projects/blockchain-network-for-the-norwegian-food-industry>

5.3 Generalisability

We conjecture that Dutkat is based on a more general architecture that is applicable in different use-case scenarios. The properties for such use-cases include some sort of continuous sensory input at a (mobile) edge node for real-time or post-analysis. This edge node can be a plane, an army vehicle, or a modern car, but also even a human being like a policeman equipped with multi-media life-logging devices or a ski athlete (like in our work [87]). For example, in self-driving cars there are multiple safety features that require constant surveillance of both the physical environment, but also the driver and passengers. In order to ensure that the privacy of the user and others in the proximity of the car is maintained, the general architecture described will be further explored in such use-case scenarios. The versatility of the Dutkat architecture can also be applied in elderly care, where continuous monitoring of the elderly individuals can be performed in a privacy-preserving manner while providing valuable insights into their well-being and potential deviations from their daily routines due to e.g. dementia.

5.4 Digitalisation and sustainable fishing

During the research of this thesis, we determined that edge computation is a fitting approach for the fishing domain by verifying the infeasibility of sending large amounts of data, generated on fishing vessels, to the mainland for processing. We have also concluded that, for certain data sources, we can apply existing machine learning approaches, however for others we either need to collect more supervised data and train new models or introduce unsupervised approaches.

One of the challenges in developing a system for the fishing industry is determining which data sources to use. There are numerous data sources available, including vessel tracking data, catch reports, and more, but not all

5.5. Legal and Ethical Considerations

of these sources may be relevant or reliable for a given task. It is important to carefully evaluate and select the most useful data sources to ensure the accuracy and effectiveness of the system.

Another challenge is the lack of labelled data, e.g., for the slipping detection method, which makes it difficult to verify the performance of methods. Without a set of labelled data to use as a reference, it is difficult to accurately assess the accuracy and reliability of the methods being developed. This can be a significant hurdle in the development of any machine learning-based system.

In the future, we require even more collaboration with both fishers and the control authorities, both for determining more specifically what is required and to acquire verifiable labels that we can evaluate our methods against. There is also a need for more research into existing data sources that we might utilise, both to assist fishers' more in catch documentation and potential feedback while they're at sea, and to detect abnormal events such as fish fraud or accidents. We would also like to look more into what more events that can be detected using satellite data, as IUU fishing is a global problem, and introducing our edge-based system everywhere is not feasible, at least on a short timescale.

5.5 Legal and Ethical Considerations

Introduction of surveillance on board fishing vessels raises several ethical considerations, particularly in regards to data privacy and protection. The General Data Protection Regulation (GDPR) in the European Union (EU) and European Economic Area (EEA), which Norway is a part of, requires that personal data is collected and used in a transparent and lawful manner, and that individuals have the right to access and control their personal data. This becomes particularly relevant when considering the use of surveillance technology on fishing vessels, as it may collect personal data of the crew

members such as their movement, location and activities on the vessel.

When introducing surveillance on fishing vessels, it is also important to consider the Norwegian Constitution paragraph 102 [1] and European Convention on Human Rights (ECHR) article 8 [53], which both guarantee the right to privacy. This means that any surveillance measures must be justified and proportional, and should not excessively intrude on the privacy of individuals. Additionally, the use of surveillance technology must be in compliance with Norwegian laws and regulations related to data protection and privacy, such as the Personal Data Act. It is essential that any surveillance measures are implemented in a transparent and lawful manner, and that individuals have the right to access and control their personal data.

Furthermore, the EU is proposing new regulations on the use of AI which aims to ensure that AI systems are safe, transparent, and respect fundamental rights. This includes provisions for human oversight, data governance, and liability for AI systems, which will have an impact on the use of AI-powered surveillance on fishing vessels [34].

Therefore, the implementation of surveillance on board fishing vessels must be done in compliance with relevant data protection laws and regulations, including GDPR and the proposed EU AI regulations, to ensure that the privacy and rights of the crew members are protected and respected. Additionally, the use of surveillance technology on fishing vessels must be balanced against the potential negative impacts on the privacy and rights of the crew, and the potential for discrimination and exploitation.

Our system, which utilises machine learning to detect illegal activities and assist in catch documentation, is a better alternative as compared to traditional surveillance methods. Machine learning algorithms can analyse large amounts of data and identify patterns that may indicate illegal activities, such as overfishing or use of prohibited gear. Furthermore, the use of machine learning can reduce human bias and error in the detection process. Additionally, our system can assist in catch documentation by automating

the process of identifying and counting fish species, reducing the potential for human error and increasing efficiency.

Moreover, our system can also provide real-time monitoring, making it possible to detect and respond to illegal activities more quickly. Also, the system can be integrated with vessel tracking technology and other sensor systems, providing a comprehensive view of fishing activities and allowing for more accurate and effective enforcement of fishing regulations. Additionally, our system can also be programmed to take into account local regulations and conservation measures, ensuring that it operates in compliance with relevant laws and guidelines.

Overall, our system offers a number of advantages over traditional surveillance methods. It can detect illegal activities more effectively, assist in catch documentation, and reduce human error, while also providing a cost-efficient, automated and real-time monitoring. Furthermore, the system can also be programmed to be compliant with the local regulations and conservation measures. In order to be effective at reducing IUU fishing nationally, however, new laws that would require our system to be present and functional on a given fishing vessel would need to be introduced.

5.6 Summary

In order to meet the sub-objectives of developing an edge-based system for automatic catch documentation and fraud detection, a subsystem for the mainland to combine and correlate data related to fishing, and evaluating and developing methods for analysing various data sources in the fishing domain, we made several contributions. Our system utilises edge computing to process and analyse data at the source, allowing for real-time catch documentation and fraud detection. We also developed a mainland subsystem that combines and correlates data from various sources, including vessel tracking data, catch reports, and market information. Through the

Chapter 5. Discussion

evaluation of various data sources, we identified those that were most useful for our task and implemented methods for effectively analysing this data.

In addition to addressing the sub-objectives, our system also facilitates the introduction of traceability into the fishing industry. By providing accurate and up-to-date documentation of catches, our system allows for greater transparency and accountability in the industry. We also discuss the potential for generalising our system to other domains beyond fishing. With the ability to process and analyse data at the edge and combine and correlate data from various sources, our system has the potential to be applied to a range of use-cases where traceability and data analysis are important.

Finally, we discuss our journey for utilising digitalisation to enforce sustainable fishing. We discuss lessons learned, what went well, what was challenging, and what will be done in future work. Additionally, we discuss ethical and legal consideration which need to be taken into account when introducing surveillance on fishing vessels.

Chapter 6

Concluding Remarks

In this concluding chapter, we summarise the main contributions of our thesis, which focuses on the design and development of Dutkat, an edge-based distributed system for privacy-preserving catch documentation and detection of illegal activities in the fishing industry. Through our work, we have made several key contributions that contribute to the field of sustainable fishing and data privacy preservation. After the conclusion, we present potential future research directions that could be investigated.

Our initial hypothesis was the following:

It is possible to develop a digital inspector that can be situated on fishing vessels for automatic documentation of catch and privacy-preserving surveillance in cases of fish fraud.

6.1 Conclusion

In this thesis, we have presented our work on the design and development of Dutkat, an edge-based distributed system for privacy-preserving catch documentation and detection of illegal activities in the fishing industry. Through our research, we have aimed to address the challenges of sustainable fishing and data privacy preservation in the industry.

We have presented the design of Dutkat, an innovative edge-based distributed system that can be used to improve catch documentation and detect illegal activities in the fishing industry. It is presented as an alternative to naive surveillance solutions that have been proposed. The system utilises machine learning to analyse large amounts of data and identify patterns that may indicate illegal activities, such as quota exploitation, overfishing, use of prohibited gear, or illegal discard of catch or bycatch. Additionally, the system can assist in catch documentation by automating the process of identifying and counting fish species, reducing the potential for human error and increasing efficiency.

Moreover, we have developed various components of the Dutkat system, and evaluated their effectiveness through experimentation. We have also explored existing data relevant to the thesis and collected new data from unique sources, which has been released openly for further research. Furthermore, we have organised multiple AI competitions to find solutions that lead to sustainable fishing and privacy preservation for fishing crews.

We have also discussed the importance of compliance-by-design perspective, ensuring the Dutkat system is in compliance with data protection laws and regulations such as GDPR and the Norwegian Constitution paragraph 102. This is particularly important when considering the use of surveillance technology on fishing vessels, as it may collect personal data of the crew members such as their movement, location, and activities on the vessel.

In conclusion, this thesis presents a comprehensive approach to address the challenges of sustainable fishing and data privacy preservation in the fishing industry, through the design and development of the Dutkat system. This confirms our hypothesis that *it is possible to develop a digital inspector that can be situated on fishing vessels for automatic documentation of catch and privacy-preserving surveillance in cases of fish fraud*. Our system offers a number of advantages over traditional surveillance methods, including automatic, privacy-preserving detection of illegal activities, assistance in

catch documentation, and reducing human error, while also providing cost-efficient, automated and real-time monitoring. Furthermore, the system can also be programmed to be compliant with the local regulations and conservation measures. We conjecture that the Dutkat system can be a valuable tool in promoting sustainable fishing practices while also ensuring the privacy and rights of fishing crews are protected.

6.2 Future Work

From a research perspective, the next steps forward involve further work to improve the accuracy of catch documentation and fraud detection in the fishing industry. This includes research into fish classification and biomass estimation, as well as exploring new data sources, algorithms, and machine learning techniques for edge computing. Despite the datasets described in Section 2.3.3, more research is needed to improve the accuracy of fish classification and biomass estimation algorithms in order to ensure the system can accurately document catches. Additionally, privacy-preserving solutions for video surveillance data need to be developed to ensure that the privacy of individuals is protected while enabling the use of this data for event detection and other purposes. Evaluating the potential for traceability in other industries, such as agriculture or supply chain management, is also an interesting area of future research.

From an industry perspective, deploying a prototype of Dutkat on a fishing trawler and gathering feedback from the crew and stakeholders are crucial. This will be done by fall 2023. The fishing industry must also explore new technologies, such as precision fishing techniques, monitoring systems, and autonomous vessels, and their potential for adoption and integration in a sustainable and cost-effective way. Moreover, strategies for adapting to and mitigating the impacts of climate change on the fishing industry need to be developed to ensure the long-term sustainability of the industry.

Chapter 6. Concluding Remarks

From the perspective of governing bodies, it is important to continue to support research into catch documentation and fraud detection in the fishing industry. This includes funding research into privacy-preserving solutions for video surveillance data and supporting the development of new technologies that can improve the accuracy of catch documentation and prevent fraud. Additionally, governing bodies must play a role in developing and implementing strategies for adapting to and mitigating the impacts of climate change on the fishing industry, including changes in fish populations and habitat loss, to ensure the long-term sustainability of the industry. This may involve the development of regulations or incentives for the industry to adopt sustainable practices, and the implementation of policies to support the growth and development of the industry in a responsible manner. Governing bodies play a crucial role in ensuring the sustainability of the fishing industry and must introduce legislation that makes it mandatory for fishing vessels to utilise a system for automatic catch documentation to promote transparency and accountability, enhance the accuracy of resource management, and prevent overfishing and illegal fishing activities.

Bibliography

- [1] “§102”. In: *Grunnloven, Updated in 2014* (1814). URL: https://lovdata.no/dokument/NL/lov/1814-05-17/KAPITTEL_6#shareModal.
- [2] W. Alpers, B. Holt, and K. Zeng. “Oil spill detection by imaging radars: Challenges and pitfalls”. In: *Remote Sensing of Environment* 201 (Sept. 2017), pp. 133–147.
- [3] Joakim Aalstad Alslie et al. “Aika: A Distributed Edge System for AI Inference”. In: *Big Data and Cognitive Computing* 6.2 (2022). ISSN: 2504-2289. DOI: 10.3390/bdcc6020068. URL: <https://www.mdpi.com/2504-2289/6/2/68>.
- [4] Mutasem K. Alsmadi and Ibrahim Almarashdeh. “A survey on fish classification techniques”. In: *Journal of King Saud University - Computer and Information Sciences* (2020). ISSN: 1319-1578. DOI: <https://doi.org/10.1016/j.jksuci.2020.07.005>. URL: <https://www.sciencedirect.com/science/article/pii/S1319157820304195>.
- [5] *Anonymized AIS data*. [Online; accessed 1-September-2022]. URL: <https://globalfishingwatch.org/data-download/datasets/public-training-data-v1>.
- [6] Martin Arjovsky. “Out of distribution generalization in machine learning”. PhD thesis. New York University, 2020.

Bibliography

- [7] The Norwegian Data Protection Authority. *Artificial intelligence and privacy*. 2018. URL: <https://www.datatilsynet.no/globalassets/global/english/ai-and-privacy.pdf>.
- [8] Edouard Auvinet et al. *Multiple cameras fall dataset*. 2010. URL: <http://www.iro.umontreal.ca/~labimage/Dataset/>.
- [9] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. “Multimodal machine learning: A survey and taxonomy”. In: *IEEE transactions on pattern analysis and machine intelligence* 41.2 (2018), pp. 423–443.
- [10] Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*. Vol. 1. MIT press Cambridge, MA, USA, 2017.
- [11] Nora Thorp Bjørnstad. *Økokrim: Fiskerikriminalitet en av de største truslene*. URL: <https://www.vg.no/nyheter/innenriks/i/mr284q/oekokrim-fiskerikriminalitet-en-av-de-stoerste-truslene>.
- [12] Ane Blázquez-García et al. “A review on outlier/anomaly detection in time series data”. In: *CoRR* abs/2002.04236 (2020). arXiv: 2002.04236. URL: <https://arxiv.org/abs/2002.04236>.
- [13] Raghavendra Chalapathy and Sanjay Chawla. “Deep learning for anomaly detection: A survey”. In: *arXiv preprint arXiv:1901.03407* (2019).
- [14] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, eds. *Semi-Supervised Learning*. The MIT Press, 2006. ISBN: 9780262033589. URL: <http://dblp.uni-trier.de/db/books/collections/CSZ2006.html>.
- [15] University of Copenhagen. *The electronic monitoring of fishermen ensures that cod aren’t tossed overboard*. [Online; accessed 21-August-2022]. URL: https://news.ku.dk/all_news/2020/07/the-

electronic-monitoring-of-fishermen-ensures-that-cod-arent-tossed-overboard/.

- [16] Christopher Costello et al. “The future of food from the sea”. In: *Nature* 588.7836 (2020), pp. 95–100.
- [17] Alberto Maximiliano Crescitelli, Lars Christian Gansel, and Houxiang Zhang. “NorFisk: fish image dataset from Norwegian fish farms for species recognition using deep neural networks”. In: *Modeling Identification and Control* 42.1 (2021), pp. 1–16.
- [18] George Cutter, Kevin Stierhoff, and Jiaming Zeng. “Automated detection of rockfish in unconstrained underwater videos using haar cascades and a new image dataset: labeled fishes in the wild”. In: *2015 IEEE Winter Applications and Computer Vision Workshops*. IEEE. 2015, pp. 57–62.
- [19] Alexander D’Amour et al. “Underspecification Presents Challenges for Credibility in Modern Machine Learning”. In: *CoRR* abs/2011.03395 (2020). arXiv: 2011.03395. URL: <https://arxiv.org/abs/2011.03395>.
- [20] Jonas Dammen et al. “FishAI: The Lodestar fishing platform”. In: *Nordic Machine Intelligence* 2.2 (2022), pp. 10–12.
- [21] Adrien Deliege et al. *Socccernet-v2*. 2021. URL: <https://socccernet.org>.
- [22] Adrien Deliege et al. “SoccerNet-v2: A Dataset and Benchmarks for Holistic Understanding of Broadcast Soccer Videos”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2021, pp. 4508–4519.
- [23] Peter J. Denning et al. “Computing as a discipline”. In: *Computer* 22.2 (1989), pp. 63–70.

Bibliography

- [24] United Nations Office on Drugs and Crime. “Fisheries Crime”. In: (2016).
- [25] Sandeep Dsouza et al. “Amadeus: Scalable, Privacy-Preserving Live Video Analytics”. In: *arXiv preprint arXiv:2011.05163* (2020).
- [26] Schahram Dustdar and Ilir Murturi. “Towards Distributed Edge-based Systems”. In: *2020 IEEE Second International Conference on Cognitive Machine Intelligence (CogMI)*. 2020, pp. 1–9. DOI: 10.1109/CogMI50398.2020.00021.
- [27] *Electronic Fisheries Technology for vessels*. July 2022. URL: <https://teem.fish/vessels/>.
- [28] *Employment in fisheries and aquaculture*. URL: <https://www.fao.org/3/cc0461en/online/sofia/2022/fisheries-aquaculture-employment.html>.
- [29] Terje Engø. *AIS avslørte fiskefusk*. [Online; accessed 1-September-2022]. URL: <https://www.kystmagasinet.no/ais-fiskefusk-landingsforskriften/ais-avslorte-fiskefusk/973900>.
- [30] ESA. *Revisit and Coverage*. [Online; accessed 21-April-2022]. URL: <https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-1-sar/revisit-and-coverage>.
- [31] ESA. *Sentinel-1 Observation Scenario*. [Online; accessed 21-April-2022]. URL: <https://sentinel.esa.int/web/sentinel/missions/sentinel-1/observation-scenario>.
- [32] *ESA news*. <https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-1/news>. Accessed: 2022-05-02.
- [33] Rick van Essen et al. “Automatic discard registration in cluttered environments using deep learning and object tracking: class imbalance, occlusion, and a comparison to human review”. In: *ICES Journal of Marine Science* 78.10 (Nov. 2021), pp. 3834–3846. ISSN: 1054-3139.

- DOI: 10.1093/icesjms/fsab233. eprint: <https://academic.oup.com/icesjms/article-pdf/78/10/3834/41772276/fsab233.pdf>. URL: <https://doi.org/10.1093/icesjms/fsab233>.
- [34] Commission European. *Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS*. URL: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%5C%3A52021PC0206>.
- [35] European Data Protection Supervisor. “EDPS Guidelines on assessing the proportionality of measures that limit the fundamental rights to privacy and to the protection of personal data”. In: (2019). URL: https://edps.europa.eu/sites/edp/files/publication/19-12-19_edps_proportionality_guidelines1_en.pdf.
- [36] FAO. *The State of World Fisheries and Aquaculture 2020. Sustainability in action*. <https://doi.org/10.4060/ca9229en>. 2020.
- [37] FAO. *What is IUU fishing?* [Online; accessed 21-August-2022]. URL: <https://www.fao.org/iuu-fishing/background/what-is-iuu-fishing/en/>.
- [38] Mark Fisher et al. “Motion stereo at sea: Dense 3D reconstruction from image sequences monitoring conveyor systems on board fishing vessels”. In: *IET Image Processing* n/a.n/a (). DOI: <https://doi.org/10.1049/ipr2.12636>. eprint: <https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/ipr2.12636>. URL: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ipr2.12636>.
- [39] Nærings- og fiskeridepartementet. “Forskrift om posisjonsrapportering og elektronisk rapportering for norske fiske- og fangstfartøy”. In:

Bibliography

- (). URL: <https://lovdata.no/dokument/SF/forskrift/2009-12-21-1743>.
- [40] Alem Fitwi et al. “Privacy-Preserving Surveillance as an Edge Service Based on Lightweight Video Protection Schemes Using Face De-Identification and Window Masking”. In: *Electronics* 10.3 (2021), p. 236.
- [41] Geoff French et al. “Deep neural networks for analysis of fisheries surveillance video and automated monitoring of fish discards”. In: *ICES Journal of Marine Science* 77.4 (Aug. 2019), pp. 1340–1353. ISSN: 1054-3139. DOI: 10.1093/icesjms/fsz149. eprint: <https://academic.oup.com/icesjms/article-pdf/77/4/1340/33649382/fsz149.pdf>. URL: <https://doi.org/10.1093/icesjms/fsz149>.
- [42] *FUSE - The Linux Kernel documentation*. <https://www.kernel.org/doc/html/latest/filesystems/fuse.html>. Accessed: 2021-08-30.
- [43] M. Gade et al. “Imaging of biogenic and anthropogenic ocean surface films by the multifrequency/multipolarization SIR-C/X-SAR”. In: *J. Geophys. Res. Oceans* 103.C9 (Aug. 1998), pp. 18851–18866.
- [44] Antonio-Javier Gallego, Antonio Pertusa, and Pablo Gil. “Automatic Ship Classification from Optical Aerial Images with Convolutional Neural Networks”. In: *Remote Sensing* 10.4 (2018). ISSN: 2072-4292. DOI: 10.3390/rs10040511. URL: <https://www.mdpi.com/2072-4292/10/4/511>.
- [45] Rafael Garcia et al. “Automatic segmentation of fish using deep learning with application to fish size measurement”. In: *ICES Journal of Marine Science* 77.4 (2020), pp. 1354–1366.
- [46] Anders Tungeland Gjerdrum. “Diggi: A Distributed Serverless Runtime for Developing Trusted Cloud Services”. In: (2020).

- [47] Cathal Gurrin, Tjalve Aarflot, and Dag Johansen. “GARDI : A Self-Regulating Framework for Digital Libraries”. In: *2009 Ninth IEEE International Conference on Computer and Information Technology* 1 (2009), pp. 305–310.
- [48] Gunnar Hartvigsen and Dag Johansen. “Co-operation in a distributed artificial intelligence environment—the stormcast application”. In: *Engineering Applications of Artificial Intelligence* 3.3 (1990), pp. 229–237.
- [49] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.
- [50] Kaiming He et al. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.
- [51] Jason Holland. *Rabobank: Global seafood trade value rebounds to USD 164 billion*. May 2022. URL: <https://www.seafoodsource.com/news/supply-trade/rabobank-global-seafood-trade-value-rebounds-to-usd-164-billion>.
- [52] Xiyue Hou et al. “FUSAR-Ship: Building a high-resolution SAR-AIS matchup dataset of Gaofen-3 for ship detection and recognition”. In: *Science China Information Sciences* 63.4 (2020), pp. 1–19.
- [53] European Court of Human Rights. *Guide on Article 8 of the European Convention on Human Rights*. Strasbourg, 2022. URL: https://www.echr.coe.int/documents/guide_art_8_eng.pdf.
- [54] *Hva er fangstid-programmet?* URL: <https://www.fiskeridir.no/Yrkesfiske/fangstid/hva-er-fangstid-programmet>.
- [55] *iFarm*. URL: <https://www.cermaq.no/ifarm>.

Bibliography

- [56] Odd Emil Ingebrigtsen. *AIS kan manipuleres*. [Online; accessed 1-September-2022]. URL: <https://www.lofotposten.no/ais-kan-manipuleres/o/5-29-686906>.
- [57] *Iridium Certus 200*. <https://www.iridium.com/services/iridium-certus-200/>. Accessed: 2021-09-08.
- [58] Glenn Jocher. *ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements*. <https://github.com/ultralytics/yolov5>. Version v3.1. Oct. 2020. DOI: 10.5281/zenodo.4154370. URL: <https://doi.org/10.5281/zenodo.4154370>.
- [59] Dag Johansen. “A distributed approach to the design of applications”. In: *Proceedings of ICCI'93: 5th International Conference on Computing and Information*. IEEE. 1993, pp. 195–201.
- [60] Dag Johansen and Gunnar Hartvigsen. “Convenient abstractions in StormCast applications”. In: *Proceedings of the 6th workshop on ACM SIGOPS European workshop: Matching operating systems to application needs*. 1994, pp. 11–16.
- [61] Dag Johansen et al. “Using Software Design Patterns to Build Distributed Environmental Monitoring Applications”. In: (1997).
- [62] Håvard Johansen, Cathal Gurrin, and Dag Johansen. “Towards Consent-Based Lifelogging in Sport Analytic”. In: *MultiMedia Modeling*. Ed. by Xiangjian He et al. Cham: Springer International Publishing, 2015, pp. 335–344. ISBN: 978-3-319-14442-9.
- [63] Jackson Kamiri and Geoffrey Mariga. “Research Methods in Machine Learning: A Content Analysis”. In: *International Journal of Computer and Information Technology(2279-0764)* 10 (Mar. 2021), pp. 2279–0764. DOI: 10.24203/ijcit.v10i2.79.
- [64] Will Kay et al. *The Kinetics Human Action Video Dataset*. 2017. arXiv: 1705.06950 [cs.CV].

- [65] Mingyu Kim et al. “Deep learning in medical imaging”. In: *Neurospine* 16.4 (2019), p. 657.
- [66] Kirstie Ball. “Electronic Monitoring and Surveillance in the Workplace. Literature Review and Policy Recommendations”. In: *Publications Office of the European Union* (2019).
- [67] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. “Revisiting Self-Supervised Visual Representation Learning”. In: *CoRR* abs/1901.09005 (2019). arXiv: 1901.09005. URL: <http://arxiv.org/abs/1901.09005>.
- [68] Dmitry A Konovalov et al. “Automatic weight estimation of harvested fish from images”. In: *2019 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE. 2019, pp. 1–7.
- [69] Hildegard Kuehne et al. “HMDB: a large video database for human motion recognition”. In: *Proceedings of the 2011 International conference on computer vision*. IEEE. 2011, pp. 2556–2563.
- [70] Andrey A Kurekin et al. “Operational monitoring of illegal fishing in Ghana through exploitation of satellite earth observation and AIS data”. In: *Remote Sensing* 11.3 (2019), p. 293.
- [71] Kystverket. *AIS Norge*. [Online; accessed 1-September-2022]. URL: <https://www.kystverket.no/navigasjonstjenester/ais/ais-artikkelside/>.
- [72] Labelbox. *Labelbox*. 2022. URL: <https://labelbox.com> (visited on 02/15/2022).
- [73] Bertrand Le Gallic and Anthony Cox. “An economic analysis of illegal, unreported and unregulated (IUU) fishing: Key drivers and possible solutions”. In: *Marine Policy* 30.6 (2006), pp. 689–695.

Bibliography

- [74] Jason S Link and Reg A Watson. “Global ecosystem overfishing: Clear delineation within real limits to production”. In: *Science Advances* 5.6 (2019), eaav0474.
- [75] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation forest”. In: *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.
- [76] Zikun Liu et al. “A high resolution optical satellite image dataset for ship recognition and some new baselines”. In: *International conference on pattern recognition applications and methods*. Vol. 2. SciTePress, 2017, pp. 324–331.
- [77] Alihan Mermer, TÜRK Meral, and Zafer Tosunoğlu. “Occupational health and safety in large-scale fishing vessels registered in Aegean ports”. In: *Ege Journal of Fisheries and Aquatic Sciences* 39.1 (2022), pp. 18–23.
- [78] Aristides Milios et al. “Automatic Fusion of Satellite Imagery and AIS data for Vessel Detection”. In: *2019 22th International Conference on Information Fusion (FUSION)*. 2019, pp. 1–5.
- [79] Ministry of Trade, Industry and Fisheries. “Framtidens Fiskerikontroll”. In: *NOU 19:21* (2019).
- [80] League of Nations. “Treaty concerning the Archipelago of Spitsbergen”. In: *League of Nations Treaty Series, Vol. 2, 8-19* (1920).
- [81] *New trawl technology supports fish sorting*. URL: <https://techtransfer.no/en/fishing-and-aquaculture/scantrol-deep-vision/>.
- [82] NORA. *Fishai: Sustainable commercial fishing*. URL: <https://www.nora.ai/competition/fishai-dataset-competition/>.
- [83] T. S. Nordmo et al. “Detection of Commercial Fishing-related Slipping Events using Multimodal Data”. In: *2022 IEEE International Symposium on Multimedia (ISM)*. Los Alamitos, CA, USA: IEEE

- Computer Society, Dec. 2022, pp. 155–156. DOI: 10.1109/ISM55400.2022.00032. URL: <https://doi.ieeecomputersociety.org/10.1109/ISM55400.2022.00032>.
- [84] Tor-Arne S Nordmo. *Njordvid: Fishing trawler video analytics task*. Nov. 2022. URL: <https://multimediaeval.github.io/editions/2022/tasks/njord/>.
- [85] Tor-Arne S. Nordmo et al. “Dutkat: A Multimedia System for Catching Illegal Catchers in a Privacy-Preserving Manner”. In: *Proceedings of the 2021 Workshop on Intelligent Cross-Data Analysis and Retrieval*. ICDAR '21. Taipei, Taiwan: Association for Computing Machinery, 2021, pp. 57–61. ISBN: 9781450385299.
- [86] Tor-Arne S. Nordmo et al. “NjordVid: A Fishing Trawler Video Analytics Task”. In: *Proceedings of CEUR Multimedia Benchmark Workshop (MediaEval)*. 2022.
- [87] Tor-Arne Schmidt Nordmo et al. “Arctic HARE: A Machine Learning-based System for Performance Analysis of Cross-country Skiers”. In: *Proceedings of the 29th ACM International Conference on Multimedia Modeling*. MMM '23. Bergen, Norway: Association for Computing Machinery, 2022.
- [88] Tor-Arne Schmidt Nordmo et al. “FishAI: Sustainable Commercial Fishing”. In: *Nordic Machine Intelligence 2.2 (2022)*, pp. 1–3.
- [89] Tor-Arne Schmidt Nordmo et al. “Fishing Trawler Event Detection: An important step towards digitization of sustainable fishing”. In: *2023 International Conference on Applied Artificial Intelligence (ICAPAI)*. 2023, pp. 1–8.
- [90] Tor-Arne Schmidt Nordmo et al. “Njord: A Fishing Trawler Dataset”. In: *MMSys '22*. Athlone, Ireland: Association for Computing Machinery, 2022, pp. 1–6. ISBN: 9781450384346. DOI: 10.1145/3458305.3463373. URL: <https://doi.org/10.1145/3458305.3463373>.

Bibliography

- [91] Aril Bernhard Ovesen et al. “File System Support for Privacy-Preserving Analysis and Forensics in Low-Bandwidth Edge Environments”. In: *Information* 12.10 (2021). ISSN: 2078-2489. DOI: 10.3390/info12100430. URL: <https://www.mdpi.com/2078-2489/12/10/430>.
- [92] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. “Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features”. In: *NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics*. 2018.
- [93] Jaeyoon Park et al. “Illuminating dark fishing fleets in North Korea”. In: *Science Advances* 6.30 (2020), eabb1197. DOI: 10.1126/sciadv.abb1197. eprint: <https://www.science.org/doi/pdf/10.1126/sciadv.abb1197>. URL: <https://www.science.org/doi/abs/10.1126/sciadv.abb1197>.
- [94] Jaeyoon Park et al. “Illuminating dark fishing fleets in North Korea”. In: *Science Advances* 6 (July 2020), eabb1197. DOI: 10.1126/sciadv.abb1197.
- [95] European Parliament. *Fishing rules: Compulsory CCTV for certain vessels to counter infractions*. [Online; accessed 21-August-2022]. URL: <https://www.europarl.europa.eu/news/en/press-room/20210304IPR99227/fishing-rules-compulsory-cctv-for-certain-vessels-to-counter-infractions>.
- [96] André Pedersen et al. *Hybrid guiding: A multi-resolution refinement approach for semantic segmentation of gigapixel histopathological images*. 2022. DOI: 10.48550/ARXIV.2112.03455. URL: <https://arxiv.org/abs/2112.03455>.
- [97] Gohar A Petrossian. “Preventing illegal, unreported and unregulated (IUU) fishing: A situational approach”. In: *Biological Conservation* 189 (2015), pp. 39–48.

- [98] Robert Pettersen, Håvard Dagenborg Johansen, and Dag Johansen. “Secure Edge Computing with ARM TrustZone”. In: *International Conference on Internet of Things, Big Data and Security*. 2017.
- [99] Itsaso Rodriguez-Moreno et al. “Video activity recognition: State-of-the-art”. In: *Sensors* 19.14 (2019), p. 3160.
- [100] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd ed. Prentice Hall, 2010.
- [101] *satellite-imagery-datasets-containing-ships: A list of radar and optical satellite datasets for ship detection, classification, semantic segmentation and instance segmentation tasks*. URL: <https://github.com/jasonmanesis/Satellite-Imagery-Datasets-Containing-Ships>.
- [102] Andrew Senior et al. “Enabling video privacy through computer vision”. In: *IEEE Security & Privacy* 3.3 (2005), pp. 50–57.
- [103] MA Al-Shabi. “Credit card fraud detection using autoencoder model in unbalanced datasets”. In: *Journal of Advances in Mathematics and Computer Science* 33.5 (2019), pp. 1–16.
- [104] Syed Zakir Hussain Shah et al. “Fish-Pak: Fish species dataset from Pakistan for visual features based classification”. In: *Data in Brief* 27 (2019), p. 104565. ISSN: 2352-3409. DOI: <https://doi.org/10.1016/j.dib.2019.104565>. URL: <https://www.sciencedirect.com/science/article/pii/S2352340919309205>.
- [105] Amir Yaghoubi Shahir et al. “Mining vessel trajectories for illegal fishing detection”. In: *2019 IEEE International Conference on Big Data (Big Data)*. IEEE. 2019, pp. 1917–1927.
- [106] *Shinkei Systems’ AI-guided fish harvesting is more humane and less wasteful*. URL: <https://techcrunch.com/2022/07/28/shinkei->

Bibliography

- systems-ai-guided-fish-harvesting-is-more-humane-and-less-wasteful/.
- [107] Keng Siau and Weiyu Wang. “Artificial intelligence (AI) ethics: ethics of AI and ethical AI”. In: *Journal of Database Management (JDM)* 31.2 (2020), pp. 74–87.
- [108] Sintef. *H2020 smartfish – trawlmonitor*. URL: <https://www.sintef.no/en/projects/2018/smartfish/>.
- [109] Sintef. *Nytt Eu Prosjekt Skal Hindre overfisking og sikre naturmangfold*. Feb. 2022. URL: <https://www.sintef.no/siste-nytt/2022/everyfish-skal-bista-naturmangfold-og-hindre-overfisking/>.
- [110] Sintef. *Testing of Catchscanner Technology*. June 2022. URL: <http://smartfishh2020.eu/testing-of-catchscanner-technology/>.
- [111] Maria Sokolova et al. “A Deep Learning Approach to Assist Sustainability of Demersal Trawling Operations”. In: *Sustainability* 13.22 (2021). ISSN: 2071-1050. DOI: 10.3390/su132212362. URL: <https://www.mdpi.com/2071-1050/13/22/12362>.
- [112] Ian Sommerville. *Software Engineering*. 9th ed. Harlow, England: Addison-Wesley, 2010. ISBN: 978-0-13-703515-1.
- [113] Paolo Spagnolo et al. “A new annotated dataset for boat detection and re-identification”. In: *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2019, pp. 1–7.
- [114] Waqas Sultani, Chen Chen, and Mubarak Shah. *Real-World Anomaly Detection in Surveillance Videos*. 2018. DOI: 10.1109/CVPR.2018.00678. URL: [%5Curl%7Bhttps://www.crcv.ucf.edu/research/real-world-anomaly-detection-in-surveillance-videos/%7D](https://www.crcv.ucf.edu/research/real-world-anomaly-detection-in-surveillance-videos/).

