Jørgen Hanssen

Expanding Our Knowledge of Maritime Trade with AIS and Explainable AI Systems

Master's thesis in Computer Science
Supervisor: Helge Langseth
July 2023

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Jørgen Hanssen

Expanding Our Knowledge of Maritime Trade with AIS and Explainable AI Systems

Master's thesis in Computer Science
Supervisor: Helge Langseth
July 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Jørgen Hanssen

Expanding Our Knowledge of Maritime Trade with AIS and Explainable AI Systems

Master's thesis, July 2023
Supervisor: Helge Langseth

Artificial Intelligence Group
Department of Computer and Information Science
Faculty of Information Technology, Mathematics and Electrical Engineering

Abstract

Recent developments in AI and machine learning models have culminated in unprecedented achievements, demonstrating decision-making capabilities that rival and, in some cases, surpass established human knowledge. The transformative potential of these systems presents unique opportunities to provide alternative perspectives on complex problems, pushing the boundaries of traditional knowledge acquisition methods.

This thesis employs AIS data, state-of-the-art models, and XAI techniques to shed new light on the intricate relationship between global fleet behavior and various economic mechanisms. The research employs several machine learning models derived from state-of-the-art research and compares their ability to do 14-day forecasts of various financial instruments in the maritime industry with varying degrees of external influence. The instruments tested are the Baltic Dry Index, the Breakwave Dry Bulk Shipping ETF, the Golden Ocean Group Ltd. stock price, and the Frontline Ltd. stock price over a sequence of experiments. Each experiment compares the various models over different feature complexities and look-back periods, and the best-performing models are explained through SHAP feature attribution using various aggregations to explain the contribution of different shipping variables (speed, draft, load, and traffic volume) between the most active ports.

The findings suggest that the best models exhibit an exceptional ability to forecast the instruments specific to the dry bulk segment while showing little proficiency in modeling the tanker segment. Furthermore, the best-performing models appear to exhibit several decision-making principles akin to established knowledge. However, ambiguity in the explanations makes them hard to evaluate, and while the exceptionally competent models do indeed provide explanations that contest established knowledge, it remains undetermined if these are accurate novelties or misleading fallacies - stressing the necessity for further research and analysis.

The thesis also includes a comprehensive appendix with extended explanations of the best models for each instrument intended for maritime analysts and anyone interested in the models' decision-making.

Preface

This thesis, completed in collaboration with Maritime Optima AS, concludes my master's degree in Artificial Intelligence, undertaken at the Department of Computer Science at the Norwegian University of Science and Technology. Spanning from August 2022 to June 2023, this journey has been one of intriguing academic exploration and practical insights.

My sincere appreciation goes to my supervisor, Professor Helge Langseth of NTNU. His invaluable feedback and insights, as well as his dedicated mentorship, significantly enriched the thesis-writing process. I consider myself privileged to have benefited from his academic guidance and mentorship.

In addition, I would like to express profound gratitude to Sven Melsom Ziegler and Kristin Omholt-Jensen from Maritime Optima. Their willingness to devote time, their professional expertise in shipping practices, and their generous assistance in evaluating the results of this thesis have been instrumental. Their availability for discussions and unwavering support throughout this project has been of immense value. It is my hope that the comprehensive set of explanations included in the appendix will prove satisfying to your keen interests and curious spirits.

Moreover, I extend my heartfelt thanks to my friends and colleagues at Maritime Optima, whose camaraderie and intrigue have served as a significant source of inspiration and support throughout this journey. Finally, I am deeply thankful to everyone who has supported me throughout this rigorous yet rewarding academic venture in the realm of Artificial Intelligence.

This master's thesis is a testament to interdisciplinary collaboration in the pursuit of knowledge and transparency; I am humbled by the experience and thankful for the opportunity.

Jørgen Hanssen
Trondheim, July 4, 2023

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence. 1, 2, 6, 9, 16, 26–28, 32, 33, 35, 39, 103, 107, 109

AIS Automatic Identification System. viii, 2–6, 9–11, 32, 35–39, 42–46, 51, 54–64, 68, 73–75, 79, 80, 87, 89, 94, 96, 101, 103–109

ARIMA Autoregressive Integrated Moving Average. 37, 74, 76, 82, 89, 96, 102, 104, 105

CCC Concordance Correlation Coefficient. 79, 80, 82, 87, 89, 94, 96, 101, 102

CNN Convolutional Neural Network. vii–ix, 23, 32, 39, 40, 42–44, 46, 48, 53, 62, 67, 68, 76, 82, 83, 89, 90, 96, 97, 103, 108

DNN Deep Neural Network. viii, 18, 46, 55, 66, 70, 76, 96

DWT Deadweight Tonnage. 60, 80, 101

ETF Exchange-Traded Fund. viii, ix, 61, 73, 87–91, 93, 103, 104

IG Integrated Gradients. 47, 70

IMO IMO Number (ship identification number). 11, 60

LOCODE UN Trade and Transport Location Code. 11, 106

LSTM Long Short-Term Memory. vii, viii, 18, 20–22, 37–45, 49, 53–55, 66–68, 76

ML Machine Learning. 2, 6, 16–19, 28, 33, 35, 42, 45, 52, 61, 65, 74, 78

MMSI Maritime Mobile Service Identity. 11, 60

MSE Mean Squared Error. 17, 77, 79, 82, 89, 96, 101, 102, 104, 105

RMSE Root-Mean-Square Error. 37

RNN Recurrent Neural Networks. 19–21, 32, 37, 39, 40, 43, 44, 46, 48, 49, 55, 66–68, 76

S-AIS Satellite-based AIS. 11

SHAP SHapley Additive exPlanations. 48, 49, 51–54, 56, 65, 69–71, 79, 99, 104, 107, 108

XAI Explainable Artificial Intelligence. 1–7, 27, 28, 35, 45, 46, 50, 51, 56, 57, 73, 74, 106, 108, 109

Chapter 1

Introduction

This chapter introduces the concept of employing AI systems to advance scientific insight and elaborates on the motivation behind its application in the field of maritime analysis to yield further insight into the interplay between the world fleet's behavior and economic mechanisms. It outlines the research goals and questions that form the foundation of the thesis, providing a roadmap for the ensuing research while also describing the research methods. Finally, the chapter discusses the significance of the research in both the field of XAI and maritime analysis before presenting an outline of the thesis structure.

1.1 Background

The rapid development of Artificial Intelligence (AI) systems has resulted in unprecedented progress across a wide range of applications. From seminal large language models (LLMs) like the *GPT* series [Radford et al., 2018] to protein folding achievements in computational biology by *AlphaFold* [Jumper et al., 2021], AI systems have demonstrated remarkable ability in solving complex problems, achieving feats previously unimaginable. As a result, the recent advancements in the AI domain have demonstrated competencies and decision-making that contest long-established human knowledge, highlighting the transformative potential of AI.

The ability of AI systems to surpass current knowledge in several domains presents a unique opportunity to leverage these systems as learning resources, providing an alternative perspective on complex problems and offering a means of expanding knowledge beyond traditional methods. By studying the decision-making processes of AI systems, researchers can use AI-derived knowledge to gain valuable insights into the patterns and relationships inherent in complex datasets, generate new hypotheses, and guide further research.

Despite the transformative potential of AI in knowledge acquisition, the lack of interpretability in AI systems' decision-making processes presents a challenge. As AI systems become more advanced and complex, their decision-making processes tend to become increasingly opaque, making it difficult to understand the rationale behind their decisions. This lack of transparency can hinder the use of AI as learning material, as such *black box* systems restrict deeper insight than the observation of their input-output behavior[1]. As a consequence, the field of Explainable AI (XAI) emerged as a counterweight, pursuing the development of interpretable AI systems that provide clear and understandable explanations for their decision-making processes. XAI

[1] The inputs and outputs (and consequently the decision) can be observed; however, the internal reasoning and decision-making process is hidden and unintelligible (black box)

has gained significant traction in recent years, with research focusing on various methods and techniques for achieving interpretability in AI systems. These techniques not only help demystify the inner workings of complex models but also contribute to better trust, transparency, and compliance with regulatory frameworks. As the adoption of AI systems in various industries grows, the need for explainable AI approaches becomes increasingly crucial to ensure that stakeholders can understand and trust AI-generated insights and predictions.

1.2 Problem and Motivation

The maritime shipping industry is a central pillar of the global economy, and with an estimated 90% of goods transported via sea, according to the International Maritime Organization, it is serving as the dominant means of freight transportation. Consequently, the heavy reliance on maritime transportation yields a strong connection between the market and the world fleet, and the prices charged for the transportation of goods, also known as freight rates, are known to impact fleet behavior [Martin Stopford, 2008; Boshoff and Fourie, 2010].

Traditionally, ship-based market predictions have been primarily informed by OECD economics, fundamental analysis, and industry experience. However, experts at Maritime Optima, the collaborating organization for this thesis, hypothesize that the mandatory installation of the Automatic Identification System (AIS) onboard commercial ships can offer untapped potential for market forecasting. Much of the data captured in the AIS, comprising real-time ship metrics such as position, speed, and draft, are known to have strong connections to market dynamics [Maanum and Selnes, 2015; Adland et al., 2018]. Despite its initial intention for search and rescue (SAR) operations, the potential of AIS for analytical purposes has become increasingly evident, and Maritime Optima has devoted considerable resources to cleaning and abstracting AIS data, resulting in a refined AIS dataset combined with comprehensive descriptive data for each vessel in the global fleet.

This thesis aims to leverage state-of-the-art ML models and XAI methods to advance our insight into the relationships between ship traffic patterns and economic mechanisms using Maritime Optima's refined AIS data. Of particular interest will be to investigate whether the decision-making generated by these models concurs with the principles of established knowledge. Moreover, by interpreting models capable of forecasting financial instruments using AIS data, the study hopes to unveil latent patterns and relationships that traditional methods may have overlooked, offering new opportunities for maritime researchers to gain insights into maritime trade. Additionally, the thesis attempts to develop predictive models for the maritime industry, enabling stakeholders, like shipping companies, port authorities, and logistics providers, to make data-driven decisions and anticipate market fluctuations, resulting in increased efficiency, competitiveness, and transparency. By utilizing the pattern-recognition capabilities of machine learning, the models might be capable of extracting actionable insights from AIS data, identifying trends that would be challenging to uncover using conventional analytical methods. A key aspect of this research is the integration of XAI methods into the developed models. The use of XAI will not only help increase the trustworthiness and acceptance of the AI-generated insights but also aid in facilitating a deeper understanding of the complex interdependencies within the maritime ecosystem. The insight yielded in this thesis, in turn, can empower maritime stakeholders to make more informed decisions and also lead to a more resilient, sustainable, and transparent industry.

1.3 Goals and Research Questions

This section outlines the two overarching goals of the research and a set of research questions to guide their inquiry. The research questions are systematically defined and justified for each goal, forming the foundation for the research. Additionally, the section addresses limitations or constraints, including industry-specific preferences forwarded by domain experts at Maritime Optima, who will ultimately lend their expertise to the interpretation and evaluation of the findings.

Goal 1 *Create a forecasting model that uses AIS data to generate proximate estimates for financial instruments within the maritime cluster.*

The objective is deemed achieved if the model is able to generate *close approximations* for various financial instruments. While a *great* accuracy is compelling, the thesis focuses on its ability to find underlying patterns in AIS for further analysis, and a proximate estimate is, therefore, deemed sufficient if it appears that the model has acquired some form of pattern recognition from the data. The domain experts have expressed a pronounced preference for a real-valued approach over a classification-based approach, attributed to its finer granularity and better fidelity. This preference remains even though the fidelity of a classification-based approach can be improved with a greater number of classes.

Research question 1.1 *What are the current state-of-the-art models for multi-feature time series prediction?*

The research question requires an investigation of the current state-of-the-art models for time series prediction, with an emphasis on utilizing the available properties in AIS data. This inquiry involves a comprehensive exploration and evaluation of existing models capable of multivariable time-series forecasting, i.e., the prediction of a single output variable over time using multiple input features.

Research question 1.2 *What existing research has been conducted that employed AIS data for forecasting?*

This inquiry seeks to explore and evaluate previous studies that have utilized AIS data for various forecasting objectives. By examining their methodologies, results, and limitations, it becomes possible to pinpoint gaps, build upon previous findings, and circumvent potential shortcomings. Constraints for this research question may include the availability of relevant literature, the comparability of different studies, and the accelerated development of technology and modeling techniques within the field.

Research question 1.3 *To what extent can AIS data be used to model financial instruments in the maritime industry successfully?*

This question addresses the extent to which AIS data can be utilized to effectively model financial instruments relevant to the maritime industry, with a particular focus on devising an effective data format and testing various models. To further measure the extent, various instruments should be tested with varying degrees of external influence, from an instrument firmly rooted in the performance of the world fleet, to something with elevated external influence. A key objective will be to derive a suitable feature format that can represent the relevant features in AIS data and apply this format to multiple forecasting models and XAI methods. The efficacy of these models will be evaluated not only based on their forecasting accuracy for the instruments but also in terms of validating the practicality and effectiveness of the proposed data format.

For the sake of this study, the predictive modeling will be conducted based solely on AIS data, without relying on additional data sources such as historical values of the financial instruments themselves or external indicators like oil prices. This approach could limit the predictive power of the models, as financial markets are influenced by a wide range of factors beyond the scope of AIS data. However, the focus is on isolating and understanding the predictive potential embedded in AIS data itself, which will contribute to a deeper understanding of how this specific data type can inform financial forecasting in the maritime industry.

Inevitably, this approach means that the models will be required to assign the effects of external influences to AIS data. However, this distinctive constraint aligns with the core intent of the research. Although the importance of external influences on financial markets is acknowledged, the purpose of this study is to illuminate the predictive power inherent within AIS data. By limiting the model inputs to AIS data exclusively, the goal is to highlight its predictive capability boundaries and establish the extent to which AIS data alone can be used to forecast these financial instruments in the maritime industry.

Goal 2 *Interpret the model achieved by goal 1 to derive both established and novel insights into the interplay between the world fleet's behavior and the maritime trade market using methods within the field of XAI.*

The objective is to interpret the model's decision-making principles, assessing its ability to follow established forecasting principles, but also to uncover new insights, which may lead to a better understanding of the underlying connections between the world fleet and financial instruments. This interpretation and validation of insights will be conducted in conjunction with a domain expert in maritime trade analytics.

Although the goal of generating new insights in the field of maritime analysis is a desirable objective, it is imperative that the model demonstrates an understanding of pre-existing knowledge to establish confidence. Limitations may arise from the complexity of the predictive model, the complexity of the AIS data, domain constraints, and the limitations of the XAI methods employed.

Research question 2.1 *What are the current state-of-the-art XAI methods for explaining multi-feature time series tasks for the resulting model?*

This question entails a thorough investigation and assessment of existing XAI methodologies that can be applied to the predictive model derived from Goal 1. Constraints may include the rapidly evolving landscape of the XAI field, which complicates the identification of the most appropriate methods, as well as the varying levels of efficacy and interpretability across different methods.

Research question 2.2 *Do the AIS-driven forecasting models provide meaningful and interpretable explanations as assessed by a domain expert within maritime trade analytics?*

This research question delves into the capacity of the AIS-based model to yield insights that a domain expert can meaningfully interpret. It is imperative to ensure that the model's outputs are not only statistically sound but also understandable and practically relevant in the context of the maritime industry. Limitations could arise from the inherent complexity and multidimensionality of AIS data and the capabilities of the chosen XAI methods.

Research question 2.3 *Do the explanations provided by the models resonate with established knowledge within the maritime trade analytics field?*

This question explores whether the models' outputs align with established knowledge within the field of maritime trade analytics. It is crucial to establish a baseline of trust in the model's output by checking if the model has learned known correlations and trends. Limitations here might stem from the potentially vast disparity between conventional wisdom and insights derived from complex AIS data, or the possibility that the models may overlook some subtle yet important factors.

Research question 2.4 *Do the models yield novel insights into the interplay between the world fleet's behavior and the maritime trade market?*

This question probes the capacity of the model to produce novel insights and contribute to a deeper understanding of the maritime trade market. Constraints include the challenge of differentiating between genuinely novel insights and artifacts of the model's assumptions or errors, as well as the difficulty of validating novel insights against real-world outcomes in the complex and dynamic maritime trade market.

1.4 Research Method

This section outlines the research method adopted to address the research goals and questions from Section 1.3, consisting of three main stages: literature review, model and data development, and model experiments. In addition, as a collaborator for the thesis, Maritime Optima has devoted itself to providing expert opinions that will be incorporated throughout the research process to ensure a rigorous and industry-relevant approach.

1.4.1 Literature Review

A literature review will be undertaken to provide the study with a requisite foundation of state-of-the-art models and approaches. The review seeks to find leading models for multivariable time series prediction and assess their applicability to AIS data. Additionally, the review will examine existing research that has utilized AIS data for forecasting purposes and investigate relevant XAI techniques, with an emphasis on effective techniques in relation to AIS-based time series data. The literature search will be conducted using relevant keywords, such as "AIS forecasting," "time series prediction," "maritime trade forecasting," and "XAI time series" in academic databases. In addition, Maritime Optima has provided valuable background knowledge through verbal lectures, as well as directed the author to relevant literature.

1.4.2 Model and Data Development

Based on the findings obtained from the literature review, the most promising and applicable state-of-the-art models and XAI methods will be identified. These models and methods will be adapted to accommodate the specific requisites of the AIS data and the forecasting objectives for the thesis. An effective data format will also be derived, intending to hold the capacity for sufficient model performance as well as good explanations.

1.4.3 Model Experiments

The selected models will be evaluated based on their ability to accurately forecast the intended variables. Several performance metrics, relevant to the field and nature of the forecasted data, will be employed to assess the model's robustness, accuracy, and reliability. While the specific metrics will be chosen based on the unique requirements of the forecasting task and the nature of the AIS data, they will generally encompass measures of error and correlation of change.

Simultaneously, the interpretability of the best models, generated by XAI techniques, will be assessed. This will involve a collaborative process with domain experts from Maritime Optima, who will evaluate the clarity, relevance, and usability of the explanations provided by the models. This step is critical to ensuring that the interpretations not only make sense in a technical context but also provide actionable insights that can be understood and used effectively by those in the maritime industry. The evaluation will strive to ensure that the generated explanations are both theoretically sound and practically meaningful.

1.5 Contributions

The research presented in this thesis contributes to both the field of Explainable AI and the field of maritime analysis, positioning itself to readers from both fields.

The primary contribution of the thesis is the addition of a case that seeks to advance insight into a new domain through the means of an AI system, using XAI techniques to derive new and confirm established knowledge. The research also provides an evaluation of established XAI techniques for the challenging case of time series forecasting using hard-to-visualize AIS data. Consequently, the research provides a new perspective on XAI in new sectors - in this case, the maritime industry. The results of this study will be valuable to researchers interested in developing and evaluating explainable AI models to uncover scientific discoveries using time series forecasting.

Additionally, the thesis contributes to the field of maritime analysis, seeking to expand our knowledge about the interplay between shipping patterns and economic mechanisms. The application of machine learning for AIS data has the potential to reveal previously unknown patterns and relationships that can improve transparency and inform strategic decision-making within the maritime industry. These results can provide valuable information to further maritime research, as well as stakeholders like shipping companies, port authorities, and policy-makers who are interested in optimizing their operations.

1.6 Thesis Structure

The remainder of the thesis is structured into five main chapters, following this introduction.

Background Theory Chapter 2 delves into the theoretical background that underpins the study. It offers a comprehensive review of critical concepts such as the Automatic Identification System (AIS), the maritime freight market, traditional maritime forecasting, and various ML paradigms and sequence-based ML models. Additionally, the chapter elaborates on explainable AI (XAI), its importance, the various approaches to explainable ai, and the evaluation of explanations and interpretations.

State of the Art Following the background theory, Chapter 3, presents a review of the current state-of-the-art research landscape. It explores the current applications of AIS in financial forecasting. Additionally, it will cover research in the broader scheme of multivariable sequence models and AIS-based research, presenting supplementary applicable models and approaches. The chapter also delves into various methods and approaches to XAI as applied to the relevant models, before concluding with a summary of the chapter's key findings.

Model and Data Engineering Then, Chapter 4 provides an elaboration on the data foundation and its processing, as well as the employed model implementations drawn from the state-of-the-art literature. Further, it details the various XAI approaches employed and their limitations.

Experiments and Results Chapter 5 presents the conducted experiments and their results, with an emphasis on providing detailed information about the experimentation process to promote reproducibility. This includes an overview of the baselines, overall experimental plan, data configurations, model parameters, training process, and performance evaluation metrics.

Evaluation and Conclusion The thesis concludes with Chapter 6, which provides an evaluation of the findings, discusses the implications of the results, and offers potential avenues for future research. This chapter serves to provide a holistic view of the research and its outcomes, contextualizing the results within the wider field of study.

Moreover, a comprehensive appendix is included, providing supplementary information and additional material that extends the main content of the thesis, but also provides an extended evaluation of each model, including additional predictions, and, more importantly, explanations for those interested.

Chapter 2

Background Theory

This chapter will provide background knowledge and theory. Section 2.1 elaborates on the AIS system, which comprises much of the data foundation of the thesis. Section 2.2 gives a high-level overview of the shipping market and how it is employed in traditional market-based analytics in Section 2.3. Section 2.4 will introduce machine learning, its applications, and its ability to learn underlying data patterns. Finally, Section 2.6 will provide an overview of the explainable AI field, including its purpose and justifications, methods of applications, and potential contributions to future scientific discoveries.

2.1 AIS

The Automatic Identification System (AIS) is a global VHF-based system of vessel-mounted transceivers that automatically relay navigational data and vessel specifications between vessels and various terrestrial- and satellite-based AIS receivers. The system's original intent was to aid search and rescue operations within the framework of the Global Maritime Distress and Safety System (GMDSS), which was designed to improve the safety of life at sea by enabling communication between vessels and onshore facilities. Over time, AIS has been incorporated into a wider range of applications, such as collision avoidance, accident investigations, and general vessel tracking. The AIS is mandatory for larger passenger and commercial cargo vessels, as required by the International Maritime Organization [1974] through the *Safety of Life at Sea (SOLAS)* Convention[1].

The AIS transceivers are divided into two classes: *Class A* and *Class B*. *Class A* transceivers are reserved for large commercial vessels following the SOLAS carriage requirements and have priority in the system. In contrast, *class B* transceivers are dedicated to lighter vessels in the commercial and leisure markets, such as yachts and small fishing vessels. For this thesis, the *Class A* messages are of interest as they are transmitted from commercial vessels involved in the global maritime trade. The AIS is broadcast near real-time through 27 different message types (see appendix A.1), with *static reports* (type 5) and *position reports* (type 1-3) being of particular interest - a subset of which can be found in Table 2.1 and 2.2, respectively. For *class A* vessels, *static reports* are typically broadcast every 6 minutes, while *position reports* are reported every 2 to 10 seconds during voyages and 3 minutes when at anchor.

[1] As of 2004, the SOLAS convention includes all passenger ships regardless of size and cargo ships of ≥ 500 gross tonnage (≥ 300 if engaged in international voyages)

Parameter	**Description**
IMO number	Unique 7-digit vessel ID number assigned during construction
MMSI number	Unique ID assigned to the onboard transceiver
ETA	Estimated time of arrival at destination port
Destination	The vessel's current voyage destination port
Draft	Distance between the waterline and vessel's hull (depth)

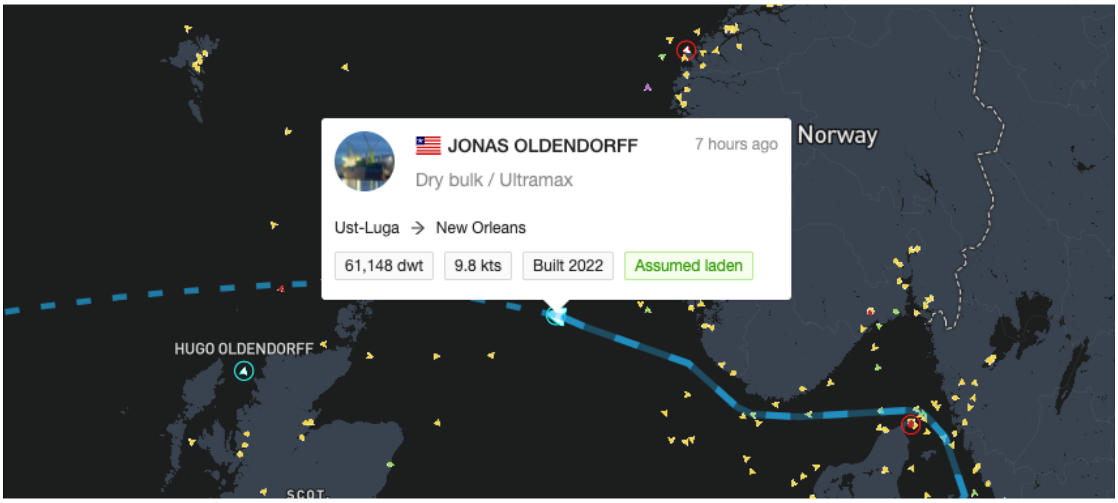
Table 2.1: A subset of type 5 *static reports*

Parameter	**Description**
MMSI number	Unique ID assigned to the onboard transceiver
Navigational status	Integer enum (underway using engine, at anchor, moored, etc.)
SOG	Speed over ground
True heading	Heading in degrees (0° to 360°, 0°=north)
Longitude	East-west position on Earth in degrees (-180° to 180°, 0°=meridian)
Latitude	North-south position on Earth in degrees (-90° to 90°, 0°=equator)
Timestamp	UTC second of when the report was generated

Table 2.2: A subset of type 1,2 and 3 *position reports*

2.1.1 AIS for Vessel Tracking

Over time, the scope of AIS applications has expanded to vessel tracking, as AIS allows near real-time monitoring of the world fleet, resulting in the development of specialized software known as *vessel tracking software*. These software platforms are specifically designed to collect, process, and present data emitted by AIS transponders, such as Maritime Optima's *ShipAtlas* software shown in Figure 2.1. This enables valuable monitoring of vessel operations, with additional functionalities being built on top, such as historical vessel tracking, trade pattern analysis, and notifications about fleet movements.



JONAS OLDENDORFF
7 hours ago
Dry bulk / Ultramax
Ust-Luga → New Orleans
61,148 dwt
9.8 kts
Built 2022
Assumed laden

Figure 2.1: Tracking of a vessel in the ShipAtlas software by Maritime Optima

Vessel trackers are widely popular among maritime enthusiasts and are repeatedly featured in news outlets that cover maritime news stories. Moreover, they have been increasingly integrated into the professional maritime industry for stakeholders such as authorities, port operators, and shipping companies. Their integration improves fleet awareness and operational efficiency by allowing direct access to operational information for both their own fleet and those of competitors, eliminating the reliance on updates directly from the vessels' crew or brokers with confidential knowledge. In addition, AIS integrates well into professional shipping software, allowing stakeholders to survey intended vs. actual performance, determine vessels' actual service speeds, and inform decisions about route distances and fuel consumption.

2.1.2 AIS Limitations

With its intention as a search-and-rescue aid, AIS is constrained by several implementation details and is thus, as documented by Emmens et al. [2021] and Harati-Mokhtari et al. [2007], prone to signal/equipment malfunctions and human error.

One fundamental limitation of AIS is its mode of reception. Initially, a terrestrial-based system was considered adequate for its intended purpose - however, it has since been supplemented with satellite-based receivers to improve tracking further out at sea. Although the addition of satellite-based AIS (S-AIS) extends the system's range and circumvents some country-specific data restrictions, it remains susceptible to deficiencies in coverage and limited temporal resolution due to weather conditions, infrequent satellite passes, computational limitations, and orbital positioning. In some instances, particularly in areas with a high volume of vessel AIS transmissions, it can result in substantial intervals between data points, causing the generated track to traverse land masses.

Moreover, while a large portion of the AIS data is generated through automated means, such as GPS, some of the information is manually entered by the crew, leaving room for human error. This has led to transmissions being contaminated with noise, either in the form of missing or incorrect values, as documented by Harati-Mokhtari et al. [2007]. Table 2.3 demonstrates some instances where the destination attributes deviate significantly from the mandated LOCODE format. Furthermore, Emmens et al. [2021] conducted a more recent analysis and found that roughly 84.1% of cargo vessels had noise in their IMO number, 35% in destination values, 54% in ETA values, and 30% in draft values.

MMSI	**Destination (AIS)**	**Received**	**Vessel Location**
538007794	"ARM GARD ON BOARD"	1st February 2023	Outside Yemen
240906000	"FORORDERS"	1st February 2023	Tyrrhenian Sea
219472000	"GUARD VESSEL FEHMARN"	1st February 2023	Fehmarn Belt
311000264	"OFFSHORE"	1st February 2023	Outside Cape Town
205763000	"===PAAML>===US WC"	1st February 2023	West of Mexico
636092819	"3"	2nd February 2023	Gibraltar

Table 2.3: Examples of noisy manually entered AIS destinations

Another complication is that the transmitters' MMSI numbers are configured manually during installation, allowing a vessel to be observed at multiple locations simultaneously, as multiple transmitters can claim the same identity. The crew may also fail to update the AIS data sufficiently or switch off the system altogether due to piracy concerns or unethical trade activities.

2.2 The Maritime Freight Market

In the book *Maritime Economics*, Martin Stopford [2008] defines a market as "any body of persons who are in intimate business relations and carry on extensive transactions in any commodity" (p. 177) and divides the shipping industry into four primary markets.

1. **The freight market**; A supply-demand driven marketplace facilitating the trade of sea transport - involving arrangements and negotiations between cargo owners and charterers to transport cargo.

2. **The sale and purchase market**; A marketplace where secondhand ships are traded to be used in the *freight market.* It allows shipowners to acquire or dispose of vessels in response to market conditions, operational requirements, or fleet expansion plans.

3. **The shipbuilding market**; A market dedicated to the design, construction, and delivery of new ships. It involves shipyards, ship designers, and equipment manufacturers, who work together to create vessels that meet the specific needs of shipowners and operators.

4. **The demolition market**; Often referred to as the *recycling market*, the demolition market deals with the disposal of end-of-life ships, in which outdated or uneconomical vessels are sold to scrapyards or recycling facilities to be dismantled, salvaging valuable materials and machinery components for reuse or recycling.

While each market plays a significant role in the shipping industry, this thesis will primarily focus on the *freight market.* However, a comprehensive explanation of the freight market is infeasible for this section; the freight market is an overwhelmingly complex system subject to countless variables driven by the interplay of supply and demand. A concise overview will be provided to maintain the feasibility of explaining it as background knowledge.

Martin Stopford describes the freight market as a venue where sea transport is bought and sold, building on the foundation of commodity trading and shipping exchange from the original mid-nineteenth century *Baltic Shipping Exchange* freight market. Today, business in the freight market is transacted via digital communication platforms, such as telephone or e-mail, and is intermediated via brokers on behalf of the parties. The maritime freight market is the largest segment in the global transportation industry, and the International Maritime Organization estimates that roughly 90% of global trade by volume is transported via sea. This is attributed to its cost-effectiveness in transporting large amounts of cargo, resulting in a highly competitive market influenced by factors such as global economic growth, fuel prices, trade policies, supply, and demand.

Most transactions on the freight market are arrangements for cargo transportation, commonly referred to as a *voyage charter.* Voyage charters are a form of contract between a cargo owner and charterer that outlines the terms of the transportation of goods between ports A and B. These agreements usually include specifications such as the type and quantity of cargo, the ports of loading and discharge, the duration of the transportation, and the payment terms. Usually, the payment term for voyage charters is usually a negotiated price per ton of cargo, known as the *freight rate.*

The other primary transaction in the freight market is a *time charter* - a contract where a charterer temporarily leases a vessel from a ship owner. In a time charter, the charterer has the exclusive use of the vessel and its crew, while the ship owner retains ownership and management responsibilities. The charterer pays the ship owner a daily hire, which covers crew wages, fuel, and maintenance costs.

2.2.1 Market Cycles and Freight Rates

The market is affected by numerous factors, including fuel costs, trade regulations, geopolitics, port capacities, and the active world fleet. However, Martin Stopford points to global economic growth and supply-demand dynamics as the most influential, as the global production and consumption of goods directly influence the demand for shipping services; growth in the global economy stimulates an elevated flow of goods, resulting in an increased need for transportation of raw materials for manufacturing or trade of manufactured products. Conversely, in times of economic downturns, the demand for transportation consequently declines.

These additive short- and long-term fluctuations, referred to as *market cycles*, constitute the overall market trend and tightly influence the freight rates. When demand for shipping services is high, also known as a *Peek/Plateau*, charterers have the leverage to charge higher rates due to the limited availability of vessels that meet the demand, forcing cargo owners to pay higher rates to secure transportation of their goods. Depending on the magnitude of the peak, ship owners may try to capitalize by expanding their fleet by acquiring new vessels in the *shipbuilding market.* The resulting increase in transport capacity may stabilize or decrease the freight rates, as cargo owners must no longer compete for limited availability, especially if there is an over-acquisitions of newbuilt vessels. The market then ultimately enters a *Collapse*, where the transport availability becomes greater than the demand. After a Collapse, the market enters a *Trough*, characterized by a surplus of shipping capacity, low freight rates, and an active demolition market, before entering a *Recovery.*

2.2.2 Influence on Fleet Behavior

Movements in the market influence the world fleet in various ways, mainly in response to changing freight rates and subsequent profit margins. One central consequence of market changes is that ship owners adjust the speed of their vessels in accordance with the freight rates, as the speed directly affects fuel consumption - the dominant cost of sea freight. In extreme troughs, vessels' net revenue might be negative, leading to slow-steaming or drifting to minimize fuel expenses. In contrast, when the market is peaking, and freight rates are high, the ship owners tend to increase their vessels' speeds to maximize profits. Similarly, the draft of vessels also impacts fuel consumption, leading to vessels being loaded below full capacity to reduce costs during times of low freight rates. Furthermore, a trough often forces vessels to compete for sub-optimal cargo sizes, resulting in lower drafts. Another aspect is that vessels tend to perform shorter voyages in a bad market. Longer voyages require a higher fuel expenditure, which may not be economically feasible during periods of low freight rates and sub-optimal cargo opportunities. Charterers will also prefer shorter voyages as they offer a quicker turnaround time and an earlier opportunity to secure new cargo.

2.2.3 Vessel Segments

The market is divided into several segments defined by the type of cargo and the cargo handling capabilities of vessels. Table 2.4 shows some of the central segments in today's industry. The segments are further divided into different *sub-segments* based on the capacity of the vessels. The definitions of sub-segments vary for each segment but are generally similar across the industry. For instance, dry bulk vessels with a deadweight of 67,000 to 78,000 MT can typically be classified as *Panamax*, which denotes the approximate size of vessels that can navigate the Panama Canal. See Appendix B for a detailed overview of the different sub-segments.

Segment	Purpose	Example commodities
Dry bulk	Transport of unpackaged bulk commodities	coal, iron ore, grain
Tanker	Transport of petroleum products	crude oil, gasoline, fuel oil
Chemical	Transport of chemical products	acids, alkalis, paint
LPG	Transport of liquefied petroleum gases	propane, butane
LNG	Transport of liquefied natural gases	methane, ethane

Table 2.4: Some central segments from Maritime Optima

2.3 Traditional Maritime Forecasting

The accurate anticipation of future market trends and conditions is a crucial determinant of success for stakeholders in the shipping industry, including shipowners, rating agencies, shipyards, bankers, and ports. These entities depend heavily on forecasting to make informed investment decisions and achieve profitability. However, the inaccuracy of shipping forecasts is widely acknowledged by the industry, following several erroneous forecasts over the years. The inadequacy of these forecasts is grounded in the difficulty of predicting the complex maritime market, and even modern, sophisticated forecasting methods can struggle [Martin Stopford, 2008].

As the pursuit of precise predictions has proven unrealistic, stakeholders must deal with uncertainty and depend on educated estimates to make decisions. As a result, forecasters supply information to mitigate uncertainty and assess risks through the development of rational forecasting systems, utilizing economic models grounded in historical patterns and trends. Consequently, forecasting strategies have shifted from attempting to predict the future to obtaining and analyzing the right information about the present.

Martin Stopford further emphasizes three fundamental components of current maritime forecasting: *relevance*, *rationale*, and *research*. The relevance element necessitates a thorough comprehension of the future aspect that the decision-maker seeks to examine. The rationale component demands a compelling justification for why the anticipated developments are likely to occur. The research element aims to minimize uncertainty and is imperative for forecasting. While these principles do not ensure complete accuracy, they establish the traditional fundamental guidelines for producing practical analyses with the primary objective of reducing uncertainty.

An important aspect is that shipowners often depend on personal experience and intuition when making decisions, as certain aspects of shipping markets are too complex for statistical models to capture, and data frequently arrives too late to be valuable. However, this decision-making method has limitations, as subjective sentiment may influence judgment, resulting in a loss of perspective. In contrast, decision-makers within prominent shipping corporations, banks, and bureaucracies delegate analysis and anticipate predictions based on established analytical techniques that can be independently verified, providing a more objective basis for their decisions.

2.3.1 Forecasting for Various Stakeholders

The various stakeholders and decision-makers in the maritime industry have distinct demands for forecasts relevant to their specific field. For instance, shipping companies require forecasts of future freight rates, newbuilding prices, and second-hand prices to facilitate informed decisions regarding long-term charters, ship acquisitions, and sales. Cargo owners are interested in future transport costs and the availability of suitable transport. Shipbuilders are focused on future demand for new ships, prices, subsidies, and competition from other shipbuilders. Bankers require forecasts of market strength, freight rates, and ship prices to evaluate loan approvals and foreclosures. Governments must balance short-term benefits against long-term risks when making decisions concerning subsidies, capacity cuts, and international shipping registers. Port authorities need to forecast the cargo volume and types of ships operated to make informed port development decisions. Machinery manufacturers must analyze trends in ship construction, future developments in operational ship management, ship operating economics, and competitor activity to decide what products to develop and how to manage capacity. Forecasts are essential in several shipping decisions, including spot-chartering ships, time-chartering ships, sale and purchase decisions, budgets, strategic and corporate planning, product development, international negotiations, government policy-making, industrial relations, and bank credit analysis.

2.3.2 Analytical Forecasting Techniques

In the domain of maritime economics, analytical forecasting techniques play a crucial role in predicting future trends and events. According to Martin Stopford [2008], there are four primary analytical techniques employed within the maritime industry: *opinion surveys*, *trend analysis*, *mathematical models*, and *probability analysis.* A summary of the various techniques can be found in Appendix C.

Opinion surveys

Opinion surveys involve consulting experts in the maritime field to gather their insights on potential future developments. As Stopford explains, the Delphi technique and other structured opinion surveys are commonly employed for this purpose, which combines the opinions of an expert panel into a consensus forecast. This approach is particularly useful for detecting emerging trends that are evident to maritime specialists but may not be apparent from historical data.

Trend analysis

Trend analysis focuses on identifying historic trends and cycles within a data series to generate future values or forecasts, also known as *time series prediction.* The simplest forms of trend analysis, such as the naive forecast or trend extrapolation, extrapolates recent trends into the future, providing a rapid assessment without accounting for possible trend changes. More sophisticated trend analysis, such as *Autotegressive Moving Average* (ARMA), involves examining underlying trends, cycles, and unexplained residuals, with forecasters tasked with determining whether past trends are likely to change.