- [115] Editorial Team. *Sjømateksporten passerte 120 milliarder kroner i 2021*. 2022. URL: <https://fisk.no/fiskeri/7553-sjomateksporten-passerte-120-milliarder-kroner-i-2021> (visited on 01/05/2022).
- [116] Enrico Tedeschi et al. “On Optimizing Transaction Fees in Bitcoin Using AI: Investigation on Miners Inclusion Pattern”. In: *ACM Trans. Internet Technol.* (Mar. 2022). ISSN: 1533-5399. DOI: 10.1145/3528669. URL: <https://doi.org/10.1145/3528669>.
- [117] Enrico Tedeschi et al. “Predicting Transaction Latency with Deep Learning in Proof-of-Work Blockchains”. In: *2019 IEEE International Conference on Big Data (Big Data)*. 2019, pp. 4223–4231. DOI: 10.1109/BigData47090.2019.9006228.
- [118] *Thales VesseLINK 200*. https://www.thalesgroup.com/sites/default/files/database/document/2021-02/2807_V1_VesseLINK200_012021.pdf. Accessed: 2021-09-08.
- [119] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition*. 4th ed. Academic Press, 2009. ISBN: 9781597492720.
- [120] Du Tran et al. “A closer look at spatiotemporal convolutions for action recognition”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018, pp. 6450–6459.
- [121] Rajesh Kumar Tripathi, Anand Singh Jalal, and Subhash Chand Agrawal. “Suspicious human activity recognition: a review”. In: *Artificial Intelligence Review* 50.2 (2018), pp. 283–339.
- [122] TV2. *Fiskefartøy skal ha dumpet 100.000 døde fisk i havet: – Sjokkerende*. [Online; accessed 10-September-2022]. URL: <https://www.tv2.no/nyheter/innenriks/fiskefartoy-skal-ha-dumpet-100000-dode-fisk-i-havet-sjokkerende/14547449/>.

Bibliography

- [123] Oguzhan Ulucan, Diclehan Karakaya, and Mehmet Turkan. “A Large-Scale Dataset for Fish Segmentation and Classification”. In: *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE. 2020, pp. 1–5.
- [124] UN. *Goal 14: Conserve and sustainably use the oceans, seas and marine resources*. [Online; accessed 21-August-2022]. URL: <https://www.un.org/sustainabledevelopment/oceans/>.
- [125] National Oceanic US Department of Commerce and Atmospheric Administration. Oct. 2016. URL: <https://oceanservice.noaa.gov/facts/oilseep.html>.
- [126] Nguyen Thanh Van, Tran Ngoc Thinh, et al. “An anomaly-based network intrusion detection system using deep learning”. In: *2017 international conference on system science and engineering (ICSSE)*. IEEE. 2017, pp. 210–214.
- [127] Deepak Vasisht et al. “FarmBeats: An IoT Platform for Data-Driven Agriculture”. In: *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. Boston, MA: USENIX Association, Mar. 2017, pp. 515–529. ISBN: 978-1-931971-37-9. URL: <https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/vasisht>.
- [128] Jiangping Wang et al. “An analysis of fishing vessel accidents”. In: *Accident Analysis & Prevention* 37.6 (2005), pp. 1019–1024.
- [129] Yuanyuan Wang et al. “A SAR Dataset of Ship Detection for Deep Learning under Complex Backgrounds”. In: *Remote Sensing* 11.7 (2019). ISSN: 2072-4292. DOI: 10.3390/rs11070765. URL: <https://www.mdpi.com/2072-4292/11/7/765>.
- [130] Global Fishing Watch. *New Fishing Data paves the way for improved analysis*. Aug. 2022. URL: <https://globalfishingwatch.org/data/new-fishing-data-improved-analysis/>.

- [131] Steve H Weingart. “Physical security devices for computer subsystems: A survey of attacks and defenses”. In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2000, pp. 302–317.
- [132] Heather Welch et al. “Hot spots of unseen fishing vessels”. In: *Science Advances* 8.44 (2022), eabq2109. DOI: 10.1126/sciadv.abq2109. eprint: <https://www.science.org/doi/pdf/10.1126/sciadv.abq2109>. URL: <https://www.science.org/doi/abs/10.1126/sciadv.abq2109>.
- [133] Thomas Winkler and Bernhard Rinner. “Security and privacy protection in visual sensor networks: A survey”. In: *ACM Computing Surveys (CSUR)* 47.1 (2014), pp. 1–42.
- [134] V. Wismann et al. “Radar signatures of marine mineral oil spills measured by an airborne multi-frequency radar”. In: *International Journal of Remote Sensing* 19.18 (1998), pp. 3607–3623. DOI: 10.1080/014311698213849. eprint: <https://doi.org/10.1080/014311698213849>. URL: <https://doi.org/10.1080/014311698213849>.
- [135] Fengwei Zhang and Hongwei Zhang. “SoK: A study of using hardware-assisted isolated execution environments for security”. In: *Proceedings of the Hardware and Architectural Support for Security and Privacy 2016*. 2016, pp. 1–8.
- [136] Tianwen Zhang et al. “Sar ship detection dataset (ssdd): Official release and comprehensive data analysis”. In: *Remote Sensing* 13.18 (2021), p. 3690.

Bibliography

Appendix A

List of Papers

A.1 Author Legend

T-A.S.N.: Tor-Arne S. Nordmo, J.A.A.: Joakim A. Alslie, M.E.E.: Martine E. Espeseth, B.H.: Birte Hansen, P.H.: Pål Halvorsen, S.A.H.: Steven A. Hicks, D.J.: Dag Johansen, H.D.J.: Håvard D. Johansen, B.A.J.: Bjørn A. Juliussen, O.K.: Ove Kvalsvik, S.O.K.: Svein O. Kvalsund, A.B.O.: Aril B. Ovesen, M.A.R.: Michael A. Riegler, V.T.: Vajira Thambawita

A.2 Paper I: Dutkat: A Multimedia System for Catching Illegal Catchers in a Privacy-Preserving Manner

Authors: T.S. Nordmo, A.B. Ovesen, H.D. Johansen, M.A. Riegler, P. Halvorsen and D. Johansen

Abstract: Fish crime is considered a global and serious problem for a healthy and sustainable development of one of mankind's important sources of food. Technological surveillance and control solutions are emerging as remedies to combat such criminal activities, but such

Appendix A. List of Papers

solutions might also come with impractical and negative side-effects and challenges. In this paper, we present the prototype of a surveillance system in lieu of current surveillance trends striking a delicate balance between privacy of legal actors while simultaneously capturing evidence-based footage, sensory data, and forensic proofs of illicit activities. Our novel approach is to assist human operators in the 24/7 surveillance loop of remote professional fishing activities with a privacy-preserving Artificial Intelligence (AI) surveillance system operating in the same proximity as the activities being surveyed. The system is primarily using video surveillance data but also other sensor data captured on the fishing vessel. Additionally, the system correlates with other sources such as reports from other fish catches in the approximate area and time, etc. Only upon true positive flagging of specific potentially illicit activities by the locally executing AI algorithms, can forensic evidence be accessed from this physical edge, the fishing vessel. Besides a more privacy-preserving solution, our edge-based AI system also benefits from much less data that has to be transferred over unreliable, low-bandwidth networks.

Author contributions (initials): **Conceptualisation:** T-A.S.N., D.J., M.A.R.; **Data collection:** T-A.S.N., M.A.R., **Methods, data analysis and interpretation:** T-A.S.N., M.A.R., **Drafting:** T-A.S.N., M.A.R., A.B.O., H.D.J, P.H., D.J., **Critical revision:** T-A.S.N., M.A.R., A.B.O., H.D.J, P.H., D.J.

Published: Proceedings of the 2021 Workshop on Intelligent Cross-Data Analysis and Retrieval (ICDAR)

Thesis objectives: Main Objective

Dutkat: A Multimedia System for Catching Illegal Catchers in a Privacy-Preserving Manner

Tor-Arne S. Nordmo
tor-arne.s.nordmo@uit.no
UiT The Arctic University of Norway
Tromsø, Norway

Aril B. Ovesen
UiT The Arctic University of Norway
Tromsø, Norway

Håvard D. Johansen
UiT The Arctic University of Norway
Tromsø, Norway

Michael A. Riegler*
SimulaMet
Oslo, Norway

Pål Halvorsen†
SimulaMet
Oslo, Norway

Dag Johansen
UiT The Arctic University of Norway
Tromsø, Norway

ABSTRACT

Fish crime is considered a global and serious problem for a healthy and sustainable development of one of mankind's important sources of food. Technological surveillance and control solutions are emerging as remedies to combat criminal activities, but such solutions might also come with impractical and negative side-effects and challenges. In this paper, we present the concept and design of a surveillance system in lieu of current surveillance trends striking a delicate balance between privacy of legal actors while simultaneously capturing evidence-based footage, sensory data, and forensic proofs of illicit activities. Our proposed novel approach is to assist human operators in the 24/7 surveillance loop of remote professional fishing activities with a privacy-preserving Artificial Intelligence (AI) surveillance system operating in the same proximity as the activities being surveyed. The system will primarily be using video surveillance data, but also other sensor data captured on the fishing vessel. Additionally, the system correlates with other sources such as reports from other fish catches in the approximate area and time, etc. Only upon true positive flagging of specific potentially illicit activities by the locally executing AI algorithms, can forensic evidence be accessed from this physical edge, the fishing vessel. Besides a more privacy-preserving solution, our edge-based AI system also benefits from much less data that has to be transferred over unreliable, low-bandwidth satellite-based networks.

CCS CONCEPTS

• **Computer systems organization** → **Distributed architectures**;
• **Computing methodologies** → **Activity recognition and understanding**; **Computer vision problems**; • **Security and privacy** → **Tamper-proof and tamper-resistant designs**.

*Also affiliated with UiT The Arctic University of Norway, Norway

†Also affiliated with Oslo Metropolitan University, Norway

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICDAR '21, August 21–24, 2021, Taipei, Taiwan

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8529-9/21/08...\$15.00

<https://doi.org/10.1145/3463944.3469102>

KEYWORDS

distributed system, AI, privacy, multimodal data analysis

ACM Reference Format:

Tor-Arne S. Nordmo, Aril B. Ovesen, Håvard D. Johansen, Michael A. Riegler, Pål Halvorsen, and Dag Johansen. 2021. Dutkat: A Multimedia System for Catching Illegal Catchers in a Privacy-Preserving Manner. In *Proceedings of the 2021 Workshop on Intelligent Cross-Data Analysis and Retrieval (ICDAR '21)*, August 21–24, 2021, Taipei, Taiwan. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3463944.3469102>

1 INTRODUCTION

Global fisheries contribute to food security for millions of people worldwide. It is estimated that the sea produces 17% of the current production of edible meat globally, and that this percentage will increase dramatically as the world's population grows [3]. Capture fisheries play a pivotal role for guaranteeing food security from the sea, but illegal fishing and over-exploitation are widespread problems that negatively impact the sustainability of wild fish stocks. Examples of such illegal activities include fishing vessels that operate without valid licenses, fishing on stocks that are depleted and close to extinction, fishing with illegal gears, and negligence in submitting correct catch data. These problems occur globally, but are perhaps particularly prominent around the poorer coastal countries that cannot launch effective counter-measures to combat such organized illegal activities.

In some geographical areas, like the northern Atlantic, evidence-based regulatory efforts have been shown to be efficient for maintaining sustainable harvesting of fish populations. A requirement, though, is that such regulations are combined with thorough fishing vessel monitoring, surveillance, and control. For instance by requiring vessels to have Automatic Identification Systems (AIS) that track position installed, have them maintain and submit logs to the authorities when fish gear is launched, when catch is landed on deck, and what species and volumes were caught, and where the catch will be delivered for further processing. Fishing vessels are also subject to an inspection at any time and without notification, both while at sea or when docking, by either the coast guard or other public inspectors.

Though such operational control regimes make cheating difficult, they are far from flawless. Failures to report a catch is one example of how criminals can cheat, under-reporting fish quantities landed

another. The monetary gain for such illegal activities can be relatively profitable; UN estimates that illegal fishing activities amount to figures in the order of billions USD a year, and involves corruption and other financial crimes, such as large-scale tax evasion and money-laundering [14].

Some argue that digital surveillance technologies can mitigate fishery crime problems. Automatic 24/7 video-surveillance systems and sensors on board the fishing vessels at sea are particularly touted as technological solutions. The Danish Fisheries Authority is already equipping tenfold fishing vessels from the country’s fleet with such equipment, a mutually agreed decision with volunteering fishermen. Norway is also planning mandatory deployment of similar surveillance cameras as one of several remedies to manage, control, and combat illegal fishing activities [13].

At a first glance, such video surveillance might sound as a plausible approach to combat crimes, but comes with a significant disadvantage with regard to privacy. Suddenly, somebody can be watching every fisherman while working on deck, which can be considered personally invasive. The proportionality law principle is also at stake striking a balance between capturing potential illicit behaviour versus intruding on personal spheres of tenfold thousand of fishermen at work. In addition, the analysis of video data in combination with other sensory data and external data is a challenge by itself. Combining different modalities for analysis comes with challenges regarding how to combine them, how to determine the importance of the different input sources, etc. Scale and performance are obvious technical problems considering that fishing vessels in the order of 1,000s will have to transfer voluminous multimedia data over satellite links or radio-based technologies to mainland operational control centers. And, imagine the tedious and labor-intensive task if this data is to be inspected in real-time by human operators.

In this paper we propose *Dutkat*: a technical solution that balances the conflicting properties of surveillance, performance, and privacy. The system provides multimodal analysis of collected data sources, where at the same time the privacy of individuals is preserved, yet secure and reliable ground truth documentation of illegal activities are captured and stored for forensics purposes. The main idea is to replace humans in the parsing of multi-sensor data and thus guarantee confidentiality and privacy, while at the same time being able to analyze and detect efficiently that suspicious activities might occur.

2 SYSTEM ARCHITECTURE

We have devised an architecture for monitoring and storing surveillance data, coupled with algorithmic control of specific physical activities in remote spatial locations. Figure 1 illustrates this distributed architecture where (1) edge nodes located in the physical proximity of interest contain physical monitoring sensors, storage, and computation devices. These edge nodes are connected to a (2) more trusted, stationary hub structure federating edge data for persistent storage. This back-end server hub structure is located in a resource rich and secure proximity and typically connects to multiple edge nodes. Different (3) analysis components can be connected to this central hub, but it is also possible to connect such analysis components directly into data streams or on edge nodes.

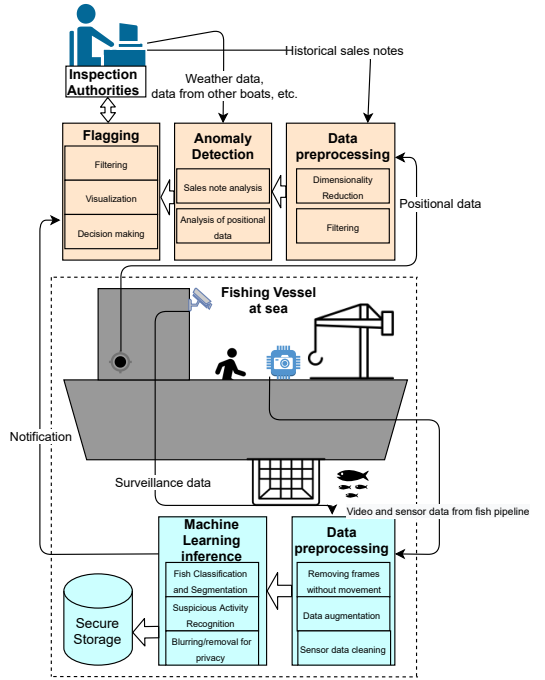


Figure 1: System architecture. The colors indicate where the computation takes place, with blue on the vessel and orange on a mainland control site.

Finally, different (4) front-end user-specific modules for operational surveillance purposes, forensics, control operations and the like are provided.

Our main approach is to replace a human being in the analysis loop of a remote video stream, with an algorithm performing the analysis in the same proximity where video and related sensor data are captured. We enhance and complement video data with other sensor data, particularly from edge located Internet of Things (IoT) devices. Captured surveillance data is not sent remotely for human inspection, which could potentially violate privacy regulations, but is continuously analyzed by local AI algorithms. Consider this algorithm as resembling a human inspector locally on each of the vessels involved. On average, the current control regime in the coastal area of Norway has resources to inspect and control in the order of 5% of 1,000 fold fishing vessels, while our approach theoretically scales this to close to 100%. In practice, our system is intended to be deployed first on the bigger trawlers, which are large vessels already fully equipped with digital technology.

We are building a first prototype implementation of this architecture. This is of *Dutkat*, a distributed, privacy-preserving surveillance system where a mainland control center is connected to a number of mobile edge devices. *Dutkat* is intended to be deployed on off-shore trawlers with license to harvest fish in the Arctic Sea. We organize

the different software modules to belong in four different planes stacked upon each other. In the bottom is a (1) data monitoring plane, next a (2) data collection and storage plane, an (3) analysis plane, and a (4) user control plane. Notice that particularly plane 2 and 3 have software modules located both on mainland and on fishing vessels.

Specifically, we target detection of suspicious activities related to quota exploitation and discard of catch, but also to estimate catch volume and species distribution to compare against sales notes. This will also provide automatic documentation for fishermen. The design allows us to analyze multiple data sources, and reduces unnecessary transmission of data over low-bandwidth satellite networks.

Video and sensor equipment will be placed on deck to detect anomalous movement and actions, and cameras will also be placed in the fish processing pipeline for catch analysis. AIS positional data will also be recorded and sent to the mainland. The analysis will be performed on the fishing vessel, and the data and results will be securely stored on the vessel. An aggregated/summarized result will be sent to the mainland for potential flagging of the vessel.

The subsystem on land will analyze AIS data and sales notes, and the results of this will be combined with the results sent from the fishing vessels. The inspection authorities will then have a graphical user interface where they can see which vessels are flagged for inspection, before the vessel even lands.

3 DISCUSSION AND RELATED WORK

The fishing domain and use of distributed AI solutions is a domain we are very familiar with. Over three decades ago, we built a series of StormCast [7–10], probably the first sensor network prototype ever with distributed expert-systems built for trawlers and coast guard ships connected through satellite communication links. StormCast was predicting suddenly erupting hazardous weather situations at sea over large areas. A mainland public version of Stormcast has been continuously operational since 1993¹.

Dutkat will consist of multiple subsystems which are dependent on different models to perform multimodal analysis. There have been multiple works on the sub-problems of privacy-preserving analytics, human activity recognition, fish analysis, and tabular analysis.

Fitwi et al. [5] describe a system for masking private information in video frames from surveillance cameras by doing detection and filtering on the edge. In their paper, they discuss the advantages and disadvantages of a cloud-based surveillance system vs. an edge-based approach. They argue that some pre-processing (i.e., motion detection for filtering of scenes) can happen on the edge, but that deep learning-based object detection methods are too computationally intensive to perform on the edge. They introduce a chaos-based encryption scheme that is used on the data before it is sent to a server for processing. There, they perform window and people detection and face recognition. A notable difference though is that our edge is relatively resourceful where entire computers can be deployed for edge analysis. Currently, our select edge platform we are exploring consists of a cluster of Nvidia Jetson Xavier NX computers.

D’souza et al. [4] describe a similar system that uses object detection for surveillance camera video streams, and whitelists classes of objects that should *not* be censored. Our system consists of many similar components that allow us to preserve privacy. However, a major difference between Dutkat and these two works is our relative greedy use of bandwidth. The papers above describe ways of reducing bandwidth requirements and encrypting data before transfer, whereas we perform the data-intensive video analysis on the edge, and only sends minimal information to the mainland site.

Rodriguez-Moreno et al. [15] describe the challenges and state-of-the-art methods for performing activity recognition from video. One needs to recognize the relevant areas that contain humans in the frames, and a variety of techniques can be applied. Hand-crafted feature extraction has been the traditional approach, while in more recent times, deep learning-based solutions have been preferred. In the more specific area of recognizing suspicious activity, the task becomes more difficult. In the Tripathy et al. [16] survey, they show usage of object detection, tracking, and activity recognition in different use-cases, however, in all these cases it is relatively simple to find examples of suspicious activity on video, as the settings are more conventional. In our case, this is not so, and we need to potentially apply transfer learning on simulated suspicious activity.

Marais [12] describes the difficulties of analyzing tabular data with ML techniques; numerical data is usually ordered and continuous, but can be discrete, while categorical data is usually unordered and discrete. This can make feature interactions difficult to learn. This is highly relevant to the analysis of the sales notes, as they are a composite of numerical and categorical features, therefore representing the data in a fitting manner is crucial.

Automatic fish analysis using ML can be difficult depending on what you aim to accomplish. All image-based ML methods are dependent on training data that reflects a large variety of conditions, e.g. lighting and water clarity. Segmentation can be particularly difficult with overlapping fish. Alsmadi et al. [1] discuss how fish classification has been done historically with traditional ML techniques, and different datasets and the data requirements of deep learning methods. We are definitely confident that deep learning methods can achieve state-of-the-art results, given a good dataset. Weight estimation of fish from images can be difficult to achieve, due to not having a 3D view and due to the density of the fish being dependent on species, feed, age, etc. Konovalov et al. [11] apply a LinkNet-34 segmentation network to segment fish in images. They then compare mathematical models for estimating weight of fish based on length, height, and/or segmented area, vs. using a convolutional network to directly estimate weight via regression. Garcia et al. [6] do measurement estimation of fish, but based on a stereoscopic view. They show that using stereo imaging makes segmentation easier in the case of overlapping fish. They utilize Mask RCNN to segment the fish, and then estimate the length of the fish via morphological operations. This could then be used in a mathematical model for weight estimation.

Baltruvisaitis et al. [2] discuss five major challenges of multimodal data analysis in ML. Of these, representation, i.e., exploiting complementarity and redundancy between different data sources, and fusion, i.e., joining information from different modalities to form a prediction are highly relevant to our use-case. The ML pipeline utilizes complementarity in e.g. the AIS and sales notes

¹<https://weather.cs.uit.no/>

data to predict some anomalous activities, and requires prediction from different models to form a final judgement on whether a fishing vessel should be inspected or not.

4 RESEARCH DIRECTIONS

The described system is addressing several aspects of a ML system pipeline. Due to the specific requirements of privacy, but also the challenges by the potential sheer volume of data and the use case itself, different possible research directions have been identified.

R1: Transparent AI system The system built for the use case needs to be transparent in all aspects. This means that the data and its distribution, how it was collected and annotated, how the analysis is performed, and how the results are presented need to be open and comprehensible. To achieve this explainability, the AI system needs to be created that is beyond most of the current black-box systems. Such a system will need to support ways of understanding data distribution, tackle domain shifting (e.g., seasonal differences in the data), and provide explainable and interpretable results.

R2: Multimodal data analysis The analysis requirements for such a system come with different challenges. One of the most important is to be able to analyze data from different sources combined. This in itself is a challenging task and not researched very well, especially in modern, deep learning based AI systems. For such a system, new methods within the field of deep learning for efficient combination of data that go beyond simple late or early fusion need to be researched and developed. This cross-data analysis also includes ways to explain the importance of the different data sources to the outcome of the analysis.

R3: Privacy-preserving AI Analysis involving data gathered from human beings potentially imposes new risks or negative consequences for individuals. Privacy is certainly at risk when, for instance, video data is collected for manual inspections or algorithmic analysis. As such, compliance issues must be a first-order design principle in next-generation novel surveillance systems like ours. This important concern for individuals right for privacy is partially covered by emerging regulations like, for instance, the EU General Data Protection Regulation (GDPR), but this is just one step in the right human-centric direction. AI is processing privacy-sensitive data, where proper directives and regulations need to come into existence. This includes that legal aspects must be integral to our computer science research.

R4: System architectures Figure 1 shows the complexity of building a complete multimodal, ML-based data analysis pipeline distributed over a large variety of computing nodes ranging from large machines ashore to smaller mobile devices on the boats at sea with limited connectivity and power. The system must integrate all the above components in an efficient way. Some data needs to be (pre)processed and filtered on the edge of the system, and more complex anomaly analyzes will be performed ashore. Developing such a system opens several research questions like what data to collect and at which frequency; placement of the processing; amount of edge processing; off-shore communication; etc.

R5: Continuous learning The described use-case comes with the specific challenge of time and location-based changes of key features. Based on the location of the boat and the season of the year the amount of fish, type of fish, and size will change. Models and datasets need to reflect this change. Ways of including this into the analysis need to be researched. Possible starting points are transfer learning, data distribution shift, and continuous learning algorithms.

R6: Data Currently most surveillance datasets are taken in cities. The chance that these datasets will provide a basis for analysis on fishing vessels is, put modestly, rather low. Specific datasets will be needed that is collected under the specific conditions on the boats. In addition, seasonal and day and night shifts should be taken into account. Regarding the fish analysis, current fish datasets are often specifically collected under perfect conditions and in areas with different types of fish. To be able to build a practically useful system one will need a fish dataset that is close to the real world conditions on the boat (fish dataset in the wild). We also conjecture the need for small data analytic datasets, one where small datasets can be useful for specific situations (relative to the typical big datasets found in, for instance, supervised learning scenarios).

The discussed research directions only present a small number of challenges and possible directions that arise, but includes what we identified as the most important ones to build a solid base for future improvements.

5 CONCLUSION

Fraud in the fishing industry is a major international problem, and it is difficult to determine when it takes place without infringing on the privacy of fishermen. Thus, we have proposed Dutkat: a distributed privacy-preserving surveillance and documentation system for detecting when crime potentially occurs. Realizing such a system requires multimodal analysis on a variety of data that is both captured on the fishing vessel, and that is federated on the mainland from other sources.

We conjecture that Dutkat is based on a more general architecture that is applicable in different use-case scenarios. The properties for such use-cases include some sort of continuous sensory input at a (mobile) edge node for real-time or post-analysis. This edge node can be a plane, an army vehicle, or a modern car, but also even a human being like a policeman equipped with multimedia life-logging devices. For example, in self-driving cars there are multiple safety features that require constant surveillance of both the physical environment, but also the driver and passengers. In order to ensure that the privacy of the user and others in the proximity of the car is maintained, the general architecture described will be further explored in such use-case scenarios.

ACKNOWLEDGMENTS

This work is partially funded by the Research Council of Norway project numbers 275516 and 263248, and Lab Nord-Norge ("Samfunnsløftet"). We particularly acknowledge contributions from Kim H. Andreassen, Arne E. Karlsen, Nandor Knust, Jon F. Mikalsen, Magnar Pedersen, Jon P. Rui, and Kjetil Robertsen.

REFERENCES

- [1] Mutasem K. Alsmadi and Ibrahim Almarashdeh. 2020. A survey on fish classification techniques. *Journal of King Saud University - Computer and Information Sciences* (2020). <https://doi.org/10.1016/j.jksuci.2020.07.005>
- [2] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2018. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence* 41, 2 (2018), 423–443.
- [3] Christopher Costello, Ling Cao, Stefan Gelcich, Miguel À Cisneros-Mata, Christopher M Free, Halley E Froehlich, Christopher D Golden, Gakushi Ishimura, Jason Maier, Ilan Macadam-Somer, et al. 2020. The future of food from the sea. *Nature* 588, 7836 (2020), 95–100.
- [4] Sandeep Dsouza, Victor Bahl, Lixiang Ao, and Landon P Cox. 2020. Amadeus: Scalable, Privacy-Preserving Live Video Analytics. *arXiv preprint arXiv:2011.05163* (2020).
- [5] Alem Fitwi, Yu Chen, Sencun Zhu, Erik Blasch, and Genshe Chen. 2021. Privacy-Preserving Surveillance as an Edge Service Based on Lightweight Video Protection Schemes Using Face De-Identification and Window Masking. *Electronics* 10, 3 (2021), 236.
- [6] Rafael García, Ricard Prados, Josep Quintana, Alexander Tempelaar, Nuno Gracias, Shale Rosen, Håvard Vågstøl, and Kristoffer Lovall. 2020. Automatic segmentation of fish using deep learning with application to fish size measurement. *ICES Journal of Marine Science* 77, 4 (2020), 1354–1366.
- [7] Gunnar Hartvigsen and Dag Johansen. 1990. Co-operation in a distributed artificial intelligence environment—the stormcast application. *Engineering Applications of Artificial Intelligence* 3, 3 (1990), 229–237.
- [8] Dag Johansen. 1993. A distributed approach to the design of applications. In *Proceedings of ICCI'93: 5th International Conference on Computing and Information*. IEEE, 195–201.
- [9] Dag Johansen and Gunnar Hartvigsen. 1994. Convenient abstractions in Storm-Cast applications. In *Proceedings of the 6th workshop on ACM SIGOPS European workshop: Matching operating systems to application needs*. 11–16.
- [10] Dag Johansen, Kjetil Jacobsen, Nils P Sudmann, J Lauvset Kare, and Werner Vogels. 1997. Using Software Design Patterns to Build Distributed Environmental Monitoring Applications. (1997).
- [11] Dmitry A Kononov, Alzayat Saleh, Dina B Efremova, Jose A Domingos, and Dean R Jerry. 2019. Automatic weight estimation of harvested fish from images. In *2019 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 1–7.
- [12] Jan André Marais. 2019. *Deep learning for tabular data: an exploratory study*. Ph.D. Dissertation. Stellenbosch: Stellenbosch University.
- [13] Ministry of Trade, Industry and Fisheries. 2019. Framtidens Fiskerikontroll. *NOU 19:21* (2019).
- [14] United Nations Office on Drugs and Crime. 2016. Fisheries Crime. (2016).
- [15] Itsaso Rodriguez-Moreno, José María Martínez-Otzeta, Basilio Sierra, Igor Rodriguez, and Ekaitz Jauregi. 2019. Video activity recognition: State-of-the-art. *Sensors* 19, 14 (2019), 3160.
- [16] Rajesh Kumar Tripathi, Anand Singh Jalal, and Subhash Chand Agrawal. 2018. Suspicious human activity recognition: a review. *Artificial Intelligence Review* 50, 2 (2018), 283–339.

A.3 Paper II: File System Support for Privacy-Preserving Analysis and Forensics in Low-Bandwidth Edge Environments

Authors: A.B. Ovesen, T.S. Nordmo, H.D. Johansen, M.A. Riegler, P. Halvorsen and D. Johansen

Abstract: In this paper, we present initial results from our distributed edge systems research in the domain of sustainable harvesting of common good resources in the Arctic Ocean. Specifically, we are developing a digital platform for real-time privacy-preserving sustainability management in the domain of commercial fishery surveillance operations. This is in response to potentially privacy-infringing mandates from some governments to combat overfishing and other sustainability challenges. Our approach is to deploy sensory devices and distributed artificial intelligence algorithms on mobile, offshore fishing vessels and at mainland central control centers. To facilitate this, we need a novel data plane supporting efficient, available, secure, tamper-proof, and compliant data management in this weakly connected offshore environment. We have built our first prototype of Dorvu, a novel distributed file system in this context. Our devised architecture, the design trade-offs among conflicting properties, and our initial experiences are further detailed in this paper.

Author contributions (initials): **Conceptualisation:** A.B.O, D.J.; **Data collection:** A.B.O., T-A.S.N., **Methods, data analysis and interpretation:** A.B.O., T-A.S.N., **Drafting:** A.B.O., T-A.S.N., M.A.R., H.D.J, P.H., D.J., **Critical revision:** T-A.S.N., M.A.R., A.B.O., H.D.J, P.H., D.J.

Published: MDPI Information, 2021

Thesis objectives: Sub-objective 1

File system support for privacy-preserving analysis and forensics in low-bandwidth edge environments

Aril B. Ovesen¹, Tor-Arne S. Nordmo¹, Håvard D. Johansen¹, Michael A. Riegler^{2,1}, Pål Halvorsen^{2,3} and Dag Johansen¹

¹ UiT The Arctic University of Norway, Tromsø, Norway

² SimulaMet, Oslo, Norway

³ Oslo Metropolitan University, Oslo, Norway

Abstract: In this paper, we present initial results from our distributed edge systems research in the domain of sustainable harvesting of common good resources in the Arctic Ocean. Specifically, we are developing a digital platform for real-time privacy-preserving sustainability management in the domain of commercial fishery surveillance operations. This is in response to potentially privacy-infringing mandates from some governments to combat overfishing and other sustainability challenges. Our approach is to deploy sensory devices and distributed artificial intelligence algorithms on mobile, offshore fishing vessels and at mainland central control centers. To facilitate this, we need a novel data plane supporting efficient, available, secure, tamper-proof, and compliant data management in this weakly connected offshore environment. We have built our first prototype of Dorvu, a novel distributed file system in this context. Our devised architecture, the design trade-offs among conflicting properties, and our initial experiences are further detailed in this paper.

Keywords: edge computing; privacy preservation; artificial intelligence; file systems; machine learning; digital forensics

Citation: Ovesen, A.B.; Nordmo, T.A.S.; Johansen, H.D.; Riegler, M.A.; Halvorsen, P.; Johansen, D. Title. *Information* **2021**, *1*, 0. <https://doi.org/>

Received:

Accepted:

Published:

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Submitted to *Information* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Numerous Internet of Things (IoT) devices are being deployed in geo-distributed locations far outside traditional computing facilities [1,2]. Examples include video surveillance cameras, home security devices, activity trackers, logistic tracking devices, and smart factory equipment. High velocity, high volume, and heterogeneous data are continuously produced by these devices at an unparalleled scale. A key challenge is to analyze and obtain trusted, timely insight from these data streams.

Distributed system architects must carefully consider structuring alternatives to centralized on-premise or public cloud services for data analysis. Moving computations closer to data sources is likely a better option than federating and centralizing all this data [3,4]. Hence, edge computing can supplement a two-tier centralized architecture by providing an additional computing infrastructure closer to the data sources, residing between IoT devices and centralized back-end services. Edge computed data can still be federated for further analysis and storage at the central locations, but now pre-processed, filtered, and transmitted in reduced volumes.

Edge devices can produce data that is too voluminous to transmit over edge networks, and too heterogeneous to adhere to a unified set of data management rules. For the purpose of tackling this challenge we introduce the Dorvu file system, a storage system that can be extended with policies and fine-grained access control to solve problems of bandwidth, privacy, and compliance.

The Dorvu file system is part of the larger Dutkat [5] framework, comprising a distributed Artificial Intelligence (AI) hybrid cloud and edge system. This system is motivated by the need for sustainable harvesting of resources from the sea, including

38 commercial fisheries in the Arctic Ocean. The world’s global population depends on
39 food obtained from the sea, and this dependence is growing [6]. This can become
40 problematic because the global sea ecosystems have been and are currently under
41 serious attacks by human activities and might not be able to meet this growing demand.
42 Challenges include overfishing and depleted fish stocks, destroyed or polluted sea
43 ecosystems, increased water temperatures, and lack of management and control regimes
44 for sustainable fisheries. According to the United Nations Office on Drugs and Crime,
45 fishery crimes are frequently transnational and organized in nature, and include illegal
46 fishing, document fraud, drug trafficking, and money laundering [7]. As a result, several
47 governments have proposed surveillance systems to track the activity of workers on
48 fishing vessels [8–10], which has been met by some with criticism and claims of privacy
49 intrusion and mass surveillance [11]. The Dutkat framework aims to provide some of
50 the proposed sustainability benefits [12], while preserving the privacy of fishing vessel
51 crew.

52 In this work, we are addressing and presenting some specific parts of the envisioned
53 Dutkat system. Specifically, the main contributions are (1) a system of how to perform
54 privacy-preserving analysis of continuously produced data is conceptualized, incorpo-
55 rating challenges such as potentially poorly connected fishing vessels moving about in
56 the Arctic Ocean, (2) we show how multi-sensory data is handled, and we showcase how
57 to use the Dorvu file system to alleviate decision-making around issues like what type
58 of data to store, in what format, and how to mandate access control and encryption on
59 it. Overall, we present an alternative approach to existing surveillance programs [9,10]
60 by replacing human inspection of video footage with a combination of automated pro-
61 cessing of sensor data, and access control policies enforced on the storage layer at the
62 time of data creation. We conjecture that incorporating Dorvu in this process better
63 preserves the privacy of people working in proximity to edge sensors, through flexible,
64 fine-grained data access and storage policies that can retain sustainability-relevant data
65 while discarding privacy intrusive footage, resulting in a less invasive system.

66 The rest of this paper will present the motivation behind our edge analysis system,
67 the architecture of our cloud and edge hybrid storage system, and the implementation
68 of a prototype of the Dorvu file system. Particular focus is on edge nodes and our design
69 choices targeting a distributed file system that tolerates failures, survives adversarial
70 attacks, and meets compliance requirements.

71 2. A Mobile AI Edge System at Sea

72 The Dorvu storage system is intended to serve as a storage layer in the larger
73 Dutkat [5] project revolved around monitoring professional fishing activities in isolated,
74 offshore areas, while retaining the privacy of those working in close proximity to areas
75 subject to surveillance. Its design involves installing robust and safe monitoring devices
76 and related software on board commercial fishing vessels with licence to fish in Norwe-
77 gian parts of the Arctic Ocean. Each such vessel has been granted a specific quota from
78 the government detailing the amount of fish allowed to catch, the fish species, fish sizes,
79 and similar.

80 Video surveillance of fishing vessels to combat sustainability issues and enforce
81 fishing quotas has been proposed by some governments [8,9] and explored by others [10].
82 The Dutkat system is designed to provide some of the sustainability benefits claimed
83 by the proposed national surveillance systems, without infringing on the privacy of
84 workers on fishing vessels.

85 2.1. Geo-Distribution

86 The overall architecture of Dutkat reflects the widely distributed and mobile nature
87 of this application domain. We will first detail the horizontal dimension of the archi-
88 tecture, which reflects physical distribution among three separate components; (1) one
89 or several back-end control centres, (2) a collection of traditional computers on board

90 each vessel, and (3) IoT devices primarily located outside on ship decks. We consider
91 each of the participating large fishing vessels as individual edge nodes in a hybrid cloud
92 computing system, where each edge node has sufficient power facilities and indoor
93 space for deployment of a collection of computers. These computers will for security
94 and fault-tolerance be configured to only run local Dutkat communication, storage, and
95 analysis software parsing locally produced data from the IoT devices outside. Consider
96 such a configuration as implementing certain features of a digital version of a local
97 fishing inspector, which will act as an algorithmic intermediary between the IoT devices
98 on deck and the back-end centralized control centres on mainland.

99 The Dutkat software must be safe-guarded and stable for 24/7 operability, while
100 at the same time ensuring compliance and non-invasion of the daily operations of the
101 vessel crew. Particularly, AI analysis performed at the edge must be able to detect local
102 anomalies and activities, and consequently persist the relevant ground truth data locally
103 while sending relevant insights to the mainland operational centres for further analysis.
104 Data persistence and analysis being performed locally aid in preserving the privacy of
105 the vessel crew.

106 A hybrid architecture is needed, with edge nodes connected to centralized struc-
107 tures. The problem at hand is complex and requires input by more than just insights from
108 a single fishing vessel. For instance, one algorithmic trigger that requires input from
109 several edge nodes is the comparison of reported catch from different fishing vessels
110 in the same offshore proximity. Anomalies can be detected through such comparisons,
111 one example being vessels reporting disproportionate amounts of fish caught relative to
112 other vessels in the same area and their allocated quota.

113 A main problem in this domain is connectivity, since digital communication between
114 these mobile vessels and mainland operational centres is primarily facilitated by satellites.
115 We conjecture that by moving computations closer to the data sources, the amount of
116 data needed to transmit over satellite links can be reduced to a practical level. This
117 is enabled by edge computing where local data filtering, analysis, and storage can be
118 carried out in real-time.

119 Evaluating and filtering data streams close to their sources are well-known concepts
120 for scaling distributed systems producing large quantities of data [13]. By this upstream
121 evaluation structuring approach, algorithms can parse and analyze entire streams of
122 data on the vessels, without adhering to the limitations of low-bandwidth satellite links.

123 2.2. Vertical distribution

124 The vertical dimension of the Dutkat architecture determines separation of concerns
125 at the individual horizontal components. The relationship between computers running
126 Dutkat software at back-end control centers and on the edge is illustrated in Figure 1.
127 Overall, (1) a persistent storage layer is in the bottom, (2) followed by a data transfer
128 layer, (3) a data consumption layer, and finally (4) a user interface layer. For edge
129 deployments, the interface layer can be omitted, and IoT devices can interface with the
130 storage system as data producers, as illustrated in Figure 1. The data storage layer will
131 be further detailed in Section 3.

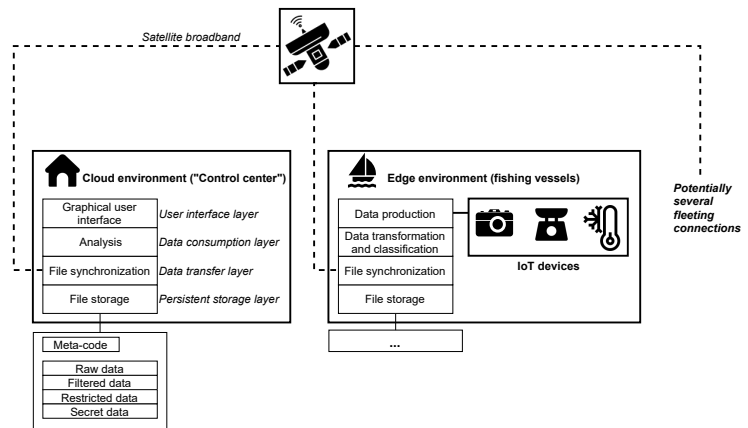


Figure 1. The horizontal and vertical distribution of data, software components and devices in the Dutkat architecture.

132 2.3. Multimedia data pipeline

133 Figure 1 shows the generic distribution of data and its relation to the software
 134 components in Dutkat. Specifically, the system is deployed to store and transmit mul-
 135 timedia data, like video and images. Edge nodes generate heterogeneous multimedia
 136 data, which can vary in content, type, and sensitivity level (i.e., the amount of private
 137 information contained in the data). For example, a collection of recordings from an edge
 138 video surveillance system may vary in sensitivity level if only a subset of the collection
 139 contains footage of people. Similarly, users responsible for generating data may have
 140 consented to different data sharing policies, while still contributing to the same dataset.
 141 At the same time, data consumers may have varying rights to view this data. In the
 142 scenarios presented in Section 1, it may vary what data local law enforcement, fishing
 143 crew, and other interested parties may consume. This results in a system of several
 144 edge nodes collaborating to produce a multimedia dataset, consumed by several nodes,
 145 both in cloud and edge deployments, that differ in their rights to view various parts
 146 of the whole dataset. Privileges can be enforced throughout the dataset by applying
 147 fine-grained access control mechanisms on individual files in the set.

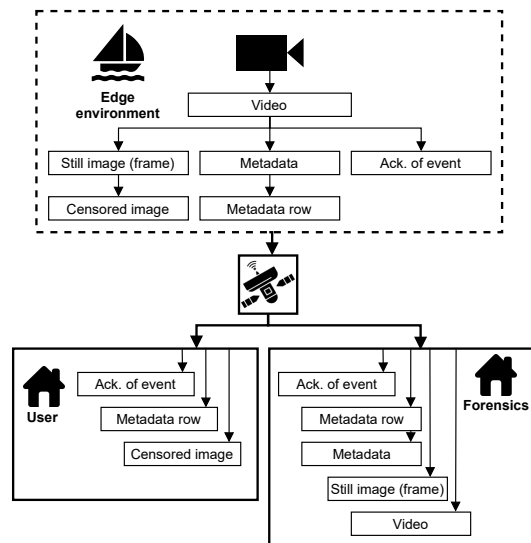


Figure 2. Example of a multimedia data pipeline transmitting information extracted from a video recording from an edge device, to two parties with differing privileges to view the data.

148 Our system facilitates real-time decision-making from land observers, based on
 149 data generated from edge devices. This is a process that involves transmitting as much
 150 meaningful data as possible from edge nodes to land nodes. This process is restricted by
 151 the low bandwidth that is expected in edge environments, such as when using satellite
 152 communication. Dutkat and the Dorvu file system are designed to support a model of
 153 decision-making in which the semantic meaning extracted from multimedia recordings
 154 is of highest priority to expend bandwidth on transmitting, and the ground truth data,
 155 i.e., the full-sized media files, is of less importance and may be retrieved eventually,
 156 when the network tolerates it. An example of the extraction of meaningful data from
 157 a larger multimedia file is shown in Figure 2. Here, the smallest significant data item
 158 extracted from a video file is the acknowledgement of the existence of a file, representing
 159 an event. Extracted metadata and still images from the video provide greater detail at
 160 the expense of more bandwidth. Finally, the original video file gives an overview of the
 161 entire event, but is not available to all parties.

162 3. System properties and architecture

163 The overall architecture of our distributed file system is based on a central hub
 164 structure with a cluster of file servers connected with multiple mobile edge nodes, each
 165 with local file storage capacity. This resembles a client-server star network with clients on
 166 the edge perimeter and servers in the centre. Another resemblance is with the distributed
 167 file system Coda [14] with back-end server clusters supporting numerous light-weight
 168 personal computers. Of particular interest is the support Coda has for disconnected
 169 operations, a situation still plausible in an offshore environment.

170 The centralized server hub is resource-rich and can use existing public cloud file
 171 systems supporting efficient, reliable, and centralized storage of multimedia data, sensor
 172 data, and machine learning results from the edge nodes. The edge nodes are less resource
 173 rich and are physically located on active fishing vessels. When along the coast and near
 174 the shore, communication options include cellular networks and radio networks, but
 175 when more distant and offshore, satellite communication is the main option. Novelty in
 176 our work is primarily related to the edge nodes, and how and what they communicate
 177 back to the mainland-located servers.

178 3.1. System requirements

179 There are special application-specific demands that need software tailoring and cus-
180 tomization. The file system is intended for use in an area with very limited computational
181 and communication resources since its mobile edge clients consist of fishing vessels
182 moving about in large ocean areas. Communication in such a widely geo-distributed
183 mobile edge computing environment is through partially disconnected, low-bandwidth
184 polar orbit satellite links. Since the fishing vessels in our system operate in the far north
185 of the globe, the geo-stationary satellite solutions we previously utilized are not adequate
186 as they do not cover the northernmost hemisphere [15]. Add to the complexity that
187 this distributed file system must be scalable, secure, fault-tolerant, and compliant with
188 particularly the EU General Data Protection Regulation (GDPR) privacy regulations [16].

189 Special properties of the select problem domain motivate the design of our system
190 as follows. First, we need to be able to continuously capture and store video and sensor
191 data for fish management, control, and forensics purposes. An example is a continuously
192 captured surveillance video of equipment stored on the deck of a fishing vessel. This is a
193 resource demanding file storage challenge, and the file system should be used to store
194 video sequences when specific activities or events are detected, and apply access policies
195 based on the contents of the events. Video data is in any case high-volume data, which is
196 challenging to reliably transmit to mainland operational surveillance and control centers
197 from the offshore mobile fishing vessels. The bandwidth delivered by satellite-based
198 solutions is not adequate to support live video streaming, even more so for networks
199 based on the low-frequency L-band.

200 Next, we need to provide redundancy for fail-safe storage of vital data, through
201 data copies at multiple local disks. The replicas are physically distributed on board each
202 vessel to reduce the probability of data corruption, loss, or unavailability. Redundancy
203 and update techniques similar to the Google file system [17] are adopted, with a master
204 control node typically administering three replicas updated in a pipeline fashion. The
205 master node will raise a flag, i.e. transmit a signal to the mainland operational centre, if
206 the replication threshold is below a certain level. Additionally, since the master node
207 might be a single point of attack or failure, we provide primary-backup replication with
208 a hot stand-by node ready to take over. Hot stand-by implies that a sequence of the latest
209 data written to disk is kept in main memory. Consider this as a large ring buffer whose
210 content will be streamed to disk upon fail-detection and fail-over.

211 The Google file system and similar master-based distributed systems [18] do not
212 provide such a replication due to complexity with consistency, impact on cost and
213 performance, the deployment in a trusted enterprise environment, and the observation
214 that master node failures seldom happen. In our case, executing on the edge in a less
215 trusted environment, we cannot tolerate a single node failure weakness in the critical
216 data storage path.

217 The hot stand-by keeps a large enough sliding window of data to be backed up in
218 case of failure so that a primary node failure will be transparently handled. Notice that
219 inconsistency problems among the data storage replicas or master replicas are to a large
220 extent avoided since data to be permanently stored on the edge nodes is tagged with its
221 timestamp and is immutable. This way no read-write conflict will appear.

222 3.2. Data classification

223 Fail-safe storage implies that there are very strict access policies affiliated with some
224 of this data. We must therefore distinguish between and classify data according to differ-
225 ent compliance, safety, and liveness properties. That is, we provide different guarantees
226 with regard to data reliability, availability, privacy, and confidentiality based on how
227 the data is classified. Data must be classified as either RAW, FILTERED, RESTRICTED, or
228 SECRET. As will become apparent, this classification differs from traditional security
229 classification schemes, as it supports implementation of non-functional aspects other

230 than security. This classification and its various properties are explained in the following
231 paragraphs.

232 Data tagged RAW contains the continuous stream of data produced by video cameras
233 and sensor devices. Select crew members on board the vessel where the data is produced
234 can gain access to this data in real-time. This can be through real-time streaming to
235 display monitors, or it might be made available as a configuration option if some of this
236 data is persisted on local disks. No encryption of the data is mandated, and only privacy-
237 preserving aspects must be handled. This might involve that vessel crew members grant
238 consent to store and access this data, or that software applies masking of any personally
239 identifiable characteristics.

240 FILTERED data consists of processed select RAW data that can be persisted to disk
241 and/or used as input to local analysis applications. Such FILTERED data can for instance
242 contain a video sequence with human activities, or sequences of video captured upon
243 other sensory input. Depending on its content, it can be enforced that this data is
244 modified before persisted to disk, in order to be used in analysis applications.

245 Data tagged RESTRICTED is not accessible by any of the crew on board the vessel
246 and is intended for surveillance operations. This data contains specific results obtained
247 from local edge analysis software processing either RAW or FILTERED data. This classifica-
248 tion also indicates that stricter access policies need to be applied, since data might be
249 annotated with additional information from analysis software, and because it is expected
250 to be transmitted over network to a central control center at some point. Examples of this
251 type of data include output from edge located machine learning applications analyzing
252 activities at the fishing vessels, select I-frames from specific surveillance videos, and
253 other sensor data detecting for instance amount of fish caught, fish types, their average
254 size, and relative distribution among species.

255 The purpose of this data classification is to provide context for central control
256 centres, and extract semantic meaning from a larger data set generated at edge nodes.
257 This serves two purposes: (1) reduce the amount of data transmitted, to support lower
258 bandwidths, and (2) transmit as little data as required to provide meaning and perform
259 forensics. This is to apply a principle of minimal privilege of access to parts of a live
260 surveillance feed, as only the data deemed necessary to provide ground truth to some
261 observation or detected event will be transmitted from the edge node.

262 The goal of providing this data overlaps with a goal of the overarching Dutkat
263 system [5], which is to provide a probabilistic and evidence-based approach to inspection
264 of fishery activities, and to shield crew members from being subjected to continuously
265 transmitted surveillance.

266 SECRET data is a complete log of RAW data that is encrypted upon storage and
267 persisted in a highly fault-tolerant manner. This data must be stored locally on the edge
268 nodes due to its sheer volume, and access to it is mandated by very restricted access
269 policies. Notably, SECRET implies that nobody on board the fishing vessel might access
270 it, and the data is immutable and cannot be altered or deleted. This data can optionally
271 be pre-processed upon storage, blurring out personal identification characteristics. The
272 data can only be accessed and decrypted by a trusted third-party with legal, explicit
273 authorization to do so. This can be a fishery inspector or other forensic parties inspecting
274 a vessel with access permissions according to existing laws and regulations.

275 In general, we build this distributed file system adhering to the proportionality
276 principle in a legal context striking a balance between human privacy rights and the
277 claimed sustainability benefits of video and sensory surveillance of fishing waters [12].
278 Invading surveillance on a physically limited area as a trawler deck impossible for the
279 vessel crew to avoid might violate privacy principles well grounded in constitutional,
280 national, and international laws. One example of this is that people in a video sequence
281 can be personally identified while working.

282 4. Implementation Details

283 The data plane described in Section 2.2 is implemented by the Dorvu filesystem,
284 to achieve support of heterogeneous data formats and pre-existing tools for analysis
285 and surveillance, through POSIX-compatibility. In this section we detail some of the
286 implementation details of our prototype.

287 Customization and adaptability are core aspects of the architecture, where the
288 software aims to provide a basic layer of traditional data storage, with the possibility
289 to interposition and add extra functionality modules between applications and the file
290 system storage. We refer to the modules attached to a file as its *meta-code*, similar to the
291 work done in [19]. This provides a means for transparently adding custom software
292 modules in the critical path of data storage.

293 The version of Dorvu implemented for this paper includes (1) Encryption as a
294 module between the user and disk, (2) file versioning based on user access rights,
295 and (3) an interface for a user to configure the encryption module and access control.
296 Additionally, we investigate the performance overhead of this functionality and the
297 userspace file system platform.

298 Interfacing with the Dorvu file system can be listed in three steps, beyond reading
299 and writing as if to a local and un-encrypted file system: (1) Users can define access
300 rights for their own files, indicating identities with public key signatures. (2) When a
301 user creates a file, a corresponding configuration file is created by Dorvu. This defines
302 the available versions of a file to the members of listed access groups. (3) When writing
303 to a file, its file extension, referenced in its corresponding configuration file, decides
304 what access control and encryption the file system will apply to the newly written data.

305 4.1. FUSE

306 Dorvu is implemented as a File System in Userspace (FUSE) application [20]. FUSE
307 is a library and Linux kernel module that enables user-level programs to function
308 as mountable file systems, by calling the FUSE kernel module via the FUSE userspace
309 library. In short, a FUSE daemon can service Linux Virtual File System (VFS) calls despite
310 running with userspace privileges. Dorvu is implemented with the Rust programming
311 language, using the Fuser library¹ to interface with the FUSE kernel module. Fuser
312 provides a userspace library that is implemented separately from the FUSE reference
313 library `libfuse`.

314 Dorvu implements storage by mirroring the contents of a directory on a local file
315 system. By using Dorvu while it is mounted to a local folder, the mirrored folder will
316 be populated with internal files and encrypted data files. These files can only be read
317 through Dorvu, or by means of manual decryption.

318 4.2. File definitions

319 Dorvu handles three different types of files internally. Interfacing with Dorvu as a
320 regular file system is done by creating, writing, and reading files. These files of arbitrary
321 content are referred to as *data files* in the context of this implementation. Creating a data
322 file automatically creates an auxiliary *configuration file*. A data file must always have a
323 configuration file in order to be visible in a Dorvu directory listing. This file contains
324 a JSON specification of the different available versions of a file (referred to as *layers*
325 in the file), in addition to a path to the access group definition to use for the corresponding
326 data file. The *group definition file* is the second type of auxiliary file used in Dorvu. Access
327 group definitions in these files are listed in JSON and require a name and a list of public
328 key SHA-256 signatures. Examples of these files are shown in Figure 3.

¹ <https://crates.io/crates/fuser>

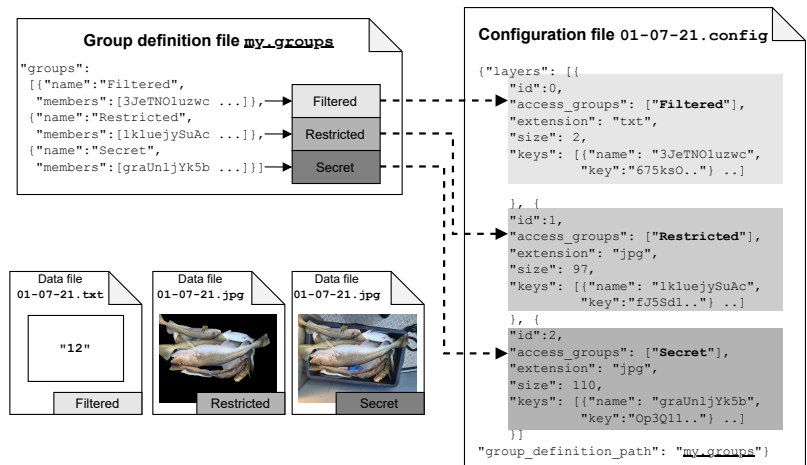


Figure 3. Definition of a configuration file, which defines access rights for a single data file, with its corresponding group definition file. The illustrated data file exists in three versions, one text file and two image files.

329 Data files are matched to their configuration files by their file stems (i.e., their file
 330 names without any directories or extensions). Files are matched to a version defined in
 331 this configuration by its extension, which is expected to be an integer matching the *ID* of
 332 a layer. When listing the contents of a directory, every configuration file at the mirrored
 333 folder corresponding to the working directory will be parsed in order to determine
 334 accessibility and file extension. Every file version whose access group includes a given
 335 user's identity, will be visible in the directory for this user. When multiple versions of
 336 the same file has the same file extension, the version with the lowest identity is deemed
 337 redundant. If a user has access to several versions (of the same file with different file types,
 338 both will be listed, as illustrated with a text file (.txt) and image file (.jpg) in Figure
 339 4. File versions enforced by access control and encryption is a generic implementation of
 340 the data classification scheme overviewed in the system requirements in Section
 341 3.2. With adequate meta-code modules and appropriate access group management, the
 342 required data classification scheme for Dutkat can be implemented in Dorvu.

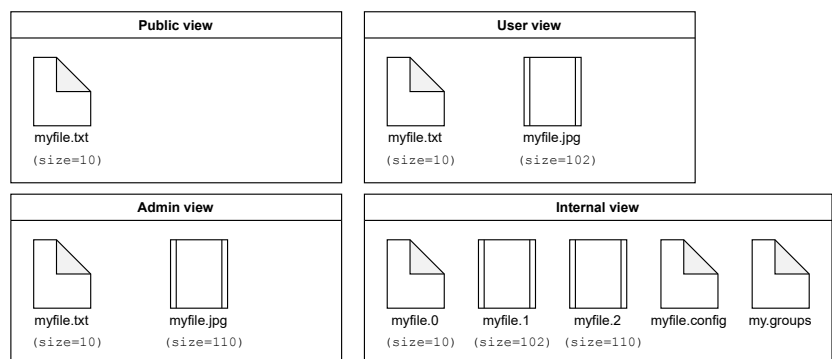


Figure 4. Multiple views of the same directory based on several users' differing access rights, with a scheme similar to Figure 3.

343 As manually maintaining configurations for every newly created file can be time
 344 consuming and increase the chance of human error, Dorvu includes the concept of
 345 *configuration templates*. A configuration template is a directory-wide configuration file

346 that is applied to every newly created file. This feature is introduced for ease-of-use; the
347 expected usage pattern of Dorvu is to set up file configurations before any automated
348 data production begins. In that case, the expected access control will be applied to newly
349 created files without the need for manual maintenance.

350 4.3. Encryption

351 With the encryption module implemented in Dorvu, files classified as SECRET are
352 stored encrypted by default, using the OpenSSL implementation² of 128 bit AES-CBC.
353 The AES key for a given file is encrypted with RSA once for each user with access to that
354 file, using their 2,048 bit public key. A base 64-encoded version of this encrypted AES
355 key is stored in the file's config, as shown in Figure 3.

356 5. Experiments and Results

357 5.1. I/O speed and overhead measurement

358 We want to gain insight into the potential overhead cost by adding Dorvu as an
359 extra layer of indirection in the critical data path for disk access. This experiment is
360 performed by measuring time taken for read and write operations on various storage
361 back-ends. The test environments are chosen to provide information about the expected
362 sources of performance overhead: the cost of encryption, the cost of file versioning
363 and access control, and the cost of utilizing a FUSE-based file system rather than a
364 kernel-integrated file system.

365 To observe the impact of encryption on read/write throughput, we deployed two
366 configurations of Dorvu, one with encryption enabled and one with all encryption fea-
367 tures disabled. To observe the costs associated with deploying a FUSE file system, we
368 implemented a simple FUSE application that forwards all operations to an ext4 file sys-
369 tem, labeled *FUSE passthrough* in our experiment. Our assumption is that the maximum
370 possible throughput of Dorvu will be that of the FUSE passthrough application, and
371 that throughput loss between the FUSE implementation and the ext4 storage back-ends
372 will be outside of the control and scope of our implementation. The difference between
373 the throughput of the encrypted and decrypted Dorvu configurations will indicate the
374 cost of encryption, and the difference between the decrypted Dorvu configuration and
375 the FUSE passthrough application will indicate the cost of file versioning and other
376 metadata operations.

377 5.1.1. Experimental Setup

378 This experiment was performed on a desktop workstation with an AMD Ryzen
379 5 3600 6-Core processor running at 3.60 GHz, and a Kingston UV400 solid state drive
380 storage device, running Ubuntu 18.04. 8 randomly generated files of varying sizes
381 were read and written 10 times per storage environment. These file sizes are chosen
382 due to our use-case of storing media files suited for low-bandwidth network transfer,
383 while we acknowledge that system overhead and inefficiencies are more easily observed
384 during longer operations with larger files. Because of this, later experiments described
385 in sections 5.2 and 5.3 utilize smaller files.

386 5.1.2. Results

387 The results from this experiment are shown in Figure 5. Results show that for
388 reading our largest files of 64 MB and 128 MB, encryption was the biggest source of over-
389 head. For every other test case, however, the difference between the FUSE passthrough
390 performance and unencrypted Dorvu performance indicate that file versioning and
391 metadata operations are the biggest software bottlenecks in Dorvu. We theorize that
392 this is particularly prevalent during file writes because these operations are split into
393 smaller operations of individual page sizes of 4,096 bytes on our test system, resulting in

² <https://crates.io/crates/openssl>

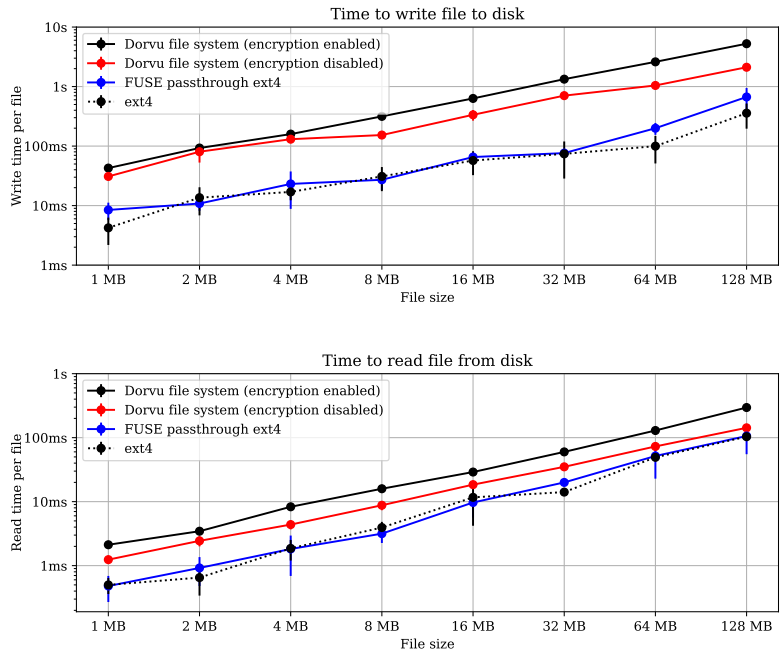


Figure 5. Time measured for read and write operations on varying file sizes and file system configurations. The top graph indicates measurements of the write operation, while the bottom graph indicates the read operation. The y-axis represents time spent on an operation, measured in seconds, plotted on a logarithmic scale. The various files used for this experiment are listed on the x-axis, represented by their file sizes. Plots include error bars, indicating standard deviation on the y-axis.

394 worse performance during writes than reads, relative to the baseline ext4 environment.
 395 We further hypothesize that, while performance penalties associated with encryption
 396 are expected, file versioning and metadata overhead observed in both read and write
 397 operations can be investigated through software profiling, and reduced by further
 398 optimization of the file system.

399 We observe that costs associated with utilizing a FUSE implementation are negli-
 400 gible in many of our observations, particularly during reads, because the majority of
 401 overhead relative to the ext4 environment is visible in the Dorvu environment, and
 402 not in the FUSE passthrough application. It is worth noting that the measurements for
 403 both the FUSE and ext4 storage deviate by up to 30% between minimum and maximum
 404 observations, but their averages are within each other's standard deviation for every file
 405 size in the experiment.

406 The FUSE passthrough comparison measurement was also performed in [19], but
 407 was re-implemented for this experiment due to our adoption of the Rust programming
 408 language and its Fuser library. A more thorough examination of the performance
 409 implications of utilizing user-space file systems is given by Vangoor et al. [21]. They
 410 show that the throughput penalty of using FUSE over ext4 can be as low as 5%, but that
 411 certain workload characteristics can severely negatively impact this performance.

412 We conjecture that, for our use-case of optimizing for low-bandwidth transfer, uti-
 413 lizing a FUSE implementation does not significantly negatively impact the performance

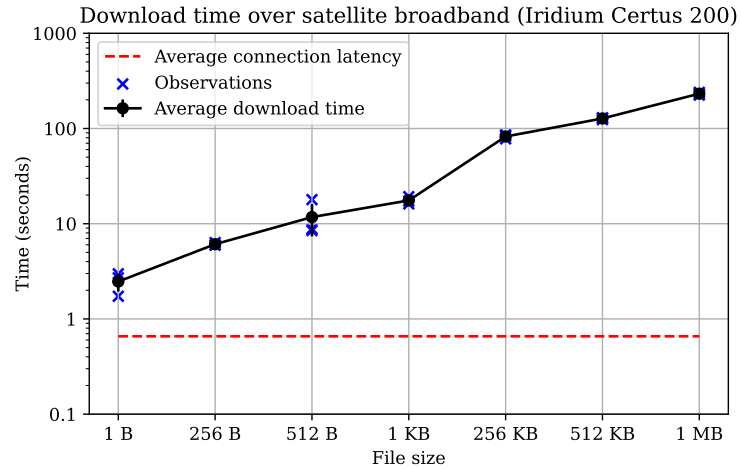


Figure 6. Observed time to download files at various sizes over a satellite broadband connection. Time is represented in a logarithmic scale on the y-axis, to show average download times of the various files, represented on the x-axis by their sizes. The average connection latency observed throughout every established connection during this experiment is also shown.

414 of Dorvu, while recognizing that a future use-case involving larger file sizes or different
 415 workload characteristics may change this outlook.

416 5.2. Satellite latency and bandwidth

417 Network communication between edge components in Dorvu and the Dutkat sys-
 418 tem will be provided by satellite broadband. Observations of the capabilities of the
 419 available satellite network are key to designing communication models and delegating
 420 tasks to edge and land components in the system. In addition to measuring the sustained
 421 average bandwidth provided by the network, we measure the transfer speed of individ-
 422 ual files of specific sizes. This is both to emulate file system usage on the network, and
 423 to give an indication of the impact of network latency when transferring small amounts
 424 of data.

425 5.2.1. Experimental Setup

426 The experiment was performed by using the Linux `curl` command to download files
 427 from a test GitHub repository with files generated for the purpose of this experiment.
 428 The experiment is performed on the Iridium Certus 200 broadband satellite service,
 429 an L-band non-geostationary satellite network claiming global satellite coverage and
 430 download and upload speeds of up to 176 kilobits per second [22].

431 We connect to this network through a Thales Avionics VesseLink 200 broadband
 432 terminal, consisting of an antenna and router for maritime use [23]. The experiment
 433 was ran on an HP EliteDesk 800G6 workstation with the VesseLink providing its only
 434 connected network. The equipment is stationed at the University of Tromsø, Norway
 435 and was tested during cloudy weather conditions with drizzle, and each download was
 436 repeated five times. This number of downloads was chosen to adhere to restrictions on
 437 network resources.

438 5.2.2. Results

439 The purpose of this experiment is to observe the capabilities and limitations of
 440 the hardware and network available to our system when deployed in the targeted en-
 441 vironment and weather conditions. The experiment is not intended as an exhaustive

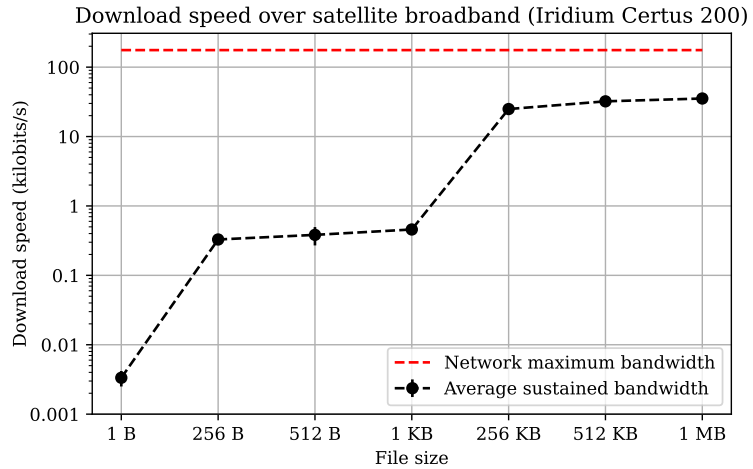


Figure 7. Observed download speed when downloading files of various sizes over a satellite broadband connection. The y-axis represents download speed, measured in kilobits per second, of the files represented on the x-axis. Additionally, the theoretical maximum network bandwidth as defined by the system provider [22,23] is shown.

442 evaluation of the feasibility of satellite broadband, or the performance of this particular
 443 satellite service in broadly defined use-cases, but to explore various network conditions
 444 that our system must handle gracefully. These results will also be applied in the
 445 experiment described in Section 5.3.

446 The resulting measurements from this experiment are shown in figures 6 and 7.
 447 Note that in both of these figures, the y-axis follows a logarithmic scale, while the x-axis
 448 follows no particular scale, rather representing a set of files. The standard deviation on
 449 the y-axis is plotted as error bars.

450 Results shown in both figures 6 and 7 indicate the same expected observation,
 451 that the average download speed throughout the download process is lower when
 452 transmitting smaller files. We assume this is a result of the time of initiating the file
 453 transfer connection and cost of transferring metadata is proportionally more significant
 454 the smaller the payload. Some storage systems are designed to handle and distribute
 455 large amounts of small files specifically, to alleviate weaknesses of existing protocols in
 456 this use-case [24,25].