Mathematical models

Mathematical models extend the concept of trend analysis by quantifying relationships between different explanatory variables within the maritime context to explain trends and predict future events, employing techniques such as regression analysis and econometric models. For instance, a mathematical model might explore the correlation between oil trade growth and global industrial production, as Stopford suggests. By estimating equations that quantify such relationships, predictive models can be developed to forecast future trends and events in the shipping industry.

Probability analysis

Probability analysis, on the other hand, estimates the likelihood of specific outcomes occurring instead of predicting exact future events in the maritime field. This technique, as detailed by Stopford, requires the calculation of probabilities in numerical terms and assists decision-makers in understanding the predictability or unpredictability of particular events in the shipping sector.

2.4 Machine Learning

Machine learning (ML) is a subfield of AI focusing on algorithms and statistical models that can learn from data and, without relying on explicit instructions, generate outputs consistent with underlying patterns and relationships. There are a multitude of different ML algorithms tailored to various applications, such as *Computer Vision* and *Natural Language Processing*, but also a wide range of general-purpose models with distinctive features and capabilities. While these algorithms ultimately share the same goal, today's algorithms are commonly categorized based on the type of learning they utilize, being *supervised*, *unsupervised*, and *reinforcement* learning.

2.4.1 Supervised Learning

Supervised learning algorithms acquire knowledge from labeled data, wherein each instance in the training dataset is annotated with a correct response (i.e., label). These algorithms aim to construct a model that maps inputs to the annotated outputs, which is achieved by iteratively adjusting the model's parameters using optimization techniques until it minimizes the difference between the correct outputs and the model's predicted outputs. Supervised learning tasks are generally separated into two problem types: *regression* and *classification.*

In *regression* tasks, the objective is to predict a real-valued output based on various input features. For instance, in the canonical *house price prediction* problem [Bai, 2022], housing prices are predicted based on house attributes like size, age, and property features. In contrast, *classification* tasks aim to predict the class of a given input by assigning it to a discrete category, such as spam or not spam, positive or negative sentiment, or different categories of objects in an image. A classic problem for classification is the *MNIST handwritten digit classification* problem [Wu and Zhang, 2010], where given a greyscale 28x28 image of a handwritten digit, the model is to categorize it as a discrete number between 0 and 9.

2.4.2 Unsupervised Learning

Unsupervised learning algorithms, unlike supervised ones, are designed to analyze and extract information from unlabeled data. These algorithms are tasked with identifying hidden patterns and relationships without prior knowledge or objectives to guide their analysis. As a result, these algorithms can reveal patterns in data exceeding what might be classified as unstructured noise [Ghahramani, 2004].

One of the most prevalent applications of unsupervised learning is *clustering*, which aims to group data based on the similarity of features by maximizing intra-cluster similarity (i.e., the similarity between instances within the same cluster) and minimizing inter-cluster similarity (i.e., the similarity between instances in different clusters). Different clustering algorithms employ a variety of methodologies to determine the number of clusters and their corresponding boundaries. For instance, hierarchical clustering methods recursively merge or split clusters based on some criteria, whereas partitioning algorithms such as K-means assign instances to fixed, non-overlapping clusters, and density-based algorithms identify clusters based on regions of high instance density.

Another application is *dimensionality reduction*, which aims to reduce the number of features in a dataset while preserving the relevant information that captures the underlying structure of the data. In many cases, datasets may contain redundant or irrelevant features that contribute to noise in the data or make it difficult to visualize and interpret. Dimensionality reduction methods aim to address these issues by transforming the data into a lower-dimensional space while retaining the essential information [Ghahramani, 2004].

2.4.3 Reinforcement Learning

Reinforcement learning is a paradigm that enables models to learn through autonomous interaction with an environment to maximize a reward signal. The algorithm is not given labeled data or explicit instructions but learns through trial and error, ultimately learning optimal actions that yield positive outcomes while avoiding actions that result in negative outcomes [Sutton and Barto, 2018]. This paradigm has lead to several advancements in robotics [Kober et al., 2013] and game playing, most notably in DeepMind's *AlphaGo* [Silver et al., 2016]

The models find optimal actions by learning a *policy* (i.e., a mapping between the observable state of the environment and the agent's performable actions) that maximizes the expected cumulative reward over time. The *policy* can be represented differently depending on the problem. For problems with smaller state spaces, it can be sufficient with a lookup table that stores each discrete state and its optimal action. However, for larger state spaces that cannot be explicitly stored in computer memory, like the number of possible states in the game of Go[2], function approximators are often utilized. These approximators enable an efficient and scalable representation of the *policy* by mapping the state to the optimal action using a mathematical function often represented by a neural network [Sutton and Barto, 2018; Xu et al., 2014].

2.4.4 The Learning Process

The learning process is the process in which an ML model adapts its internal representation to optimize its predictive performance. A majority of ML models utilize an optimization algorithm to facilitate the learning process, which is achieved by minimizing a loss function $\mathcal{L}$ [Hastie et al., 2009; Goodfellow et al., 2016]. The loss function quantifies the difference between the model's predictions and the correct output (label), providing a measure of the model's performance during training. One widely used loss function for regression problems is the *mean squared error (MSE)*, as shown in Equation 2.1. The MSE penalizes large differences between the prediction x and target y, emphasizing larger differences when computing the loss.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2 \tag{2.1}$$

Different models employ different optimization techniques to find the optimal set of weights for the input features. For instance, linear regression models can use such loss functions directly, minimizing the difference between the predicted and correct outputs. On the other hand, decision trees utilize a divide-and-conquer strategy to recursively split the data based on feature values, optimizing a criterion such as information gain [Quinlan, 1986].

Learning in Neural Networks

Gradient-based models, like neural networks, have become increasingly prevalent in the development of state-of-the-art models due to their role as universal function approximations, making them capable of representing *any* function, as established by Hornik et al. [1989]. The models learn by computing the gradient of a loss function with respect to the model's parameters ω [Goodfellow et al., 2016]. The gradient $\nabla_\omega \mathcal{L}$ defines the direction in which the parameters should be modified to achieve the steepest decrease in the loss function. The iterative adjustment of the model parameters in the direction of the negative gradient leads to the model's convergence towards a minimum in the loss function, improving its predictive performance. A high-level algorithm of the learning processes for neural networks can be found in Algorithm 1.

[2] With a board size of 19x19 and 3 possible states per tile, Go has $3^{19 \cdot 19} \approx \boldsymbol{1.74e^{172}}$ possible board states

Algorithm 1 General learning process for neural networks

```
$M \leftarrow$ Neural network model
$O \leftarrow$ Optimizer
$L \leftarrow$ Loss function
$E \leftarrow$ Number of epochs
$B \leftarrow$ Batch size
$X \leftarrow$ Set of training samples
$y \leftarrow$ Targets for samples in $X$
Ensure: $E > 0$, $B > 0$, $|X| = |y|$
Initialize model $M$ with random parameters
for epoch $\leftarrow 1$ to $E$ do
    Shuffle training data $(X, y)$
    batches $\leftarrow (X, y)$ divided into batches of size $B$
    for all batch $(X_{batch}, y_{batch})$ in batches do
        $P \leftarrow M$'s predictions on $X_{batch}$
        $\mathcal{L} \leftarrow$ Computed loss using $L$ on $P$ and $y_{batch}$
        $\nabla_\omega \mathcal{L} \leftarrow$ Computed gradients of $\mathcal{L}$ w.r.t. $M$'s parameters $\omega$
        $\omega \leftarrow \omega - \nabla_\omega \mathcal{L}$ using $O$                ▷ Update M's parameters
    end for
end for
return Trained model $M$
```

2.5 Sequence-Based Machine Learning Models

Sequence-based ML models, especially of the deep neural network (DNN) category, have emerged as powerful alternatives to traditional statistical methods for time series forecasting, and have demonstrated exceptional performance in a range of forecasting applications [Tang et al., 2022; Makridakis et al., 2022]. In contrast to traditional methods, which necessitate a series of assumptions about the underlying data-generating process[3] [Box et al., 2015], ML and DNN models are data-driven, learning to map inputs to the correct outputs directly from data. This approach allows them to handle complex, non-linear relationships and adapt to changing dynamics within the data, resulting in improved forecasting accuracy. Moreover, using these methods for time series forecasting allows for larger volumes of data with an increasingly diverse range of input features, including both structured and unstructured data. This feature-rich approach has been particularly advantageous for applications such as financial market prediction, where multiple sources of information, such as economic indicators, sentiment analysis, or in this case, ship movement data, can be leveraged to improve forecasting accuracy [Mehta et al., 2021].

In the past decade, researchers have proposed various DNN state-of-the-art architectures specifically designed to address temporal dependencies in sequential data. These include Recurrent Neural networks and their advanced variations, such as LSTM networks. Moreover, Convolutional Neural Networks and the recent Transformer model have been effectively employed in tackling time-series challenges. To account for the many state-of-the-art models explained in Chapter 3, this section will provide some theoretical foundation on which many of these sequence models stand.

[3] The data-generating process refers to underlying mechanisms, such as seasonality and random fluctuations, that make up the observed data

2.5.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a class of ML models specifically designed for processing sequential data. The concept of RNNs emerged in the late 1980s and early 1990s, with the pivotal contributions of Jordan [1986] and Elman [1990] shaping the development and popularization of RNNs. Their work derived the *the simple recurrent network*, a type of RNN characterized by its ability to model temporal dependencies and maintain a form of memory across time steps using hidden states. Since their inception, RNNs have risen to prominence in a wide array of time series forecasting applications, serving as the foundational component of later state-of-the-art sequential models.

The Recurrent Structure and Hidden States

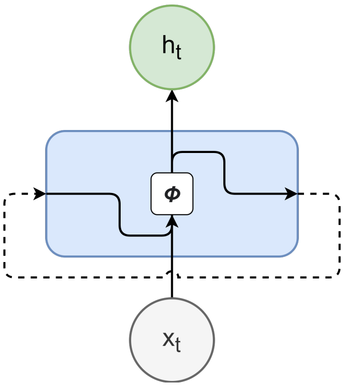


Figure 2.2: The cell of a simple recurrent network.

The fundamental building block of an RNN is the recurrent neuron, or cell, as depicted in Figure 2.2. The hidden state, represented by the stapled line, serves as a form of memory that allows the network to learn patterns across sequences of inputs. At a given time step t, a cell's output h_t is calculated from the input x_t for the given time step in conjunction with the hidden state h_{t-1} from the previous time step [Elman, 1990]. The calculation of the hidden state is expressed in Equation 2.2, wherein W and U denote the weight matrices corresponding to the hidden state and input respectively [Goodfellow et al., 2016, p. 374]. The term b is the bias term, and ϕ is an activation function, typically a non-linear function like the hyperbolic tangent $tanh$ (Appendix 1a).

$$h_t = \phi(b + Wh_{t-1} + Ux_t) \tag{2.2}$$

To provide a more comprehensive understanding, Figure 2.3 provides an illustration of an unrolled RNN, which highlights the network's structure across three sequential time steps $t-1$, t, and $t+1$. Each cell in the unrolled RNN has its own independent input and output, with arrows between the cells denoting the transfer of hidden states across time steps.

Temporal Dependencies and Gradient Problems

While the transfer of hidden states occurs between adjacent cells, it is important to emphasize that the network is capable of retaining information that spans multiple time steps, allowing

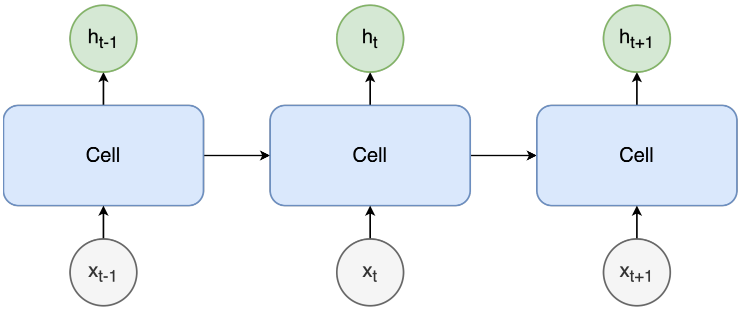


Figure 2.3: Unrolled cell of the simple recurrent network across three sequential timesteps.

information to flow through several intermediate cells. This renders RNNs particularly suitable for sequential problems like language modeling and time series forecasting, where temporal dependencies can span several time steps. For instance, when considering the phrase "It was spicy, but I ate the _", both "Jalapeño" and "rice" can be valid completions. Yet, "Jalapeño" has a higher probability due to the influence of the earlier word "spicy", which would have to retain its information over the next four words. However, RNNs can face challenges when dealing with long-range dependencies due to the vanishing and exploding gradient problem.

The vanishing gradient problem arises when the gradients of the loss function diminish significantly, leading to minimal updates in the network's weights and impeding the learning process, particularly over extended sequences. This issue is primarily attributed to the repeated multiplication of small values during the backpropagation process, resulting in exponentially smaller gradients as they propagate backward through the network. This can cause earlier layers in the network to learn at a slower pace compared to the later layers, effectively hindering the model's ability to capture long-range dependencies.

Conversely, the exploding gradient problem occurs when the gradients become excessively large, causing instability and potentially divergent behavior in the learning process. The exploding gradients can result in erratic updates to the model parameters, hindering convergence and sometimes leading to a complete collapse of the learning process.

Several techniques have been proposed to address the vanishing and exploding gradient problem in RNNs. One of the most notable approaches is the introduction of architectures capable of regulating the flow of information, like the LSTM cell, which is further discussed in Section 2.5.2. Other approaches include $L1$ and $L2$ regularization to encourage smaller network weights, as well as *gradient clipping*. Gradient clipping, proposed by Pascanu et al. [2013], sets a threshold to limit the magnitude of gradients, effectively preventing the exploding gradient problem while still allowing the network to learn from large gradient values.

Backpropagation In Recurrent Networks

Training RNNs necessitates specialized algorithms that can accommodate their unique structure and the temporal dependencies they model. Two of the most prominent algorithms used for this purpose are *Real-Time Recurrent Learning* (RTRL) and *Backpropagation Through Time* (BPTT).

The RTRL algorithm, which Elman [1990] utilized to train his simple recurrent network, is an online learning algorithm[4] that updates the network's weights at every time step based on

[4] Online algorithms update models continuously as data become available, in contrast to offline algorithms,

the partial derivatives of its output with respect to its weights. RTRL offers advantages such as continuous learning and resilience to the vanishing and exploding gradient problem, but its significantly high computational cost limits its practicality for large-scale applications.

Meanwhile, BPTT, an adaptation of the standard backpropagation algorithm for feedforward neural networks, has been tailored to accommodate the sequential nature of RNNs. Proposed by Werbos [1990], BPTT involves unrolling the RNN through time and treating it as a deep feedforward network with shared weights across each time step. This unrolling facilitates a more traditional backpropagation where the gradients are accumulated over the entire sequence and subsequently used to update the network's weights. This allows the BPTT algorithm to be more computationally efficient but has elevated susceptibility to gradient problems than RTRL. To mitigate these issues, techniques such as truncated BPTT have been developed, which limit the backpropagation to a fixed number of time steps, reducing computational complexity and lessening the impact of long-range dependencies [Williams and Zipser, 1995]. Overall, BPTT is generally more widely adopted due to its computational efficiency, especially with the advent of more resilient variations.

2.5.2 Long Short-Term Memory Networks

The Long Short-Term Memory (LSTM) network, introduced by Hochreiter and Schmidhuber [1997], is a specialized type of RNN that addresses the challenge of the gradient problems evident in conventional RNNs. By employing a more attentive cell structure, LSTMs can learn long-range dependencies in time series data more effectively, making them particularly suitable for problems involving longer sequences and greater temporal dependency spreads.

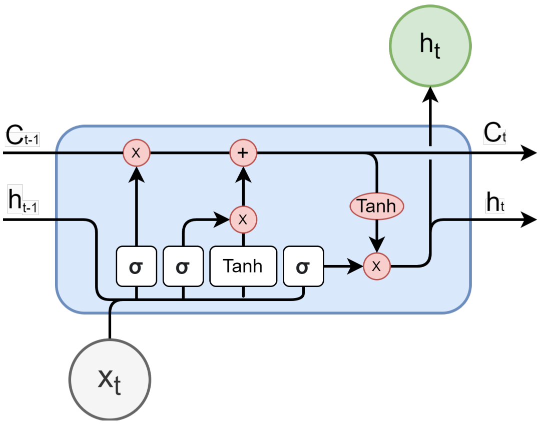


Figure 2.4: A single LSTM cell.

The LSTM network improves on the simple recurrent network by replacing its cell with a more sophisticated one, which Hochreiter and Schmidhuber [1997] refers to as the *memory cell.* As illustrated in Figure 2.4, an LSTM cell is composed of several interconnected components, with the *forget*, *input*, and *output* gate being responsible for regulating the information flow through the cell, enabling selective retention or discarding of information based on its relevance to current data. Specifically, they interact with the cell state C, illustrated as the horizontal line running through the top of the figure, serving as an additional "memory" to better retain

which trains a model once on an accumulated dataset.

relevant information from earlier time steps, reducing the effects of short-term dependencies. This is in contrast to the simple recurrent network, which only relies on a single hidden state.

The gates are represented with sigma (σ), as they specifically utilize the sigmoid activation function (Appendix 1b), where values of 0 block signal pass and 1 allow for signal pass-through, effectively opening and closing gates for different values. Equation 2.3, 2.4, and 2.5 gives the forget (f), input (i), and output (o) gate, respectively, for a time step t.

$$\mathbf{f_t} = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{2.3}$$

$$\mathbf{i_t} = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2.4}$$

$$\mathbf{o_t} = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{2.5}$$

In each gate, the hidden state from the previous cell h_{t-1} and the current input x_t are concatenated before being transformed by the gate's respective weight W and bias b. By subsequently applying a sigmoid activation function to the resulting transformation, the gate yields an activation vector with values between 0 and 1 that best optimizes its overall function in the cell.

From left to right in Figure 2.4, the first cell is the forget gate, which controls which values of the existing cell state that should be maintained or forgotten. The *input gate* then determines the extent to which the incoming information should be incorporated into the cell state. This operation is given in Equation 2.6, where the forget gate discards irrelevant values from the previous cell state before it is added to relevant candidate values $\tilde{C_t}$ from the input. Note that the candidate values have undergone the transformation described in Equation 2.7 before their relevant values are selected by the input gate.

$$C_t = \mathbf{f_t} \odot C_{t-1} + \mathbf{i_t} \odot \tilde{C_t} \tag{2.6}$$

$$\tilde{C_t} = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{2.7}$$

Lastly, the *output gate* modulates the output of the cell, which is derived from a transformation of the updated cell state, given in Equation 2.8, which is also passed on to the next layer for the subsequent time step.

$$h_t = \mathbf{o_t} \odot tanh(C_t) \tag{2.8}$$

These gating mechanisms are important for the network's ability to learn long-range dependencies, as they allow the LSTM to control the information flow adaptively, retaining only the most salient features across time steps. Their attentive design has been successfully applied to a wide range of sequential problems, including time series forecasting problems, demonstrating their ability to effectively model complex, non-linear relationships and adapt to changing dynamics within the data, ultimately leading to their ability to outperform traditional methods [Siami-Namini et al., 2018].

2.5.3 Convolutional Neural Networks

Convolutional Networks (CNNs) offer a different approach to capturing temporal dependencies in time series data compared to recurrent architectures. CNNs employ one-dimensional convolutional layers to capture patterns in time series data, making them suited for forecasting applications [Bai et al., 2018]. This section will elaborate on the underlying concepts and components of the CNN architectures and their application in modeling temporal sequences.