457 It is observed in Figure 7 that the achieved download speed is considerably lower
 458 than the network maximum, which can be influenced by several factors, such as weather
 459 conditions, relative satellite location, and network traffic [26].

460 5.3. Machine Learning Workloads on the Edge vs. a Centralized Hub

461 We argue that analysis should be performed on the edge to preserve privacy. Addi-
 462 tionally, based on our end-to-end satellite communication experiments in Section 5.2,
 463 we conjecture that transferring raw video data from the edge nodes to a centralized
 464 mainland hub has its performance limitations. We would therefore like to evaluate such
 465 a centralized system to see if it is feasible. The typical workload in our fishery use-case
 466 is activity recognition based on video data to determine whether e.g. discard of fish has
 467 occurred. Therefore we will test if a centralized system could work based on an activity
 468 recognition workload.

469 In order to evaluate the throughput of a centralized system we focus on the bitrate
 470 required to run inference on videos in real-time and compare against the average band-
 471 width measured for the satellite connection. We send different levels of compressed

472 video data over our satellite connection and identify the top-1 video-level accuracy of
473 models trained on this data. We compress the video by reducing the resolution and/or
474 reducing the frame rate. We aim to see how the compression affects the accuracy of the
475 machine learning model chosen. If the accuracy decreases significantly from the base
476 case (112x112, 30 fps), then it is not feasible to send data over the satellite connection and
477 the inference should be performed locally.

478 If we use the best average results for bandwidth from Figure 7, we get an average
479 bandwidth of ≈ 35 kbps over the satellite connection. Given that the video files, on
480 average, are much smaller than 1 MB, this is a conservative estimate. The optimal
481 bandwidth is taken from documentation sheets for the satellite router [22,23]. The
482 bandwidth required to send data over the satellite connection should be lower than the
483 average bandwidth measured.

484 5.3.1. Experimental Setup

485 In our experiments, we utilize a 18-layer R(2+1)D network, introduced by Tran et
486 al. [27], to perform action recognition. The network was pretrained on the Kinetics-400
487 dataset [28] and then fine-tuned on the HMDB51 dataset [29]. The video data's resolution
488 and frame rate are reduced to various degrees, while measuring required bandwidth.
489 The network is finetuned over 50 epochs and the weights from the epoch which gave
490 the best validation accuracy are kept. This network is then run on a test set giving the
491 final accuracy in Figure 8.

492 We train on 16-frame clips, as was done during pre-training on the Kinetics-400
493 dataset. If the framerate is too low, we repeat the last frame until we fill the tensor. The
494 frames are consecutive and we apply temporal jittering while training. The video-level
495 accuracy is calculated by taking the average prediction of 20 different clips from the same
496 video, and then we choose the top-1 result. The bandwidth required for the different
497 levels of compressed video data was calculated by taking the average bitrate of all
498 compressed videos in the HMDB51 dataset.

499 We implemented the experiment using PyTorch [30], and the model was trained on
500 an Nvidia RTX 2080 Ti. The model was imported from the `torchvision` module. The
501 frames are extracted from the videos and are resized and combined into a tensor in the
502 batch generator. The frames are augmented randomly using horizontal flips and affine
503 translations before they are normalized according to the means and standard deviations
504 of the Kinetics-400 dataset [31].

505 5.3.2. Results

506 We hypothesized that compressing data, in order to adhere to bandwidth restric-
507 tions, would lead to lower accuracy for the action recognition model. As we can observe
508 in Figure 8, reducing the resolution results in a dramatic decrease in model performance.
509 Reducing the frame rate also decreases the accuracy, but not to the same degree. The
510 highest possible accuracy we achieve that requires a bandwidth lower than the average
511 bandwidth is at 43.81 %, which is much lower than our highest accuracy at 69.3 %.
512 Assuming we want a high-performing model, with results as close to state-of-the-art
513 as possible (see Figure 8), we conclude that performing inference on a centralized hub
514 is infeasible. The reason for the discrepancy in our highest accuracy and the accuracy
515 documented in [27] might be due to numerous factors, such as training time, different
516 augmentation schemes, learning rate scheduling, etc.

517 Based on our experiments, upstream evaluation is a more realizable design option
518 for our application scenario [13]. As our system should support real-time monitoring,
519 we will choose the evaluation scheme and inference location based on the capabilities of
520 transferring results from edge nodes in real-time. Hence, data should be analyzed close
521 to its source with inference on video data performed on the edge, on board the fishing
522 vessel where the video camera is located. This design choice also complies with the

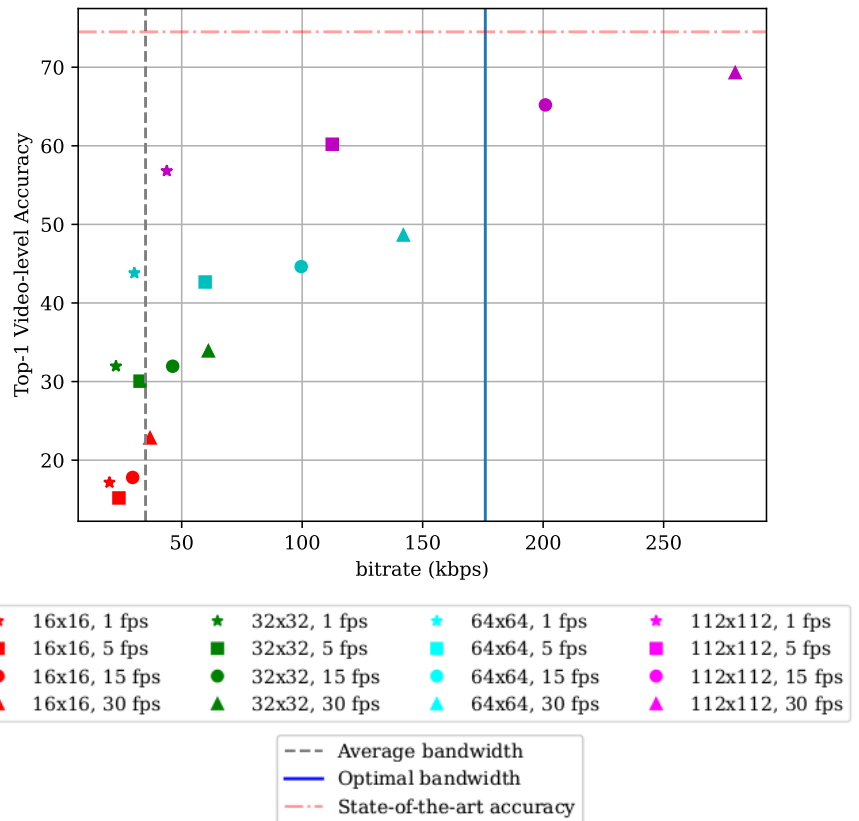


Figure 8. Plot over top-1 video-level accuracy (y-axis) vs. bitrate required to perform real-time inference (x-axis), including vertical lines indicating average and optimal bandwidth possible on our satellite connection. The different colors represent different resolutions of the video data used in fine-tuning the machine learning model, and the shapes are different frame rates. The horizontal line represents the top-1 video-level accuracy achieved by fine-tuning on HMDB51 in [27].

523 privacy-preserving design of the system, with the edge nodes performing privacy-critical
524 operations.

525 **6. Related work**

526 *6.1. Privacy-preserving surveillance*

527 A surveillance system with built-in video processing and access control based
528 on video analysis was presented in IBM's PrivacyCam [32]. This surveillance system
529 provides cameras with the capability to re-render an input video stream with features
530 such as persons or objects removed. Unedited output streams can be provided for
531 authorized users. Various techniques were applied in similar surveillance systems, for
532 instance by removing distinctive facial features [33], encrypting faces [34], or obscuring
533 people based on specific visual markers [35].

534 *6.2. File systems*

535 Numerous systems provide encryption as a transparent file system feature or as
536 software on top of a traditional file system. Cryptfs [36] and its successor eCryptfs [37]
537 are cryptographic file systems included in the Linux kernel that provide encryption to
538 files located in local or remote file systems, by storing cryptographic metadata in headers
539 of individual files. Similarly, software like TrueCrypt [38], VeraCrypt [38], and Apple
540 FileVault provide a decrypted view of an encrypted directory mounted elsewhere in the
541 file system.

542 Several projects provide cryptographic file systems implemented in FUSE. EncFS [39]
543 runs in userspace and mounts to a directory in a local file system and encrypts all data
544 written, storing encrypted versions of these files in a separate location. Gocryptfs [40]
545 is a similar project implemented in the Go programming language with the Go-FUSE
546 kernel bindings library. SecureFS [41] is a C++ FUSE project that aims to provide similar
547 features as EncFs and Gocryptfs to multiple operating systems. Common for these FUSE
548 file systems and Dorvu is that they are implemented as an overlay file system, providing
549 a layer of indirection before writing to a separate local or remote file system. The design
550 of generalized layered file systems and the technology that enables them on various
551 platforms are reviewed and discussed by Zadok et al. [42].

552 *6.3. Extensibility*

553 Architecting extensible software in the offshore domain resembles how we structured
554 our StormCast system [15] [43], which further motivated the early mobile agent
555 system TACOMA [44] built for shipping code and state around in a network for remote
556 installation. Our current work utilizes the meta-code concept [19] for extending and
557 customizing remote nodes where remote software can be configured with mobile code.

558 Our previous Balava file system [45] was built with FUSE and meta-code for man-
559 aging computations that coupled multiple public clouds together transparently, and
560 involved data with confidentiality constraints. Meta-code as a structuring toolkit is used
561 as in Dorvu, but not in a weakly connected, mobile edge environment. Meta-code is used
562 in Balava for transparently gluing together a hybrid cloud system that interconnects
563 private environments with public clouds such as Microsoft Azure and Amazon Web
564 Services.

565 *6.4. Data transmission*

566 To avoid transmitting irrelevant and redundant data over the bandwidth-limited
567 links from the remote edge devices to the central cloud-based servers, we aim to ap-
568 ply several data reduction mechanisms. By performing most of the analysis locally,
569 transmission of large amounts of data can be reduced. This is especially important for
570 bandwidth-hungry data types like images and videos. Multiple approaches have been
571 explored for reducing the amount of data generated and for reducing data transmission.
572 For instance, Gurrin et al. [46] propose a system that detects action in images and keeps

573 only images where action is detected. Ji et al. [47] extract features from both the spatial
574 and the temporal dimensions by performing 3D convolutions, thereby capturing the
575 motion information encoded in multiple adjacent frames. Such approaches are used to
576 reduce data, both for storage, transmission, and later analysis. Further reductions can
577 be achieved reducing image or frame dimensions and sizes without losing important
578 information [48], and analysing the tradeoffs between better quantization and reducing
579 the frame rate [49].

580 Compression of video data using machine learning is also something we investi-
581 gated and compared against in Section 5.3. Related approaches include Nvidia Max-
582 ine [50,51], a recently developed tool for massive compression of video data for video
583 conferences. This application domain involves videos of faces with typically static back-
584 grounds. It is challenging to apply this approach to our application domain, with video
585 data depicting general activities, because it requires large amounts of data to train an
586 equivalent generative adversarial network. Similar video analysis must be performed
587 for privacy, e.g., avoiding to show faces or objects that should for some reason be pro-
588 tected. For example, Fitwi et al. [52] describe a system for masking private information
589 in video frames from surveillance cameras by doing detection and filtering on the edge.
590 Moreover, D'souza et al. [53] describe a similar system that uses object detection for
591 surveillance camera video streams, and whitelists classes of objects that should *not* be
592 censored. Thus, such approaches will both reduce bandwidth, but also provide support
593 for privacy preservation. Neto et al. [54] describe an edge-based system for smart city
594 applications. They describe a system for real-time processing of data that preserves
595 privacy, that also utilizes a workload balancer to balance tasks across multiple edge
596 nodes. However, this workload distribution is not applicable for our application, since
597 edge nodes are expected to be physically distant from each other.

598 6.5. Centralized data analysis

599 We have proposed that the desired rate of data production in our system is larger
600 than the targeted satellite communication link can transfer in real-time. Despite band-
601 width restrictions, it can still be advantageous to analyze data from multiple nodes and
602 sensors, and potentially in combination with additional data collected from third-parties,
603 such as sales notes and weather data.

604 Multimodal analysis of data is usually leading to better and more accurate results
605 as recent work shows, but comes with additional costs regarding the hardware needed
606 [55,56]. Especially ensembles of experts models work well with multimodal data streams
607 and complex task analysis [57,58] which makes them a good alternative for the presented
608 use case. For future work we can use pre-analyzed streams of data that will act as input
609 to an expert ensemble model in which each of the expert sub-networks will focus on
610 learning the specific patterns of that particular data stream.

611 7. Conclusions

612 We are developing a geo-distributed, loosely coupled AI system for surveillance of
613 fishing activities in the Arctic Ocean. The development and deployment of this system
614 comes with several challenges, due to the nature of the data produced and the targeted
615 edge environment. For example, continuous production of multimedia data requires
616 privacy compliance and fault-tolerance, while the bandwidth of edge networks hinders
617 data transmission and real-time monitoring from non-edge components in the system.
618 We observe that our available satellite broadband networks are not suitable for real-time
619 video transmission for activity recognition, and we propose a system for analysis and
620 data storage on the edge to facilitate this.

621 We have presented details of a prototype of Dorvu, a geo-distributed file system
622 with support for fine-grained access control policies and software modules. Our pro-
623 totype demonstrates an implementation of encryption and file versioning based on
624 access rights, and we outline the expected I/O overhead of encryption and metadata

625 operations associated with these capabilities. The deployed version of this system will be
626 spanning edge nodes on fishing vessels out at sea connected with mainland centralized
627 file servers, to utilize a combination of data filtering, analysis, and access control, to
628 serve as a privacy-preserving alternative to manual video surveillance.

629 **Author Contributions:** Conceptualization, D. Johansen; methodology, A.B. Ovesen and D. Johansen; software, A.B. Ovesen, T.A.S. Nordmo and M.A. Riegler; validation, A.B. Ovesen and
630 T.A.S. Nordmo, investigation, A.B. Ovesen and T.A.S. Nordmo; data curation, T.A.S. Nordmo;
631 writing—original draft preparation, A.B. Ovesen, T.A.S. Nordmo, M.A. Riegler, P. Halvorsen and D.
632 Johansen; writing—review and editing, A.B. Ovesen, T.A.S. Nordmo, H.D. Johansen, M.A. Riegler,
633 P. Halvorsen and D. Johansen; visualization, A.B. Ovesen and T.A.S. Nordmo; supervision, H.D.
634 Johansen and D. Johansen; project administration, H.D. Johansen, P. Halvorsen and D. Johansen;
635 funding acquisition, H.D. Johansen and D. Johansen.

637 **Funding:** This work is partially funded by the Research Council of Norway project numbers
638 274451 and 263248, and Lab Nord-Norge ("Samfunnsløftet").

639 **Acknowledgments:** We particularly acknowledge contributions from Kim H. Andreassen, Arne
640 E. Karlsen, Nandor Knust, Jon F. Mikalsen, Magnar Pedersen, Jon P. Rui, and Kjetil Robertsen.

641 **Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the
642 design of the study; in the collection, analyses, or interpretation of data; in the writing of the
643 manuscript, or in the decision to publish the results.

References

1. Satyanarayanan, M.; Simoens, P.; Xiao, Y.; Pillai, P.; Chen, Z.; Ha, K.; Hu, W.; Amos, B. Edge analytics in the internet of things. *IEEE Pervasive Computing* **2015**, *14*, 24–31.
2. Wang, S.; Zhang, X.; Zhang, Y.; Wang, L.; Yang, J.; Wang, W. A survey on mobile edge networks: Convergence of computing, caching and communications. *Ieee Access* **2017**, *5*, 6757–6779.
3. Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C.; Konečný, J.; Mazzocchi, S.; McMahan, H.B.; others. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046* **2019**.
4. Zhang, Y.; McQuillan, F.; Jayaram, N.; Kak, N.; Khanna, E.; Kislal, O.; Valdano, D.; Kumar, A. Distributed Deep Learning on Data Systems: A Comparative Analysis of Approaches. *Proc. VLDB Endow* **2021**, *14*.
5. Nordmo, T.A.S.; Ovesen, A.B.; Johansen, H.D.; Riegler, M.A.; Halvorsen, P.; Johansen, D. Dutkat: A Multimedia System for Catching Illegal Catchers in a Privacy-Preserving Manner. Proceedings of the 2021 Workshop on Intelligent Cross-Data Analysis and Retrieval; Association for Computing Machinery: New York, NY, USA, 2021; ICDAR '21, p. 57–61.
6. Costello, C.; Cao, L.; Gelcich, S.; Cisneros-Mata, M.Á.; Free, C.M.; Froehlich, H.E.; Golden, C.D.; Ishimura, G.; Maier, J.; Macadam-Somer, I.; Mangin, T.; Melnychuk, M.C.; Miyahara, M.; de Moor, C.L.; Naylor, R.; Nøstbakken, L.; Ojea, E.; O'Reilly, E.; Parma, A.M.; Plantinga, A.J.; Thilsted, S.H.; Lubchenco, J. The future of food from the sea. *Nature* **2020**, *588*, 95–100.
7. UNODC. Fisheries Crime: transnational organized criminal activities in the context of the fisheries sector **2016**.
8. Øystein Ingilæ. Fiskere settes under overvåkning. *Kyst og Fjord*.
9. Márcia Bizzotto. Fishing rules: Compulsory CCTV for certain vessels to counter infractions. *European Parliament Press Release*.
10. Ministry of Trade, Industry and Fisheries. Framtidens Fiskerikontroll. *NOU 19:21* **2019**.
11. Martinussen, T.M. Danske fiskere samler seg mot kamera-overvåkning i fiskeriene. *Fiskeribladet*.
12. van Helmond, A.T.; Mortensen, L.O.; Plet-Hansen, K.S.; Ulrich, C.; Needle, C.L.; Oesterwind, D.; Kindt-Larsen, L.; Catchpole, T.; Mangi, S.; Zimmermann, C. Electronic monitoring in fisheries: lessons from global experiences and future opportunities. *Fish and Fisheries* **2020**, *21*, 162–189.
13. Carzaniga, A.; Rosenblum, D.S.; Wolf, A.L. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems (TOCS)* **2001**, *19*, 332–383.
14. Satyanarayanan, M.; Kistler, J.J.; Kumar, P.; Okasaki, M.E.; Siegel, E.H.; Steere, D.C. Coda: A highly available file system for a distributed workstation environment. *IEEE Transactions on computers* **1990**, *39*, 447–459.
15. Johansen, D. StormCast: Yet another exercise in distributed computing. *Distributed Open Systems in Perspective* **1993**.
16. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union* **2016**.
17. Ghemawat, S.; Gubioff, H.; Leung, S.T. The Google file system. Proceedings of the nineteenth ACM symposium on Operating systems principles, 2003, pp. 29–43.
18. Dean, J.; Ghemawat, S. MapReduce: Simplified data processing on large clusters **2004**.
19. Johansen, H.D.; Birrell, E.; Van Renesse, R.; Schneider, F.B.; Stenhaug, M.; Johansen, D. Enforcing privacy policies with meta-code. Proceedings of the 6th Asia-Pacific Workshop on Systems, 2015, pp. 1–7.

20. FUSE - The Linux Kernel documentation. <https://www.kernel.org/doc/html/latest/filesystems/fuse.html>. Accessed: 2021-08-30.
21. Vangoor, B.K.R.; Tarasov, V.; Zadok, E. To FUSE or not to FUSE: Performance of user-space file systems. Proceedings of 15th USENIX Conference on File and Storage Technologies, 2017, pp. 59–72.
22. Iridium Certus 200. <https://www.iridium.com/services/iridium-certus-200/>. Accessed: 2021-09-08.
23. Thales VesseLINK 200. https://www.thalesgroup.com/sites/default/files/database/document/2021-02/2807_V1_VesseLINK200_012021.pdf. Accessed: 2021-09-08.
24. Yu, L.; Chen, G.; Wang, W.; Dong, J. Msfs: A storage system for mass small files. Proceedings of 11th International Conference on Computer Supported Cooperative Work in Design. IEEE, 2007, pp. 1087–1092.
25. Thain, D.; Moretti, C. Efficient access to many small files in a filesystem for grid computing. Proceedings of 8th IEEE/ACM International Conference on Grid Computing. IEEE, 2007, pp. 243–250.
26. Gerard, M.; Bousquet, M. *Satellite communications systems*; Teubner, 1993.
27. Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; Paluri, M. A closer look at spatiotemporal convolutions for action recognition. Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2018, pp. 6450–6459.
28. Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; Suleyman, M.; Zisserman, A. The Kinetics Human Action Video Dataset, 2017, [arXiv:cs.CV/1705.06950].
29. Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; Serre, T. HMDB: a large video database for human motion recognition. Proceedings of the 2011 International conference on computer vision. IEEE, 2011, pp. 2556–2563.
30. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc., 2019; pp. 8024–8035.
31. Ashley, K. *Applied Machine Learning for Health and Fitness*; Springer, 2020.
32. Senior, A.; Pankanti, S.; Hampapur, A.; Brown, L.; Tian, Y.L.; Ekin, A.; Connell, J.; Shu, C.F.; Lu, M. Enabling video privacy through computer vision. *IEEE Security & Privacy* **2005**, 3, 50–57.
33. Newton, E.M.; Sweeney, L.; Malin, B. Preserving privacy by de-identifying face images. *IEEE transactions on Knowledge and Data Engineering* **2005**, 17, 232–243.
34. Boulton, T.E. PICO: Privacy through invertible cryptographic obscuration. Proceedings of Computer Vision for Interactive and Intelligent Environment (CVIIIE). IEEE, 2005, pp. 27–38.
35. Schiff, J.; Meingast, M.; Mulligan, D.K.; Sastry, S.; Goldberg, K. Respectful cameras: Detecting visual markers in real-time to address privacy concerns. In *Protecting Privacy in Video Surveillance*; Springer, 2009; pp. 65–89.
36. Zadok, E.; Badulescu, I.; Shender, A. Cryptfs: A stackable vnode level encryption file system. Technical report, Technical Report CUCS-021-98, Computer Science Department, Columbia University, 1998.
37. Halcrow, M.A. eCryptfs: An enterprise-class encrypted filesystem for linux. Proceedings of the 2005 Linux Symposium, 2005, Vol. 1, pp. 201–218.
38. VeraCrypt - free open source disk encryption software. <https://veracrypt.fr/>. Accessed: 2021-08-15.
39. encFS - an Encrypted Filesystem. <https://vgough.github.io/encfs/>. Accessed: 2021-08-15.
40. gocryptfs - simple. secure. fast. <https://nuetzlich.net/gocryptfs/>. Accessed: 2021-08-15.
41. Filesystem in userspace (FUSE) with transparent authenticated encryption. <https://github.com/netheril96/securefs/>. Accessed: 2021-08-15.
42. Zadok, E.; Iyer, R.; Joukov, N.; Sivathanu, G.; Wright, C.P. On incremental file system development. *ACM Transactions on Storage (TOS)* **2006**, 2, 161–196.
43. Hartvigsen, G.; Johansen, D. Co-operation in a distributed artificial intelligence environment—the stormcast application. *Engineering Applications of Artificial Intelligence* **1990**, 3, 229–237.
44. Johansen, D.; Van Renesse, R.; Schneider, F.B. Operating system support for mobile agents. Proceedings of 5th Workshop on Hot Topics in Operating Systems (HotOS-V). IEEE, 1995, pp. 42–45.
45. Nordal, A.; Kvalnes, Å.; Hurley, J.; Johansen, D. Balava: Federating private and public clouds. Proceedings of 2011 IEEE World Congress on Services. IEEE, 2011, pp. 569–577.
46. Gurrin, C.; Aarflot, T.; Johansen, D. GARDI : A Self-Regulating Framework for Digital Libraries. Proceedings of the IEEE International Conference on Computer and Information Technology, 2009, pp. 305–310.
47. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2013**, 35, 221–231.
48. Ng, S. Principal component analysis to reduce dimension on digital image. *Procedia Computer Science* **2017**, 111, 113–119.
49. McCarthy, J.D.; Sasse, M.A.; Miras, D. Sharp or Smooth? Comparing the Effects of Quantization vs. Frame Rate for Streamed Video. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2004, p. 535–542.
50. AI Can See Clearly Now: GANs Take the Jitters Out of Video Calls. <https://blogs.nvidia.com/blog/2020/10/05/gan-video-conferencing-maxine/>. Accessed: 2021-09-08.
51. Wang, T.C.; Mallya, A.; Liu, M.Y. One-shot free-view neural talking-head synthesis for video conferencing. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10039–10049.

52. Fitwi, A.; Chen, Y.; Zhu, S.; Blasch, E.; Chen, G. Privacy-Preserving Surveillance as an Edge Service Based on Lightweight Video Protection Schemes Using Face De-Identification and Window Masking. *Electronics* **2021**, *10*, 236.
53. Dsouza, S.; Bahl, V.; Ao, L.; Cox, L.P. Amadeus: Scalable, Privacy-Preserving Live Video Analytics. *arXiv preprint arXiv:2011.05163* **2020**.
54. Rocha Neto, A.; Silva, T.P.; Batista, T.; Delicato, F.C.; Pires, P.F.; Lopes, F. Leveraging Edge Intelligence for Video Analytics in Smart City Applications. *Information* **2021**, *12*.
55. Hosseini, M.P.; Tran, T.X.; Pompili, D.; Elisevich, K.; Soltanian-Zadeh, H. Multimodal data analysis of epileptic EEG and rs-fMRI via deep learning and edge computing. *Artificial Intelligence in Medicine* **2020**, *104*, 101813.
56. Lu, R.; Cai, Y.; Zhu, J.; Nie, F.; Yang, H. Dimension reduction of multimodal data by auto-weighted local discriminant analysis. *Neurocomputing* **2021**, *461*, 27–40.
57. Zhai, Y.; Ye, Q.; Lu, S.; Jia, M.; Ji, R.; Tian, Y. Multiple expert brainstorming for domain adaptive person re-identification. Proceedings of 16th European Conference on Computer Vision, Part VII 16. Springer, 2020, pp. 594–611.
58. Zhang, W.; Yang, D.; Zhang, S.; Ablanedo-Rosas, J.H.; Wu, X.; Lou, Y. A novel multi-stage ensemble model with enhanced outlier adaptation for credit scoring. *Expert Systems with Applications* **2021**, *165*, 113872.

A.4. Paper III: Fishing Trawler Event Detection: An important step towards digitization of sustainable fishing

A.4 Paper III: Fishing Trawler Event Detection: An important step towards digitization of sustainable fishing

Authors: T.S. Nordmo, A.B. Ovesen, H.D. Johansen, P. Halvorsen, M.A. Riegler, and D. Johansen

Abstract: Detection of anomalies within data streams is an important task that is useful for different important societal challenges such as in traffic control and fraud detection. To be able to perform anomaly detection, unsupervised analysis of data is an important key factor, especially in domains where obtaining labelled data is difficult or where the anomalies that should be detected are often changing or are not clearly definable at all. In this article, we present a complete machine learning based pipeline for real-time unsupervised anomaly detection that can handle different input data streams simultaneously. We evaluate the usefulness of the proposed method using three well-known datasets (fall detection, crime detection, and sport event detection) and a completely new and unlabelled dataset within the domain of commercial fishing. For all datasets, our method outperforms the baselines significantly and is able to detect relevant anomalies while simultaneously having low numbers of false positives. In addition to the good detection performance, the presented system can operate in real-time and is also very flexible and easy to expand.

Author contributions (initials): **Conceptualisation:** T-A.S.N., M.A.R.; **Data collection:** T-A.S.N., **Methods, data analysis and interpretation:** T-A.S.N., **Drafting:** T-A.S.N., M.A.R., A.B.O., H.D.J, P.H., D.J., **Critical revision:** T-A.S.N., M.A.R., A.B.O., H.D.J, P.H., D.J.

Published: (In Submission) International Conference on Applied Artificial Intelligence (ICAPAI) 2023

Thesis objectives: Sub-objective 4

Fishing Trawler Event Detection: An important step towards digitization of sustainable fishing

1st Blinded
Blinded

Abstract—Detection of anomalies within data streams is an important task that is useful for different important societal challenges such as in traffic control and fraud detection. To be able to perform anomaly detection, unsupervised analysis of data is an important key factor, especially in domains where obtaining labelled data is difficult or where the anomalies that should be detected are often changing or are not clearly definable at all. In this article, we present a complete machine learning based pipeline for real-time unsupervised anomaly detection that can handle different input data streams simultaneously. We evaluate the usefulness of the proposed method using three well-known datasets (fall detection, crime detection, and sport event detection) and a completely new and unlabelled dataset within the domain of commercial fishing. For all datasets, our method outperforms the baselines significantly and is able to detect relevant anomalies while simultaneously having low numbers of false positives. In addition to the good detection performance, the presented system can operate in real-time and is also very flexible and easy to expand.

Index Terms—Event detection, Unsupervised, Sustainable Fishing

I. INTRODUCTION

Data streams containing video, images and sensor data are being generated in multiple areas of society. Humans and machines are constantly observed at different frequencies and qualities. Examples include traffic or security video surveillance, monetary fraud transaction monitoring, or detection of errors in industry production facilities. To be able to build useful and efficient analysis applications in such domains, one usually requires data streams with labelled data like, for example, bounding boxes around people in surveillance videos or annotations of events in sensor data streams. For some applications and data streams, this is feasible, but for others, it is not. This often depends on the amount of data (too much data to annotate, requiring large amounts of manual labor), or the simple fact that one does not know what to annotate. Proper datasets are key to development of machine learning applications. Problems in many existing datasets include completeness, quantity, validity, and correctness of data, and correct labelling of data for supervised learning approaches. For some application domains, proper datasets are not available at all, or have very limited value. For instance, finding outlier values in a series of data, that is to temporally or spatially localize the anomaly events in a time-series sequence, can be challenging.

These challenges are specifically relevant for anomaly detection, because it is often not known what anomalies can happen, and it would require a huge number of annotations.

In addition, feature extraction comes with a computational cost that might be a bottleneck in, for instance, a real-time surveillance context. The current trend in research is to find alternative solutions often incorporating semi-, self- or unsupervised [1]–[3] learning. Most of these solutions are focused on one specific type of data stream, i.e., video or sensor data only, although some application scenarios provide several data streams simultaneously that can be analysed to produce a better result (e.g., multiple video streams from different angles, video and sensor data from traffic, or sensors measuring different bio signals from an intensive care unit patient).

To address the challenge of anomaly detection when multiple data streams are provided, we propose an unsupervised anomaly detection system that is able to handle several, multimodal data streams simultaneously. The system is designed based on the specific use-case given in the Dutkat project targeting sustainable harvesting of marine resources off-shore [4]. This is a multi-billion dollar industry worldwide, but one that also comes with serious problems according to, for instance, the United Nations and their sustainability focus [5]. The main goal of Dutkat is to detect potential criminal activity on large off-shore fishing vessels by introducing robust, privacy-preserving edge-computing systems [6], and multimodal data streams on each vessel. Although the Dutkat use-case was one of the main motivators for developing the system, we also show that it is very flexible and can be applied to totally different use-cases involving data streams and anomaly detection. We demonstrate that our system is relatively application domain agnostic by using datasets from the surveillance, elderly care (unexpected falls), and sport domains.

The main contributions of this paper are as following: (i) we propose an unsupervised anomaly detection system that can handle several, multimodal data streams simultaneously in the environment of a fishing trawler that come with specific requirements and limitations, (ii) we evaluate our system on three different datasets with temporal annotations; and (iii) we apply it on unlabeled data from a fishing trawler's surveillance system, where we manually validate the results.

Our experiments show that the proposed system is able to detect anomalies completely unsupervised in an efficient manner, and utilizes information from different data streams if available. In addition, we show that it outperforms several core baselines for the labeled datasets. Finally, and most important we show that it can be used for the specific use

case of anomaly detection on fishing trawlers that comes with requirements.

II. RELATED WORK

Anomaly detection has been researched extensively in the past years. Most novel methods rely on deep-learning based approaches [7] that require a lot of training data. Less research has been performed in the direction of unsupervised machine learning for anomaly detection. This is probably due to several factors such as difficulty to verify and evaluate the output and the general lower performance of unsupervised methods compared to supervised ones [8]. Some methods also rely on a combination of supervised and unsupervised learning [9].

[10] present an unsupervised anomaly detection algorithm for traffic video data that achieves an F1 score of 0.5926 on the NVIDIA AI CITY 2020 challenge test dataset [11]. [12] proposed another unsupervised method using autoencoders to detect anomalies in high-performance computing systems. The challenge in this specific area is that the available datasets are rather small and supervised methods cannot easily be used. In addition to the challenges of lacking enough training data, detecting unknown abnormalities in real-time has attracted focused research. This is often an important requirement for systems that intend to be used in real world scenarios that are time-critical [13], [14].

Applying anomaly detection in real-world scenarios comes with several challenges that we are tackling with the presented work here. First, a real-world capable system needs to handle the input data in real-time and optimally should also be able to deal with new data or additional data streams. Furthermore, depending on the application, one might not know what different types of anomalies can happen and how these anomalies might change over time. Thus, a complete system needs to be agile and able to react to changes and new anomalies. This comes with a trade-off in terms of recall and precision, that is whether all relevant events are captured versus how many of the captured events reported actually are relevant ones. In scenarios like capturing potential fraudulent activities on fishing vessels, one rather would like to detect all suspicious events and accept a larger number of potentially false positives [4]. Additionally, the fishing trawler use case also comes with limitations such as very limited bandwidth due to satellite connections, computational power limitations and that the detected anomalies will change over time due to different reasons such as change of crew or behaviour on the boat. This rules out most existing related work since we cannot rely on a model that is trained on annotated data or methods that are computational heavy. Thus, we present in this paper a method that is light weight and specifically designed for the application on fishing trawlers.

III. SYSTEM

Our goal is to provide real-time anomaly detection from multiple data streams using an analytics engine running on the edge nodes onboard fishing vessels. For this, we propose an efficient and highly modular pipeline system consisting of

three steps: (1) feature extraction, (2) an optional embedding layer, and (3) a moving average-based anomaly detection method. The pipeline allows for analysis of multidimensional input, such as multicamera-based datasets, or combinations of anomaly detection methods. We emphasize that our design is modular and can be configured with multiple inputs, feature extraction methods, and embedding processes, depending on the target context. An example configuration of our use-case can be seen in Figure 1, with multiple inputs, different pretrained feature extraction approaches, with the result being combined at the end. The combination of the anomaly detection outputs can also be performed in numerous ways, for example by using simple boolean OR/AND operations, or more clever weighting strategies. Below, we describe the specific components used for our evaluation.

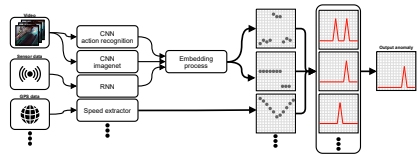


Fig. 1. Example of pipeline configuration. Notice how multiple, multimodal data streams can be processed in different ways and run through the anomaly detection algorithm, before being combined. The embedding process is also dependent on the feature extractor, and thus is not always necessary.