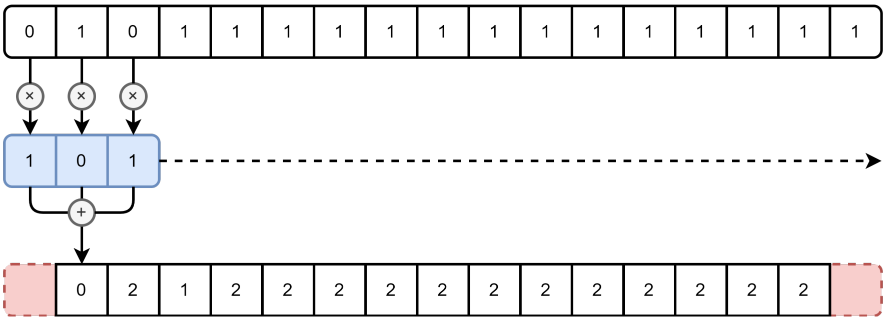


Figure 2.5: Convolution of a vector using a $3x1$ kernel.

CNNs represent a category of deep learning models that draw inspiration from how local receptive fields in neurons of the visual cortex process visual information [Lindsay, 2021]. CNNs have demonstrated remarkable success in various computer vision tasks since their introduction for handwriting recognition by LeCun et al. [1998]. They are comprised of convolutional layers, which convolve a series of filters, referred to as kernels, on the input data. This process facilitates the learning of local patterns, which are further abstracted into spatial hierarchies and concepts. Figure 2.5 illustrates this process, showcasing a vector undergoing convolution as a $3x1$ kernel slides across the input sequence, computing the element-wise product and sum at each position. Consequently, the output vector depicted in the figure exhibits a reduced length compared to the input vector, while potentially containing a higher abstraction of the underlying patterns. This abstraction process is particularly evident in CNNs employed for computer vision tasks; initial layers typically detect edges, which are subsequently abstracted into shapes and concepts by succeeding layers. Equation 2.9 gives the convolution of a vector X of size L using a kernel K of size N.

$$\text{convolved} = X \circledast K_N = \left\{ \sum_{n=0}^{N-1} K[n]X[i+n] \mid i = 0, 1, \ldots, L-1 \right\} \tag{2.9}$$

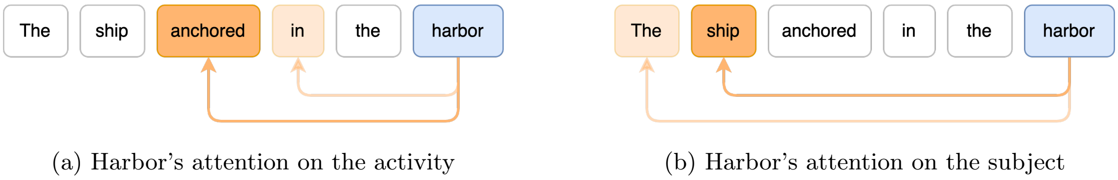
In relation to tabular time series data, one-dimensional (1D) convolutional layers can be employed to learn temporal patterns through the application of convolutions along the time axis.

2.5.4 Transformers

The transformer, introduced by Vaswani et al. [2017], has emerged as a powerful alternative to recurrent and convolutional networks for processing sequential data, excelling at modeling more nuanced and longer-range temporal dependencies using its unique attention mechanisms. While originally proposed for natural language processing (NLP), transformer models have demonstrated exceptional performance for other tasks, including time series forecasting [Han et al., 2022; Zhou et al., 2021]. This section will first cover the fundamental mechanisms of the transformer, followed by an explanation of its overall structure, and finally, an elaboration of recent work tailoring the architecture to forecasting problems.

Attention and Multi-Head Attention

The attention mechanism inherent in transformers allows them to attend to all elements of the input sequence simultaneously, thereby establishing a relationship among these elements that signify their relative importance. To achieve this, each element in the sequence is assigned a distinct weight with respect to all other elements, with the most relevant dependencies having a higher score, helping the transformer focus on the most important information for a specific element. As exemplified in Figure 2.6a and 2.6b, the words relating to the activity occurring in the port and the subject performing the activity might exhibit elevated relevancy for the word "harbor".



(a) Harbor's attention on the activity

(b) Harbor's attention on the subject

Figure 2.6: Hypothetical attention for the word "harbor" for relevant words in a sentence, where more relevant dependencies have a stronger color.

Attention is founded upon three components: the *query*, *key*, and *value* vectors. For a particular element in a sequence, its *query* (Q) representation is used to "query" or "ask" about its relevance with respect to other elements in the sequence, including itself. Its *key* (K) representation serves as an encoding that a compatibility function can utilize together with a query to yield a "relatedness" measurement between two elements. Its *value* (V) representation functions as an internal representation encapsulating its inherent content or information. The lengths of the Q and K vectors are denoted as d_k, while V is denoted as d_v, which signifies the dimensionality of the model (d_{model}) and is frequently proportional to its expressive capacity.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.10}$$

The attention calculation, denoted by Equation 2.10, computes the attention scores by taking the dot product of the Q and K for each element in the sequence, scaling them by $\sqrt{d_k}$, and subsequently applying a softmax normalization. The final output is obtained by calculating the weighted sum of the values based on the attention scores. During calculation, queries, keys, and values are represented as matrices to facilitate concurrent calculation of all elements in the input sequence, resulting in a performant contextualized representation of the entire sequence.

Multi-head attention extends the basic attention mechanism to improve the expressiveness and learning capacity of the model. The primary distinction between multi-head attention and basic attention lies in the application of the attention mechanism multiple times, utilizing different learned linear projections of the Q, K, and V representations. This process computes several sets of attention weights and values for the sequence, allowing the model to focus on different aspects of the input sequence. For instance, Figure 2.10 illustrates two different attentions - one focusing on the subject (2.6b), and the other focusing on the activity (2.6a). When employing multiple heads, the d_{model} is converted to d_k and d_v by dividing the overall d_{model} by the number of parallel attention heads (h). In the original paper, the authors set $d_{model} = 512$ and $h = 8$, resulti $d_k = d_v = d_{model}/h = 64$. Due to the reduced dimension of each head, the total computational cost is comparable to that of single-head attention with full dimensionality, but with the advantage of combined attention to information from different representation subspaces at various positions.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \tag{2.11}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{2.12}$$

Equation 2.11 gives the multi-head attention for Q, K, and V by concatenating h attention heads, given by Equation 2.12. The values for Q, K, V, and the final concatenated output are transformed through the learnable projection matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ respectfully.

Positional Encoding

As the attention mechanism is inherently position-agnostic, positional information must be encoded into the sequence to inform about the relative or absolute positions of elements within the input sequence. To solve this problem, Vaswani et al. [2017] used the periodic sine and cosine functions given in Equation 2.13 and 2.13.

$$PE(pos, 2i) = \sin\left(\frac{pos}{1000^{\frac{2i}{d_{model}}}}\right) \tag{2.13}$$

$$PE(pos, 2i + 1) = \cos\left(\frac{pos}{1000^{\frac{2i}{d_{model}}}}\right) \tag{2.14}$$

Pos denotes the position of a specific element in the sequence, where i is the dimension so that each dimension of the positional encoding corresponds to a sinusoid. The periodic functions allow the generation of unique positional encodings for each element within the input sequence. The generated encodings are then added to the initial embeddings of the input elements, thereby allowing the model to incorporate positional information in subsequent attention computations. A key advantage of this encoding technique is its ability to represent a wide range of positions while maintaining the capacity to differentiate between adjacent positions.

Transformer Architecture

Figure 2.7 illustrates the architecture of the transformer as proposed by Vaswani et al. [2017] in the original paper. The model comprises an encoder and decoder stack, each of which may contain multiple identical layers, where the number of layers is denoted as N (N_e and N_d respectively).

In the *encoder* stack, each layer has two sub-layers; a multi-head attention mechanism followed by position-wise feed-forward networks, with residual connections and layer normalization applied throughout. Residual connections allow the model to maintain information from previous layers more effectively, mitigating the vanishing gradient problem and improving the overall learning capacity of the network. The position-wise feed-forward network is a special type of fully connected feed-forward network that is applied to each position separately and identically, consisting of two linear transformations with a *ReLU* activation (Appendix 1c) in between, as given by Equation 2.15.

$$FFN(x) = ReLU(xW_1 + b_1)W_2 + b_2 \tag{2.15}$$

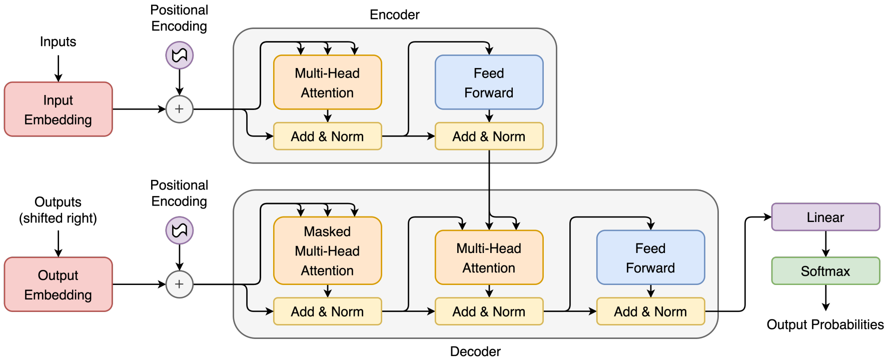


Figure 2.7: Original transformer architecture.

The *decoder* layers share a similar structure, with the incorporation of an encoder-decoder attention mechanism that facilitates interaction between the encoder and decoder stacks, allowing the capture of cross-dependencies. The output from the encoder serves as the K and V values for the secondary attention layer within the decoder, which is processed after the decoder's own multi-head attention layer. The initial attention layer within the decoder is *masked*, specifically designed to mitigate non-causal attention, i.e., the model's ability to access future elements in the sequence during training, as it may result in information leakage and reduced effectiveness in generating coherent predictions in tasks that require autoregressive behavior. As exemplified in Equation 2.16 showing a mask for sequences of length 4, this masking mechanism involves the application of a triangular mask to the attention scores, setting the upper triangular portion of the scores to negative infinity as $softmax(-\infty) = 0$. As such, the masking effectively forces the model to generate predictions based solely on the information available up to the current time step, maintaining its autoregressive nature.

$$mask = \begin{bmatrix} 1 & -\infty & -\infty & -\infty \\ 1 & 1 & -\infty & -\infty \\ 1 & 1 & 1 & -\infty \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{2.16}$$

2.6 Explainable AI

While recent advancements in AI technology have resulted in outstanding accomplishments, pursuing these achievements have often overshadowed the priority for explainability, resulting in complex AI models that achieve exceptional feats but lack transparency in their decision-making. This trend is the result of models' interpretability being inversely proportional to that of their predictive ability, as stated by Molnar [2022] in his book *Interpretable Machine Learning* and further illustrated in Figure 2.8. With new state-of-the-art models getting increasingly complex, driven by the aspiration for improved capabilities, methods for sufficient human interpretation get ever more relevant, especially for higher-risk domains where inaccurate decisions can have excruciating consequences [Molnar, 2022; Doshi-Velez and Kim, 2017; Miller, 2019].

Consequently, the field of Explainable AI (XAI), a subfield focusing on AI's explainability and human interpretation, has become an increasingly popular research area constituting more than 18,000 publications since last year[5] - almost as much as Doshi-Velez and Kim [2017] saw in the past five years in 2017.

The terms *explainability* and *interpretability* are highly interchangeable, but for this thesis, the slightly nuanced definitions by Miller will be adopted. Miller defines *interpretability* as the degree to which a human observer can understand the cause of an AI's decision and *explainability* as the method through which the observer can attain the understanding.

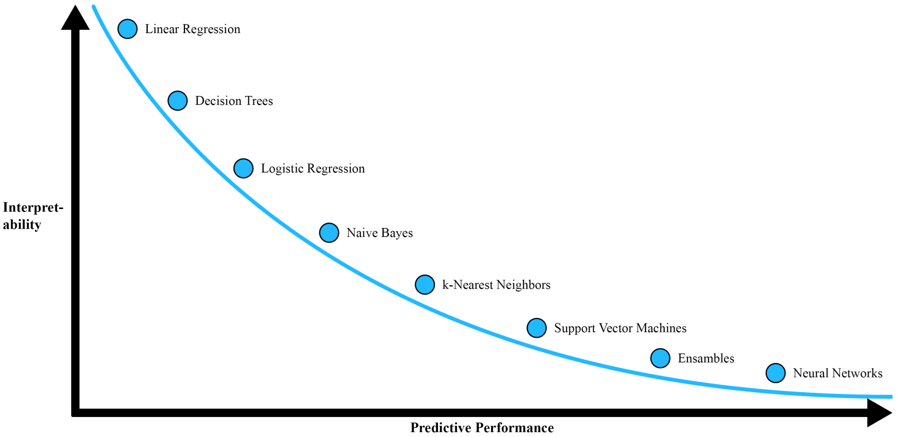


Figure 2.8: The typical relationship between interpretability and performance for different ML methods explained by Molnar [2022].

2.6.1 Importance of Explainable AI

Black box AI has been widely incorporated into various consumer products, including mobile applications, e-commerce platforms, and multimedia streaming services. Despite their lack of interpretability and explanation capabilities, the ramifications of their decisions are often insignificant. However, this is not the case in safety-critical fields, where the consequences of incorrect or biased decisions made by black-box AI can be severe and even life-threatening. Molnar discusses several fields in which interpretation is a vital component of an AI system, like healthcare, finance, and autonomous driving, in which it is imperative that the decisions made by AI systems are explainable, transparent, and can be justified.

While Molnar does not point to any real-world representatives of these scenarios, Samek and Müller [2019] has compiled several cases disclosing *Clever Hans*[6] predictors using XAI methods. For example, the winner of the PASCAL VOC competition [Everingham et al., 2009] excelled due to detecting a copyright watermark consistent in images of horses, the "unique" presence of water in images of boats, and rails in images of trains. So while the model could classify the images correctly, its detection was grounded in recurring contexts in the images that were not directly related to the objects. This tendency is also seen in other cases, such as a model distinguishing between wolves and huskies based on the presence of snow [Ribeiro et al., 2016] and another model categorizing dumbbells by association with the arm lifting it [Mordvintsev

[5] *About* 18,100 results in Google Scholar for "Explainable AI" since 2022

[6] Clever Hans predictors are cases of AI finding spurious correlates in the data to make predictions, originating from the horse *Clever Hans* who was thought to possess counting abilities, but ultimately "counted" by deriving the correct answer from the questioner's reaction.

et al., 2015]. These cases demonstrate how easily AI systems can be misled and that such flaws can stay undetected without interpretation, highlighting the potential limitations of deploying these systems in real-world scenarios.

Although much of the motivation for explainable AI is rooted in the need for trust and accountability in safety-critical domains, it can also facilitate the discovery of novel scientific insights and contribute to advancements in human knowledge. A seminal example of this can be seen in the work of Silver et al. [2016], where their reinforcement agent *AlphaGo*, learning from self-play, played a previously unconsidered "non-human" move in the game of Go, which ultimately proved to be decisive in securing its victory. The discovery of this move challenged the existing understanding of experts in the field and proved the potential of AI to generate new and groundbreaking knowledge. Based on these findings and recent advancements in XAI for more powerful non-linear models, like neural networks, we might be able to develop novel strategies and accelerate future research in other domains such as healthcare, drug development, and material sciences [Samek and Müller, 2019].

2.6.2 Interpretable Models

Interpretable models are a category of ML algorithms whose inner representations are inherently transparent and explainable, often being directly mapped to human-understandable concepts. This key characteristic allows for a high degree of transparency, and their decision-making process can be easily followed and understood by humans. However, due to their inherent interpretability, these models tend to sacrifice some degree of predictive accuracy in exchange for simplicity and interpretability, following the trend highlighted in Figure 2.8. To elucidate further and exemplify the inherent interpretability of these models, the ensuing discussion will focus on linear regression and decision trees, two prototypical models that exemplify this category's characteristics.

Linear Regression

Linear regression is a fundamental yet powerful interpretable model for predicting a continuous variable based on one or more input features. It assumes a linear relationship between the input features and the target variable, attempting to find the optimal set of weights for each feature to minimize the sum of squared residuals between the predicted and target values. Linear regression models have long been favored by researchers addressing quantitative problems, including statisticians and computer scientists.

$$y_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} + \epsilon_i, \quad i = 1, \ldots, N \tag{2.17}$$

Linear regression predicts the target as a weighted sum of the input features, as given in Equation 2.17, where for the i-th sample, y_i is the target variable, $X_{i1}, X_{i2}, \ldots, X_{ip}$ are the input features, β_0 is the intercept term, $\beta_1, \beta_2, \ldots, \beta_p$ are the weights corresponding to each feature, and ϵ_i is the residual error, denoting the difference between the prediction and the actual outcome. The equation is applicable for all samples in the dataset, where the index i ranges from 1 to the number of samples (N). To further illustrate, Figure 2.9 depicts a linear regression for a single feature X_1 using six samples.

Expanding on the interpretability of linear regression models in the context of XAI, one of the primary advantages of these models is their inherent linearity, i.e., that the relationship between input features and the target variables is linear. This characteristic makes estimation procedures simple and allows for an easy-to-understand interpretation at the modular level (i.e., the weights), as they indicate the contribution of each feature in predicting the target variable, allowing for understanding and quantification of the relationship between each feature and the

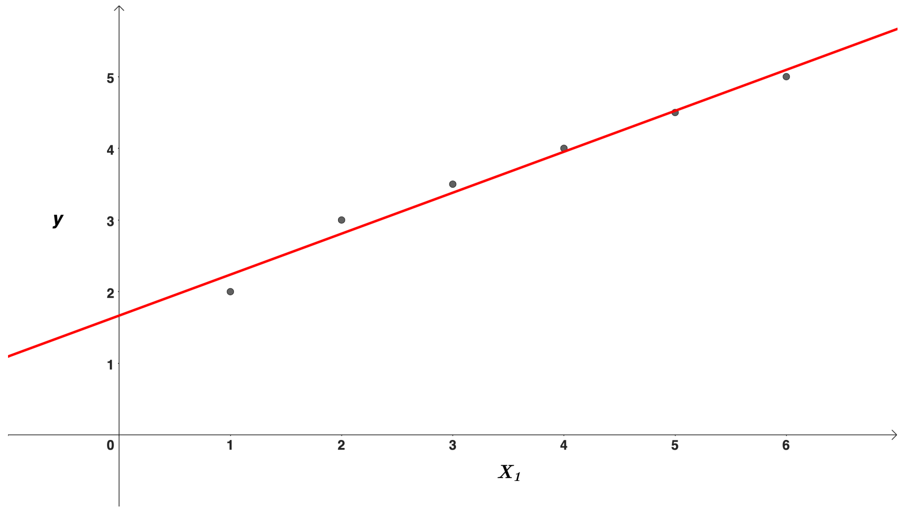
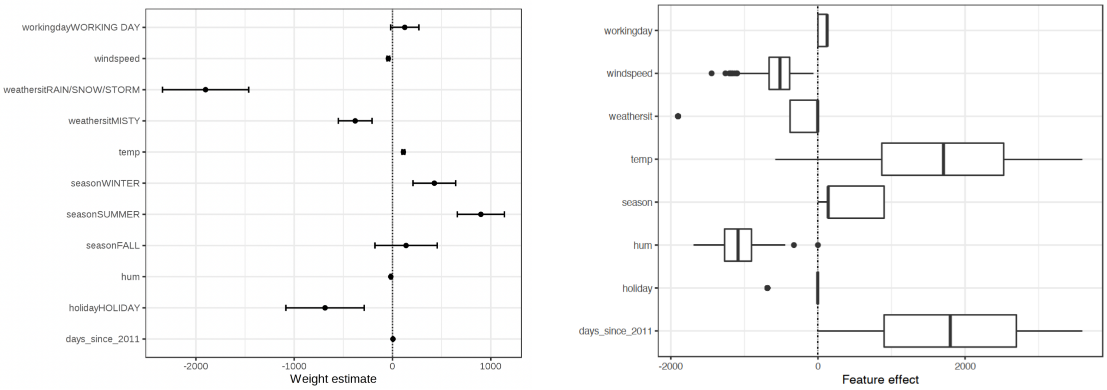


Figure 2.9: Linear regression for one feature (X_1) where $y = 1.67 + 0.57X_1$

target [Molnar, 2022]. Furthermore, these models facilitate the calculation of confidence intervals for the estimated weights, which provide a range for the weight estimates that cover the "true" weight with a certain confidence level, further improving their interpretability.