A. Feature Extraction

The feature extraction step of our pipeline can be based on a general machine learning model like a pretrained neural network. Such a model does not have to be trained on the use-case. For example, in this work we use an action recognition network trained on tasks not relevant for anomaly detection on a fishing trawler. We chose this model due to the fact that it is not well known which actions happen on a fishing boat and a more general action recognition dataset might be the best to generalize to our case. The specific predictions do not matter in isolation, only the change in predictions over time. To perform action recognition, we utilize an 18-layer R(2+1)D network, introduced by [15]. The network was pretrained on the Kinetics-400 dataset [16], which consists of 400 action classes. For the used architecture, the pooling layer outputs a 512-dimensional feature vector that is fed to the fully-connected layer with softmax. The output from this network functions as feature extraction, where the predicted class of a subsequence of the video, is a data point in the transformed time-series.

An important thing to note is that we chose this particular pretrained network due to our Dutkat use-case where we are attempting to identify unexpected and potentially illicit actions of the fishermen. However, due to the lack of data of such actions, we are particularly interested in changes of actions which the pretrained network predicts. These actions will often not be correct, but changes between different actions will most likely reflect a change in the actual actions as well.

B. Embedding of labels

The output vector from the pretrained neural network is ordered alphabetically. That is, the indices of the output vector are ordered by the alphabetical order of the corresponding labels. Thus, values that are close together in the output vector will not represent any realistic ordinal relationship. In our experiments we observed that this can cause the anomaly detection method to misclassify certain data points due to the output value changing dramatically, while the underlying labels are semantically similar. For example, the output value might change from 112 to 353, which corresponds to indices that are sorted alphabetically, but the underlying labels corresponding to these values could be, for instance, “eating chips” and “tasting food”, which are semantically close. To address this problem, we applied an embedding on the labels to investigate if it can lead to better results. This embedding would place semantically similar labels close together and thus should lead to better input for the anomaly detection.

The embedding process is shown in Figure 2. The textual labels that correspond to the output vector indices are first embedded via a *sent2vec* model, devised and implemented by [17], that was trained on a Wikipedia corpus. Then, ten high-intensity and low-intensity activities were chosen from the labels to function as “support-vectors”, analogous to how a support vector machine works. A line is then drawn between the means of the low-intensity support-vectors and the high-intensity support-vectors. This line represents an intensity axis. Every point is then projected onto this line, and they are then max-min scaled to values between 0 and 1.

The axis used is based on the assumption that anomalies in videos containing people only arise when a drastic change in intensity occurs. We conjecture that such changes are context dependent and might change depending on the dataset on which the method is to be used.

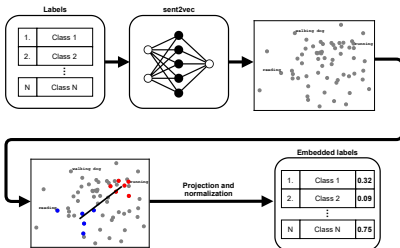


Fig. 2. Illustration of embedding process. Note how the classes are distributed when they are embedded by *sent2vec*, and that in the embeddings are in a different order than the index provided in the beginning.

C. Anomaly detection method

Most anomaly detection methods evaluate if a data point is part of the dataset distribution, and if not, labels it as an anomaly. We, however, are interested in an unexpected transition in the values of a time-series from single or multiple

data streams, where the values are still part of the distribution. These are known as changing points [18].

Since we are interested in changing points, i.e., points in a time-series where the sequence changes for an interval of time, we chose a moving average approach. This approach compares the median of the previous n points to the current point in a time-series. If the current point is outside of the historical interquartile range, it is deemed an anomaly.

IV. DATASETS

The system was applied on three different labeled datasets for evaluation, and an additional unlabeled dataset for testing. The datasets depict varied situations and actions, with different video durations and potential artifacts. The datasets also consist of very different camera positions and movements, with static, moving, and multi-positional cameras depending on the dataset. The action recognition model was pretrained on the Kinetics-400 dataset [16].



Fig. 3. Frames from different videos of the Falling dataset [19]. Notice the different angles and objects in the scene. (The images are from publicly available videos.)

The *Multiple cameras fall* (Falling) dataset by [19] consists of 24 videos with eight cameras filming the scenarios. Of these, 23 depict a person performing several activities, before falling onto a mattress or chair. The remaining video does not contain a fall. The videos have a resolution of 720×480 and a framerate of 30 frames per second (FPS).

The Falling dataset is different from the other datasets that are explored in this paper, because it is based on eight cameras capturing each fall from different angles. The different angles can be seen in Figure 3. This allows us to detect anomalies that might otherwise be obscured by the orientation of the person in the video and test the system for its multiple streams capabilities.

The *Real-world Anomaly Detection in Surveillance Videos* (AV) dataset, created by [20], consists of 1,900 videos containing for example actions like abuse, arrests, arson, assaults, and accidents. The videos are from static surveillance cameras and contain varying backgrounds and numbers of people. The videos have a resolution of 320×240 and a framerate of 25 FPS. Despite the videos having the same resolution they might have different letterboxing, as can be seen in Figure 4.

We will only utilize the videos that have corresponding temporal annotations, as this is needed for the evaluation. This reduces the dataset size to 304 videos. Moreover, the majority of these videos are the “Normal” videos (i.e., they do not contain any anomalies), so these are also ignored, which reduces the dataset further to 154 videos on which we can evaluate our system.

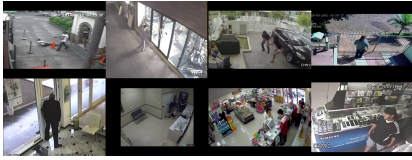


Fig. 4. Frames from different classes within the AV dataset [20], e.g., image three is depicting a robbery and image six is depicting an arrest. (The images are from publicly available videos.)



Fig. 5. Frames illustrating the variety in the Soccernet-v2 dataset [21], with close-ups, wide-angle shots, and the camera following a specific player/referee. (The images are from publicly available videos.)

The Soccernet-v2 dataset ([21], [22]) consists of 500 broadcast soccer games with 300,000 temporal annotations including action and camera labels. We randomly chose a subset of all the games from 2015-2016, which totals 45 games. Each video lasts for approximately 45 minutes, which is much longer than the videos of the two previous datasets discussed above. The videos have a resolution of 398×224 and a framerate of 25 FPS.

Two groups of events from the videos are labeled; the camera movements and actions. The camera movements consist of events like close-ups on players/referee or behind the goal, and when switching between different cameras. The action labels consist of soccer events like goals, free-kicks, and fouls.

A major difference with this dataset compared to the two above is that the camera often is far away from the players to capture the coordinated flow of the game involving multiple players and their opponents. Therefore, detecting actions can be relatively easy in certain cases, such as in the fourth picture in Figure 5, but more difficult in other cases.



Fig. 6. Frames from the FT dataset. Since the videos are from a live-stream, several unnatural artifacts and subsequences are included. Notice the eighth image containing an overlay explaining where the boat is at the time, but it obscures the background which we are interested in. (The images are from publicly available videos.)

The fishing trawler dataset consists of a collection of 30 videos of approximately one hour each. They are a series of live-streams from 2019 filmed aboard the Hermes¹ fishing trawler during operation in the Arctic Sea. The videos contain

ten different camera positions, including a camera on front of the trawler, another angled on the deck, and a camera on the factory level of the ship. These are alternated between at semi-regular intervals. The videos have a resolution of $1,280 \times 720$ and a framerate of 25 FPS.

The FT dataset contains many artifacts since it is from a live-stream, This can be, for instance, as overlay map explaining where they are or information about what they are catching, or even external sequences which are not part of the video stream captured on board. This can be seen in Figure 6.

V. EVALUATION

Evaluating an unsupervised anomaly detection system is difficult. Comparing against labeled datasets can be potentially misleading, due to temporal labels only reflecting when a human detects a change in the time-series. However, the change in the flow of the time-series might have occurred earlier. Therefore, it is important to consider the interval around a temporal label that decides whether a prediction is classified correctly or not. In addition an unsupervised system might detect anomalies that are not detected by the human or are not normal from a data perspective but from a human perspective depending on the use case nothing special (e.g., a bird flying trough the scene).

A. Experimental Setup

We implemented the system in Python using PyTorch ([23]) and the Anomaly Detection Toolkit (ADTK)². The pretrained neural network model was imported from the torchvision module. The frames are extracted from the videos and are resized and combined into a tensor in the batch generator. The feature extraction was performed on each dataset, and the results were saved in comma-separated files. Then, the anomaly detection was applied to the extracted features. Each data point corresponds with an 8-frame clip, which is one of the accepted inputs to the action recognition network used. Due to the specific use case the system is designed for and the requirements coming with it, it would not provide any insights if we compare the system with other existing anomaly detection methods. But to compensate for this we tested the system on three datasets not coming from the fishing domain to show that it is able to detect anomalies at a acceptable level.

All datasets except the Falling dataset were analyzed using a simple linear pipeline which consists of the input, a pre-trained action recognition network, an embedding layer, and the anomaly detection algorithm. The Falling dataset contains multiple data streams capturing the same event, therefore the configuration is slightly different. Multiple data sources, corresponding with the different camera angles, are fed into the same pretrained network and embedding layer, before being the processed data is run through the anomaly detection algorithm separately. These detection streams are then combined naively using a simple boolean-OR combination, i.e., at the current timestep, if an anomaly is detected in any of the streams, it is regarded as an anomaly in the combined output.

¹<https://www.hermesas.no>

²<https://arundo-adtk.readthedocs-hosted.com/en/stable/>

The evaluation was prohibitively expensive on the Socccernet-v2 dataset due to the length of the videos, and required optimization. The evaluation of each predicted point, which consists of a comparison against temporal annotations, was divided across multiple nodes. This allowed us to evaluate detected anomalies in a parallelized fashion.

All the experiments were run on a Linux machine with Ubuntu 20.04 LTS distribution having a 3.70GHz Intel Xeon CPU E5-1620, 64GB DDR3 RAM, and an OCZ-VERTEX 4 512GB SSD. The feature extraction was run on an NVIDIA RTX 2080Ti GPU.

B. Evaluation Methodology

For evaluation purposes, we compare our system to multiple baseline classifiers, namely a uniform random baseline and two constant baselines (i.e., classifying every data point as an anomaly or non-anomaly). We compare accuracy, recall, precision, F1-score, and Matthew’s correlation coefficient (MCC)³. The metrics are calculated as macro-averages over the two classes (anomaly/non-anomaly). Due to the imbalanced nature of the datasets, the accuracy will generally be quite high and misleading, but we include it for completeness. For the AV dataset, we further evaluate the performance per class, to determine whether our approach is better at detecting specific types of anomalies. Finally, we evaluate how the interval range of whether a point is deemed as a true positive or a false positive affects the results.

Furthermore, we benchmark our system in terms of processing performance to evaluate if it can be applied in a real-time scenario, i.e., processing the videos at the speed of the framerate. However, the ADTK framework is not designed for real-time analysis, i.e., it expects a complete Pandas dataframe before performing any analysis. Thus, we have chosen to first generate a dataframe that is the length of the comparison window of the anomaly detection method, then actively append new data points and remove the oldest data points to maintain a fixed size. The average framerate is calculated based on ten random videos within each dataset.

Finally, we manually inspect and analyze the results of applying our system on the FT dataset, to gain some insight as to how it handles this specific scenario. We particularly focus on what type of events are classified as anomalies, and also carefully investigate what the system misses.

C. Results and Discussion

In this section and the respective subsections we present the results and discussions around them. We start with a more general perspective followed by deeper analysis of different interesting aspects of our evaluation.

1) *Compared to baseline: Falling Dataset.* As documented in Table I, our system achieves the best results on the Falling dataset. We conjecture this is due to multiple reasons; first, this dataset consists of multiple cameras filming the scene from different angles, thus erroneous changes in predicted actions

³Note: MCC cannot be calculated for the constant baselines due to division by zero.

Falling Dataset					
Method	Recall	Precision	F1-score	Accuracy	MCC
Uniform	0.49	0.49	0.46	0.50	0.009
Constant 1	0.50	0.38	0.43	0.23	N/A
Constant 0	0.50	0.12	0.19	0.77	N/A
Ours	0.76	0.82	0.79	0.86	0.57
AV Dataset					
Method	Recall	Precision	F1-score	Accuracy	MCC
Uniform	0.50	0.50	0.34	0.50	-0.003
Constant 1	0.50	0.01	0.01	0.01	N/A
Constant 0	0.50	0.49	0.50	0.99	N/A
Ours	0.58	0.57	0.57	0.99	0.15
Socccernet-v2 Dataset					
Method	Recall	Precision	F1-score	Accuracy	MCC
Uniform	0.50	0.50	0.43	0.50	-0.001
Constant 1	0.50	0.07	0.12	0.14	N/A
Constant 0	0.50	0.43	0.45	0.86	N/A
Ours	0.50	0.82	0.48	0.87	0.10

TABLE I
COMPARISON OF OUR SYSTEM AGAINST BASELINES. EACH TABLE CORRESPONDS TO THE RESULTS FROM A SPECIFIC DATASET. BOLD INDICATES THE BEST PERFORMANCE.

will be filtered out due to the boolean-OR combination approach in the anomaly detection algorithm, which is different from the other datasets which do not require combination of multiple data sources. Secondly, each video only contains a single person throughout the majority of the duration which makes it easier for the algorithm to focus on the specific action.

AV Dataset. The results on the AV dataset are a bit closer to the baselines compared to the Falling dataset, but we still get better results for all metrics which show that the proposed method is able to detect anomalies even in difficult scenarios. The reason for the results being worse than for the Falling dataset is most likely due to the variety of the content and quality of the videos. Some actions are visually easier to detect than others. A more detailed discussion on this specific challenge is provided in subsection V-C2.

Socccernet-v2. Our system was closest to the baseline when applied on the Socccernet-v2 dataset. We assume this is due to the large distance between the camera and the players in the majority of the duration of the videos. The greatest difference is for the precision, which implies that the system is performing well at detecting events/anomalies, while detecting few false positives.

With regard to the MCC, as stated previously, we can only compare the uniform classifier to our system. The MCC is a metric that handles imbalanced data well, taking all elements of the confusion matrix into account. A value of zero would indicate a classifier predicting correctly 50% of the time. A value of one would indicate a perfect classifier. We see that the uniform classifier has an MCC of approximately zero. The largest difference in the MCC is for the Falling dataset, with 0.57 for our system. Though MCC might not be a reliable indicator of how good a classifier is [24], it clearly correlates to a certain degree and gives a better understanding of the performance when the datasets are biased. For all three datasets, the MCC was above random predictions for our system.

From this first analysis, we can see that the system is per-

forming well on the task of unsupervised anomaly detection. Specifically, we can see that multiple data streams appear to lead to better results (Falling vs the other two). It is apparent that the type of classifier that produces the input for the anomaly detector is important. The basic action recognition model used was best suited for the falling detection whereas the event in the other two datasets where not specifically part of the actions. Using more specific models most probably would lead to better results, for example, a soccer or crime event-specific model for the respective dataset (or even combination of models capturing different aspects of the stream).

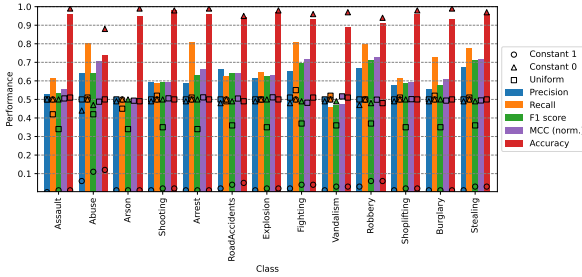


Fig. 7. Per class performance on the AV dataset. The circle, triangle and square markers indicate the results of the baseline classifiers for each metric.

2) Deeper analysis on AV and Soccermet-v2 datasets:

We assumed that certain classes in the AV dataset would be difficult to detect, such as the shoplifting class. Shoplifters try to be inconspicuous while performing their illicit activity, and as such, the detected actions might not change very much. Results documented in Figure 7 demonstrate that our system performs closest to the baseline on the Arson and Vandalism classes of videos. We can also observe a difference between the Constant 1 and Constant 0 baseline showing that Constant 1 is performing worse. This indicates that the dataset is containing a lot of normal frames and not so many positive frames which makes it again a difficult dataset. Most classifiers would easy start focusing on the normal frames and basically become a Constant 0 classifier. Our method avoided this and was able to detect some of the anomalies correct which is shown in the MCC with 0.15. Visually inspecting videos in these classes, we saw that, for the Arson videos, the actions of the arsonist do not change significantly over the course of the videos except for when they flee the scene. The system usually detects when the fire has reached a certain size, but not when it is ignited in the beginning. With the Vandalism videos, the system’s performance highly depends on the type of vandalism and the distance from the camera to the perpetrator. Videos of tagging or videos where the camera is far away are more difficult for the system.

Regarding the Soccermet-v2 dataset, there are a total of 14,969 camera labels and 10,008 action labels. Due to the distance between the camera and the players, our hypothesis was that it might be difficult for our system to detect the

Soccermet-v2: Camera vs Actions					
Labels	Recall	Precision	F1-score	Accuracy	MCC
Camera labels	0.51	0.78	0.49	0.90	0.11
Action labels	0.51	0.67	0.51	0.95	0.08

TABLE II
PERFORMANCE OF OUR SYSTEM ON THE SOCCERMET-V2 DATASET USING THE CAMERA LABELS VS. THE ACTION LABELS.

actions described by the action labels. Looking at Table II, we can see the the recall is the same for both sets of labels, which implies that the system that the same relative number of relevant anomalies/events are detected. However, the precision is quite different, with our system achieving better precision using the camera labels. This implies that fewer false positives are generated. This might align with our hypothesis, because the camera events are easier to detect.

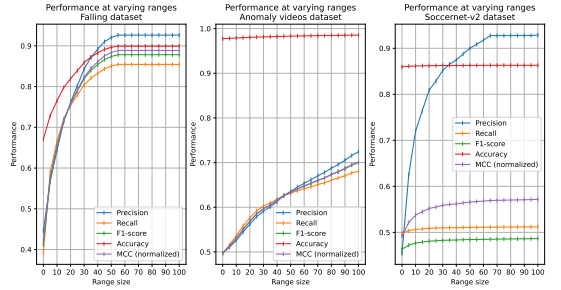


Fig. 8. How the interval range around a temporal label affects the results of our system. Due to a data point representing 8-frame clips, an interval range of 100 ($x - 100, x + 100$) corresponds with ± 30 seconds around a temporal label for a 25 FPS video.

3) *Interval range for evaluation:* In Figure 8, we are evaluating how the interval range around a true label affects the performance on the different datasets. As we remarked in the beginning of this section, evaluating time-series detection methods is difficult. Using only the true labels timestamp as a correct classification would be equal to results when range size is set to zero. This would not be a realistic goal to optimize. As long as the system manages to detect an event/anomaly within a certain interval, this should be classified as a true positive. The range size in Figure 8 ranges from 0 to 100 data points in either direction around a true label. 100 data points correspond to approximately 32 seconds, so that would allow for any anomaly detected within a minute around a true label to be classified as a true positive.

In the Falling dataset, we can see that the results plateau at a range size of about 55, which corresponds with 20 seconds. As the longest video in this dataset is 44.6 seconds (average 11.1 seconds), it makes sense that the results plateau around this value, because at this point every predicted anomaly will correspond with one of the labels. We see a similar plateauing in the soccermet-v2 dataset, however the duration of these videos are always approximately 45 minutes. We assume this

might instead be due to the camera movement, and a player rarely being in focus for a long time.

For both the AV and the soccernet-v2 dataset, the accuracy is consistently high regardless of range size. This is probably due to the imbalance of anomalies/events versus non-events being relatively greater than for the Falling dataset. This is because the videos in the Falling datasets are usually very short, thus the section of the videos containing the fall makes up a significant portion of the video.

FPS Benchmark

Dataset	#Input streams (res.)	Preprocessing	Avg. Framerate
Falling	8 (720 × 480)	Act. Recog.	108.2
AV	1 (320 × 240)	Act. Recog.	219.3
Soccernet-v2	1 (398 × 224)	Act. Recog.	200.9
FT	1 (1280 × 720)	Act. Recog.	157.6

TABLE III

RESULTS FROM FPS BENCHMARK. AS WE CAN SEE, THE SYSTEM CAN BE APPLIED ON REAL-TIME LIVE-STREAMED DATA.

4) *Real-time benchmark*: Based on the results in Table III, we observe that the system can be utilized when analyzing live data streams. Due to differences in FPS and resolution between the different datasets, the preprocessing causes the results to differ. Due to the Falling dataset consisting of multiple data streams per video, this obviously causes the framerate to be lower. The data streams are processed sequentially through the pretrained network and embedding layer, but the anomaly detection algorithm is applied in parallel. The results on the FT dataset are also quite lower than the AV and Soccernet-v2 datasets. This is most likely due to the resolution of the videos being higher and the duration being longer. However, despite the different results, we conclude that the system can be applied in a real-time context and deployment.

5) *False Positives*: It is important to note that events that are classified as false positives are not necessarily wrong. The system might detect events that were not deemed important by the annotators of their dataset. Nevertheless, these points might still be interesting, as they might indicate event such as the change in flow of a soccer match, or new people entering the frame in a surveillance video. With unsupervised approaches it is therefore impossible to be completely sure what is a false positive versus a true positive, but for the evaluations in the previous sections we have assumed the temporal annotations to be correct. However, based on our system’s precision in Table I, we can see that, generally, the system does not introduce too many false positives.

6) *FT dataset*: Finally, we look at the manual inspection of the results on the FT dataset trying to detect anomalies on the fishing boat. We have applied the system on the entire dataset, and manually gone through and evaluated the predicted anomalies. The majority of events that are detected are changes between different cameras/scenes. After that, human activity is the next largest group, which contains actions such as “talking”, “talking on the phone”, “Leaning forward”, “working”, etc. Of the ten different camera positions, the angles within the bridge are the ones that depict most scenes with people on-screen. Then, we have lighting changes, which

Anomalies detected in FT dataset

Event Type	Percentage
Scene Change	34.4
Human Activity	25.6
Lighting Change	7.6
Overlay	3.9
High Seas	5.7
Camera	2.8
Other	22.0

TABLE IV

MANUAL INSPECTION RESULTS FROM FT DATASET.

are either due to electric lights being turned on/off, or the sun lighting up different areas of the vessel due to rocking waves. The “High Seas” tag is also dependent on large waves rocking the vessel, which for example causes the horizon to cover more/less of the screen, or the sun to move in and out of frame. As such, there may be some overlap between these two classes. There are two types of overlays that are added over the videos; small notes containing factoids related to fishing and the sea, and larger overlays that cover most of the screen and show a map of where the vessel is at the moment. Finally, camera movement refers to a specific angle where the camera can be manually controlled by the skipper to show interesting events such as whale sightings or other vessels. Regarding the predicted events under “Other”, these include all events that are relatively rare, like movement of cranes/heavy machinery, whale spottings, compression artifacts, and “nothing”, which are events where we do not observe any obvious change and constitute 14.3% of the detected events. Some of these anomalies can be seen in Figure 9.



Fig. 9. Example frames of anomalies found in the dataset, containing (from left to right) “talking on the phone”, “high seas”, “whale spotting”, and “working”. (The images are from publicly available videos.)

17.2% of the scene changes were not detected. False negatives other than scene changes are difficult to evaluate, but certain events that are detected previously, like overlays, but not later are relatively easy to spot during manual inspection (2.1% of these are missed).

Overall, we can observe that the proposed system has capabilities to detect interesting and relevant anomalies. This makes it a possible alternative to check all content or not checking at all for the specific use-case. In addition, we only tested it with one data stream and a general action recognition model. If we combine different streams from trawlers and build more specific models, the performance will increase most certainly. Furthermore, it can be used to build datasets since it can be used to determine which possible events can occur, label them and make them usable for building supervised models that again can be used in the system.

VI. CONCLUSION

In this paper, we presented an unsupervised machine learning approach for anomaly event detection of multimodal data streams. The system is highly modular and can combine the results from multiple data streams, using different pretrained feature extraction methods that are not necessarily trained for the use-case. We apply our system on three labeled datasets, and one unreleased, unlabeled dataset. The system outperformed the baselines on all labeled datasets examined in this paper, however on the Soccernet-v2 dataset, which consists of far-away video shots of multiple players, performed the worst and was slightly above baseline.

In the future, we would like to try a more complex configuration of the system pipeline. The system pipeline can be expanded to utilize more pretrained networks for feature extraction, e.g., using a network trained on the Imagenet dataset [25] could make it easier to detect road accidents. It would also be interesting to use a detection network to detect and crop around individuals within the videos to see whether that could result in better performance, particularly on the videos where the camera is filming multiple people from a distance.

We aim to apply our system when annotating the FT dataset, and we will release this dataset in the near future. We will also apply our system when we deploy the Dutkat system on fishing vessels in the near future. In the actual use-case, we will utilize surveillance video data, sensor data from scales and other tools, etc. to try to incorporate as many sources as possible for the anomaly detection. Based on our preliminary analysis of the problem of detecting criminal/anomalous activity on fishing vessels based on video data, we have concluded that acquiring data of such specific events is infeasible. Therefore, we will in the future use multiple different data streams and perform anomaly detection on these combined, using the system described in this paper. These data streams would potentially consist of surveillance video data, sensor data estimating catch biomass, AIS (automatic identification system) data, etc. The resulting events and anomalies we detect from this data will be used to incrementally build labelled datasets that can be used in the system to improving performance and detect other anomalies.

ACKNOWLEDGMENTS

We particularly would like to thank Signor Antonsen of Hermes AS, Kim H. Andreassen, Jon F. Mikalsen, and Kjetil Robertsen.

REFERENCES

- [1] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Asian conference on computer vision*. Springer, 2018, pp. 622–637.
- [2] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, "Cutpaste: Self-supervised learning for anomaly detection and localization," in *IEEE/CVF CVPR*, 2021, pp. 9664–9674.
- [3] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLoS one*, vol. 11, no. 4, p. e0152173, 2016.
- [4] T.-A. S. Nordmo, A. B. Ovesen, H. D. Johansen, M. A. Riegler, P. Halvorsen, and D. Johansen, "Dutkat: A multimedia system for catching illegal catchers in a privacy-preserving manner," in *Proceedings of the 2021 Workshop on Intelligent Cross-Data Analysis and Retrieval*, ser. ICDAR '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 57–61.
- [5] United Nations Office on Drugs and Crime, "Fisheries crime," 2016.
- [6] A. B. Ovesen, T.-A. S. Nordmo, H. D. Johansen, M. A. Riegler, P. Halvorsen, and D. Johansen, "File system support for privacy-preserving analysis and forensics in low-bandwidth edge environments," *Information*, vol. 12, no. 10, 2021.
- [7] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.
- [8] B. R. Kiran, D. M. Thomas, and R. Parakkal, "An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos," *Journal of Imaging*, vol. 4, no. 2, p. 36, 2018.
- [9] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft, "Deep semi-supervised anomaly detection," *arXiv preprint arXiv:1906.02694*, 2019.
- [10] K. Doshi and Y. Yilmaz, "Fast unsupervised anomaly detection in traffic videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.
- [11] M.-C. Chang, C.-K. Chiang, C.-M. Tsai, Y.-K. Chang, H.-L. Chiang, Y.-A. Wang, S.-Y. Chang, Y.-L. Li, M.-S. Tsai, and H.-Y. Tseng, "Ai city challenge 2020-computer vision for smart transportation applications," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 620–621.
- [12] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini, "Anomaly detection using autoencoders in high performance computing systems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 9428–9433.
- [13] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017.
- [14] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *IEEE CVPR*, 2018, pp. 3379–3388.
- [15] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.
- [16] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," 2017.
- [17] M. Pagliardini, P. Gupta, and M. Jaggi, "Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features," in *NAACL 2018*, 2018.
- [18] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *CoRR*, vol. abs/2002.04236, 2020.
- [19] E. Auvinet, C. Rougier, J. Meunier, A. St-arnaud, and J. Rousseau, "Multiple cameras fall dataset," 2010. [Online]. Available: <http://www.iro.umontreal.ca/~labimage/Dataset/>
- [20] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," pp. 6479–6488, 2018.
- [21] A. Deliege, A. Cioppa, S. Giancola, M. J. Seikavandi, J. V. Dueholm, K. Nasrollahi, B. Ghanem, T. B. Moeslund, and M. Van Droogenbroeck, "Soccernet-v2," 2021. [Online]. Available: <https://soccer-net.org>
- [22] —, "Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos," in *IEEE/CVF CVPR*, June 2021, pp. 4508–4519.
- [23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.
- [24] D. Chicco, N. Tötsch, and G. Jurman, "The matthews correlation coefficient (mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation," *BioData mining*, vol. 14, no. 1, pp. 1–22, 2021.
- [25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

A.5 Paper IV: Njord: A Fishing Trawler Dataset

Authors: T.S. Nordmo, A.B. Ovesen, B.A. Juliussen, S.A. Hicks, V. Thambawita, H.D. Johansen, P. Halvorsen, M.A. Riegler and D. Johansen

Abstract: Fish is one of the main sources of food worldwide. The commercial fishing industry has a lot of different aspects to consider, ranging from sustainability to reporting. The complexity of the domain also attracts a lot of research from different fields like marine biology, fishery sciences, cybernetics, and computer science. In computer science, detection of fishing vessels via for example remote sensing and classification of fish from images or videos using machine learning or other analysis methods attracts growing attention. Surprisingly, little work has been done that considers what is happening on board the fishing vessels. On the deck of the boats, a lot of data and important information are generated with potential applications, such as automatic detection of accidents or automatic reporting of fish caught. This paper presents Njord, a fishing trawler dataset consisting of surveillance videos from a modern off-shore fishing trawler at sea. The main goal of this dataset is to show the potential and possibilities that analysis of such data can provide. In addition to the data, we provide a baseline analysis and discuss several possible research questions this dataset could help answer.

Author contributions (initials): **Conceptualisation:** T-A.S.N., D.J., M.A.R.;

Data collection: T-A.S.N., A.B.O., B.A.J., S.A.H., V.T., **Methods, data**

analysis and interpretation: T-A.S.N., S.A.H., V.T., **Drafting:** T-A.S.N.,

S.A.H., V.T., A.B.O., H.D.J, P.H., M.A.R., D.J., **Critical revision:** T-A.S.N.,

S.A.H., V.T., A.B.O., H.D.J, P.H., M.A.R., D.J.

Published: ACM Multimedia Systems (MMSys) 2022

Thesis objectives: Sub-objective 3

Njord: A Fishing Trawler Dataset

Tor-Arne Schmidt Nordmo
UiT The Arctic University of Norway

Aril Bernhard Ovesen
UiT The Arctic University of Norway

Bjørn Aslak Juliussen
UiT The Arctic University of Norway

Steven Alexander Hicks
SimulaMet, Norway

Vajira Thambawita
SimulaMet, Norway

Håvard Dagenborg Johansen
UiT The Arctic University of Norway

Pål Halvorsen*
SimulaMet, Norway

Michael Alexander Riegler†
SimulaMet, Norway

Dag Johansen
UiT The Arctic University of Norway

Abstract

Fish is one of the main sources of food worldwide. The commercial fishing industry has a lot of different aspects to consider, ranging from sustainability to reporting. The complexity of the domain also attracts a lot of research from different fields like marine biology, fishery sciences, cybernetics, and computer science. In computer science, detection of fishing vessels via for example remote sensing and classification of fish from images or videos using machine learning or other analysis methods attracts growing attention. Surprisingly, little work has been done that considers what is happening on board the fishing vessels. On the deck of the boats, a lot of data and important information are generated with potential applications, such as automatic detection of accidents or automatic reporting of fish caught. This paper presents Njord, a fishing trawler dataset consisting of surveillance videos from a modern off-shore fishing trawler at sea. The main goal of this dataset is to show the potential and possibilities that analysis of such data can provide. In addition to the data, we provide a baseline analysis and discuss several possible research questions this dataset could help answer.

CCS Concepts

• **Computing methodologies** → **Machine learning**; *Cross-validation*; *Supervised learning*; • **Applied computing** → **Law**.

Keywords

Fishing Trawler, Surveillance, Slow TV, Machine Learning, Artificial Intelligence, Dataset

ACM Reference Format:

Tor-Arne Schmidt Nordmo, Aril Bernhard Ovesen, Bjørn Aslak Juliussen, Steven Alexander Hicks, Vajira Thambawita, Håvard Dagenborg Johansen, Pål Halvorsen, Michael Alexander Riegler, and Dag Johansen. 2022. Njord: A Fishing Trawler Dataset. In *13th ACM Multimedia Systems Conference (MMSys '22)*, June 14–17, 2022, Athlone, Ireland. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3524273.3532886>

*Also affiliated with Oslo Metropolitan University, Norway

†Also affiliated with UiT The Arctic University of Norway

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

MMSys '22, June 14–17, 2022, Athlone, Ireland

© 2022 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9283-9/22/06.

<https://doi.org/10.1145/3524273.3532886>



Figure 1: Example of concurrent videos stream observations on the command bridge of the Hermes trawler.

1 Introduction

A modern fishing vessel is infused with high-tech digital technologies. The bridge of a trawler operating in, for instance, the Arctic contains numerous terminals visualizing geographical position and other vessels in the vicinity, weather conditions and predictions, fish finder sonar data and the like. Video streams from the deck and production line under deck are also frequently displayed so that the officer in-charge has real-time information when making operational decisions. Figure 1 is an example of different video streams observed simultaneously on a fishing vessel. The video stream is used, for instance, in a safety context for the crew members alone on deck or working somewhere along the heavy machinery constituting a production line. Accidents in this industry are not an exception and an important problem to consider [6, 12].

The constant collection of voluminous, multimodal data on modern commercial fishing vessels leads to interesting possibilities for the application of advanced analysis methods. For example, using Artificial Intelligence (AI) to analyze this data could lead to new insights supporting more energy-efficient locations of fish to catch,

sustainable catching, and a safer working environment for the fishermen. Add to this the potential such technologies can have from a resource control and global management perspective.

AI-relevant technologies are already being applied in this domain. One example is support for sustainable fishing operations [5, 7, 9], another is publishing of fish datasets relevant for developing new models in this specific domain [4, 8, 15]. Fishing vessel and boat detection is yet an example where AI technologies can replace tedious and labor-intensive manual operations [10, 13, 14]. The list is longer, but what is missing is labeled datasets from the internal activities on board a commercial fishing vessel. The work presented in this paper is a first contribution to fill this void.

We present an open and novel dataset called Njord, which was collected from cameras on a high-end commercial fishing vessel operating in the Arctic Ocean and annotated with bounding box and classification annotations. The current dataset contains 71 annotated videos and 127 videos that are not annotated from live-streams that aired in 2019. We envision that the presented dataset can lead to a myriad of new research and a better understanding of a completely unexplored but important area. The dataset is also meant to be updated over time with more videos and additional data sources.