While humans often struggle to mentally grasp linear models with feature dimensions that exceed the three dimensions of our physical environment, linear models facilitate the generation of various visualizations, such as weight plots and effect plots, which greatly contribute to improved interpretability. Both types of plots allow a better understanding of the contributions of each feature to the predictions in the dataset.



(a) Weight plot.

(b) Effect plot.

Figure 2.10: Linear regression plots from Molnar [2022] for a linear regression model estimating the number of rented bikes on a particular day.

Weight plots provide a visual representation of the weights and their variances, which can be interpreted as the estimated impact of each feature on the response variable. As seen in Figure 2.10a, weights are displayed as points, and the 95% confidence intervals are represented as lines. This plot is easily interpretable, and it clearly indicates that bad weather (especially the *RAIN/SNOW/STORM* feature) has a negative effect on the number of bike rentals, while the *days_ since_ 2011* feature has a negligible impact. These plots may have limitations when features are measured on different scales, which can be addressed by scaling the features before fitting the linear model.

Effect plots, on the other hand, offer an even more meaningful analysis by focusing on the contributions of individual features to the prediction of the response variable. By calculating instance-specific effects, which are obtained by multiplying feature values with their corresponding weights, effect plots facilitate a comparative analysis of the effects for each instance relative to the overall distribution. These plots effectively convey the range of contributions for each feature, offering insights into the relationship between features and the response variable.

Decision Trees

Decision trees establish a hierarchical structure of decisions for predicting a target variable and are applicable to both regression and classification problems. In the context of regression, these decision trees are commonly referred to as regression trees. Figure 2.11 illustrates a typical regression tree, wherein its nodes represent decisions based on the feature values, while the branches represent the potential outcomes of these decisions. To predict the value of a sample, one initiates a traversal at the tree's root node and follows the path of decisions matching the sample's feature values. At the end of the path, the *leaf node* suggests the associated value or class of the sample. It is important to emphasize that Figure 2.11 is simplified to show one value per leaf node; in practice, leaf nodes usually contain a range of values, as samples with similar features in the dataset may end up in the same leaf node but with different values. This is inversely proportional to the complexity of the tree, as a more complex tree allows for more leaf nodes and, thus, fewer accumulated samples per leaf node. Additionally, the number of decisions per decision node could exceed two, depending on the data and algorithm employed to grow the tree.

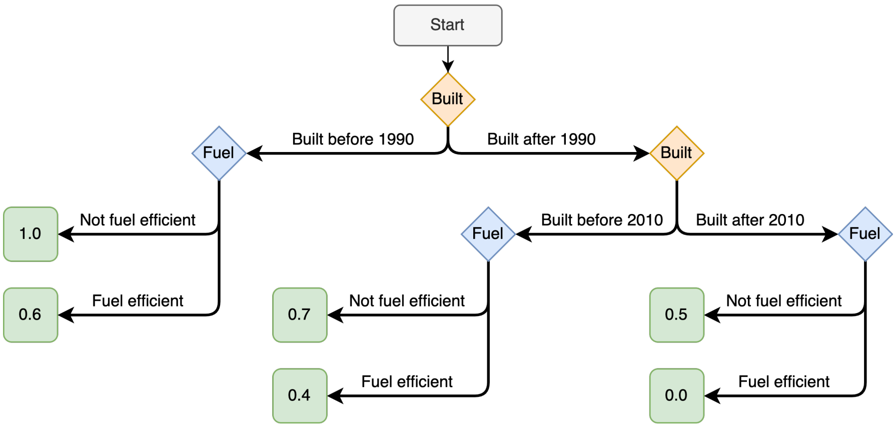


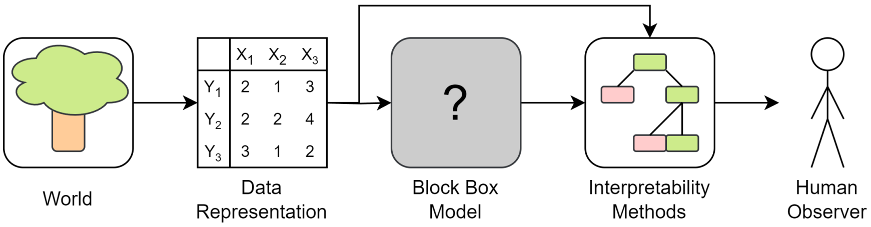
Figure 2.11: A hypothetical regression tree for vessels' daily cost (normalized between 0 and 1) based on its age and fuel efficiency.

Decision trees show good interpretability, as their decision-making process bears a close resemblance to human decision-making, making them easy to follow and comprehend on a per-sample basis. One can argue that decision trees offer better interpretability than linear regression, as the latter relies on a combination of coefficients and variables that can be harder to understand and visualize intuitively, especially as the number of features increases. Moreover, feature importance is inherently baked into their representation; as the tree is constructed, the algorithm prioritizes the most impactful features first. Consequently, features situated higher in the tree hierarchy tend to exert greater influence on the decision-making process. Additionally, decision trees provide a direct way of examining counterfactual explanations, as the exploration of "what if" scenarios is as simple as traversing alternative paths within the tree.

Similarly to linear models, decision trees exhibit a number of limitations affecting their predictive power. For instance, while they succeed in modeling non-linear relationships, they struggle to effectively handle linear relationships, which they approximate through multiple splits, resulting in a suboptimal "stepped" function. To exemplify this issue, consider the case presented in Figure 2.11, wherein the model is unsuccessful in establishing a linear relationship for the build year. As a consequence, a marginal one-year difference between a fuel-efficient ship constructed in 1989 and 1990 leads to a discrepancy in its daily cost of 0.2, which remains constant until the year 2010, after which it experiences a new sudden reduction of 0.4. Another critical limitation is that their interpretability is significantly hurt as their depth increases; the maximum number of leaf nodes in a binary decision tree is 2^D where D is the depth and closely related to the number of features. Thus, a binary tree of depth 6 could have $2^6 = 64$ leaf nodes, making them exponentially hard to interpret.

2.6.3 Model-Agnostic Interpretation

While some models are inherently interpretable, like the linear regression and decision tree models exemplified in Section 2.6.2, they tend to sacrifice predictive ability for their interpretable qualities. Conversely, more powerful models with a high degree of predictive ability, like the ones explained in Section 2.5, are often of a highly complex nature and do not provide adequate means of inherent interpretability.



	X_1	X_2	X_3
Y_1	2	1	3
Y_2	2	2	4
Y_3	3	1	2

Figure 2.12: The overarching process of translating the real world into human-understandable explanations using black box models and model-agnostic methods.

As illustrated in Figure 2.12, model-agnostic methods seek to tackle this problem by separating the explanations from the model, yielding a flexible approach to the interpretation of any model, regardless of its complexity or lack of inherent interpretability. Moreover, having the explanations separated from the models allows for a wider array of explanation forms, which can range from linear formulas to rule sets or feature importance plots.

Model-agnostic methods are usually *post-hoc*, meaning that they are applied after the model has been trained and operate independently of the model. Post-hoc methods generate explanations based on the feature values and the model's subsequent predictions, offering a flexible and wide-reaching approach to interpretation. Therefore, such methods constitute versatile tools for extracting interpretability from already existing complex models rather than needing to design and train new models with intrinsic interpretability.

Moreover, model-agnostic explanations can generally be framed within two main scopes, namely *local* and *global* explanations, facilitating different granularity of explanations.

Local Explanations

Local explanations focus on explaining an AI system's decision on the micro level, facilitating the interpretation of the system's decisions for a specific instance or prediction. In the context of AIS-based forecasting, local explanations could clarify why a model predicted a specific change in value for a financial instrument given a specific set of variables found in the AIS data. For instance, given a sequence of AIS observations and a model's estimate of a sudden drop in value, local explanations can help pinpoint which days and what variables, such as fleet speed and draft, were influential for the model's decision. Despite the detailed insights provided by local explanations, their applicability may not extend beyond the specific instances they explain, a factor warranting careful interpretation to avoid over-generalization.

Global Explanations

On the other hand, global explanations focus on the macro level, providing an overall understanding of the model's decision-making process. Global explanations offer an overview of how different features interact and impact each other across the model, contributing to an understanding of the model's overall strategy, important features, general trends, and biases. By considering global explanations, one can discern how the model consistently evaluates different features, such as attributes related to vessel movement, route patterns, and the temporal significance of past days, in order to predict future movements in a given instrument. While global explanations play a vital role in comprehending the broad behavior of the model, their comprehensive nature entails an elevated level of intricacy. Moreover, instances where local explanations substantially diverge from global behavior highlight the significance of both explanation types in attaining a well-rounded understanding of AI systems.

2.6.4 Model-Specific Interpretation

Model-specific methods constitute an approach that can be regarded as bridging the gap between intrinsically interpretable and model-agnostic methods by utilizing the internal structure of the specific model they are designed for to generate explanations. As opposed to the generality of model-agnostic methods, model-specific methods are deeply entrenched in the model's architecture, exploiting the internal mechanisms of the model to produce explanations.

While the model-specific approach bears a close resemblance to that of interpretable models, the main distinction is that the models themselves are not necessarily interpretable. Instead, these methods are "affixed" to the model, either intrinsically or post-hoc, to abstract the complex internal structures and processes into intelligible explanations, akin to those produced by model-agnostic methods. For instance, in the context of AIS-based forecasting with deep learning models, model-specific methods could elucidate the importance of specific time steps or specific AIS attributes by examining the model's internal state, such as hidden states in RNNs, feature maps in CNNs, or attention weights in Transformers.

However, although model-specific interpretation methods can provide detailed insights into the decision-making process of complex models, they are tied to the specific type of model used, limiting their applicability to other models and impede an agile interchange of models.

2.6.5 Evaluation of Explanations

Assessing the efficacy and accuracy of explanations is integral in ensuring the reliability, validity, and interpretability of AI systems. In the end, the primary purpose of explanations is to communicate the decision-making processes of AI systems in a comprehensible manner to human users. However, the evaluation of such explanations, which hinges on subjectivity and context, presents several challenges, and, as expressed by Molnar [2022], there is no real consensus on how to measure interpretability. That being said, he points to research by Doshi-Velez and Kim [2017] that proposes three main approaches to the evaluation of interpretability. Firstly, a *real-task domain expert evaluation* involves domain experts evaluating the explanations as given in the end product, leveraging the expertise of domain specialists in evaluating interpretability. Secondly, a *simplified layperson evaluation* involves non-experts evaluating the comprehensibility of simplified explanations, which acknowledges the importance of making explanations accessible to individuals without specialized knowledge. Lastly, a *function-level proxy evaluation* adopts metrics that have been collectively agreed upon as effective in quantifying interpretability. For instance, the interpretability of decision trees can be evaluated by their depth, as shorter trees are easier to understand.

Measuring Explanation Efficacy

An evaluation of explanation methods, i.e., the algorithms that generate explanations, can be derived from their various properties, including expressive power, translucency, portability, and algorithmic complexity [Molnar, 2022]. The *expressive power* of an explanation method refers to its ability to convey the logic and reasoning behind a model's decisions, which can be gauged by examining how effectively it communicates the relationship between features and predictions. *Translucency*, on the other hand, indicates the extent to which the explanation method analyzes the underlying ML model's parameters. An effective explanation should balance high and low translucency - the former providing more information to generate explanations and the latter ensuring the method's portability across different models. *Portability* is a measure of the range of ML models compatible with the explanation method. Greater portability implies broader applicability, hence potentially increasing the method's efficiency. The *algorithmic complexity* of an explanation method, which pertains to its computational demands, is also a critical consideration, particularly when computation time is a constraint. An efficient explanation method should strike a balance between providing insightful explanations and managing computational complexity.

Similarly, the explanations themselves also merit evaluation based on several properties, each contributing to their quality and effectiveness. Molnar [2022] outlines these properties as accuracy, fidelity, consistency, stability, comprehensibility, certainty, importance, novelty, and representativeness. *Accuracy* assesses the effectiveness of an explanation in terms of its capacity to predict unseen data accurately. It could be quantitatively gauged by using a separate test dataset to ascertain how successfully the explanation can predict outcomes. However, Molnar [2022] stresses that the appropriateness of accuracy as a measurement could be contextual, varying based on the accuracy of the underlying model. *Fidelity*, which can either be global or local, is paramount, as it assesses the extent to which an explanation mirrors the model's prediction; an explanation with low fidelity fails in its primary objective of illuminating the workings of the ML model. *Consistency* and *stability* both provide comparative analysis, though in differing contexts. Consistency is concerned with the degree of similarity between explanations when different models trained on the same task are employed. The desirable level of consistency can be tricky and could depend on whether the models utilize similar or different features to arrive at similar predictions. Conversely, stability pertains to how much the explanations for similar instances vary within a single model. High stability, indicating minor changes in explanation, even with slight feature variations, is generally desirable. *Comprehensibility*, while challenging to quantify, remains a critical determinant of an explanation's utility, gauging its understandability to humans. *Certainty* measures an explanation's capacity to reflect the model's prediction confidence, thus contributing to understanding the reliability of predictions. The *importance* assesses an explanation's ability to depict the significance of features, offering a deeper insight into which features contributed the most to the prediction. *Novelty* measures the explanatory capacity of determining whether an instance originates from a region distinct from the training data distribution, while *representativeness* evaluates the breadth of instances an explanation can cover, covering either the entire model or individual predictions.

Chapter 3

State of the Art

In this chapter, the current literature and research in the field of multi-feature time series prediction and AI interpretation will be presented. The focus will be on studies and methodologies relevant to the research objectives outlined in Section 1.3. The intent is to illuminate the current state of these research areas and identify applicable conclusions and approaches.

Of particular interest is research within the distinct scope of AIS-based financial forecasting, owing to its direct relevance to the research goals, which will be explored in Section 3.1. In Section 3.3, a wider array of AIS-based forecasting research will be considered; while these studies may not be immediately related to financial forecasting, their application to AIS data could potentially yield additional insights or approaches to AIS-based financial forecasting. Subsequently, the scope will be broadened to incorporate a wider range of research within multi-feature sequence modeling in Section 3.2. This expansion aims to supplement potential gaps in existing AIS-based research with newer relevant approaches and methodologies that have yet to be applied in this specific domain. A variety of modeling techniques will be explored, with an emphasis on those that comply with the constraints set by the research goal and have demonstrated efficacy when dealing with the complex, high-dimensionality temporal data that can be found in the AIS. Finally, XAI research proposing or evaluating methods for the resulting applicable models will be presented in Section 3.4.

3.1 AIS-Based Financial Forecasting

The utilization of AIS data for forecasting financial instruments within the maritime cluster is a relatively under-explored area within the field of academic research. However, it has gained increasing attention in recent years; a small but growing number of studies have employed ML models, primarily recurrent neural networks, to predict financial indices and freight rates using AIS and other relevant statistical data. Although additional work exists relating to financial forecasting, including ones related to maritime industries, this section will specifically focus on those where AIS is the main focus of the research. By examining such studies, we aim to provide an overview of existing work that closely aligns with our research objective of investigating the potential of AIS data in forecasting financial instruments in the maritime domain.

3.1.1 Forecasting the Baltic Dry Index

Research conducted by Kanamoto et al. [2019] and Sarantopoulos [2021] examines the viability of incorporating AIS data in conjunction with financial statistics to improve the predictability of the Baltic Dry Index (BDI). The BDI is a prominent financial index that serves as a benchmark for the cost of shipping unpacked raw materials, such as coal, iron ore, and grain. It is a critical indicator of the global shipping market's overall health and primarily reflects the costs of transporting these commodities on dry bulk vessels, such as the Capesize, Panamax, and Supramax segments.

Kanamoto et al. [2019] examined the Baltic Capesize Index (BCI) specifically, which is a subset of the BDI for the Capesize segment, i.e., dry bulk vessels with a deadweight capacity exceeding 180,000 tons. The study employed a baseline dataset consisting of ten fundamental macroeconomic variables, including prices of steel, iron ore, crude oil, as well as the BCI itself. The authors then investigated the effects of incorporating AIS data, partitioned into four distinct geographic regions, namely the Indian Ocean, Australia, Brazil, and the world at large, as illustrated in Figure 3.1.

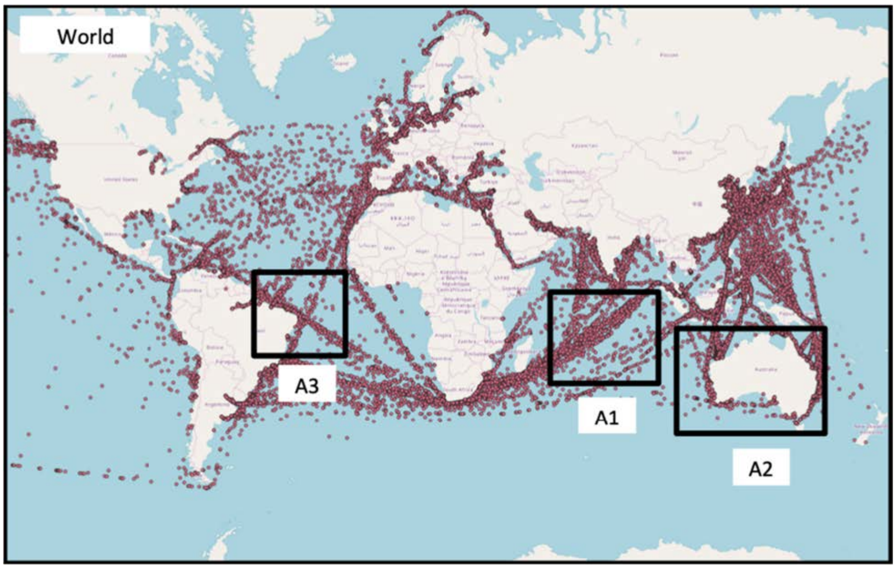


Figure 3.1: Target regions employed by Kanamoto et al. [2019] (from the original paper).

To ensure higher data reliability, a speed condition was imposed, which limited the dataset to vessels with a reasonable range of speeds between 3 and 30 knots. Moreover, an additional condition was applied to the vessels' course to restrict the analysis to those en route to import countries, such as Japan, China, and Australia. For each of the regions, the authors computed the vessels' average speed, the sum of deadweight, and the sum of the product of deadweight and speed, which were then incorporated into the features. These calculations were incorporated as 14-day moving averages to mitigate the impact of daily fluctuations and noise in the raw S-AIS data, which also helps encapsulate longer-term trends. When constructing their model, the authors *opted for a deep neural network instead of a recurrent one* due to their limited dataset size of 900 samples (639 training, 159 validation, 102 testing).