Therefore, the main contributions of this paper are:

- (1) We compile and publish a unique, fully open dataset containing surveillance video data based on live-streams from a fishing trawler. A large part of the dataset is thoroughly annotated with both bounding box and classification labels.
- (2) We provide domain knowledge about the specific use case including a discussion of legal aspects and current open challenges.
- (3) We provide a set of baseline machine learning experiments to benchmark the released dataset and evaluate its technical validity.
- (4) We discuss and suggest possible future research directions and application scenarios using the dataset.

The remainder of this paper is organized as follows. In Section 2, we describe how we have structured the dataset. Then, in Section 3, we describe the details of what the dataset contains and how it was collected and annotated. This is followed by Section 4, which gives an overview of legal aspects surrounding the prospect of surveillance in a fishing trawler scenario and datasets in general. After this, we describe potential applications and usage scenarios for the dataset in Section 5. In Section 6, we then describe suggested metrics that are relevant for the dataset and perform some baseline experiments. Finally, we conclude and describe some future work in Section 7.

2 Dataset Structure

The dataset is organized as follows. The root directory contains a *readme.txt* file and a *videos* directory. The *readme.txt* file gives a brief description of the included data and annotations. The *videos* directory contains a subdirectory for each annotated video that contains the video in *.mp4* format and two annotation files, one file for the bounding box annotations and one file for the timeline annotations. The two annotation files are structured as *.csv* files using a semi-colon as the delimiter. The bounding box contains one line per bounding box annotation with the following seven

values; class, frame number, center x position, center y position, the bounding box's width, and the bounding box's height. The width and height have been normalized by dividing each by the video's width and height, respectively. The timeline annotation file contains one line per annotated class and includes the following two values; the class of the frame and the frame number of the corresponding video. The *videos* directory also contains a *unannotated* subdirectory containing all videos that have not been annotated yet.

3 Dataset Details

As previously described, the Njord dataset contains surveillance videos from the *Hermes*¹ fishing trawler that were live-streamed online in 2019 as "Slow TV" [1] entertainment. The videos are from a trip from the western shores of Greenland to Norway, documenting their fishing journey. There are a total of 29 live-stream videos that are, on average, 1 hour in duration. These were downloaded from YouTube using the *youtube-dl* CLI tool. They were then split up into 10-minute segments to be easier to deal with both in the labeling and the benchmarking process. At the time of submission, we have annotated a subset of these. This results in a dataset with 71 videos that have been annotated so far and 127 videos that are not annotated. 71 annotated videos, each with a frame rate of 25 fps and a duration of approximately 10-minutes, results in approximately 1,065,000 frames with annotations. The videos have a resolution of $1,280 \times 720$ and run at 25 fps. The videos have varying lighting conditions with complex, moving backgrounds due to the trawler being at sea. The videos consist of eight different fixed-camera scenes plus a view with a manually-operated camera for showing particularly interesting events, such as whale observations and other boats. The cameras are changed between on a fixed schedule but can also be manually changed by the captain. This sometimes results in scenes having varying durations. There are overlays that sometimes appear on-screen. These show general information about what is being caught, information about the vessel in general, and statistics related to the catch. They also sometimes show a map overlay with the current location of the trawler along with its speed and orientation of it.

For each video, we have labeled bounding boxes around people, other boats, nets, and fish. The temporal annotations consist of when scene changes occur, when overlays are turned on and off, when Events of Interest (EoI) occur, and when the intro plays. We also have labels that denote whether it is daytime or nighttime, and, due to the videos being from a live-stream, labels for parts of the videos that are before the introduction and after the end of the relevant live-stream. The bounding boxes for fish label groups of fish due to the scenes on deck showing fish being far away from the camera. The bounding boxes for the nets both label nets in use and those lying in heaps on deck.

The labels were manually created using Labelbox [3]. Labelbox is a platform for annotating datasets. It has a simple interface that allowed us to label bounding boxes and temporal annotations. For the bounding boxes, it linearly interpolates between keyframes, allowing for a faster annotation operation.

The dataset is anticipated as continuously growing and expanding (in terms of annotations, but also amount of data), and currently,

¹<https://www.hermesas.no>

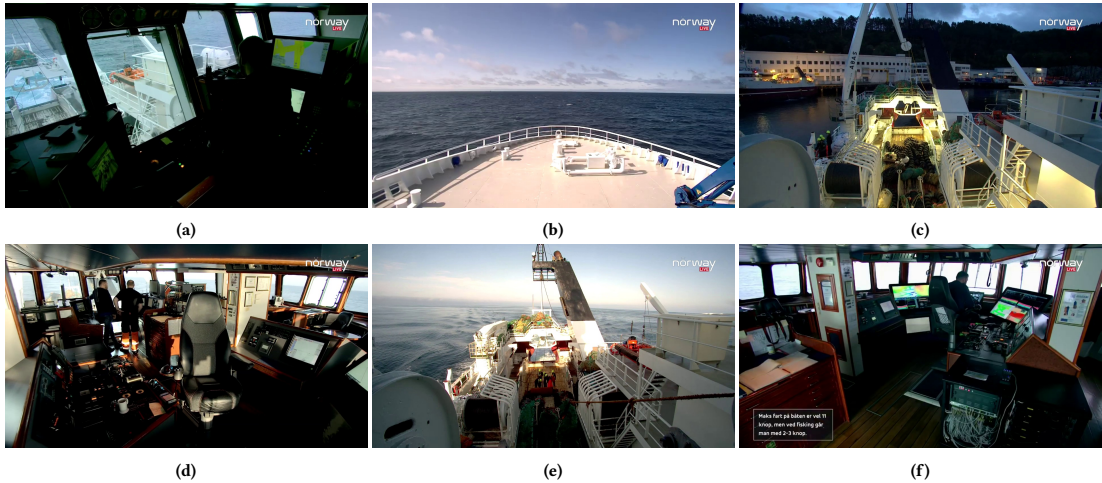


Figure 2: Sample frames from different videos of the dataset. (a) - A view from the bridge looking down at the deck, (b) - A view from the front of the vessel, (c) - A view of the deck as the trawler moves from port, (d) - A view of the bridge, (e) - A view of workers on deck, and (f) - A view of the bridge from another angle with an overlay.

it contains 71 fully annotated videos and 127 videos without annotations. The not annotated video can also be useful for unsupervised or self-supervised learning experiments.

Njord is licensed under Creative Commons Attribution Non-Commercial 4.0 International (CC BY-NC 4.0), and is available for download at <https://doi.org/10.5281/zenodo.6284673>.

4 Legal aspects

A fishing vessel is a secluded environment where people often work and live for several weeks at a time. Introducing video surveillance and video surveillance combined with machine learning in such an environment has privacy and data protection aspects. Besides fundamental privacy rights, the use of surveillance cameras on board vessels needs to comply with European Data Protection Regulations and emerging European AI regulations. Article 4 (1) of the General Data Protection Regulation (GDPR) defines personal data as “any information relating to an identified or identifiable natural person”. A picture of a natural person in a surveillance video stream could identify the person and would fall under the definition of personal data in the GDPR. The GDPR requires the processor of personal data to have a valid legal basis (consent, performance of a contract, a vital interest of the data subject, a legal obligation or a public interest) for the processing to be lawful. The lawfulness of the processing of surveillance video data on board a fishing vessel depends on the purpose of the processing. Processing to prevent accidents would likely need to rely on another legal basis than processing to prevent and deter illegal fishing. The legal basis for the processing would also depend on whether, for instance, a fishing company or a public control authority is the processor under Article 4 (8) GDPR.

A proposal for an *Artificial Intelligence Act (AIA)*² is currently negotiated in the European Parliament. If surveillance video data is combined with machine learning to detect anomalies in the video stream, the system would be included in the definition of an AI system in Article 3 (1) of the act. The proposed act classifies AI systems after their purpose, where systems that pose a risk of adverse impact on fundamental rights are subject to stricter requirements. An AI system intended to be used by law enforcement authorities for risk assessments or crime analytics is defined as high-risk AI systems under the AIA Article 6 (2). If surveillance video data from fishing vessels is combined with machine learning to report and prevent infringements of fishing regulations, the system might be included in the definition of a high-risk AI system. A high-risk AI system is required to comply with risk assessing procedures throughout its lifetime. Article 10 of the AIA lays down specific quality criteria for datasets applied in high-risk AI systems. According to Article 10 (3) and (4), training, validation and testing datasets shall be “relevant, representative, free of errors and complete.” Moreover, datasets applied in high-risk systems shall take into account the characteristics or elements that are particular to the specific geographical, behavioral or functional setting within the high-risk AI system is intended to be used.

Both European data protection regulations and the emerging specific AI regulation in the European Union, in essence, are assessments of the proportionality of the interference in natural persons rights balanced against the purpose of the processing of personal data and the purpose of the AI system. The principle of proportionality requires an interference in a fundamental right, such as the

²Regulation of the European Parliament and of The Council Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts COM(2021) 206 Final.

right to privacy or data protection, to pursue a legitimate aim, be necessary and suitable to the aim pursued, and the interference and the fundamental right must pass a balance test. For the interference to be proportionate, the reasons to interfere must outweigh the interference in the fundamental right. The lawfulness of a surveillance system on fishing vessels will therefore depend on the aim pursued by the video surveillance, security for the workers, prevention of accidents, automatic documentation of catches or prevention and deterrence of illegal fishing etc., and the extent to which the privacy and data protection rights of the individuals working on board are affected.

Due to these concerns, it can be interesting to look at the proportion of the videos containing people. Based on the videos currently annotated, 44.8% of the frames contain a person. Depending on the use-case, it can potentially be possible to only utilize frames that do not contain people, however, in other use-cases, such as when analyzing fishing procedure, they are crucial. Therefore, looking into anonymization approaches can be useful.

5 Applications and Usage Scenarios

The purpose of publishing this unique dataset is to motivate the machine-learning research community to explore new aspects of the fishing domain. As a starting point, we foresee this dataset to have several applications and usage scenarios. A few examples are presented in the following:

- General object detection with complex backgrounds and lighting conditions;
- automatic documentation of the fishing procedure;
- surveillance of persons and their activities;
- privacy research; and
- detection of Events-of-Interest (EoIs).

Due to the complexity of the backgrounds and the varying lighting conditions, this dataset can be used as a difficult benchmark for object detection. There are multiple scenes where people and other objects overlap. All of this results in a complex scene where an object detection algorithm can be put to the test.

Automatically documenting the progress of the fishing procedure, i.e., from catching the fish to processing and storing of the catch, can be useful for process efficiency. From the different scenes of the videos, one can see all of the different stages of this pipeline. Learning what to focus on in the different scenes can be useful for optimizing the fish processing procedures. With areas of fish being labeled, detecting catch biomass can also be an interesting endeavour. Provided one manages to get a relatively accurate measurement, this can provide an indicator whether the trawler is fishing within their quotas or not.

Surveillance of the fishing vessel and the crew is important, specifically to ensure that action is taken when accidents occur, for example, if a fisherman falls or a net falls on top of them. Surveillance can also document whether a proper procedure regarding the handling of catch and bycatch (i.e., a catch of species that are not allowed to be caught) is being followed. In Norway, all catch and bycatch need to be brought to shore. In addition, the data can also be used to explore privacy aspects of surveillance data and algorithms for privacy-related research by, for example, using it to learn how to obfuscate faces, etc.

Table 1: Evaluation results of the baseline experiments.

Model	Precision	Recall	mAP_0.5	mAP_0.5:0.95
YOLOv5n	0.698	0.502	0.527	0.265
YOLOv5s	0.732	0.545	0.543	0.271
YOLOv5m	0.697	0.552	0.569	0.277
YOLOv5x	0.621	0.570	0.550	0.264

Considering these videos are from slow-TV live-streams, it might be interesting to detect highlights/events that are the most interesting parts (i.e., EoIs). This could be used to build models that can be used to notify viewers if something interesting happens, etc. Highlight detection has been done previously in sports [11].

Finally, due to the sheer volume of data and labels, applying machine learning pipelines on this dataset is non-trivial. Therefore it can be interesting, from a systems point-of-view, to explore different approaches on how to deal with such voluminous data.

To showcase one possible use case for the data, we perform a set of baseline experiments using the object detection scenario mentioned above in Section 6.

6 Example Use Case Experiments

Together with the development and collection of the presented dataset, we performed a series of experiments meant to create a baseline for future researchers to measure against. The experiments use the bounding box annotations included in the dataset to detect specific interest points in the videos. Specifically, we aim to detect people, fishing nets, fish, and passing boats. As the dataset is made up of two different types of annotations, the appropriate metrics used to measure predictive performance vary based on the task. For detection (bounding box prediction), metrics such as precision, recall, and mean average precision (mAP) are most appropriate. For the timeline annotations, classification metrics such as precision, recall, f1-score, and Matthews correlation coefficient should be used.

Not all videos contain bounding box annotations and, for this experiment, we ended up using 58 of the 71 videos. Each of these videos consists of approximately 15,000 frames. To speed up processing, we decided to only analyze only frame per second of video content, resulting in approximately 6,000 frames per video.

We use a YOLOv5-based [2] object detection approach, for which the implementation used to perform all experiments is presented in the dataset's official GitHub repository³ and is based on the official YOLOv5 implementation⁴. We experimented using four different versions of the YOLOv5 architecture (YOLOv5s, YOLOv5n, YOLOv5m, and YOLOv5x), where transfer learning was performed from the official coco weights that are included in the aforementioned YOLOv5 repository. Each model was trained for a maximum of 300 epochs, stopping if the model did not improve $mAP(0.5 : 0.95)$ on the validation dataset for the previous 10 epochs. All experiments were performed using three-fold cross-validation to ensure that each data sample was used in both training and validation. The experiments were run on what can be considered consumer-grade

³<https://github.com/simula/njord>

⁴<https://github.com/ultralytics/yolov5>

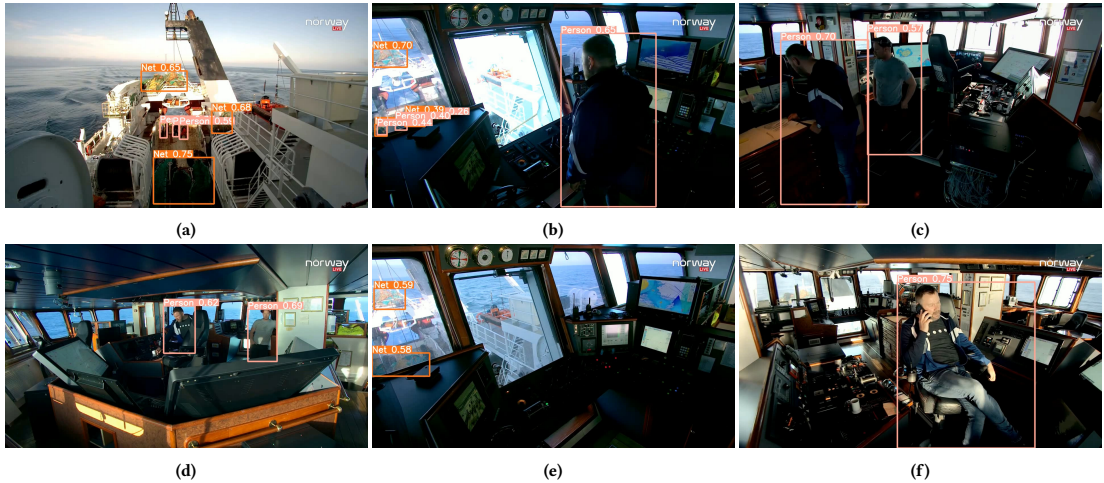


Figure 3: Sample predictions made by the YOLOv5m model. (a) - we see several workers on deck where the model is able to detect the workers in addition to the nets, (b) - The captain overlooking the workers on deck for which both the captain and workers are detected, (c) - A view from the trawler bridge with two workers detected by the model, (d) - A view of the bridge with a detected worker and captain, (e) - A view of the deck where the model detects the nets, and (f) - The captain sitting on the bridge which is detected by the model.

hardware consisting of an RTX 3090 Nvidia GPU and an Intel i9 CPU.

According to the results presented in Table 1, the best $mAP(0.5 : 0.95)$ value of 0.277 and $mAP(0.5)$ of 0.569 are achieved by the YOLOv5m model. However, the YOLOv5s model can obtain a higher precision with a value of 0.732, while the YOLOv5x model shows the best recall with a value of 0.570. Examples of the predictions made by the best model (YOLOv5m) according to the mAP values are depicted in Figure 3. The baseline results are promising and show that the dataset can be used to perform interesting analysis tasks. Nevertheless, the results are far from perfect and future work is needed.

In terms of detection speed, it took on average 192.22 frames per second using the smallest model (YOLOv5n) and on average 86.35 frames per second using the largest one (YOLOv5x). For training, the YOLOv5n model took on average 34 minutes to train per fold, and the YOLOv5x model trained for approximately 2 hours per fold. We see the potential for improvement and several areas for interesting research questions, especially considering the large amount of data being processed.

7 Conclusion and Future Work

In this paper, we describe a novel dataset from the commercial fishing domain, which has not been explored yet. Datasets have previously been published related to sustainable fishing and boat and fish detection, but not specifically about activities and processes happening on board fishing vessels. With this dataset, we contribute to opening up this new and challenging domain of rapidly growing interest. We present a baseline experiment on object detection using

the dataset, which shows promising results but holds potential for improvement. In addition, we point at several possible research directions that can take advantage of the Njord dataset.

Specifically for the presented dataset, we will continue to annotate the remaining 127 videos and update the dataset continuously that we have available and also extend the dataset further with new data. There might be other objects or features of these videos that could be interesting to have annotated. Due to the videos coming from a live-stream, the captain sometimes addresses the audience and gives general information about where they are, what the plans are for that day, and other general information. Labeling this could be interesting for general natural language processing tasks such as question answering or translation. For the fish annotations, it would be interesting to have fine-grain annotations such as segmentations per fish that could, for example, be used to train models that automatically approximate the biomass of the entire catch.

Acknowledgements

We particularly thank Hermes staff and owners for relevant discussions, meetings, and for allowing us to annotate and publish this dataset. This work is partially funded by the Research Council of Norway project number 274451 and Lab Nord-Norge ("Samfunnsløftet").

References

- [1] Gerard Gilbert. 2014. *Slow Television: The latest Nordic trend*. <https://www.independent.co.uk/arts-entertainment/tv/features/slow-television-chess-trains-and-knitting-9122367.html>
- [2] Glenn Jocher. 2020. *ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements*. <https://doi.org/10.5281/zenodo.4154370>

- [3] Labelbox. 2022. *Labelbox*. <https://labelbox.com>
- [4] Daoliang Li, Qi Wang, Xin Li, Meilin Niu, He Wang, and Chunhong Liu. 2022. Recent advances of machine vision technology in fish classification. *ICES Journal of Marine Science* (2022).
- [5] Mi-Ling Li, Yoshitaka Ota, Philip J Underwood, Gabriel Reygondeau, Katherine Seto, Vicky WY Lam, David Kroodsmas, and William WL Cheung. 2021. Tracking industrial fishing activities in African waters from space. *Fish and Fisheries* 22, 4 (2021), 851–864.
- [6] Alihan Mermer, TÜRK Meral, and Zafer Tosunoğlu. 2022. Occupational health and safety in large-scale fishing vessels registered in Aegean ports. *Ege Journal of Fisheries and Aquatic Sciences* 39, 1 (2022), 18–23.
- [7] Jaeyoon Park, Jungsam Lee, Katherine Seto, Timothy Hochberg, Brian A Wong, Nathan A Miller, Kenji Takasaki, Hiroshi Kubota, Yoshioki Oozeki, Sejal Doshi, et al. 2020. Illuminating dark fishing fleets in North Korea. *Science advances* 6, 30 (2020), eabb1197.
- [8] Alzayat Saleh, Issam H Laradji, Dmitry A Kononov, Michael Bradley, David Vazquez, and Marcus Sheaves. 2020. A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis. *Scientific Reports* 10, 1 (2020), 1–10.
- [9] Monique Simier, Jean-Marc Ecoutin, and Luis Tito de Morais. 2019. The PPEAO experimental fishing dataset: Fish from West African estuaries, lagoons and reservoirs. *Biodiversity Data Journal* 7 (2019).
- [10] Paolo Spagnolo, Francesco Filieri, Cosimo Distante, Pier Luigi Mazzeo, and Paolo D'Ambrosio. 2019. A new annotated dataset for boat detection and re-identification. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 1–7.
- [11] Kaiyu Tang, Yixin Bao, Zhijian Zhao, Liang Zhu, Yining Lin, and Yao Peng. 2018. AutoHighlight : Automatic Highlights Detection and Segmentation in Soccer Matches. In *2018 IEEE International Conference on Big Data (Big Data)*. 4619–4624. <https://doi.org/10.1109/BigData.2018.8621906>
- [12] Jiangping Wang, Anthony Pillay, YS Kwon, AD Wall, and CG Loughran. 2005. An analysis of fishing vessel accidents. *Accident Analysis & Prevention* 37, 6 (2005), 1019–1024.
- [13] Tianwen Zhang, Xiaoling Zhang, Xiao Ke, Chang Liu, Xiaowo Xu, Xu Zhan, Chen Wang, Israr Ahmad, Yue Zhou, Dece Pan, et al. 2021. HOG-ShipCLSNet: A novel deep learning network with hog feature fusion for SAR ship classification. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), 1–22.
- [14] Tianwen Zhang, Xiaoling Zhang, Xiao Ke, Xu Zhan, Jun Shi, Shunjun Wei, Dece Pan, Jianwei Li, Hao Su, Yue Zhou, et al. 2020. LS-SSDD-v1.0: A deep learning dataset dedicated to small ship detection from large-scale Sentinel-1 SAR images. *Remote Sensing* 12, 18 (2020), 2997.
- [15] Yue Zhang, Masato Yamamoto, Genki Suzuki, and Hiroyuki Shioya. 2022. Collaborative Forecasting and Analysis of Fish Catch in Hokkaido from Multiple Scales by Using Neural Network and ARIMA Model. *IEEE Access* (2022).

A.6 Paper V: Detection of Slipping Events using Multimodal Data

Authors: T.S. Nordmo, M.M. Espeseth, B.A. Juliussen, M.A. Riegler and D. Johansen

Abstract: “Slipping”, or deliberate release, of dead or dying fish is a potentially illegal act, depending on the fish species and jurisdiction where the fish was caught. These events are extremely difficult to regulate, due to fishing vessels being far out to sea and spread out across a large area. Therefore, detecting such events manually using inspection boats is infeasible. However, slipping events can lead to release of fish oil, if the fish dies as part of the release process, which can be detected in satellite imagery given good detection capabilities. We propose an approach to detect fishing vessels that might have deliberately released fish they have caught. The method we propose is to analyze multiple sources of data simultaneously and combine it, using positional data, sales notes, oil slick detection and satellite-based vessel detection. We evaluate our approach, discuss results and aspects related to each data source, and describe legal aspects regarding slipping. Our experimental results show that we are able to provide a ranking of suspicious vessels that can be used for further investigations.

Author contributions (initials): **Conceptualisation:** T-A.S.N., D.J.; **Data collection:** T-A.S.N., M.M.E., **Methods, data analysis and interpretation:** T-A.S.N., M.M.E., **Drafting:** T-A.S.N., M.M.E., B.A.J., M.A.R., D.J., **Critical revision:** T-A.S.N., M.M.E., B.A.J., M.A.R., D.J.

Published: IEEE International Symposium on Multimedia (ISM) 2022

Thesis objectives: Sub-objective 2,3,4

A.7 Paper VI: **Áika**: A Distributed Edge System For AI Inference

Authors: J.A. Alslie, A.B. Ovesen, T.S. Nordmo, H.D. Johansen, P. Halvorsen, M.A. Riegler and D. Johansen

Abstract: Sustainability challenges presented by several governing bodies have outlined the need for surveillance of commercial fisheries in world oceans. Traditional video monitoring systems may not be suitable due to limitations in the offshore fishing environment, with unstable, low bandwidth, high latency satellite connections, and issues of preserving the privacy of crew members. Some of these challenges can be solved by moving applications to the edge, close to the data source. This paper presents **Áika**, a robust system for executing distributed Artificial Intelligence (AI) applications on the edge, designed for monitoring and surveillance in privacy-sensitive offshore environments. Faults are handled through replication and a distributed checkpointing scheme, with a dedicated monitoring node continuously evaluating the system state. At the same time, flexible access policies at the storage level enforce privacy limitations on data replication and transfer. Experiments show that **Áika** is feasible as a platform for distributed AI applications in this scenario, as it is able to maintain computation tasks to completion in unstable environments.

Author contributions (initials): **Conceptualisation:** J.A.A., D.J., A.B.O., T-A.S.N.; **Data collection:** J.A.A., **Methods, data analysis and interpretation:** J.A.A., **Drafting:** J.A.A., A.B.O., T-A.S.N., H.D.J., P.H., M.A.R., D.J., **Critical revision:** J.A.A., A.B.O., T-A.S.N., H.D.J., P.H., M.A.R., D.J.

Published: MDPI Information, 2022

Thesis objectives: Sub-objective 1



Article

Áika: A Distributed Edge System for AI Inference

Joakim Aalstad Alslie¹, Aril Bernhard Ovesen^{1,*}, Tor-Arne Schmidt Nordmo¹, Håvard Dagenborg Johansen¹, Pål Halvorsen^{2,3}, Michael Alexander Riegler^{1,2} and Dag Johansen¹

- ¹ Department of Computer Science, UiT The Arctic University of Norway, 9037 Tromsø, Norway; jal029@post.uit.no (J.A.A.); tor-arne.s.nordmo@uit.no (T.-A.S.N.); havard.johansen@uit.no (H.D.J.); michael@simula.no (M.A.R.); dag.johansen@uit.no (D.J.)
- ² Holistic Systems Department, SimulaMet, 0164 Oslo, Norway; paalh@simula.no
- ³ Department of Computer Science, Oslo Metropolitan University, 0130 Oslo, Norway
- * Correspondence: aril.b.ovesen@uit.no

Abstract: Video monitoring and surveillance of commercial fisheries in world oceans has been proposed by the governing bodies of several nations as a response to crimes such as overfishing. Traditional video monitoring systems may not be suitable due to limitations in the offshore fishing environment, including low bandwidth, unstable satellite network connections and issues of preserving the privacy of crew members. In this paper, we present Áika, a robust system for executing distributed Artificial Intelligence (AI) applications on the edge. Áika provides engineers and researchers with several building blocks in the form of Agents, which enable the expression of computation pipelines and distributed applications with robustness and privacy guarantees. Agents are continuously monitored by dedicated monitoring nodes, and provide applications with a distributed checkpointing and replication scheme. Áika is designed for monitoring and surveillance in privacy-sensitive and unstable offshore environments, where flexible access policies at the storage level can provide privacy guarantees for data transfer and access.



Citation: Alslie, J.A.; Ovesen, A.B.; Nordmo, T.-A.S.; Johansen, H.D.; Riegler, M.A.; Halvorsen, P.; Johansen, D. Áika: A Distributed Edge System for AI Inference. *Big Data Cogn. Comput.* **2022**, *6*, 68. <https://doi.org/10.3390/bdcc6020068>

Academic Editor: Tzung-Pei Hong

Received: 29 April 2022

Accepted: 14 June 2022

Published: 17 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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/>).

Keywords: edge computing; edge inference; computation frameworks; sensor networks

1. Introduction

In attempts to decrease latency, and increase security and reliability, some new AI solutions are gradually being deployed and executed on the edge. Stoica et al. [1] and Carcillo et al. [2] discuss challenges related to the field of AI, and list both edge computing and security as important topics of research. The term *edge intelligence* is often used to describe the confluence of the edge computing and AI fields [3].

Moving AI solutions to the edge may be required for systems that operate in environments where access to high-speed internet connections is either limited or non-existent, and available bandwidth is too low for effective data transportation. For instance, on-boat computer systems that operate in remote locations such as international waters and Antarctica rely on satellite connection to communicate with on-land services outside their environment. The development of 5G networks may improve connectivity in the future, but global coverage is not likely due to its short range [4].

Deploying an AI system on the edge comes with certain challenges, particularly for monitoring and surveillance systems that are concerned with privacy regulations and agreements. Systems deployed at physically remote edge locations may be prone to attacks from malicious actors in the environment, and the consequences of data leaks are more severe when sensitive and private data are being generated and stored. AI systems that deal with private data of users also need to satisfy the requirements of privacy-governing laws such as the General Data Protection Regulation (GDPR) [5] and constitutional rights to privacy. This places restrictions on the monitoring and data-collection process, while also heightening requirements of confidentiality and resilience to hostile attacks.

This article presents Áika, a robust system for executing distributed AI applications on the edge. Áika is developed and evaluated in a scientific context as a concrete edge computing platform specially targeting support for real-time AI systems with special security and fault-tolerance properties. A key property of Áika is how it remains active and performs continuous analysis of data during various component failures. We investigate how to provide efficient data analytics in an unstable and non-trusted edge environment.

Our work on the Áika system is motivated by the need for highly automated continuous AI-based monitoring and privacy-preserving surveillance of fishing vessels at sea. Fish is considered one of the most important food sources in the world, and it is estimated that it currently makes up around 17% of the current production of edible protein on a global scale [6]. At the same time, the fishing industry has fallen victim to crime in the form of illegal fishing and over-exploitation. According to the United Nations Office on Drugs and Crime, crimes in the fishery industry are typically organized and transnational in nature, and include money laundering, illegal fishing, document fraud, and drug trafficking [7]. The system is developed and evaluated in the context of surveillance and monitoring off the shore of Norway for the purpose of enforcing sustainable resource management and fish harvesting in the Arctic.

This computing environment is assumed to be both unstable and untrusted. Systems operating in it have elevated risk of faults and intrusions compared to stationary or cloud-connected systems. Factors that contribute to this risk include low and unpredictable bandwidth, high latency, unstable connections, remote locations, and uncertain time frames between physical interaction with the components of the system, in addition to the potential threat of malicious actors. The challenging weather conditions in the Arctic that such fishing vessels operate in are also a concern. This increases the importance of designing a robust and secure system that not only is able to tolerate faults, but can also detect and monitor them.

The main contributions of the work presented in this paper encompass the design of the Áika system, which enables the expression and execution of distributed AI applications, by utilizing a set of reusable design patterns for structuring distributed computations on the edge. We show that AI solutions can be executed in untrusted edge environments through a graph computation model, through our fault-tolerant middleware system that can detect, recover from, and report abnormal behavior.

In the following sections, we will discuss the motivation and architecture of our proposed system, followed by experiments, related work, and discussions.

2. Background

The sustainability issues and economic challenges related to illegal fishing have caused several governments to propose surveillance systems to track the activities of workers [8–10], which has been characterized as privacy infringing and mass surveillance by some of those working on fishing vessels [11]. The Dutkat framework [12] was designed to retain some of the sustainability benefits of these programs [13] while preserving the privacy of fishing vessel crew.

Solutions for distributed video analysis have been presented in several works related to surveillance, traffic monitoring, and smart city applications [14]. Some deploy lightweight edge sensors that encrypt data before transmission [15], while others choose to perform object detection and privacy preservation directly at edge nodes [16]. Reliance on connectivity to a cloud service is common for these monitoring platforms; it is sufficient to perform transformations to reduce bandwidth usage or ensure privacy at the edge before performing the most computationally intense tasks in a more centralized and connected environment. However, the domain of fishing monitoring comes with a greater challenge of connectivity, and the bandwidth available for offshore fishing vessels is not sufficient to provide a satisfactory real-time transfer of video data [17]. This calls for a system specialized for this domain, which can be deployed on resourceful vessels that

continuously move in and out of edge environments with low bandwidth, high latency, and unreliable connectivity.

Nodes are expected to move into areas where they are unreachable from mainland services, which heightens requirements of fault tolerance and fault detection. Aika aims to recover from faults quickly in order to resume any interrupted computation processes and restore the system state. Manual inspections of the system during runtime may be challenging, or prohibitively expensive, due to the remote locations of nodes. As such, detected faults must be logged and classified for evaluation of its severity and probability of resulting from interference of malicious actors, to aid in decision making regarding the need for manual intervention or inspection.

Loading AI models into memory can be time consuming, which might negatively affect the recovery time. The system should therefore support resilient schemes by redundancy where replicated components process the same data, simultaneously. If one component fails, other components should still be processing its data, ensuring that throughput remains stable despite failures. The system should support configuration of resilient algorithms such as triple-modular redundancy, if a user wants to apply them to the system.

The domain of maritime surveillance involves storage and processing of data from various sources, including video and sensor data. Two practical storage challenges arise from this use-case and environment: First, nodes are geographically distributed and will be hindered by latency and bandwidth during retrieval of remotely stored data. It cannot be assumed that every node has access to all relevant data at all times. Second, the physical location of nodes, and the information they store locally, can be subject to varying juridical requirements and agreements. It is assumed that input data can contain sensitive information that is prone to privacy agreements and legal regulations. A distributed computation pipeline in an unstable environment may require geographic replication and redundancy to remain operative. At the same time, a system processing sensitive data may depend on agreements and regulations that restrict data consumption and movement, based on the information contained in input streams, particularly those containing videos and images of humans.

There are several systems that provide data storage and processing facilities. Client-Edge-Server for Stateful Network Applications (CESSNA) is a protocol developed to achieve resilient networking on a stateful edge computing system [18]. The protocol provides consistency guarantees for stateful network edge applications and allows offloading of computations from clients and servers. This leads to a reduction in response latency, backbone bandwidth, and computational requirements for the client.

FRAME is a fault-tolerant and real-time messaging system for edge computing [19] based on a publish-subscribe architecture, with a duplicated broker to avoid having a single point of failure. FRAME leverages timing bounds to schedule message delivery and replication actions to meet needed levels of assurance. The timing bounds are thus able to capture relation between traffic/service parameters and loss-tolerance/latency requirements. The architecture is implemented on top of the TAO real-time event service [20].

Norwegian Army Protocol (NAP) is a scheme for fault tolerance and recovery of itinerant computations [21]. The runtime architecture resolves around having a landing pad thread and a failure detection thread within each process. The landing pad is responsible for maintaining a NAP state object that stores information about mobile agents hosting execution or serving a rear guard. The landing pad thread is responsible for informing the failure detection thread which landing pad needs to be monitored. NAP uses a linear broadcast strategy that refines the strategy implemented by Schneider et al. [22].

Falcon Spy [23] provides distributed failure detection and recovery using a network of spies in the application, operating system, virtual machine, and network switch layers on the system being monitored. The spies are deployed at the different layers to hinder disruption. The motivation behind Falcon Spy is to enable effective failure detection and improve the previous method (end-to-end timeouts).

Dryad is a general-purpose distributed execution engine developed by Microsoft, used to execute coarse-grained data-parallel applications [24]. One of Dryad's key features is to allow the user to construct an execution DAG through a custom graph description language. The Dryad runtime maps the graph onto physical resources. The graph vertices allow an arbitrary number of inputs and outputs, unlike MapReduce [25], which only supports single inputs and outputs. A job manager contains the application-specific code used to construct the communication graph. It also schedules work across available physical resources, which are maintained by a name server.

Cogset [26] is a high-performance engine that builds on the principles of MapReduce [25], but uses static routing while preserving non-functional properties. Cogset provides a few fundamental mechanisms for reliable and distributed storage of data and parallel processing of statically partitioned data at its core.

The Staged Event-Driven Architecture (SEDA) [27] simplifies the construction of well-conditioned and highly concurrent Internet services. SEDA applications are constructed as a network of stages. A stage is defined as an application component that consists of three sub-components: an event queue that handles incoming events, an event handler, and a thread pool.

3. System Overview

Áika is designed to execute multiple distributed AI pipelines in a Directed Acyclic Graph (DAG) computation format on edge clusters, supporting a large range of complex distributed analytics executing on edge devices. Because Áika is intended to run in potentially hostile edge environments, actors cannot be fully trusted to faithfully execute any specified protocol and access to high-speed Internet may be limited. Fault tolerance is necessary to ensure that the system does not fail at runtime. Limited Internet connections can also lead to data being generated at a higher rate than the connection can transport. The solution is to move the analytical process to the edge, which is where the data are generated by sensors.

To support our requirements, we design Áika as a hierarchical system where a controller is responsible for monitoring the remaining part of the system and executing recovery when a component fails. The overall system is composed of multiple processes that communicate in a cluster. Each process has a specific role and it is not changed during runtime. The processes are as following:

Agent The agent is responsible for processing data. The agent can either fetch this data from the disk itself, or receive data from other agents.

Local Controller The local controller is responsible for monitoring the agents that reside on the same physical node as the local controller itself. Each physical device has at least one local controller running. A local controller without any agent to monitor is referred to as a *replica*. This type of local controller can be used for recovery if an entire physical node fails.

Cluster Controller The cluster controller is responsible for managing the entire cluster. It communicates with the local controllers to ensure that each physical node is running. If a physical node shuts down, the cluster controller is responsible for initiating recovery, either directly on the node where the failure occurred, or on a replica.

Monitor This is an additional process that is meant to be physically located on a trusted location, unlike the system itself. The monitor is responsible for communicating with the system to ensure that it is up and running. In the case of failure, the monitor may notify personnel or authorities about this.

The different roles will be covered in greater depth in Sections 3.2–3.4.

3.1. System Components Structure

We design Áika using a hierarchical structure, as shown in Figure 1. This structure also applies to each individual components. Each individual component is designed as a

multi-threaded process. A main thread is responsible for initializing and monitoring child threads, where the child threads execute some type of behavior in the system. This can, for example, be to initialize a multi-threaded server, communicate with another component in the system through a client, or execute some form of custom work. This varies from component to component. If any of the child threads fail, they will be restarted by the main thread.

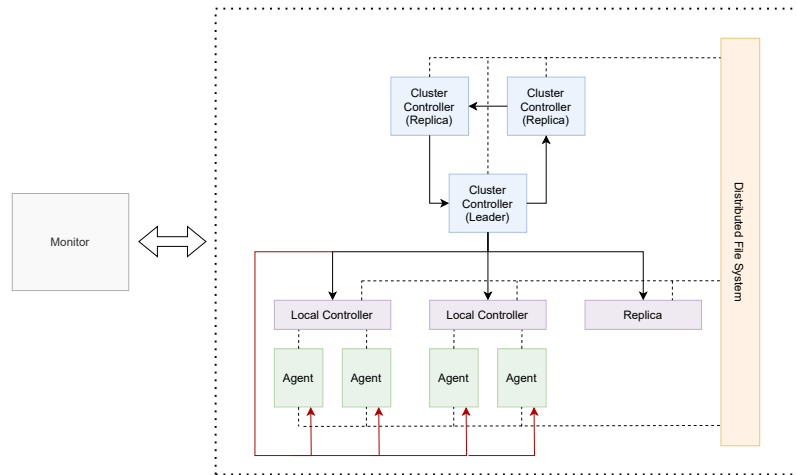


Figure 1. Áika’s architecture. Arrows represent client/server communication. Red arrows represent communication that may only occur during recovery. The figure does not include communication between agents. All nodes in the cluster are connected to a distributed file system that enables file sharing across the nodes. This is practical for recovery.

If a main thread shuts down, its child threads are also shut down and the entire process will fail. This approach ensures that servers are shut down if the main thread fails. This simplifies the recovery process, since no thread can be partially available and occupy resources after failure. Figure 2 illustrates how processes can be organized into a hierarchy of threads.

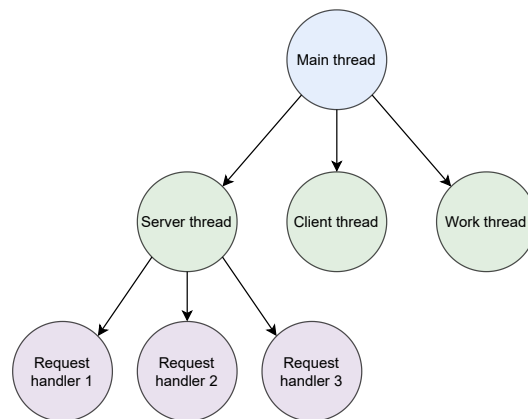


Figure 2. The general process structure of Áika’s components. Each process is organized in a hierarchy of threads, where a main thread starts several child threads. Child threads are restarted by the main thread if they fail. Servers spawn multiple request handler threads to enable requests from multiple sources to be handled concurrently.

3.2. Controllers

Aika implements a hierarchical multi-layered Controller/Agent design, where agents are managed by local controllers residing on each physical device. The local controllers are managed by a cluster controller that manages the remaining components in the system.

3.2.1. Local Controller

The local controller is responsible for monitoring the agents residing on the same physical nodes. It ensures that each agent is running and, in the case of crash failure, restarts the agent that crashed. It also logs the crash and the time of detection. In the event of a physical device crashing, a replica local controller will be responsible for restarting and recovering the local agents that crashed. This type of recovery process is initiated by a cluster controller.

3.2.2. Cluster Controller

The cluster controller is responsible for monitoring the entire cluster of computers that runs the system. The cluster controller has a Controller/Agent relationship with the local controller, where the local controller functions as the agent. Whenever a remote monitor attempts to connect to the cluster controller, it must provide a response to it to ensure the monitor that the system is running.

If a local controller fails, the cluster controller will attempt to recover it. The cluster controller initiates node failure recovery if it fails to recover the failed local controller. This means that the configuration of the failed local controller is forwarded to a replica local controller.

The cluster controller is duplicated to avoid complete system failure in the case where it crashes. The cluster controllers are organized into a chain (see Figure 3) where each cluster controller responds to ping requests from their predecessor while pinging their successor. Each cluster controller has the full system configuration and therefore knows about all components. If the cluster controller in the chain fails completely and cannot be recovered, the predecessor will remove the cluster controller from its configuration and move to the next cluster controller in the chain.

If the main cluster controller fails, the duplicated controller that monitors the leader will attempt to recover it. If it fails the attempted recovery, it will instead become the new leader.

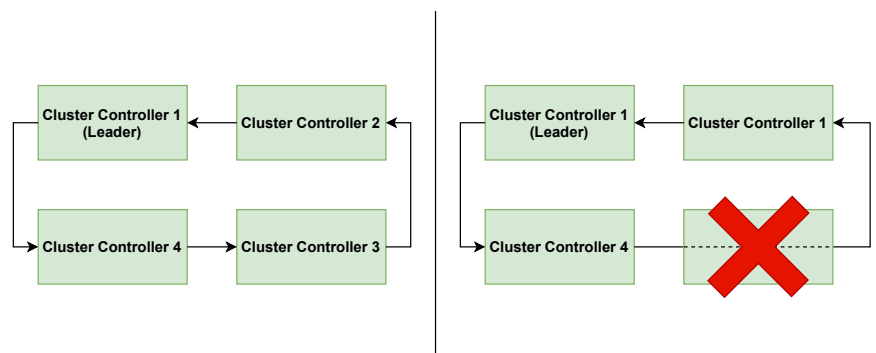


Figure 3. The cluster controller is replicated and connected in a chain.

3.3. Agents

The agents are responsible for working with and processing the data, and they make up the core building blocks of the DAG computation model. The general task of each agent is thus to receive or fetch work (either from another agent or from file), then process data based on the work received before passing the results further ahead in the graph. Work items are transferred over client/server connections.

Figure 4 shows the general structure of the agents. A client or server thread is used to request or receive data from the previous agent. The thread continuously puts work items received on an input queue, which a work thread retrieves items from, before performing some type of work on the item. The result is put on an output queue. Another client or server thread is then used to forward the result to the next agent in the graph.

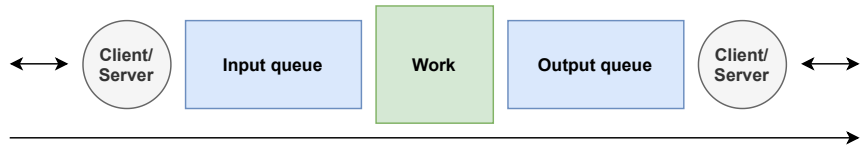


Figure 4. Shows the general structure of the agents.

Preserving data integrity despite failure is achieved through the use of persistent queues, as they continuously write items to file as they are being inserted into the queue. In the case of failure, an agent will always be able to resume from the previous checkpoint upon recovery, as long as it is connected to the distributed file system. By writing the persistent queues to a file, a replica local controller will also be able to resume the work if the physical node shuts down, since it also will be connected to the same file system. A mechanism in the queue enables items to only be removed from file after work on the item has been completed and the result has been forwarded to the next queue. This mechanism is used both during work and during communication between agents to ensure that items are not lost.

The agents constitute the building blocks for a DAG, which is configured by the user. The DAG can be configured to be complex and, for instance, consist of nodes that receive data from multiple sources, or pass data forward to multiple sources. A set of base agents has therefore been created, which uses different combinations of client and server threads at each end. Each agent has particular use cases where they are useful. Note that the figures of each individual agent has abstracted the persistent queues between client and work away for simplicity.

3.3.1. Left Worker Agent

The left worker agent is composed of a server thread on the left side, where items are received, and a client on the right side, where items are forwarded. This is illustrated in Figure 5, where other agents can put items on the left agent’s input queue by making a request of the left agent’s server. The left agent is responsible for forwarding the item to another agent itself by performing its own remote enqueue call to the agent.

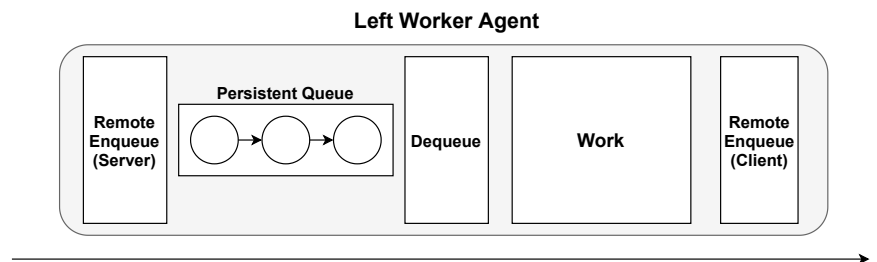


Figure 5. Left worker agent. A worker agent that dequeues messages from a local queue before the analysis process, but enqueues the result remotely.

This type of agent is useful in cases where data are received from multiple sources. The client on the right side enables the agent to forward the same item to multiple sources. An example use case for the left agent could be to use it as a voter agent for implementing N-modular redundancy.

3.3.2. Right Worker Agent

The right worker agent is composed of a client on the left side and a server on the right side (see Figure 6). This means that the agent fetches items itself from a single agent through a remote dequeue call, while items are forwarded when other agents request them.

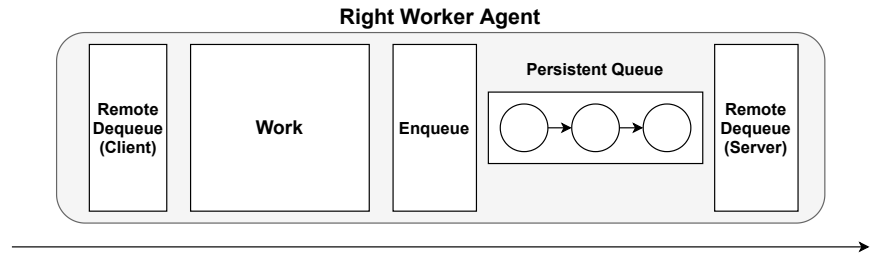


Figure 6. Right worker agent. A worker agent that dequeues messages from a remote queue before the analysis process, but enqueues the result on a local queue.

The server on the right side enables the agent to scatter items to different agents. This is useful in situations where load balancing is required due to upcoming computation heavy work. The consequence of using the right worker agent is that the client enables it to fetch data from a single source only.

3.3.3. Double Worker Agent

This type of agent contains servers both before and after processing the item (see Figure 7). This makes the agent completely passive, as messages are only received and forwarded through requests from other agents. This type of agent can be useful in cases where it receives messages from and scatters messages to multiple sources.

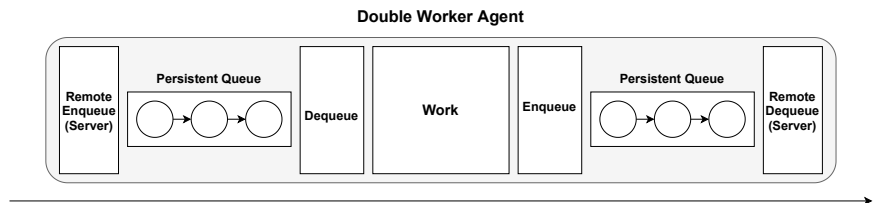


Figure 7. Double worker agent. A worker agent containing servers both before and after processing the item.

3.3.4. Initial Worker Agents

The purpose of the initial agent is to fetch data from some location in a custom manner (implemented by the application developer), before forwarding them to the next pipeline stage. The data flow is illustrated in Figure 8. It is meant to be used as the first stage in the pipeline. Initial agents can either use a client or a server to forward items further into a pipeline. In Section 4.5, we demonstrate how the initial server agent can be used for load balancing for counting words in a textual document.

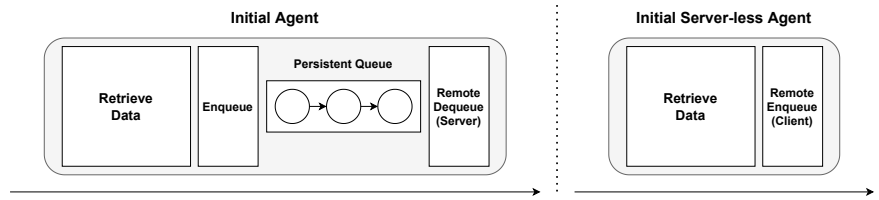


Figure 8. Initial worker agents. This type of agent is used to initiate one or several pipelines. This is achieved by having the agent continuously retrieve data from a source and then forward it to either a local (see left figure), or a remote (see right figure) queue.

3.3.5. Final Worker Agents

The purpose of the final worker agent is to carry out the final work on an item at the end of a pipeline within the DAG. Because of this, it does not have any output queue, or client/server thread after work. The final agent can utilize a client for fetching items, or a server for receiving them. These two types of final worker agents are illustrated in Figure 9.

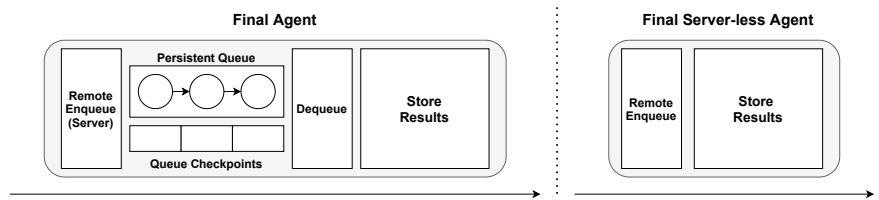


Figure 9. Final worker agents. This type of agent is used to finalize one or several pipelines. The agent retrieves the end results from either a local or a remote queue, then handles the result in a customized manner.

3.3.6. Queue Agent and Server-less Agent

The queue agent is only composed of one scheduling queue, which leaves the responsibility of enqueueing and dequeueing messages entirely up to other agents. It is passive, like the double queue agent, and it is also useful in similar cases where work is not required to be performed on the item in between. It can be used as a collection point for data from multiple sources that are scattered afterwards.

The server-less agent contains only clients and is therefore responsible for both fetching items from another agent and forwarding items after processing (see Figure 10). In Section 4.6, we demonstrate how a server-less agent can be used to retrieve items from a load balancing queue on another agent and forward the item to feature extraction models on multiple agents to increase performance.

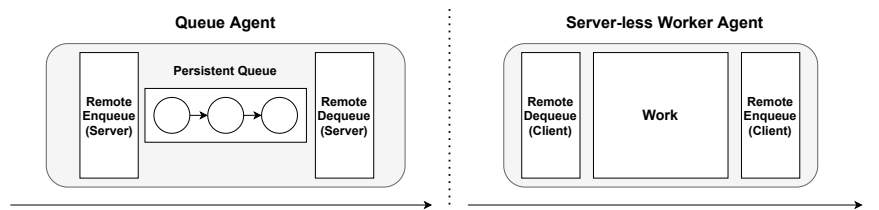


Figure 10. Queue agent and server-less worker agent. The agent to the left contains a single queue without any analysis. The agent to the right does not have any servers and receives items by making requests through clients on both sides.

3.4. Monitor

The remote monitor resides at a physically remote location relative to the edge system and is responsible for communicating with the system, potentially over low bandwidth. The reason for using a remote monitor is that it is not possible to fully control the physical equipment completely from an untrusted edge. The equipment is subject to potential physical harm and signal interference, and can be difficult to reach and recover.

The remote monitor is used as a safety device that resides within a safe location that can monitor the system state. By continuously communicating with a cluster controller in the system, the monitor can receive information such as which physical devices are running, if any known crashes have occurred, or additional information regarding the system. If the remote monitor fails to reach the system, it may classify this as an abnormal event and report it to system administrators or authorities. Information such as the context surrounding the failure can aid in determining the need for further investigation or inspection of the remote nodes.

3.5. File System

All nodes connect to the same distributed file system for storage and retrieval of input data and checkpoints. As data generated at edge locations can contain sensitive information, privacy agreements determining the permitted handling and use of data must be respected both by end-users and by data movement in software. The Dorvu file system [17] is used as a storage platform for data and checkpoints in Áika for this reason.

While the communication and recovery scheme of Áika assumes the total access of requested data through a file system interface, the storage platform can enforce access control policies that respect restrictions such as the permitted geographical storage locations of files. Policies can be configured to only apply for certain files depending on the semantic information contained in them, enabling a fine-grained access control scheme. For instance, policies expressed in Dorvu can allow the transfer of files to certain locations on the condition that specific information is removed beforehand. Dorvu achieves this by executing custom policy programs at the time of file creation, and encrypting various parts of the file with different keys depending on access level. This transparently enforces policy compliance for each component in the Áika system.

4. Experimental Setup and Evaluation

In this section, we describe several experiments for evaluating the performance of Áika. All experiments are carried out at on a local Rocks cluster (version 7.0) running CentOS 7. The internal transfer rate between machines in the cluster is expected to be approximately 100 MB/s. The experiments are performed on a subset of homogeneous cluster nodes consisting of 55 Lenovo P310 computers with one Intel Core(TM) i7-6700 @ 3.40 GHz with 4 cores, 32 GB RAM, and a Nvidia GM107GL (Quadro K620) GPU each.

4.1. Micro-Benchmarks

To gain insight on the performance and overhead of the individual component of Áika and how time is spent on the various tasks, we ran several micro-benchmarks. These benchmarks are summarized in Table 1.

Table 1. Results from micro-benchmarks.

Micro-Benchmark	Time (milliseconds)
Local Controller Initial Startup	300
Local Controller Further Startup	150 (per local controller after the first)
Agent Startup	20–40
Pass integer item from agent to agent	7–9

We experienced that the cluster controller took approximately 300 milliseconds to start a single local controller on a separate node with an SSH client. After setting up one local controller, the time increases with approximately 150 milliseconds per additional local controller. The local controller, however, takes approximately between 20 and 40 milliseconds to start an agent. This demonstrates that the local controller not only can be used to offload the workload of the cluster controller, but can also manage to recover agents in a shorter amount of time, leading to a more efficient recovery procedure. This is especially relevant if agents shut down often.

Passing integer items from one agent to the next in a pipeline takes approximately 8 milliseconds. During this time, the item moves through two persistent queues and one TCP stream.

4.2. Persistent vs. In-Memory Queues

To understand the overhead of the persistent queues used by Áika to store information between computation steps, we measure the end-to-end throughput of the system within regular time frames at the end of a pipeline. The effects of runtime failures on performance is simulated by inducing crashes.

For these experiments, we configure Áika in a single pipeline setup consisting of one initial agent, five worker agents, and one final agent. The initial agent creates the workload data by forwarding integers into the pipeline. Worker agents retrieve and increment these integers before forwarding them to the next pipeline steps. To simulate work time, a worker agent may also sleep after an integer has been incremented. The final agent receives the data and increments its counter of received requests. A separate processing thread writes the number of requests received every five seconds to a log file. By batching these disk writes, we reduce the impact measurement logging has on system performance.

The observed number of requests received when running the pipeline configuration for 1200 s (20 min) can be seen in Figure 11. Because the data are highly varied, the figure also includes the moving average using a sliding window size of $n = 5$.

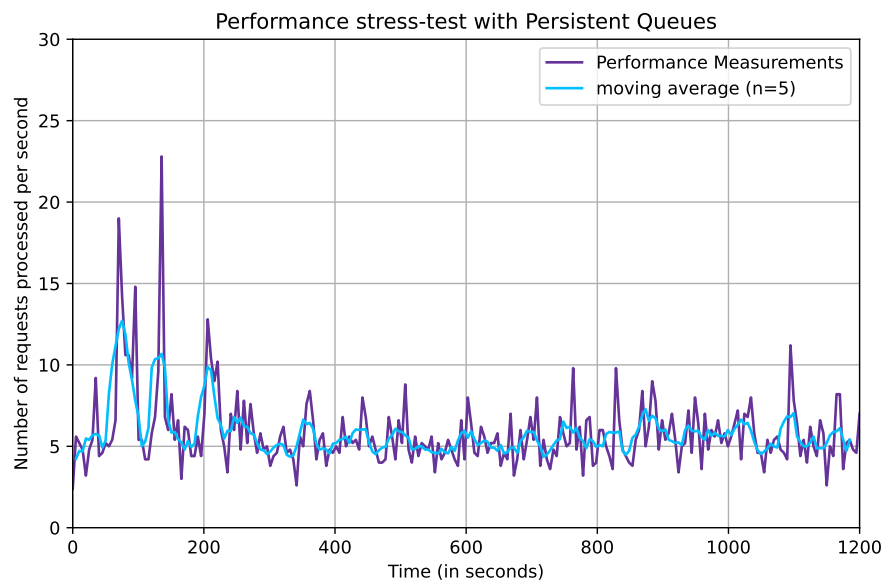


Figure 11. The obtained results from stress-testing the system with persistent queues over the course of 1200 s. The number of requests are measured and reset every 5 s. The plot shows the average number of requests processed per second over each 5 s interval. The moving average is computed with a window size $n = 5$.

From the figure, we can observe that the system has a high throughput in the initial stages of the process, before it becomes relatively stable at around 250 s, where the number of requests processed ranges between four and eight requests per second. This demonstrate the system's maximum throughput on our current hardware infrastructure.

Next, we perform a similar experiment, but using in-memory queues in the agents instead of persistent disk-based queues. The in-memory queue is configured to hold a maximum of 100 items. The observed end-to-end throughput is plotted in Figure 12.

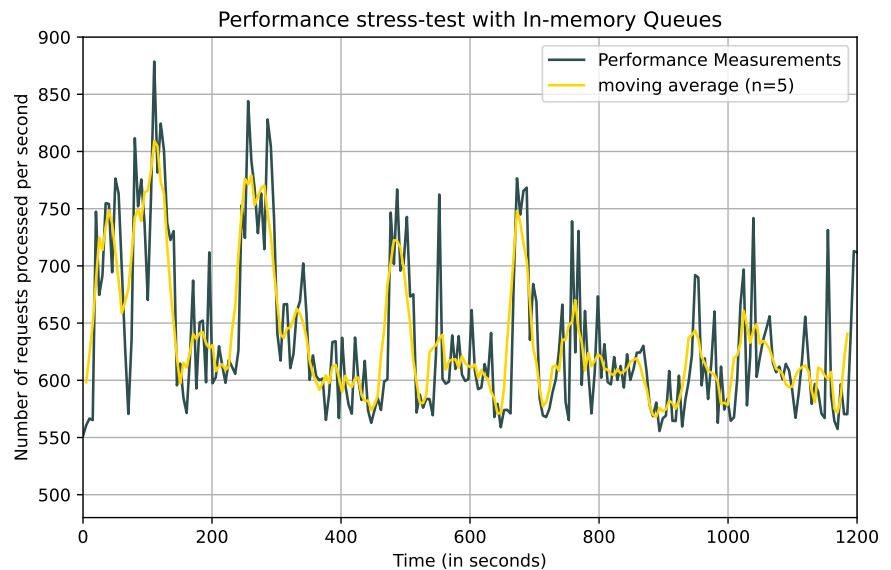


Figure 12. The results obtained by stress testing the system with in-memory queues over the course of 1200 s. Number of requests are measured and reset every 5 s. The plot shows the average number of requests processed per second over each 5 s interval. The moving average is computed with a window size $n = 5$.

As can be observed in the figure, the system clearly has a higher performance when using in-memory queues. This is not unexpected, as any form of computation on a data item should be performed faster when the item is fetched from volatile memory instead of disk. The figure shows that the performance with in-memory queues can be up to 100 times better compared to the system with persistent queues.

4.3. Computationally Demanding Workloads

To simulate applications with a higher computational load, we configure each worker to sleep for between 0.9 and 1.1 s before incriminating and forwarding the integers. The observed throughput is shown in Figure 13 for both persistent queues and in-memory queues.

Contrary to our previous observations, persistent queues performs better than in-memory queues when jobs are more computationally demanding. The performance also appears more stable, ranging mostly between 0.6 and 1 request per second for persistent queues and between 0.4 and 1 requests per second for in-memory queues.

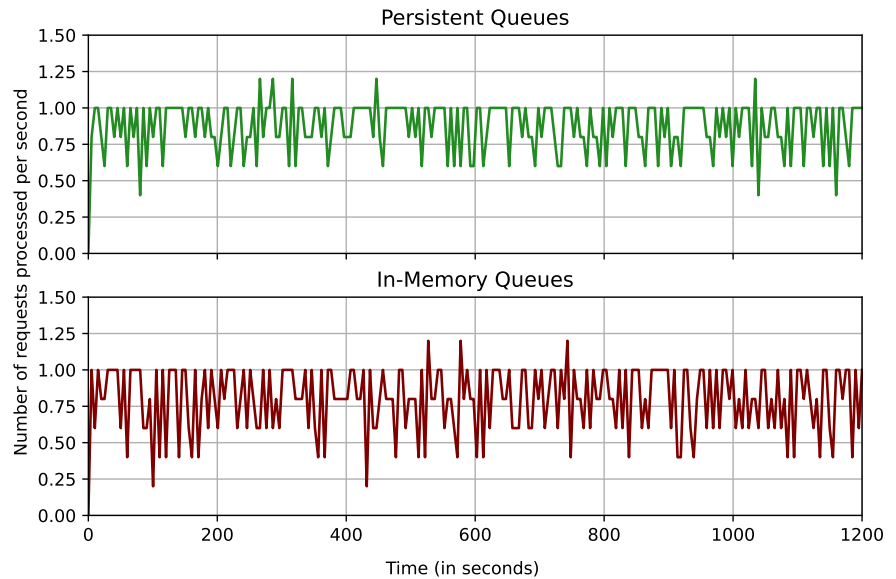


Figure 13. Results obtained when measuring the number of requests received where worker agents in the pipeline sleep for one second after processing an item. Measurements are carried out with persistent queues (top) and in-memory queues (bottom). The number of requests received is measured and reset every 5 s. The plot shows number of requests per second over each 5 s interval.

The results illustrate that the overhead created when writing the queues to file continuously becomes negligible with regards to the overall performance. Despite this, the queues for these measurements only contain integers. With such small values, the time spent writing to file will be significantly reduced compared to, for instance, images. It is therefore important to consider the size of the data when working out a system configuration with optimal performance for carrying out the desired task.

4.4. Failure Experiment

The purpose of the failure experiment is to gain insight into how the performance is affected when working agents are being regularly shut down. We also investigate to what degree the system manages to remain stable when agents are being shut down and recovered.

Figure 14 illustrates the different performance results obtained when stress testing the system while simulating failures. In our failure simulation experiments, a Killer process is instructed to pick one random worker agent or initial agent at 15 s intervals and terminate that process. The Killer runs for the entire duration of the experiment when deployed.

Measurements with and without the Killer process are similar both in terms of the series shape and the performance. When the Killer is deployed, the measurements do, however, become slightly worse. Despite the agents being terminated every 15 s, the performance still remains relatively similar in terms of stability. Agents being terminated every 15 s and the number of requests received being measured every 5 s means that the performance should decrease in every third measurement. This could explain the lack of visible dips in performance.

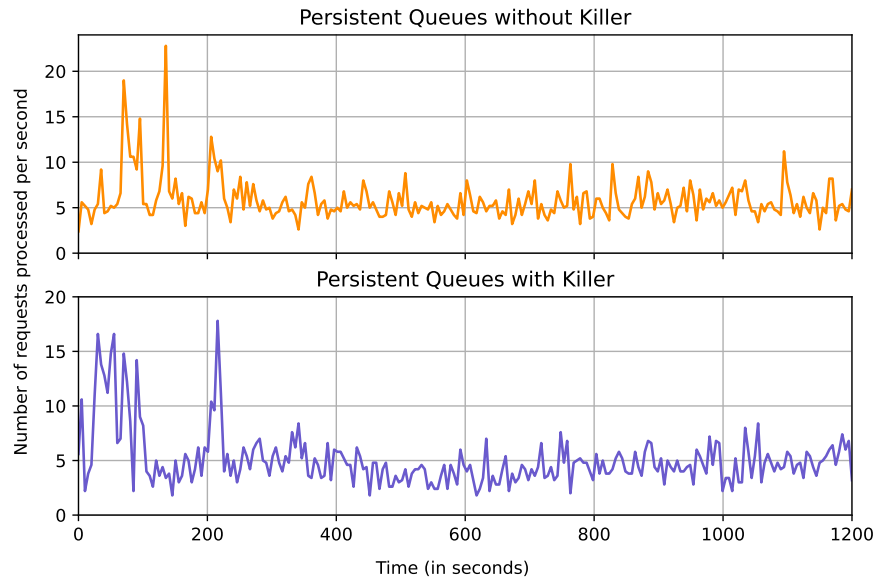


Figure 14. Results obtained when measuring the number of requests received during stress-testing over the course 1200 s without (top) and with (bottom) the killer deployed. The number of requests received is measured and reset every 5 s. The plot shows the average number of requests per second over each 5 s interval.

Table 2 summarizes the observation for the various experiments. These measurements confirm the overhead created when using persistent queues becomes negligible when the workload is increased. Not only are the results more stable for persistent queues (standard deviation 0.18 versus 0.22), it also performs better overall with an average of 0.88 requests processed per second compared to 0.80 s for the in-memory queue. The performance does decrease slightly, from 5.82 requests per second during the stress test to 5.08 requests per second during the same test with the Killer component deployed. The stability of the system also seems to decrease slightly, as the standard deviation changes from 2.29 to 2.58 requests per second when the Killer component is deployed.

Table 2. A summary derived from the continuous performance experiments explained previously. The table contains the minimum, maximum, mean, median, and standard deviation of all requests per second measured for each benchmark.

Queue Type Used	Min Value	Mean Value	Standard Deviation	Median Value	Max Value
Persistent Queue (No sleep)	2.40	5.82	2.29	5.40	22.80
In-memory Queue (No sleep)	551.40	636.89	68.67	612.30	878.60
Persistent Queue (Kill every 15 s)	1.8	5.08	2.58	4.6	17.8
Persistent Queue (Sleep 0.9–1.1 s)	0	0.88	0.18	1	1.2
In-memory Queue (Sleep 0.9–1.1 s)	0	0.80	0.22	0.8	1.2

It is important to note that despite the performance gain from using in-memory queues, the potential consequence of this is that the system remains unable to properly recover from faults if any component shuts down during the runtime. A local controller may resume an agent that crashes, enabling it to continue working. When using in-memory queues, the data stored will, however, be lost if the agent itself is to crash. This requires a complex recovery routine, where the agent initializing the pipeline has to re-retrieve the specific items that have been lost during the crash and propagate them through the entire pipeline

all over again. In the case where the system operates in a trusted environment where the runtime does not last long or is communication intensive, it could benefit the user to use in-memory queues in favor of persistent queues.

The observations in Figure 14 show that Áika provides a stable performance despite worker agents being terminated. The local controllers are constantly monitoring the agents and could therefore explain why the system manages to remain stable. Despite this, the performance of the system is still affected, and having processes killed between small intervals could be fatal for the performance of the system.

4.5. Distributed Word Counter

To evaluate if Áika can handle simple analytical tasks that do not necessarily involve highly complex computations, we deploy a distributed application with a computation pipeline that follows a MapReduce pattern. While MapReduce-computations are often conceptually simple, specialized frameworks for expressing these problems are needed to orchestrate data distribution, parallelization, and failure handling [28]. To implement a MapReduce-like problem on Áika, we deploy a distributed word-counting application. The application counts the words contained in the input text file. The input dataset is generated by sampling random words from the list of the 1000 most common words in the English language according to Education First [29]. The purpose of this experiment is to show that Áika can be configured to express and perform general MapReduce-like operations. Three types of agents are used:

- Split worker divides into distinct and independent units.
- Map workers tokenize input text and iterate counting every word.
- The counted values are stored in a simple Python dictionary.
- Reduce workers combine input dictionaries from the mappers into a total dictionary.

We configure Áika to use a single splitter to initialize the word counting, between 1 and 16 number of mappers, and a single reducer that combines the results from all the mappers. The system is configured with static load balancing, where each map worker fetches one job from the split workers queue, respectively. This also means that each worker performs their work in one single iteration. The experiment is repeated 15 times so that the stability of the system can be measured as well.

The results obtained from running word counting on a 100 MB and a 300 MB dataset can be found in Figure 15. The figure indicates that the system is able to scale well in terms of handling embarrassingly parallel algorithms, due to the slope having an expected concave shape. Despite this, the slope starts to flatten at around 8–9 s. One reason for this is that creating more map workers is more time consuming for the system, as the workers are instantiated by local controllers, which are further instantiated from a single cluster controller. In addition to this, the use of a single reducer could lead to a minor bottleneck, since it is responsible for combining results from all map workers.

As the micro-benchmarks shows, the cluster controller spends approximately 0.30 s starting up a single local controller. Afterwards, it spends approximately 0.15 s extra for each additional local controller. The local controller, on the other hand, spends approximately between 0.02 and 0.04 s to start a single agent. This in total makes up over 1 s of overhead when the number of map workers is six or more, which partly explains the overhead observed in the graph. Furthermore, it is necessary to take into account communication overhead, file reading and writing (due to the persistent queues), and the startup wait time for each agent. A part of the overhead could also be explained by the fact that each mapper loads their entire partition into memory before mapping. When the size of the dataset increases, the overhead may therefore also increase.