Their model estimated the value for BCI_{t+30} using nine hidden layers and incorporated the sequential inputs $D_{t-60}, \ldots, D_t$ as illustrated in Figure 3.2, where D_t represents the data at time t and consists of $\{S_t, Io_t, Au_t, Br_t, Wo_t\}$, wherein S denotes the set of macroeconomic variables, and the remaining terms represent the aggregated AIS data for each of the four regions for the given timestep t. While the model estimates the absolute value of the BCI, the authors

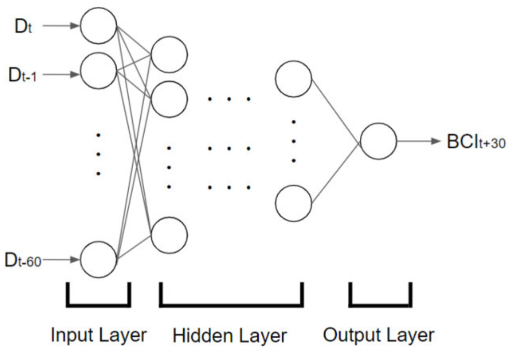


Figure 3.2: Structure of the network employed by Kanamoto et al. [2019] (from the original paper).

emphasize that the differential value $\Delta BCI_{t+30} = BCI_{t+30} - BCI_t$ is more important, and used this as the preferred evaluation for the model. Additionally, due to the high number of features when $|D_t| = 21$ and $N = 60$ ($21 \times 61 = 1281$ features), the authors used the Maximal Information Coefficient measurement to remove the least relevant features to mitigate potential model overfitting with excessive input variables. They found that increasing the number of features led to improved results on the test dataset but worsened results on the validation set. Specifically, an increase in the number of features from 315 to 1187 resulted in an increase in the validation RMSE from 237 to 371. In contrast, the test set RMSE decreased from 712 to 554. Concluding their study, the authors saw a **clear improvement in accuracy when including AIS data**, with an RMSE reduction from 1124 to 554 on their test dataset.

Another study with a similar objective conducted by Sarantopoulos [2021] combined macroeconomic and shipping-related variables to forecast the BDI. Although this study does *not use data derived from the AIS directly*, it does incorporate data about vessel scrap value, time charter rates for 32,000 deadweight ships, and the Bulkvarrier Newbuilding Price Index into various RNN models. Among the models tested, the LSTM model demonstrated the best performance. Additionally, the paper presents an effective baseline ARIMA model that can be utilized as a benchmark for BDI forecasting.

3.1.2 Forecasting Freight Rates

In a study conducted by Næss [2018], it was observed that the accuracy of an LSTM model in predicting short-term fluctuations in freight rates **improved with the inclusion of AIS data**. Specifically, the AIS-based LSTM model outperformed three other models, namely, VAR, MLP, and a persistence baseline model. Despite the noted improvement, the difference in performance between the AIS and non-AIS LSTM was **not significant**, with a reduction in RMSE from 13.03 to 12.46. The study focused on the LPG shipping market, and specifically, the propane spot price in Mont Belvieu, a major trading and storage hub for natural gas liquids in Texas, USA, that is often used as a benchmark for the global LPG market. The models they tested were tasked with predicting $t + 1$ using $t - 20, \ldots, t$, where the temporal resolution of t was one week, and the target was the weekly average of the spot price. For input features, price and market features such as the oil price and various other LPG spot prices were included, in addition to several features derived from the AIS data, which, similarly to Kanamoto et al. [2019], was aggregated into different regions. In the case of this study, the regions of interest were the *Atlantic*, *Far East*, *Arabian Gulf*, *East Pacific*, *North West Europe*, *Indian Ocean*, and the *Mediterranean*. The derived AIS data included vessel counts and overall capacity in the regions to understand the supply-demand relationship in the market, sailing distance from a vessel's current position to a particular area or port to comprehend the implications of global vessel positioning and the effects of geographical price arbitrage, flux in and out of regions to provide additional market insight, and fleet sailing speed and variance to reflect market rates.

Similar research by Århus and Salen [2018] focused on the Tanker market and the freight rates for oil transport between the Arabian Gulf and Singapore, also intending to assess the potential improvement in predictive performance when including AIS data. Their foundational non-AIS features consisted of the freight rate itself, variables related to supply/demand, and a range of macroeconomic variables. To model supply, they used the bunker price in Singapore, while an approximation of refinery profitability in Asia, i.e., the difference between petrol price in Singapore and crude oil price in Dubai, was used to model demand. Their macroeconomic variables consisted of the Baltic Exchange Dirty Tanker Index (BDTI), U.S. dollar - Saudi Riyal exchange rate, U.S. dollar - Singapore dollar exchange rate, and an oil spread variable modeling the spread between Dubail crude oil 1-month and 3-month futures. For the AIS-derived data, they looked at variables believed to explain fleet productivity (supply), fleet activity (supply), and overall tonne-mile demand. Fleet productivity was modeled by deriving the average speed and load factor of both the global Very Large Crude Carrier (VLCC) fleet and the VLCCs in the vicinity of and between the two ports. The fleet activity was modeled by calculating the number of VLCCs in both ports, the number of VLCCs en route to both ports, and the number of VLCCs in the area between the two ports. Tonne-mile demand was modeled by aggregating the tonne-mile demand for all VLCCs heading to Singapore, as well as for VLCCs heading to other major ports (Arabian Gulf, China, Japan, USA, and West Africa). These 17 features were then utilized in an LSTM model to forecast the freight rate for $t+1$, $t+5$, and $t+10$, testing a variety of different feature configurations using a rolling window of $t-21,\dots,t$. The LSTM model was benchmarked against a multivariate linear regression model, and the results indicated that the LSTM outperformed the benchmark for the $t+10$ time horizon, although the baseline model was competitive at shorter horizons. Interestingly, **the inclusion of AIS features did not yield a significant improvement in performance**. The authors outlined several factors for future investigation, including the quality of the AIS data and possible refinements to the model itself.

3.2 Multivariable Sequence Modeling

Despite the early stages of academic literature devoted to AIS-driven machine learning for financial forecasting specifically, the superset of multivariable sequence modeling problems offers a plethora of auxiliary models and methodologies that demonstrate applicability to the problem. Numerous studies have experimented with multivariate regressive models tailored to temporal sequences and have seen success in many domains, such as weather forecasting, traffic prediction, and energy consumption, rendering them suitable candidates for our problem.

This section provides an overview of the current landscape within the academic literature concerning multivariable sequence forecasting, with an emphasis on research that is applicable to the problem at hand. The intention behind this overview is to extend the toolset of successful sequence modeling techniques that may prove proficient in exploiting the predictive capacity of AIS data, even when constrained by a limited dataset consisting of large vectors and a shortened time horizon.

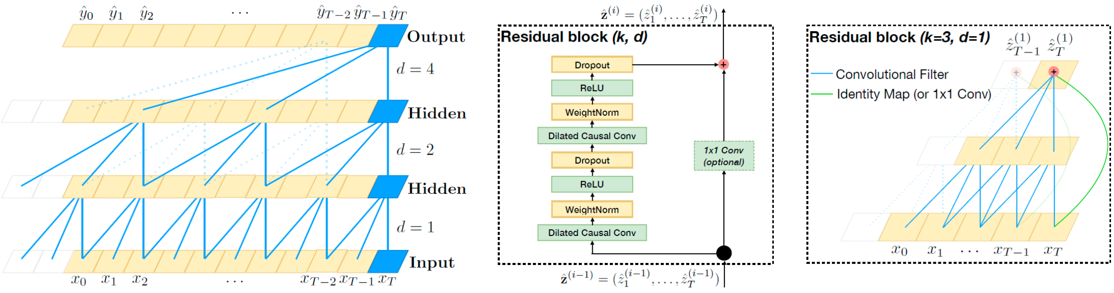
3.2.1 Recurrent Neural Networks

Recurrent Neural Networks, particularly their more advanced variations such as LSTMs, have established themselves as the backbone of sequence modeling for years. Even in the current era of rapid advancements in AI and machine learning, these networks continue to deliver results that are often considered best in class, a testament to their robustness and versatility [Smagulova and James, 2019; Tax et al., 2020]. The relevance and applicability of RNNs, primarily LSTMs, are clearly observable in the existing literature specifically related to AIS-based financial forecasting, as discussed in Section 3.1. These models have been heavily employed in research within this niche, indicating their suitability for handling the intricacies of the problem at hand.

3.2.2 Temporal Convolutional Networks

Convolutional Neural Networks (CNNs), initially purposed for modeling spatial hierarchies in spatially correlated data such as images, have demonstrated great efficacy in sequence modeling. For instance, early works from the likes of Waibel et al. [1989] confirmed the practicality of CNNs in the domain of speech recognition, and Collobert and Weston [2008] highlighted the proficiency of CNNs across a wide range of language processing tasks. Currently, the application of CNNs has expanded to a vast collection of sequence modeling tasks, such as natural language processing, machine translation, audio synthesis, and video-based action segmentation, as substantiated by the works of Dos Santos and Zadrozny [2014]. Kalchbrenner et al. [2014], Zhang et al. [2015], Gehring et al. [2017]. Oord et al. [2016], Dauphin et al. [2017], and Lea et al. [2017].

The work of Bai et al. [2018] presented a convolutional architecture for sequence prediction, which was pitted against prevalent canonical sequence models, such as the LSTM model, on various benchmarks commonly used to evaluate recurrent networks. Their devised model, referred to as a Temporal Convolutional Network (TCN), is rooted in the principle of simplicity and generality, drawing from the successful best practices and network designs from recent research within convolutional models. Their design yielded a sequence-to-sequence model, deploying causal and dilated convolutions to generate an equal-length output sequence from the input without allowing future data to leak into the past. Figure 3.3 illustrates the architectural elements of their model.



$\hat{y}_0$ $\hat{y}_1$ $\hat{y}_2$ $\cdots$ $\hat{y}_{T-2}$ $\hat{y}_{T-1}$ $\hat{y}_T$
Output
$d = 4$
Hidden
$d = 2$
Hidden
$d = 1$
Input
x_0 x_1 x_2 $\cdots$ x_{T-2} x_{T-1} x_T
$\mathbf{z}^{(i)} = (\hat{z}_1^{(i)}, \ldots, \hat{z}_T^{(i)})$
Residual block (k, d)
Dropout
ReLU
WeightNorm
Dilated Causal Conv
Dropout
ReLU
WeightNorm
Dilated Causal Conv
1x1 Conv (optional)
$\mathbf{z}^{(i-1)} = (\hat{z}_1^{(i-1)}, \ldots, \hat{z}_T^{(i-1)})$
Residual block (k=3, d=1)
$\hat{z}_{T-1}^{(1)}$ $\hat{z}_T^{(1)}$
Convolutional Filter
Identity Map (or 1x1 Conv)
x_0 x_1 $\cdots$ x_{T-1} x_T

(a) Dilated causal convolutions (b) Residual block (c) Residual connection

Figure 3.3: Architectural elements of the TCN proposed by Bai et al. [2018] (from the original paper).

The proposed TCN by Bai et al. [2018] hinges upon three central principles: causality, dilation, and residuals. Causality, as depicted in Figure 3.3a, is maintained through the implementation of no-leaking convolution on equal-length outputs. This is achieved via a 1D fully-convolutional network, wherein each layer retains the length of the input layer, convolving an output at time t only with elements from time t and prior in the preceding layers. Dilated convolutions improve the model's ability to capture longer historical contexts efficiently, considering that a convolutional network's access to earlier timesteps is linearly proportional to the depth of the network without dilation. This is evident in Figure 3.3a, demonstrating three layers with distinct dilation factors $d = 1, 2, 4$ and a kernel size $k = 3$. Finally, the model implements residual connections, as shown in Figures 3.3b and 3.3c, to improve the stability and convergence speed of the model, particularly for deeper networks. Each residual block comprises two dilated convolutional layers, each subject to weight normalization before a non-linear ReLU activation. The output from each activation undergoes dropout regularization, ensuring the whole channel is zeroed out at each training step. This transformation is then combined with the input residual via a $1x1$ convolution instead of direct addition to account for variations in input-output widths.

The primary findings from their research underlined the superiority of TCNs over RNNs in terms of processing speed, attributed to the parallel nature of CNNs, better horizon size control via dilation and layer depth parameters, stability of gradients, and compatibility with variable length inputs. However, TCNs were found to demand higher memory and necessitate parameter modifications for domain transfers compared to RNNs. Their TCN was able to outperform an LSTM, GRU, and RNN model for a majority of sequence modeling tasks, with the exception of a Word-level PTB task that required a large model size (13M), in which an LSTM proved superior.

Research conducted by Yan et al. [2020] used a similar model to Bai et al. [2018] and found similar results; the TCN saw success in predicting ENSO[1] 1, 3, 6, and 12 months into the future, yielding a better overall prediction than LSTM.

3.2.3 Transformers

The landscape of Natural Language Processing (NLP) was radically transformed following the recent advent of transformer models and attention-based architectures, as proposed in the seminal paper by Vaswani et al. [2017]. The subsequent emergence of large language models, such as BERT [Devlin et al., 2018] and the GPT model series [Radford et al., 2018], has further extended the scope of capabilities within the NLP field. These models have demonstrated unprecedented progress in a broad range of NLP tasks, from sentiment analysis and machine translation to more complex tasks like text summarization and generation of human-like conversational responses. While originally introduced for NLP tasks, transformers have seen a significant adoption for multivariable sequence forecasting, covering domains such as traffic flow [Reza et al., 2022; Chen et al., 2022; Xu et al., 2020], energy-related problems [L'Heureux et al., 2022; Qu et al., 2022; Zhao et al., 2021], and disease outbreaks [Li et al., 2022b; Wu et al., 2020].

In their evaluation of the Transformer architecture, Lara-Benítez et al. [2021] conducted a comparison with LSTM and CNN networks, examining both accuracy and computational efficiency in univariate time series forecasting across 12 datasets. Their adaptation of the Transformer model removes the encoder present in the original design. Instead, they introduce the positionally-encoded inputs directly to each decoder in a decoder stack. Their results indicate that the Transformer model achieved state-of-the-art forecasting accuracy similar to that of the LSTMs while outperforming the CNNs. The Transformer model was faster to train than LSTMs but was significantly slower than the CNN networks. Additionally, they found that the inference time was slow due to the single-step prediction scheme used, making inference significantly slower than the other architectures.

[1] The El Niño-Southern Oscillation (ENSO) is a climatic phenomenon in the tropical Pacific Ocean characterized by periodic variations in sea surface temperatures and atmospheric pressure.

Conversely, another study by Zerveas et al. [2021] proposing a generic Transformer-based framework for multivariate problems consistently saw superior results from their Transformer-based model, outperforming other state-of-the-art models such as LSTMs, XGBoost, and Rocket, **even with a limited set of training samples**. Their work offers a counterpoint to Lara-Benítez et al. [2021], who advocated a decoder-only architectural approach. In contrast, Zerveas et al. [2021] adopted an encoder-only model, arguing that the decoder module is predominantly suited for generative tasks, whereas the encoder module allows for versatile and adequately adaptable addressing of a broader range of tasks, including classification, regression, imputation, and also generative problems like forecasting. Additionally, this approach incidentally eliminates approximately half of the model parameters, producing advantageous computational and learning implications.

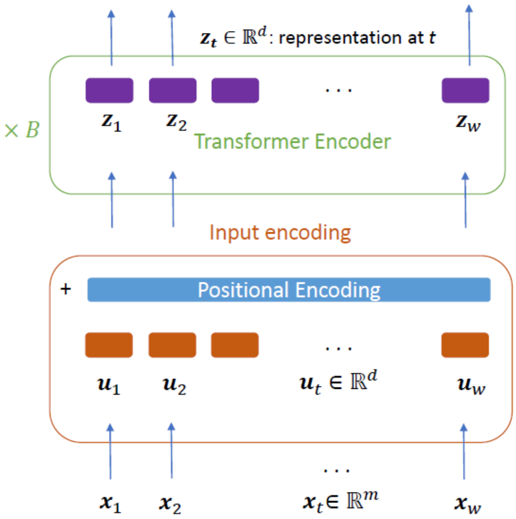


Figure 3.4: Generic Transformer-based architecture employed by Zerveas et al. [2021] (from the original paper).

Their architecture, as illustrated in Figure 3.4, employs a stack of Transformer encoders to model the temporal dependencies within a training sample X, where $x_t \in \mathbb{R}^m : X \in \mathbb{R}^{w \times m} = [x_1, x_2, \ldots, x_w]$. Each training sample has w historical observation vectors x_t consisting of m variables. Each observation x_t is first linearly projected onto a d-dimensional vector space through a linear transformation $u_t = W_p x_t + b_p$, where d is the dimension of the transformer model sequence element representation (model dimension). A positional encoding is then added to the vector representation u_t, making up the input to the encoder, corresponding to the word vectors found in the NLP transformers. It is worth noting that **the authors observed improved performance for all datasets using a fully learnable positional encoding instead of the deterministic sinusoidal encodings** found in the original architecture. Moreover, the authors used batch normalization after the encoder block instead of layer normalization, seeking to mitigate the effect of outlier values in time series not found in NLP word embeddings. The output from the encoder $\bar{z} \in \mathbb{R}^{d \times w} = [z_1, \ldots, z_w]$ is versatile and can be used for several objectives. For instance, for regression and classification tasks, the authors simply employed a linear transformation $\hat{y} = W_o \bar{z} + b_o$.

3.3 Broader AIS-Based Research

Despite the scarcity of research into the application of AIS for financial forecasting specifically, numerous studies have delved into the employment of AIS in ML models for alternative objectives. A vast majority of these AIS-centric machine learning studies have focused on analyzing shipping networks and patterns, destination prediction, and estimation of ship movements and trajectories. This section will elaborate on relevant research that employs AIS data for purposes beyond financial forecasting, with the intent of uncovering additional approaches to AIS incorporation, quality mitigation, and the exploration of alternative models that may prove advantageous for the objective of this thesis.

3.3.1 Two-Dimensional AIS Representations With CNNs

CNNs have demonstrated exceptional performance in various image recognition and processing tasks by effectively identifying patterns and extracting features from two-dimensional (2D) data. In the realm of AIS-based research, some studies have employed CNNs to analyze AIS data by transforming it into a 2D representation, capitalizing on the inherent spatial nature of AIS data. Given that AIS conveys positional information for vessels, it is reasonable to represent this data on a cartographic projection, allowing researchers to interpret AIS data as an image wherein each element of the grid can encapsulate multiple channels of information analogous to color channels in traditional image data. This approach facilitates the application of well-established image processing techniques, such as CNNs, for the analysis and processing of AIS data.

Chen et al. [2020] devised a Ship Movement Image Generation and Labelling (SMIGL) process for sampling vessels' AIS data into images, which they subsequently fed into a CNN tasked with categorizing their movement behavior. The images were derived from the SMIGL process by sampling the vessels' trajectories over fixed time intervals, which the authors claim is beneficial in mitigating the long-tail effect [Bellingham et al., 2010]. Each trajectory sample was processed into a 244x244 image, and the vessels' positions for the trajectory were rasterized into the image, such that each pixel represented a spatial representation of the trajectory. This rasterization process is illustrated in Figure 3.5.

The red, green, and blue color values of the pixels were used to decode different information about the movements. Specifically, red pixels signified sections of the trajectory where the vessel was stationary, green pixels denoted normal navigation states, and blue pixels indicated a state of maneuvering. This information was inferred from changes in the vessels' course and speed. A significant change in course between AIS points suggested that a vessel was maneuvering (blue), high speeds signified a normal navigation state (green), and slower speeds indicated stationary behavior (red). Their results show that the CNN was superior over other classification algorithms (K-nearest neighbor, support vector machine, and decision trees), with an average F1 score of 76.38%.