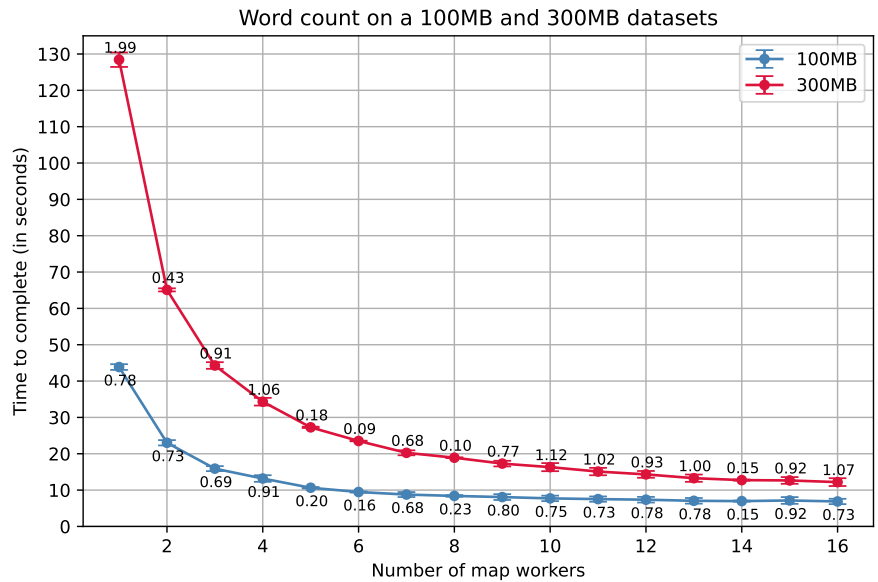


Figure 15. Results obtained from counting the words in datasets with sizes of 100 MB and 300 MB. Standard deviation is shown as an error bar, and the corresponding standard deviation value is displayed above each data point.

4.6. Distributed Deep Feature Extraction

The purpose of the Distributed Deep Feature Extraction experiment is to evaluate Áika's ability to extract features from image data with multiple deep learning models, using two approaches. Another purpose of the experiment is to investigate the system's ability to scale with these two approaches.

For these experiments, we use the STL-10 image dataset to perform the feature extractions [30]. The STL-10 dataset is inspired by the CIFAR-10 dataset [31], although there are some differences between them, such as the images having a higher resolution (96×96 instead of 28×28). The feature extractions are performed on the entire dataset of 113,000 images in batches, with 500 images per batch. The use of batches could give an indication of how the system performs with larger scaled images.

The system extracts features from the data with the use of three pre-trained Keras [32] models. We evaluate two different approaches:

1. All three models are loaded into N workers that perform feature extraction on all three models, sequentially.
2. The three models are distributed among three workers, such that the feature extraction process can be performed in parallel (This experiment was performed on CPUs instead of GPUs due to compatibility issues).

The sequential feature extraction graph consists of an initial agent that divides work among a set of workers. Each worker extracts features from the given data with three feature extraction models, sequentially. The distributed feature extraction graph replaces the workers with a sub-graph where each feature extraction model is put on a respective worker. This enables the feature extraction process to be performed in parallel.

Feature extractions on all 113,000 images take approximately 3336 s when configured to run on a single machine. This experiment was repeated three times. The computation time for the same task run on Áika in batches of 500 images using the two approaches described above can be found in Figure 16. Both approaches seem to scale at a similar rate in terms of number of sub-graphs. The distributed feature extraction approach is clearly more efficient, but also requires more workers.

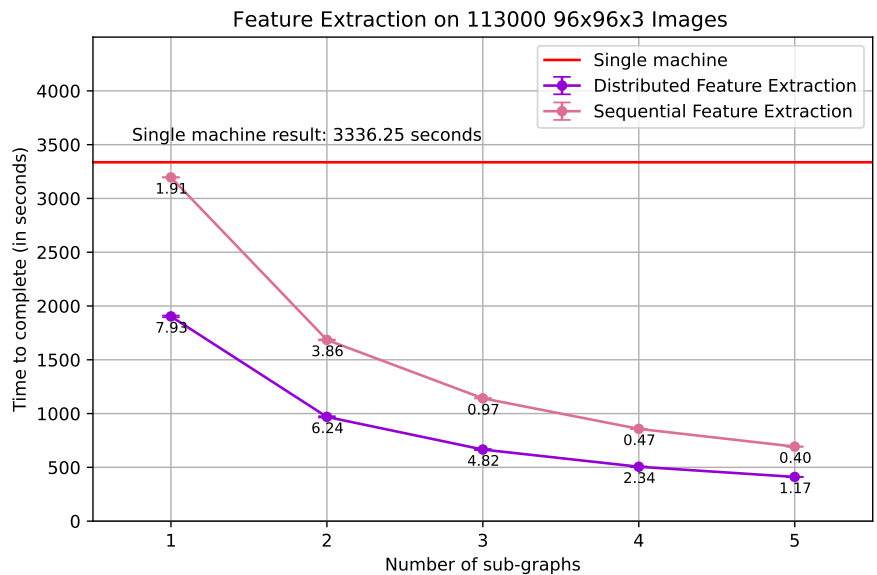


Figure 16. Results obtained from performing feature extraction with pre-trained VGG-16, DenseNet-121, and ResNet-50 models, where the models are either distributed among three workers (Distributed Feature Extraction), or put in a sequence on a single worker (Sequential Feature Extraction). The images have been processed with a batch size of 500. The standard deviation is shown as an error bar and value at each data point.

One interesting finding is that the sequential feature extraction approach performs better compared to the single machine benchmark, even when using a single worker. At the data point with one sub-graph, the sequential feature extraction experiment has the same work distribution as the single machine benchmark, while expected to have additional overhead that the sequential feature extraction approach receives due to latency. The reason could also be due to varying processing frequency or memory management in Python.

These experiments show that Áika's method of distributing the feature-extraction models across several workers is beneficial for performance. It is, however, important to note that the distribution sub-graph requires three workers instead of a single one. The results from running the experiment with three single workers instead of a sub-graph of three workers proves to be more beneficial in terms of pure performance. The reason for this is that the VGG-16 model spends more time extracting features compared to the other two models, which makes the system scale less evenly. However, if the system was to further utilize a classifier that blocks until all features extracted for the batch of data have been received, the distributed approach may prove to be more beneficial for rapidly classifying features.

5. Conclusions

Monitoring of resources in world oceans and the Arctic is a technological challenge that requires domain-specific systems. The edge environments that offshore fishing vessels traverse are characterized by their lack of stable connections, high latency, low bandwidth, and difficulty of manual intervention. This paper describes Áika: a prototype system for executing distributed AI applications in these domain-specific edge environments. We investigate how a system supporting AI inference in these environments can be built to support a wide range of computational graphs through a DAG computation model, while making the system tolerant to failures.

Áika provides application developers with generalized building blocks that can be used to construct complex distributed computational tasks with robustness and privacy

properties. Through a hierarchical design, we utilize local controllers on physical nodes to perform quick recovery when failure occurs. A cluster controller is used to further invoke node recovery, where agents from a failed physical node are moved to a replica. The cluster controller is replicated in a chain to avoid having a single point of failure, and communicates with a remote monitor that logs failures and classifies the system state when sufficient bandwidth is available.

Our experimental evaluations demonstrate that Áika has a stable throughput despite potential agent crashes. For data-intensive tasks, we show that persistent queues can be beneficial compared to in-memory queues, in cases where the workload on an item exceeds the time spent transporting the item. We demonstrate how Áika can be used to create DAGs of varying complexity with load balancing, and distribute work among agents to optimize performance. We implement two experiments with computational tasks relevant for the targeted domain, namely a MapReduce task and a deep-learning-based feature extraction task. The results from these experiments demonstrate that Áika is scalable and supports different computational graph designs that can be used in the domain of fishery monitoring and surveillance.

Author Contributions: Conceptualization, J.A.A., A.B.O. and D.J.; methodology, J.A.A. and D.J.; software, J.A.A. and A.B.O.; validation, J.A.A., investigation, J.A.A., A.B.O. and D.J.; data curation, J.A.A.; writing—original draft preparation, J.A.A. and M.A.R.; writing—review and editing, J.A.A., A.B.O., T.-A.S.N., H.D.J., M.A.R., P.H. and D.J.; visualization, J.A.A.; supervision, H.D.J. and D.J.; project administration, H.D.J., M.A.R., P.H. and D.J.; funding acquisition, H.D.J. and D.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially funded by the Research Council of Norway project numbers 274451 and 263248, and Lab Nord-Norge (“Samfunnsløftet”).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Stoica, I.; Song, D.; Popa, R.A.; Patterson, D.A.; Mahoney, M.W.; Katz, R.H.; Joseph, A.D.; Jordan, M.; Hellerstein, J.M.; Gonzalez, J.; et al. *A Berkeley View of Systems Challenges for AI*; Technical Report UCB/EECS-2017-159; EECS Department, University of California: Berkeley, CA, USA, 2017.
2. Carcillo, F.; Le Borgne, Y.A.; Caelen, O.; Kessaci, Y.; Oblé, F.; Bontempi, G. Combining unsupervised and supervised learning in credit card fraud detection. *Inf. Sci.* **2021**, *557*, 317–331. [\[CrossRef\]](#)
3. Deng, S.; Zhao, H.; Yin, J.; Dustdar, S.; Zomaya, A.Y. Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence. *IEEE Internet Things J.* **2020**, *7*, 7457–7469. [\[CrossRef\]](#)
4. Liu, J.; Sheng, M.; Liu, L.; Li, J. Network Densification in 5G: From the Short-Range Communications Perspective. *IEEE Commun. Mag.* **2017**, *55*, 96–102. [\[CrossRef\]](#)
5. European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Off. J. Eur. Union* **2016**, *119*, 1–88.
6. Costello, C.; Cao, L.; Gelcich, S.; Cisneros-Mata, M.A.; Free, C.M.; Froehlich, H.E.; Golden, C.D.; Ishimura, G.; Maier, J.; Macadam-Somer, I.; et al. The future of food from the sea. *Nature* **2020**, *588*, 95. [\[CrossRef\]](#) [\[PubMed\]](#)
7. UNODC. *Fisheries Crime: Transnational Organized Criminal Activities in the Context of the Fisheries Sector*; UNODC: Vienna, Austria, 2016.
8. Ministry of Trade, Industry and Fisheries. *Framtidens Fiskerikontroll*; NOU 21:19; Ministry of Trade, Industry and Fisheries: Oslo, Norway, 2019.
9. Márcia Bizzotto. Fishing Rules: Compulsory CCTV for Certain Vessels to Counter Infractions. European Parliament Press Release. Available online: <https://www.europarl.europa.eu/news/en/press-room/20210304IPR99227/fishing-rules-compulsory-cctv-for-certain-vessels-to-counter-infractions> (accessed on 8 August 2021).

10. Ingilæ, Ø. Fiskere Settes Under Overvåkning. Kyst og Fjord. Available online: <https://www.kystogfjord.no/nyheter/forsiden/Fiskere-settes-under-overvaakning> (accessed on 8 August 2021).
11. Martinussen, T.M. Danske Fiskere Samler Seg Mot Kamera-overvåkning i Fiskeriene. Fiskeribladet. Available online: <https://www.fiskeribladet.no/nyheter/danske-fiskere-samler-seg-mot-kamera-overvakning-i-fiskeriene/2-1-839478> (accessed on 8 August 2021).
12. Nordmo, T.A.S.; Ovesen, A.B.; Johansen, H.D.; Riegler, M.A.; Halvorsen, P.; Johansen, D. Dutkat: A Multimedia System for Catching Illegal Catchers in a Privacy-Preserving Manner. In Proceedings of the 2021 Workshop on Intelligent Cross-Data Analysis and Retrieval, Taipei, Taiwan, 21 August 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 57–61. [\[CrossRef\]](#)
13. van Helmond, A.T.; Mortensen, L.O.; Plet-Hansen, K.S.; Ulrich, C.; Needle, C.L.; Oesterwind, D.; Kindt-Larsen, L.; Catchpole, T.; Mangi, S.; Zimmermann, C. Electronic monitoring in fisheries: Lessons from global experiences and future opportunities. *Fish Fish.* **2020**, *21*, 162–189. [\[CrossRef\]](#)
14. Ananthanarayanan, G.; Bahl, P.; Bodík, P.; Chintalapudi, K.; Philipose, M.; Ravindranath, L.; Sinha, S. Real-time video analytics: The killer app for edge computing. *Computer* **2017**, *50*, 58–67. [\[CrossRef\]](#)
15. Fitwi, A.; Chen, Y.; Zhu, S.; Blasch, E.; Chen, G. Privacy-preserving surveillance as an edge service based on lightweight video protection schemes using face de-identification and window masking. *Electronics* **2021**, *10*, 236. [\[CrossRef\]](#)
16. Dsouza, S.; Bahl, V.; Ao, L.; Cox, L.P. Amadeus: Scalable, Privacy-Preserving Live Video Analytics. *arXiv* **2020**, arXiv:2011.05163.
17. Ovesen, A.B.; Nordmo, T.A.S.; Johansen, H.D.; Riegler, M.A.; Halvorsen, P.; Johansen, D. File System Support for Privacy-Preserving Analysis and Forensics in Low-Bandwidth Edge Environments. *Information* **2021**, *12*, 430. [\[CrossRef\]](#)
18. Harchol, Y.; Mushtaq, A.; McCauley, J.; Panda, A.; Shenker, S. CESSNA: Resilient Edge-Computing. In Proceedings of the 2018 Workshop on Mobile Edge Communications, Budapest, Hungary, 20 August 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 1–6. [\[CrossRef\]](#)
19. Wang, C.; Gill, C.; Lu, C. FRAME: Fault Tolerant and Real-Time Messaging for Edge Computing. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–9 July 2019; pp. 976–985. [\[CrossRef\]](#)
20. Harrison, T.H.; Levine, D.L.; Schmidt, D.C. The Design and Performance of a Real-Time CORBA Event Service. *SIGPLAN Not.* **1997**, *32*, 184–200. [\[CrossRef\]](#)
21. Johansen, D.; Marzullo, K.; Schneider, F.; Jacobsen, K.; Zagorodnov, D. NAP: Practical fault-tolerance for itinerant computations. In Proceedings of the 19th IEEE International Conference on Distributed Computing Systems (Cat. No.99CB37003), Austin, TX, USA, 5 June 1999; pp. 180–189. [\[CrossRef\]](#)
22. Schneider, F.B.; Gries, D.; Schlichting, R.D. Fault-tolerant broadcasts. *Sci. Comput. Program.* **1984**, *4*, 1–15. [\[CrossRef\]](#)
23. Leners, J.; Wu, H.; Hung, W.L.; Aguilera, M.; Walfish, M. Detecting failures in distributed systems with the Falcon spy network. In Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles, Cascais, Portugal, 23–26 October 2011; pp. 279–294.
24. Isard, M.; Budiu, M.; Yu, Y.; Birrell, A.; Fetterly, D. Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks. *SIGOPS Oper. Syst. Rev.* **2007**, *41*, 59–72. [\[CrossRef\]](#)
25. Dean, J.; Ghemawat, S. MapReduce: Simplified Data Processing on Large Clusters. In Proceedings of the OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, USA, 6–8 December 2004; pp. 137–150.
26. Valvåg, S.V.; Johansen, D.; Kvalnes, Å. Cogset: A high performance MapReduce engine. *Concurr. Comput. Pract. Exp.* **2013**, *25*, 2–23. [\[CrossRef\]](#)
27. Welsh, M.; Culler, D.; Brewer, E. SEDA: An Architecture for Well-Conditioned, Scalable Internet Services. *SIGOPS Oper. Syst. Rev.* **2001**, *35*, 230–243. [\[CrossRef\]](#)
28. Dean, J.; Ghemawat, S. MapReduce: Simplified data processing on large clusters. *Commun. ACM* **2008**, *51*, 107–113. [\[CrossRef\]](#)
29. 1000 Most Common Words in English. 2011. Available online: <https://www.ef.com/wwen/english-resources/english-vocabulary/top-1000-words/> (accessed on 11 November 2021).
30. Coates, A.; Ng, A.; Lee, H. An analysis of single-layer networks in unsupervised feature learning. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 215–223.
31. Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*; Citeseer: Princeton, NJ, USA, 2009.
32. Keras: Deep Learning for Humans. 2015. Available online: <https://github.com/fchollet/keras> (accessed on 28 April 2022).

A.8 Paper VII: FishAI: Sustainable Commercial Fishing

Authors: T.S. Nordmo, O. Kvalsvik, S.O. Kvalsund, B. Hansen, D. Johansen, H.D. Johansen and M.A. Riegler

Abstract: FishAI: Sustainable Commercial Fishing is the second challenge at the Nordic AI Meet following the successful MedAI, which had a focus on medical image segmentation and transparency in machine learning (ML)-based systems. FishAI focuses on a new domain, namely, commercial fishing and how to make it more sustainable with the help of machine learning. A range of public available datasets is used to tackle three specific tasks. The first one is to predict fishing coordinates to optimize catching of specific fish, the second one is to create a report that can be used by experienced fishermen, and the third task is to make a sustainable fishing plan that provides a route for a week. The second and third task require to some extent explainable and interpretable models that can provide explanations. A development dataset is provided and all methods will be tested on a concealed test dataset and assessed by an expert jury.

Author contributions (initials): **Conceptualisation:** T-A.S.N., M.A.R.;

Data collection: T-A.S.N., **Methods, data analysis and interpretation:** T-A.S.N., M.A.R., **Drafting:** T-A.S.N., O.K., S.O.K., B.H., S.A.H., H.D.J, P.H., M.A.R., D.J., **Critical revision:** T-A.S.N., O.K., S.O.K., B.H., S.A.H., H.D.J, P.H., M.A.R., D.J.

Published: Nordic Machine Intelligence, 2022

Thesis objectives: Sub-objective 3



FishAI: Sustainable Commercial Fishing Challenge

Tor-Arne Schmidt Nordmo¹, Ove Kvalsvik², Svein Ove Kvalsund², Birte Hansen³, Pål Halvorsen⁴, Steven A. Hicks⁴, Dag Johansen¹, Håvard Dagenborg Johansen¹, Michael A. Riegler^{1,4}

1. UiT The Arctic University of Norway, Norway
2. Vekstlandet, Norway
3. NORA—Norwegian Artificial Intelligence Research Consortium, Norway
4. SimulaMet, Norway

Abstract

FishAI: Sustainable Commercial Fishing is the second challenge at the *Nordic AI Meet* following the successful *MedAI*, which had a focus on medical image segmentation and transparency in machine learning (ML)-based systems. *FishAI* focuses on a new domain, namely, commercial fishing and how to make it more sustainable with the help of machine learning. A range of public available datasets is used to tackle three specific tasks. The first one is to predict fishing coordinates to optimize catching of specific fish, the second one is to create a report that can be used by experienced fishermen, and the third task is to make a sustainable fishing plan that provides a route for a week. The second and third task require to some extent explainable and interpretable models that can provide explanations. A development dataset is provided and all methods will be tested on a concealed test dataset and assessed by an expert jury.

Keywords: artificial intelligence; machine learning; fishing; automatic reporting

Introduction

With a fishing zone spanning 2.1 million square meters, Norway is considered Europe's largest fishing and aquaculture nation. Every year, commercial vessels catch fish with a total value of around 20 billion NOK from the Norwegian fishing zone.

The overall migration patterns of the major fish species are relatively predictable and common knowledge. A fisherman knows, for example, that the mackerel season starts in mid-September and plans accordingly. On a daily basis, however, fish populations can move over large distances, and with the main decision-making tool being the captain's experience and intuition, boats often search for days or even weeks before making a catch. The

number of boats is not negligible; there are currently around 1,100 Norwegian vessels over 11 meters involved. It is estimated that each vessel burns around 2,000–2,500 liters of fuel per day, which translates to approximately 5 million kg CO₂-equivalents per day.

Although the fishing fleet over time has shown an impressive ability to renew itself, the core operation of searching and catching fish clearly has room for improvements in a sustainability context. Specifically, a more energy-efficient commercial fishing practice and operation should be a goal. In other words, there are great environmental benefits and opportunities in optimizing commercial fishing activities by reducing unnecessary transport distances. With the recent release of catch data made available by the Norwegian Directorate of Fisheries, a significant potential of applying artificial intelligence opened up, which we want to explore with this challenge.

Dataset Details

We provide the participants with a collection of four publicly available datasets: a catch note dataset, a temperature dataset, a salinity dataset, and a moon phase dataset. All datasets can be used in all tasks and can be downloaded via: <https://tinyurl.com/54w5bvxa>. For the GPS coordinates predictions, the catch notes dataset is the ground truth. Participants are also encouraged to use other data sources if they are public available. In the following we provide a more detailed description of each dataset and what the participants can expect for the evaluation of their results.

Catch Notes Dataset

The catch notes data contains catch notes collected by the Norwegian Fishing Directorate from 2000 to today for vessels larger than 15 meters. The notes consist

of information about the catch that is manually logged during landing, e.g., when it was caught, where it was caught, what equipment was used, and the species distribution of the catch. There are approximately 130 data fields and around one million notes each year. Fields that might be of interest include information about where they fished ("Hovedområde", "Lon", "Lat", etc.) and information about fish caught and how they were caught ("Art - FDIR", "Bruttovekt", "Redskap").

The catch notes are in Norwegian. For most of the variables this is not relevant. In case it might be relevant participants can translate the data as part of their data preparation pipeline. The dataset from each year can be found, along with documentation and metadata, at: <https://www.fiskeridir.no/Tall-og-analyse/AApne-data/Fangstdata-seddel-koblet-med-fartoyedata>.

Temperature, Salinity, and Moon Phase Datasets

The temperature, salinity, and moon phase datasets are meant to be auxiliary datasets that might give more information regarding fish migration. Both the temperature and salinity datasets are presented in netCDF4 formats. Therefore, we recommend to use the `netCDF4` Python module for extracting the data.

Temperature Data

Sea surface temperature (SST) from 1981 to present has been collected by National Oceanic and Atmospheric Administration (US). It contains daily estimates of SST globally. The data was collected from satellite observations, and consists of daily data at 0.25 degree latitude \times 0.25 degree longitude resolution [1]. We have included the subset of data from 2000 to present day. The dataset is published at: <https://www.psl.noaa.gov/data/gridded/data.noaa.oisst.v2.highres.html>.

Salinity Data

Monthly averages of salinity data from 2015 to present day is provided from the SMAP Salinity V4 dataset [2]. Salinity (in combination with temperature) affects the growth rate of microalgae. This can potentially affect the migration patterns of fish. Eight-day running averages are also possible to obtain if needed (<https://salinity.oceansciences.org/data-smap-v4.htm>).

Moon Phase Data

The moon phase data consists of dates and exact times of full moon from 1900 to 2050. Lunar phases affect the migration and behaviour of fish due to water levels changing. Therefore, it is potentially possible to use this data source for modeling of the movement of fish. The dataset is published at <https://www.kaggle.com/datasets/lsind18/full-moon-calendar-1900-2050>.

Task Descriptions

We present three subtasks: the catching optimization task, the reporting task, and the planning task. Each task targets different aspects of the data. The participants are encouraged to submit solutions for all three subtasks, but can also just focus on specific tasks.

Task 1: Catching Optimization and Prediction Task

Build a model that can predict which coordinates a vessel should prioritize in order to maximize the likelihood of catching a type of fish of your choosing (haddock or mackerel is most valuable for the industry partners). The prediction can be based on historical data.

Task 2: Report Generation Task

Create a report of your analysis that can be read by experienced fishermen; an user-friendly visualization that a captain can read to make an assessment of where the vessel should search for fish the next day.

Task 3: Sustainability Fishing Plan Task

Make a Sustainability Fishing Plan; a weekly plan that suggests the routes the fisherman should follow to optimize fish caught and fuel consumption.

The aim is to build a tool that will help fisherman make decisions about where to search for fish in the immediate future.

This could include features such as a heatmap indicating the largest likelihood for catch of a specific type of fish, recommendations based on predicted catch volume relative to distance from current location, or similar.

Evaluation Methodology

Task one will be evaluated using an unseen test set. We will use standard metrics such as precision, F1-score, accuracy, mean absolute error, etc., to evaluate the performance of the methods. Tasks two and three will be evaluated by an expert team consisting of experienced fishermen and data scientists which will provide an overall ranking of the submitted report and fishing plan. Submissions to task two and three are evaluated using a qualitative approach compared to task one. The quality of the report and plan are measured by attributes like readability, presentation, and usefulness. Each team will receive a report from the expert team on their performance. There will be one first place and one second place based on a combination of the evaluations gathered from each of the three sub-tasks.

Summary

This article presents the *FishAI: Sustainable Commercial Fishing* challenge held at the 2022 Nordic AI Meet. The challenge aims to open up for research and innovation in the commercial fishing domain to increase sustainability. FishAI is providing three subtasks that range from catch optimization to automatic report generation. We hope that this challenge inspires established and young researchers and people interested in innovation to explore

an important and interesting topic contributing to more sustainable commercial fishing activities.

Conflict of interest

There is no conflict of interest.

References

1. Reynolds RW, Smith TM, Liu C, Chelton DB, Casey KS, and Schlax MG. Daily High-Resolution-Blended Analyses for Sea Surface Temperature. *Journal of Climate* 2007; 20:5473–96. DOI: 10.1175/2007JCLI1824.1. Available from: <https://journals.ametsoc.org/view/journals/clim/20/22/2007jcli1824.1.xml>
2. Meissner T, Wentz FJ, and Le Vine DM. The Salinity Retrieval Algorithms for the NASA Aquarius Version 5 and SMAP Version 3 Releases. *Remote Sensing* 2018; 10. DOI: 10.3390/rs10071121. Available from: <https://www.mdpi.com/2072-4292/10/7/1121>

A.9 Paper VIII: NjordVid: A Fishing Trawler Video Analytics Task

Authors: T.S. Nordmo, A.B. Ovesen, H.D. Johansen, D. Johansen and M.A. Riegler

Abstract: Fishing is one of the most important food sources globally. Commercial fishing can potentially be more efficient, precise, and accountable, and if artificial intelligence should be one of the remedies for improvement, one need a better understanding of inner details and what processes are happening on a fishing trawler. The Njord task aims to encourage researchers to tackle this challenge in addition to preserving the privacy of the people working on these boats. The participants are asked to detect events in videos taken on the fishing trawler and to enhance privacy for the people visible in the videos.

Author contributions (initials): **Conceptualisation:** T-A.S.N., M.A.R.; **Data collection:** T-A.S.N., **Methods, data analysis and interpretation:** T-A.S.N., **Drafting:** T-A.S.N., A.B.O., H.D.J, M.A.R., D.J., **Critical revision:** T-A.S.N., A.B.O., H.D.J, M.A.R., D.J.

Published: Multimedia Benchmark Workshop, 2022

Thesis objectives: Sub-objective 3

NjordVid: A Fishing Trawler Video Analytics Task

Tor-Arne S. Nordmo^{1,*}, Aril B. Ovesen¹, Håvard D. Johansen¹, Dag Johansen¹ and Michael A. Riegler^{1,2}

¹UiT: The Arctic University of Norway, Norway

²SimulaMet, Norway

Abstract

Fishing is one of the most important food sources globally. Commercial fishing can potentially be more efficient, precise, and accountable, and if artificial intelligence should be one of the remedies for improvement, one needs a better understanding of inner details and what processes are happening on a fishing trawler. The NjordVid task aims to encourage researchers to tackle this challenge in addition to preserving the privacy of the people working on these boats. The participants are asked to detect events in videos taken on the fishing trawler and to enhance privacy for the people visible in the videos.

1. Introduction

Surveillance on board fishing vessels has been argued to be a necessity for sustainable fishing practices and for our ability to fight fraud in the fishery industry [1, 2]. Fishing vessels are secluded environments where a small group of people work and live together in a constrained space, often for several weeks at a time. Introducing video surveillance in such environments, in particular combined with machine learning, has raised new privacy and data protection aspects that need to be addressed. This task provides a unique opportunity to gain insight into the inner workings of a commercial fishing vessel while at sea, its part in the food production pipeline, and the living and working conditions of the crew onboard. Understanding these elements are essential for the development and usage of practical automated surveillance systems.

With this competition we hope to achieve a better understanding of the processes that happen on a fishing trawler and in addition we want to encourage the community to work on this important topic.

2. Dataset

The Njord dataset [3] contains surveillance videos from the Hermes fishing trawler that were live-streamed online in 2019 as slow-TV entertainment. The dataset consists of 71 videos that have been annotated so far and 127 videos that are not annotated. The videos have a resolution of 1, 280 × 720 and run at 25 frames-per-second. The videos have varying lighting conditions with complex, moving backgrounds due to the trawler being at sea. The videos consist of eight different fixed-camera scenes plus a view with a manually-operated camera for showing particularly interesting events, such as whale observations and other boats. The cameras are changed between on a fixed schedule but can also be manually changed by the captain. This sometimes results in scenes having varying durations. There are overlays that sometimes appear on-screen. These show general information about what is being caught, information about the


MediaEval'22: Multimedia Evaluation Workshop, January 13–15, 2023, Bergen, Norway and Online

*Corresponding author.

✉ tor-arne.s.nordmo@uit.no (T. S. Nordmo)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

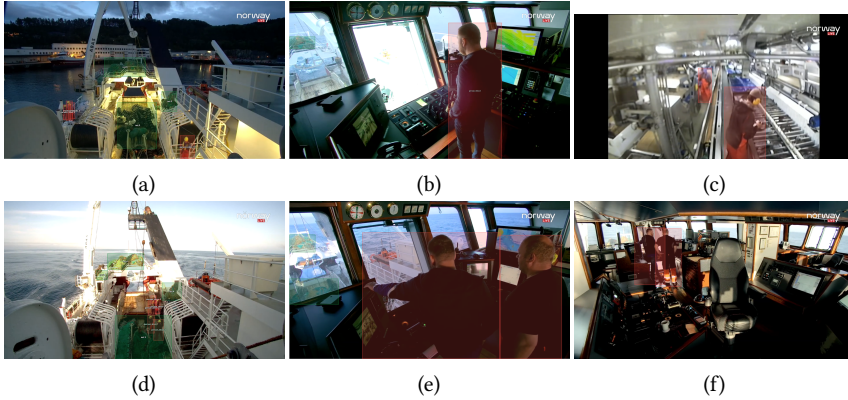


Figure 1: Sample frames from the dataset with overlaying bounding box annotations.

vessel in general, and statistics related to the catch. They also sometimes show a map overlay with the current location of the trawler along with its speed and orientation.

For each video, we have labeled bounding boxes around people, other boats, nets, and fish. The temporal annotations consist of when scene changes occur, when overlays are turned on and off, when Events of Interest (EoI) occur, and when the intro plays. We also have labels that denote whether it is daytime or nighttime, and, due to the videos being from a live-stream, labels for parts of the videos that are before the introduction and after the end of the relevant live-stream. The bounding boxes for fish label groups of fish due to the scenes on deck showing fish being far away from the camera. The bounding boxes for the nets both label nets in use and those lying in heaps on deck.

The dataset is organized as follows. The videos directory contains a subdirectory for each annotated video that contains the video in .mp4 format and two annotation files, one file for the bounding box annotations and one file for the timeline annotations. The two annotation files are structured as csv-formatted files using a semi-colon as the delimiter. The bounding box contains one line per bounding box annotation with the following seven values; class, frame number, center x-position, center y-position, the bounding box's width, and the bounding box's height. The width and height have been normalized by dividing each by the video's width and height, respectively. The timeline annotation file contains one line per annotated class and includes the following two values; the class of the frame and the frame number of the corresponding video. The videos directory also contains an unannotated subdirectory containing all videos that have not been annotated yet.

The dataset Njord is publicly available under the CC BY-NC 4.0 International license.

3. Tasks

The NjordVid task consists of two different subtasks, which can be tackled independently depending on your research area of interest. The dataset consists of 198 surveillance videos from a fishing trawler, of which 71 are annotated with bounding boxes and temporal annotations. The goal of the task is to both gain insight into what is happening on fishing vessels and also investigate methods for preserving the privacy of the fishing crew.

Table 1

Baseline experiments for the detection of events and people in the development dataset.

Model	Precision	Recall	mAP_0.5	mAP_0.5:0.95
YOLOv5n	0.698	0.502	0.527	0.265
YOLOv5s	0.732	0.545	0.543	0.271
YOLOv5m	0.697	0.552	0.569	0.277
YOLOv5x	0.621	0.570	0.550	0.264

3.1. Subtask 1

Detection of events on the boat: The participants are asked to detect events on the boat like people moving, fish caught, etc. In addition to simple detection of the events we also ask the participants to provide an interestingness score which relates to how uncommon the event is. The score should be between 0 to 1 where 0 determines a very common event and 1 a very uncommon event.

3.2. Subtask 2

Privacy of onboard personnel: For this task the participants are asked to develop methods to preserve the privacy of the people working on the boat, which includes anything that can identify the person (face, nametags, etc). At the same time the privacy preserving measurements should have as little impact on the analysis as possible.

4. Evaluation

For the evaluation of subtask 1 we will use the standard metrics Precision, Recall, F1 score and Matthew correlation coefficient. The interestingness score provided by the participants will be used to weight the resulting scores.

For subtask 2 we will have a group of manual evaluators checking the privacy aspects on the test dataset (basically is the person still identifiable by a human observer or not). In addition we will calculate some metrics before and after the method was applied. Specifically, we will apply an object detection model and evaluate with classic regression metrics before and after the privacy-preserving method is applied.

5. Baseline Results

In this section we present some baseline results obtained by training a simple object-tracking model using YOLOv5 on the development dataset. Table 1 shows the performance metrics based on the ground truth given in the development dataset and Figure 1 provides some example images with resulting bounding boxes.

6. Discussion and Outlook

The task focuses on the exploration of a completely unknown area where automatic multimedia analysis can have an important impact. We hope that the task will lead to new insights and research questions in addition to inspiring researchers to work on this important topic. For the

future we envision a more complex and multimodal dataset that also contains sensor readings and other additional information.

We particularly thank Hermes staff and owners for relevant discussions, meetings, and for allowing us to annotate and publish the Njord dataset and use it for MediaEval.

References

- [1] Ministry of Trade, Industry and Fisheries, Framtidens fiskerikontroll, NOU 19:21 (2019).
- [2] P. Release, Fishing rules: Compulsory CCTV for certain vessels to counter infractions, 2021. <https://www.europarl.europa.eu/news/en/press-room/20210304IPR99227/fishing-rules-compulsory-cctv-for-certain-vessels-to-counter-infractions>.
- [3] T.-A. S. Nordmo, A. B. Ovesen, B. A. Juliussen, S. A. Hicks, V. Thambawita, H. D. Johansen, P. Halvorsen, M. A. Riegler, D. Johansen, Njord: a fishing trawler dataset, in: Proceedings of the 13th ACM Multimedia Systems Conference, 2022, pp. 197–202.