3.3.2 LSTMs with Convolutions for Improved Feature Extraction

Syed and Ahmed [2023] derived a model that combines a 1D CNN layer with an LSTM architecture to improve the *marine vessel track association process*. This process is a critical process in marine traffic monitoring with the objective of grouping successive positional reports as tracks and connecting them to their corresponding vessel. With the spatiotemporal nature of AIS data, the underlying premise of their approach was that the application of a 1D CNN layer before LSTM layers leverage the spatial modeling capabilities of the CNN in combination with the LSTM's proficiency in modeling temporal contexts. Consequently, their approach aims to improve the accuracy and robustness of the model's predictions by exploiting the complementary

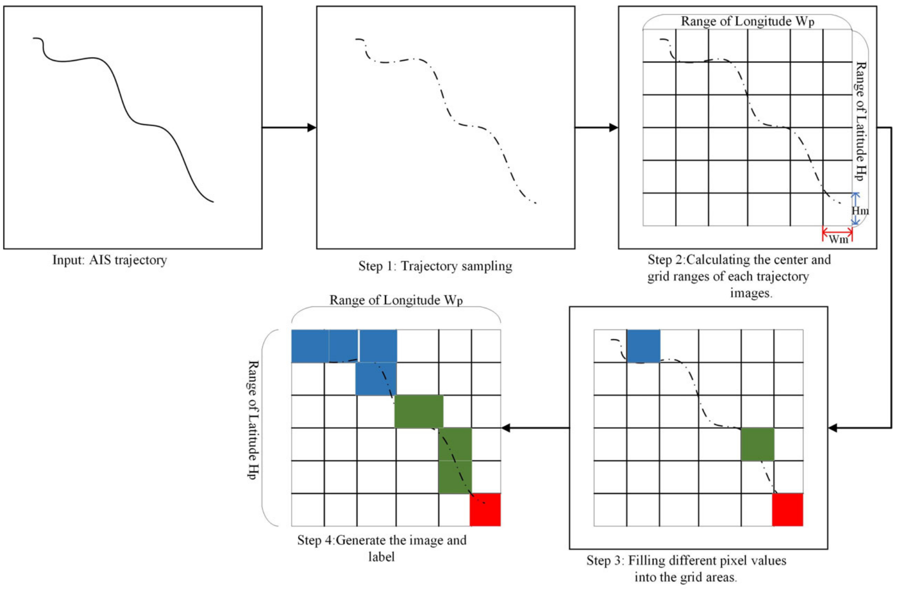

Input: AIS trajectory
Step 1: Trajectory sampling
Range of Longitude Wp
Range of Latitude Hp
Hm
Wm
Step 2:Calculating the center and grid ranges of each trajectory images.
Step 3: Filling different pixel values into the grid areas.
Range of Longitude Wp
Range of Latitude Hp
Step 4:Generate the image and label

Figure 3.5: Illustration of the rasterization process employed by Syed and Ahmed [2023] (from the original paper).

strengths of the two architectures, treating the track association process as a multivariate time series problem for the multiple vessels.

VID	SEQUENCE_DTTM	LAT	LON	SPEED	COURSE
vessel 1	2020-02-29T22:00:01Z	37.8567	23.5374	0	0
vessel 2	2020-02-29T22:00:01Z	37.9483	23.6411	0	349.9
vessel 3	2020-02-29T22:00:02Z	37.9315	23.6804	0.1	170.1

Table 3.1: Structure of AIS data used by Syed and Ahmed [2023] (values adjusted).

From their dataset, as exemplified in Table 3.1, each row is treated as an observation in a time series. They assigned the vessel IDs (VID) as target variables and used the geographical information (LON and LAT) and dynamic parameters (SPEED and COURSE) as the primary input parameters for the model. They emphasize that many of the vessels in the training dataset did not include a sufficient number of reports, and the number of viable vessels in the dataset was, therefore, reduced from 327 to 23.

The structure of the CNN-LSTM model, as depicted in Figure 3.6, is divided into two main parts. The initial part applies the 1D CNN layer to recognize spatial dependencies. The spatial features are subsequently forwarded to the second part, where they are processed by two LSTM layers, which capture the temporal dynamics within the processed spatial features. The data is then fed into a fully connected layer before a softmax layer with an output for each vessel.

Another study conducted by Spadon et al. [2022] also explored the use of a CNN-RNN architecture; however, for the purpose of predicting the content of forthcoming AIS messages from multiple vessels simultaneously despite temporal irregularities. In their work, they conducted a comprehensive experiment of many models, and their CNN-RNN architecture was tested for

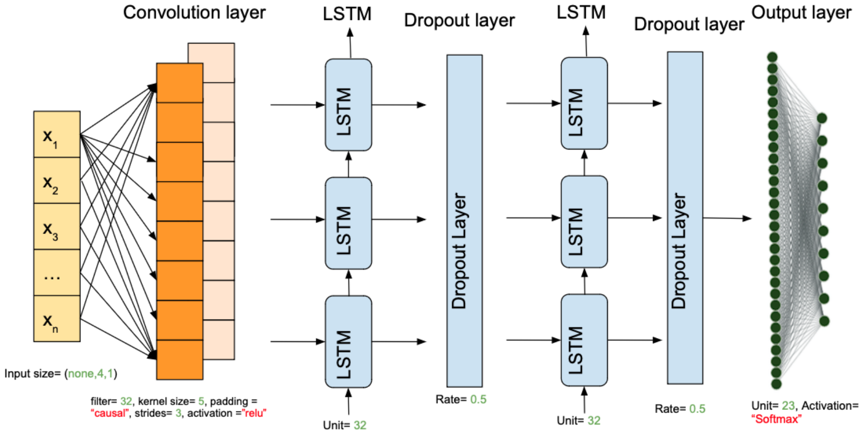


Figure 3.6: The CNN-LSTM architecture proposed by Syed and Ahmed [2023].

several RNN variations. Similarly to the findings of Syed and Ahmed [2023], the findings in this paper demonstrate that incorporating CNNs prior to RNN layers enhances the feature extraction process, yielding more accurate predictions of vessel trajectories while also demonstrating greater stability across diverse time horizons when compared to alternative models. Additionally, they conclude that their model is highly capable of dealing with AIS noise, which they define as intra-message errors and temporal irregularities.

The architecture put forth by Spadon et al. [2022] presents distinct contrasts to the model proposed by Syed and Ahmed [2023]; instead of having a single CNN layer in the front of recurrent layers, the model consists two CNN-RNN blocks. Each block is comprised of a one-dimensional CNN layer, serving as a feature-extraction mechanism, positioned ahead of a single-layered RNN variant. Additionally, each block has a linear feed-forward shortcut connecting its input and output in a residual-like connection with trainable parameters.

Spadon et al. [2022] compared their CNN+RNN variations to over 60 different traditional and state-of-the-art baseline algorithms adapted for their trajectory AIS transmission task, including standard RNN variations, transformer-based auto-encoders, temporal convolutional networks, and additional deep learning architectures. They compared the models over three different complexity cases: low, medium, and high. The low complexity case sought to predict the subsequent 5 messages using the past 15, the medium complexity case sought to predict the subsequent 25 messages using the previous 15, while the high complexity case used the previous 30 messages to predict the next 50 messages. Their results show that their CNN+LSTM model outperformed all other baselines for the medium and high complexity cases, whereas their CNN+GRU model was superior for the low complexity case.

3.3.3 Generative Transformers for Trajectory Prediction

Nguyen and Fablet [2021] successfully applied a Transformer model for the purpose of forecasting vessel trajectories using AIS data, significantly outperforming other state-of-the-art models like LSTMs and CNNs.

Their AIS trajectories were represented as sequences of AIS observations $\{x_{t_0}, x_{t_1}, \dots, x_{t_T}\}$, where $x_t \triangleq \{lat, lon, SOG, COG\}^T$, signifying latitude (in degrees), longitude (in degrees), speed over ground (in knots), and course over ground (in degrees), respectively. Furthermore, the observations were interpolated to ensure a consistent time step size of 10 minutes. One of the major obstacles acknowledged by the authors in trajectory prediction, particularly in the context of AIS trajectory prediction, is the heterogeneous and multimodal character of motion data given the low-dimensional observations. Consequently, the authors expanded this representation into a higher-dimensional embedding vector using a discretized one-hot vector, as illustrated in Figure 3.7. Their embedding resolution per attribute e_t^{lat}, e_t^{lon}, e_t^{SOG}, and e_t^{COG} was set to 256, 256, 128, and 128, respectively, resulting in a value resolution of 0.01° for lat and lon, 1 knot for SOG, and 5° for COG. With these resolutions, the whole embedding vector e_t resulted in a dimensionality of 768.

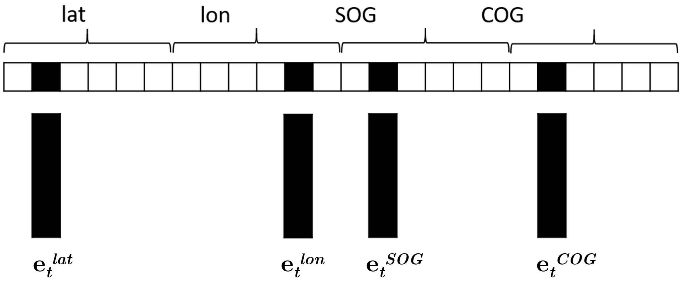


Figure 3.7: Proposed AIS observation representation used by Nguyen and Fablet [2021].

Their model was derived from a conventional sequence-to-sequence transformer architecture comprising eight layers, each equipped with eight attention mechanisms. This model was provided with AIS sequences for various vessels, each of which encompassed its sequential AIS observations over a period of three hours. The model was then tasked with generating subsequent observations that extended from one to fifteen hours into the future. For evaluation, they compared the distance (haversine) between the true positions and the predicted positions.

The results of their experiments demonstrate the **superior performance of the transformer model** when compared to its counterparts. For instance, with respect to 3-hour-ahead predictions, the transformer model achieved a substantial reduction in the mean prediction error at -85.9% relative to the LSTM model, which translates to a performance increase by a factor of 7.1.

3.4 XAI Approaches to Sequence Models

Research within the field of XAI has derived an array of approaches and methods for investigating the decision-making process of complex ML models, including sequence-based models, where interpretation and understanding of model decisions can be complicated due to sophisticated architectures, hidden states, and intricate temporal dependencies. Although these methods are not explicitly tied to AIS data or financial forecasting, their applicability extends to any sequence model and, therefore, constitutes a crucial part of the toolset for this thesis.

This section will explore the landscape of XAI methods in the context of the applicable sequence models established in this chapter. Therefore, in keeping with the research objectives and subsequent domain-specific constraints, several XAI methodologies will be deliberately excluded from the discussion owing to their lack of relevance or being infeasible to retrofit towards the domain constraints. The omitted methods are either tailored for classification tasks, such as *Class Activation Maps*, *Anchors* and *Shaplets*, solely offer local explanations, as is the case with *LIME.* Moreover, perturbation-based methods [Ivanovs et al., 2021] are considered infeasible in this study due to the complexity of the time series data. Simulating random fluctuations that are naturally occurring and temporally valid is challenging and outside the scope of this research. Furthermore, employing total randomness would generate explanations based on synthetic and potentially flawed data, which would not provide meaningful insights into the model's decision-making process. Further, methods reliant on a simplification of the input features, such as *Symbolic Aggregate Approximation*, are also not considered due to the complex nature of AIS data and apprehensions regarding its potential slow rate of change together with the relatively short-spanned dataset used in the research.

3.4.1 Gradient-Based Methods

Gradient-based methods constitute a category of methods specifically designed for models within the DNN domain. These methods leverage the gradients inherent in these models to measure the sensitivity of the output with respect to each input feature, typically by exploiting the backpropagation algorithm [Ancona et al., 2017]. Consequently, the majority of gradient-based methods generate feature attribution explanations that assign responsibility to individual input dimensions for a given output. While some methods are generally applicable across a wide variety of deep-learning models, including CNNs, RNNs, and Transformer models, there are also some model-specific methods tailored to the specific architectures.

Furthermore, there are several additional gradient-based methods and extensions that are left out of this section; however, the ones most commonly implemented in state-of-the-art XAI libraries will be discussed.

Gradient*Input

In the realm of gradient-based explanatory methods, one of the fundamental and simpler approaches is the Gradient*Input method, which has laid the groundwork for the development of more sophisticated gradient-based techniques. Its intent was to increase the fidelity of sensitivity analysis through the generation of more refined attribution maps using gradients [Ancona et al., 2017]. Given an input x and the prediction given by $F(x)$, where F is the function implemented by the model, the gradients of the output prediction with respect to the input features ∇_x can be computed using the conventional techniques employed in the backpropagation algorithm. The underlying premise is that ∇_x constitutes a local sensitivity map for x, as it defines the change in $F(x)$ across all feature dimensions [Baehrens et al., 2010]. By interpreting the steepness of these gradients per feature dimension, the relative importance of each corresponding feature can be ascertained, with steeper gradients suggesting a higher impact on $F(x)$ and thereby indicating greater feature importance. Consequently, each element of the computed gradients can be multiplied with the corresponding element of the input features, yielding the contribution of each feature in the input as $x \times \nabla_x F(x)$.

Due to its simplicity, the Gradient*Input method is highly computationally efficient. However, it is primarily suitable for local explanations; while it does offer an understanding of how different input features collectively impact the output, it may not accurately capture the unique contribution of each individual feature. This limitation arises due to the non-linear nature of DNNs, as well as the possibility of complex feature interactions and dependencies that can't be adequately represented at a single input point. While this makes the method less desirable for our research goal, its role as a foundational method and base for more relevant gradient-based techniques necessitates its inclusion.

In fact, most methods extending the Gradient*Input method introduce the idea of averaging the gradients between a baseline and an input, whereas Gradient*Input uses the instant gradient for the input.

Integrated Gradients

Integrated Gradients (IG) is a method proposed by Sundararajan et al. [2017] that extends the Gradient*Input method with the goal of overcoming its limitations while retaining its beneficial characteristics. It maintains the simplicity of Gradient*Input while augmenting it with a procedure that generates a more comprehensive and holistic picture of feature importance across the entire input space. This is achieved by incorporating the gradients over a straight-line path in the input space from a baseline instance to the given input instance.

Given an input x and a baseline x', for which Sundararajan et al. [2017] recommends the absence of contributing features ($\nabla_x F(x') \approx 0$), IG computes the gradients at a sequence of generated points along the path from x' to x and integrates (averages) them to obtain the final attribution scores. Specifically, for each feature i, the method computes the integral of the gradients with respect to the feature i along the path from x' to x, as shown in Equation 3.1.

$$IG_i(x) = (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha \tag{3.1}$$

To further illustrate, Figure 3.8 shows the path intersection between x' and x on a two-dimensional gradient plane composed of two features, with several points along it to approximate the integral using their derivatives. The figure also presents the immediate gradient used by the Gradient*Input method for point x in blue, allowing for comparison.

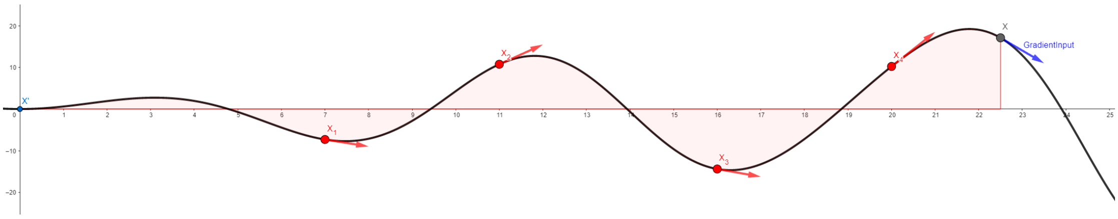


Figure 3.8: A straight-path intersection of a 2D gradient plane between a baseline x' and an input x used by the Integrated Gradients method to integrate (average) several derivatives to attribute feature importance.

Several studies have demonstrated the effectiveness of IG on time series data, some of which included recurrent architectures.

Pirie et al. [2022] utilized IG to extract feature importance attributions for each day over three different time frames: the entire two-week period, the final week, and the final three days. These attributions were then incorporated into a template-based natural language generation approach to generate explanations in the form of a weather forecast report.

Layer-wise Relevance Propagation

Bach et al. [2015] introduced the Layer-wise Relevance Propagation (LRP) as a gradient explanation method that traces back the contributions of all neurons in the network to the final output. In Figure 3.9, lrp-original introduces a neural network architecture and portrays both a forward pass (right) and relevance propagation (left). The relevance propagation initiates from the output layer, from which it propagates the output back through the preceding layers, distributing a relevance value $R_i^{(l)}$ to each neuron along the way, where i is the unit of layer l. Finally, the relevance propagates to the input layer, in which each input is attributed the corresponding relevance value.

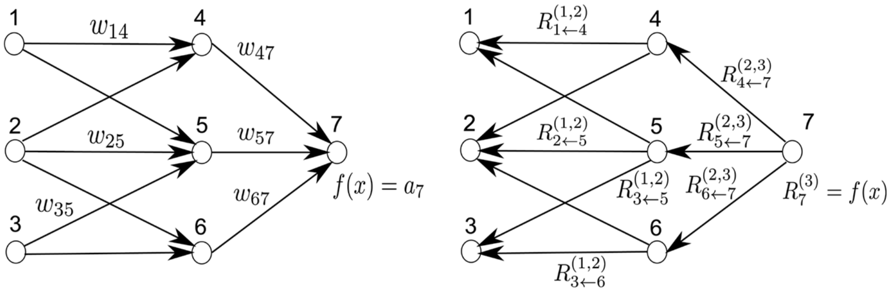

1
w_{14}
4
w_{47}
2
w_{25}
5
w_{57}
7
w_{35}
w_{67}
$f(x) = a_7$
3
6
1
$R^{(1,2)}_{1\leftarrow 4}$
4
$R^{(2,3)}_{4\leftarrow 7}$
2
$R^{(1,2)}_{2\leftarrow 5}$
5
$R^{(2,3)}_{5\leftarrow 7}$
7
$R^{(1,2)}_{3\leftarrow 5}$
$R^{(2,3)}_{6\leftarrow 7}$
$R^{(3)}_7 = f(x)$
3
6
$R^{(1,2)}_{3\leftarrow 6}$

Figure 3.9: A simple neural network undergoing both a forward pass and relevance propagation (illustration from Bach et al. [2015]).

LRP operates on a fundamental principle of conservation, asserting that the aggregate amount of relevance, or equivalently, the contribution of each neuron to the ultimate prediction must remain invariant during the propagation process. For exemplification, referring to Figure 3.9, the relevance of the output layer is directly correlated to the model function, resulting in the equality $R_7^{(3)} = R_4^{(2)} + R_5^{(2)} + R_6^{(2)} = f(x)$.

The LRP method has gained considerable traction as an explanation technique for sequence models, owing to its robust interpretability and efficacy. It provides an intricate view of how different neurons contribute to the output, thus enabling greater comprehension of the complex decision-making processes in complex deep-learning models.

Jung et al. [2021] proposed an extended selective version of the method to explain CNN and RNN models while also asserting the effectiveness of the original method.

Ullah et al. [2021] applied LRP to a 1D-CNN for Credit Card Fraud detection and Telecom Customer Churn using tabular sequence data. Their findings indicate that LRP yields more effective explanations than both LIME and SHAP, with a significant computational time advantage.

LRP has also been applied to Transformer models. Voita et al. [2019] successfully applied LRP to a Transformer to evaluate the degree to which different heads at each layer contribute to the top-1 logic predicted by the model.

Li et al. [2022a] developed a feature-level spatial-temporal layer-wise relevance propagation (ST-LRP) method for the purpose of quantitatively obtaining the correlation of multiple input feature data to the energy consumption prediction results in both spatial and temporal dimensions. Their method allowed for the visualization found in Figure 3.10, which shows feature relevance over both the temporal and spatial dimensions.

DeepLIFT

DeepLIFT (Deep Learning Important FeaTures) was proposed by Shrikumar et al. [2017], which, similarly to LRP, propagates relevance scores backward through the network. However, in DeepLIFT, each unit is assigned an attribution that represents the relative effect of the unit activated at the original network input compared to the activation at some 'reference' input.

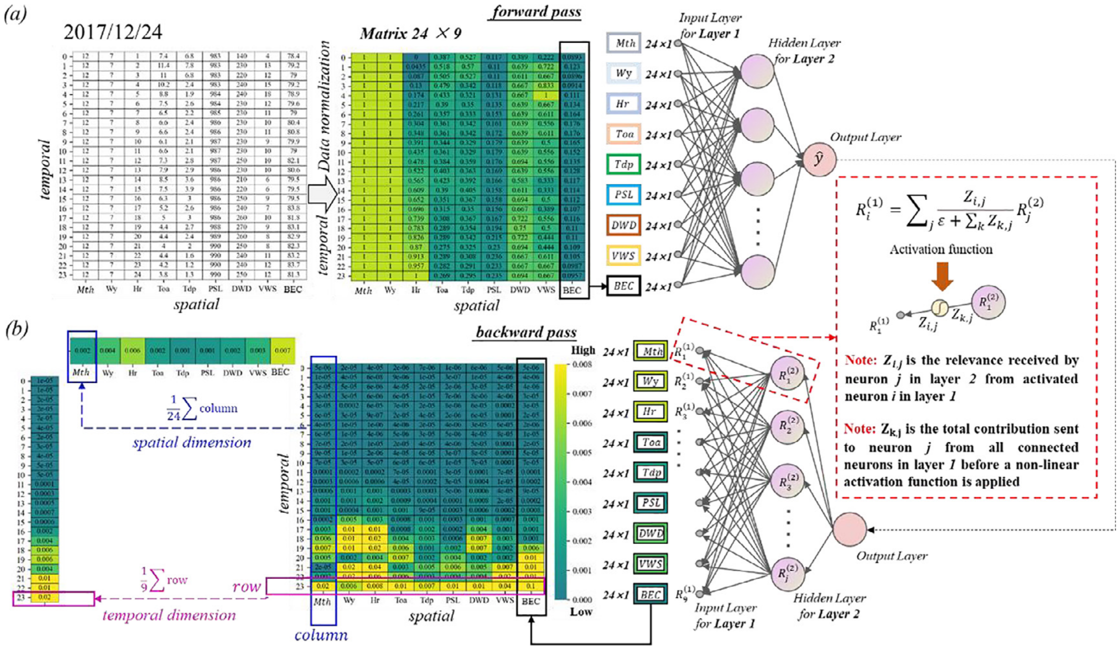

(a)
2017/12/24
temporal
spatial
Mth Wy Hr Toa Tdp PSL DWD VWS BEC
Data normalization
forward pass
Matrix 24 × 9
Input Layer for Layer 1
Hidden Layer for Layer 2
Output Layer
$\hat{y}$
24×1
$R_i^{(1)} = \sum_j \frac{Z_{i,j}}{\varepsilon + \sum_k Z_{k,j}} R_j^{(2)}$
Activation function
Note: $Z_{i,j}$ is the relevance received by neuron j in layer 2 from activated neuron i in layer 1
Note: $Z_{k,j}$ is the total contribution sent to neuron j from all connected neurons in layer 1 before a non-linear activation function is applied
(b)
backward pass
$\frac{1}{24}\sum$ column
spatial dimension
$\frac{1}{9}\sum$ row
row
temporal dimension
column
High
Low
Output Layer
Hidden Layer for Layer 2
Input Layer for Layer 1

Figure 3.10: Visualizations yielded by the ST-LRP (from Li et al. [2022a]).

This reference constitutes a neutral input that represents an appropriate default for the problem at hand, which will be fed forward through the network to establish reference values for all units in the network. For any target neuron t of interest whose computation is dependent on neurons $x_1, \ldots, x_n$, its difference-from-reference can be defined as $\Delta t = t - t^0$, where t^0 is the reference value of t. This delta is then used to assign contribution scores $C_{\Delta x_i \Delta t}$ to Δx_i, as shown in Equation 3.2, such that the sum of the contribution scores for all x_i equals Δt.

$$\Delta t = \sum_{i=1}^{n} C_{\Delta x_i \Delta t} \tag{3.2}$$

Concluding their study, Shrikumar et al. [2017] states that DeepLIFT could hold a particular utility for models such as RNNs that employ *saturating* activations like sigmoid or tanh. They argue that the difference-from-reference approach facilitates the transmission of information even when the gradient is zero, an assertion supported by subsequent studies. For instance, Wang et al. [2019] used *Deep SHAP*, a compositional approximation of SHAP values to compute feature importance, which generated satisfactory explanations. Another study by Vega García [2019] also found *DeepSHAP* to provide adequate explanations for an LSTM model while also outperforming other gradient-based SHAP implementations, such as *GradientSHAP*, in computation efficiency.

SmoothGrad

Smilkov et al. [2017] introduced a technique for reducing noise in the gradient of models and their input to help visually sharpen gradient-based sensitivity maps. Although it may not have a direct correlation with the problem being examined, SmoothGrad's versatility makes it applicable to a broad spectrum of challenges, and its utility is such that it is included in various comprehensive libraries specifically designed for a model explanation. The central concept of SmoothGrad is the reduction of gradient noise through an averaging process applied over multiple instances of noise-perturbed input data. This technique can effectively address the high sensitivity of saliency maps to input data, a recurrent issue in gradient-based explanations. The result is a more refined and potentially more interpretable visual representation. Despite its benefits, the implementation of SmoothGrad entails numerous calculations as it requires the computation of gradients over multiple instances with added noise, resulting in increased computational expenditure. It is also important to note that the addition of random noise to time-series data, as already mentioned, may not be inherently logical.

However, given its pre-existing implementation within the XAI library functions used in this study, along with its status as a state-of-the-art regularization technique, an elaboration of SmoothGrad is included herein.

3.4.2 Attention-Based Methods

The prominence of attention-based explanations has increased in tandem with the progress of Transformer models. These explanations capitalize on their inherent attention mechanisms, which serve as the component for attending to the most important temporal observations. Attention can be employed as a means of explanation as it assigns varying weights of relevance to different parts of the input. These weights can be considered as a measure of feature importance, with higher weights indicating higher relevance to the output. However, the attention weights do not provide an exact measure of the contribution of each input to the output; they instead offer a means of understanding what parts of the input the model is primarily oriented towards.

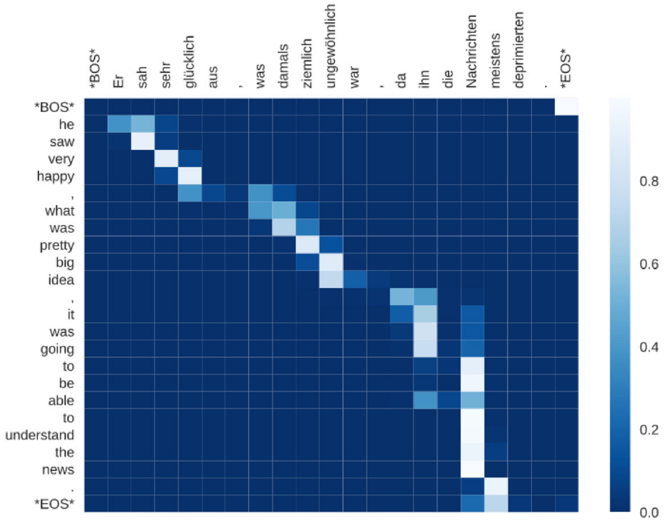


Figure 3.11: Attention heatmap of a German-English translation (from Bükk and Hoang [2022]).

These attention mechanisms are commonly visualized in *attention heatmaps*, as illustrated in Figure 3.11. The figure depicts a German-English translation where the Transformer model's attention weights allow us to discern what source words (German) were considered most influential when generating each target word (English). Each row corresponds to a target word, and each column corresponds to a source word. The intensity of the color in the cell at the intersection of a row and a column reflects the magnitude of the attention weight corresponding to the source word when generating the target word. Lighter cells indicate a stronger attention weight, signifying that the model assigned a higher relevance to that source word when generating the target word.

In the context of AIS-based time series sequence-to-sequence modeling, a similar visualization could be derived where each row and column constitute a time step. For a sequence-to-one regression approach, it could be visualized as a one-dimensional array where each index is a time step. Additionally, as the feature space in this study is homogenous across samples, it might be possible to average attention weights over a global scope, which could explain overarching temporal importance.

3.4.3 Shapley Additive Explanations

Shapley Additive Explanations (SHAP) has emerged as a prominent XAI approach that has garnered substantial attention and widespread adoption within the XAI research community, including for time series problems [Dikshit and Pradhan, 2021; Zhang et al., 2022; García and Aznarte, 2020]. Renowned for its reputedly straightforward and understandable explanations and model-agnostic applicability, SHAP has become a preferred method for interpreting complex machine-learning models. SHAP derives its theoretical foundation from cooperative game theory, specifically drawing from the concept of the Shapley Value, developed by Shapley et al. [1953], a Nobel laureate in economics.

Foundation of the Shapely Value

The Shapley Value offers a solution for the distribution of a cooperative game's *payoff* among its participants. It operates on the principle that each player's share of the total gains should mirror their contribution to the total value. A cooperative game is formally defined by a pair (N, v), where N denotes a set of players and v is a characteristic function that assigns a real number $v(S)$ to every coalition $S \subseteq N$, indicating the total worth achievable by the players in S. The Shapley value φ for a specific player i is computed as shown in Equation 3.3. It calculates their average marginal contribution to all potential coalitions, where n is the total number of players and the sum extends over all coalitions not containing player i.

$$\varphi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} (v(S \cup \{i\}) - v(S)) \tag{3.3}$$

Figure 3.12 offers a visual representation of this concept, illustrating how each player (Circle, Square, and Triangle) contributes to the total payout of four coins. The worth v of each possible coalition $S \subseteq N$ is represented by a number of coins, from none (0 coins) to all players involved (4 coins), where Circle and Square individually contribute one coin each and Triangle contributes two coins. The figure also highlights the concept of *efficiency*, a property ensuring that the values attributed to players sum up to the total payoff, thereby facilitating a fair distribution of all gains among the players. This principle remains valid across several games, where each player's Shapley value is the cumulative sum of the Shapley values for each game.

The Shapley Value in the Context of Machine Learning Models

In the context of machine learning, the Shapley value provides a mechanism to quantify the contribution of each feature to the predictive outcomes of a model. By conceptualizing each sample as a game, wherein the features assume the role of players N, and the payoff v represents

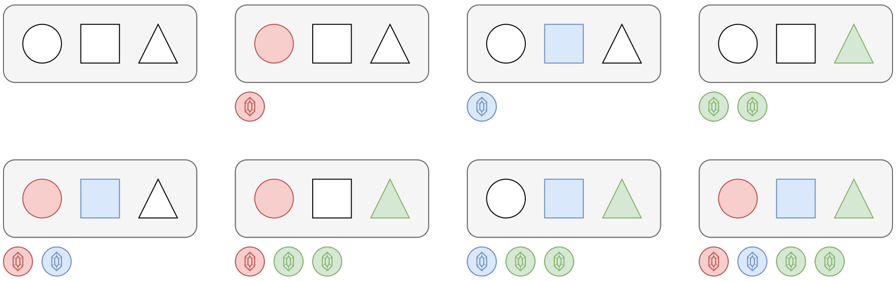


Figure 3.12: Illustration of the different coalitions and subsequent worth (represented as coins) within a cooperative scenario involving three players: $N = \{\text{Circle}, \text{Square}, \text{Triangle}\}$.

the model's prediction value, the Shapley value establishes a means to gauge the average marginal contribution of a given feature to the overall prediction. The properties of SHAP ensure that the explanation for the prediction is fully specified by the feature importances and that ineffective features are attributed zero contribution.

The complexity of state-of-the-art sequential models necessitates approximations when computing SHAP values, and gradient-based explanation methods like those detailed in Section 3.4.1, are often used to approximate the SHAP values. In these approximation methods, SHAP values can be interpreted as gradient-based attributions adjusted to distribute the difference in output between the instance being explained and a reference instance among the features. The specific adjustment method and integration manner depends on the underlying gradient-based explanation method.

Local and Global Explanation Scope

SHAP provides both local and global explanations for ML models. Locally, it provides an explanation for individual predictions, quantifying the effect of each feature on the model's output. There are several ways of visualizing local explanations; Figure 3.13 shows a commonly used *force plot*, which graphically depicts how each feature pushes the prediction away from the base value (average prediction) towards the actual prediction.

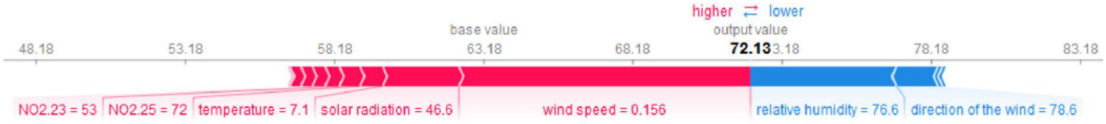


Figure 3.13: An example of a SHAP force plot for a local explanation (illustration from García and Aznarte [2020]).

Globally, SHAP enables understanding the overall behavior of the model by aggregating local explanations over the entire dataset. These aggregated explanations, commonly visualized using *summary plots*[2], show the importance and contribution of each feature and their effect on the model's predictions. They can reveal global feature importance, where larger absolute Shapley values denote more important features, as well as feature impact direction, where positive Shapley values increase prediction and negative values decrease it.

[2] Summary plots can include violin plots, bar charts, heatmaps, and other per-feature plots.

3.4.4 SHAP for Sequence Models

Sequence models increase the complexity of SHAP-based feature attribution due to the introduction of temporal dependencies, where a feature's relevance depends not only on itself but also on its position and relationship with other features in the sequence. Nonetheless, the application of SHAP to sequence models has been documented in several academic studies.

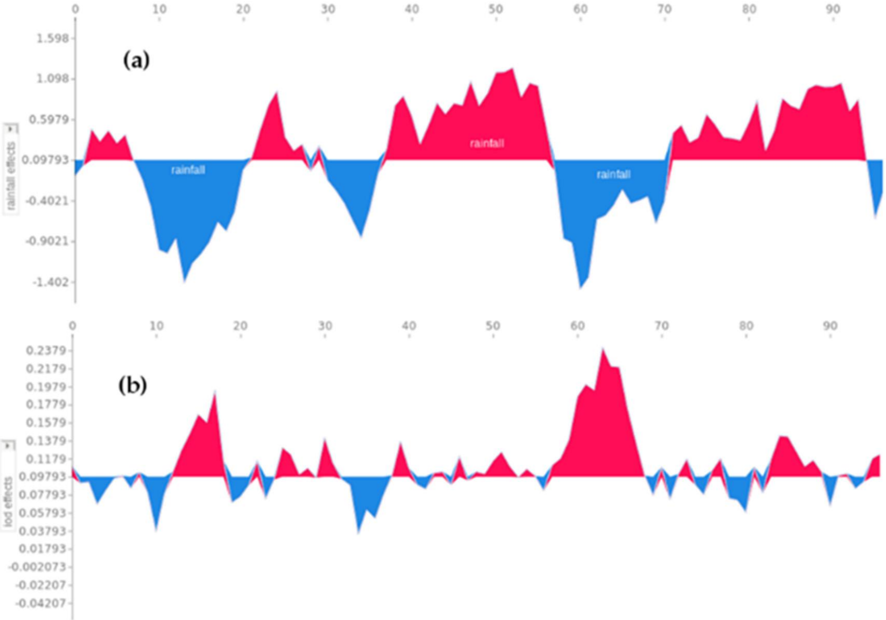


Figure 3.14: Dikshit and Pradhan [2021] illustrate the impact of two features in a time period using collective force plots (illustration from the original paper).

Dikshit and Pradhan [2021] applied SHAP to a CNN-BiLSTM sequence model, aiming to derive insight into how different climate variables interact with drought, specifically the Standardized Precipitation Index (SPI-12). Their motivation stemmed from recent revelations suggesting that the inclusion of climatic variables in data-driven prediction models significantly enhances their predictive accuracy. However, the basis of these assertions primarily rested on the statistical metrics used to evaluate model accuracy. Hence, they sought to delve deeper, utilizing SHAP to gain a more nuanced understanding of the underlying mechanisms and the interactions between variables. Their study focused on five locations in New South Wales (Australia), for which a historical dataset between 1901 to 2018 was compiled containing several climate variables, such as the Southern Oscillation Index (SOI), Indian Ocean Dipole (IOD), sea surface temperature, Nino indices, along with rainfall data. Their CNN-BiLSTM model was able to achieve an R^2 score of around 0.90 in all locations, even without pre-processing or normalization. They subsequently utilized SHAP force plots, as explained in Section 3.4.3, focusing on global explanations for different drought periods. Consistent for all locations, their SHAP force plot explanations indicated that the rainfall feature had the most significant impact on the SPI-12 predictions. They also showed a collective force plot of two individual predictors, as illustrated in Figure 3.14, which gives an indication of feature impact over a time series. They note that the rainfall and IOD features are correlated and that the additive and independent nature of SHAP inflated the overall contribution of rainfall, leading to a more major contribution of rainfall for events that were primarily dependent on IOD. They further discuss that the SHAP library they employed had developmental challenges, especially for its *DeepExplainer*[3], which they argue has inadequate performance as the model complexity increases. They conclude that these local and collective force plots can yield a broad understanding of a model's reasoning and that the SHAP values were able to provide valuable information for their time series problem.

The application of SHAP for time-series data has also been explored in the domain of smartphone app usage forecasting, where Zhang et al. [2022] employed it to explain the usage times of various applications. In contrast to Dikshit and Pradhan [2021], who used a recurrent neural network, Zhang et al. [2022] found LightGBM, a gradient-boosting decision-tree model, to deliver superior performance for their problem, which was then explained using SHAP. They conclude that SHAP was able to provide satisfactory explanations.

Another study by García and Aznarte [2020] saw similar satisfaction in SHAP's explanations on an LSTM model forecasting the atmospheric concentration of NO_2 in Madrid, using features such as wind speed and solar radiation. They tested three different explanation methods for their model, namely Kernel SHAP, GradientSHAP, and DeepSHAP. They concluded that the DeepSHAP implementation showed better performance than GradientSHAP for deep learning models and that KernelSHAP could not be configured for their LSTM model. Their derived explanations were deemed satisfactory, and they aligned with the current understanding of the phenomena.

Other studies have compared SHAP to other interpretation frameworks and found that SHAP yields superior explanations. For instance, Ozyegen et al. [2020] applied a *Perturbation Curve for Regression* and *Ablation Percentage Threshold* metric, which saw favorable fidelity and preferable global mean replacement in SHAP, while a study by Saluja et al. [2021] saw both SHAP and LIME provide good explanations.

3.5 Summary and Key Findings

This section will serve as a concise yet comprehensive overview of the most significant and noteworthy findings from the extensive discussion on the current state-of-the-art literature. It will discuss key studies, theories, methodologies, and empirical evidence that directly contribute to the research goals of this thesis.

3.5.1 AIS as Features

Temporal Size and Resolution

The studies presented in Section 3.1 that used AIS data specifically for financial forecasting typically adhere to a daily temporal resolution, whereas Næss [2018] opted for a coarser resolution of one week. The adoption of coarser temporal resolutions may be rationalized by the slow changes in shipping patterns that are consequent to the relatively slow speed of vessels and the extensive distances they travel. However, it is important to note that the adoption of coarser resolutions for time series data, especially with short time spans, will limit the amount of data that can be utilized without introducing contamination between the sequences of training and test sets. To counteract this, the sequences would have to be shortened to reduce the overall time span in the sequences, reducing their length. Therefore, a one-day temporal resolution, which has been validated as successful by the other studies, appears to be a suitable choice.

[3] The *DeepExplainer* module from the SHAP Python library is based on the DeepLift algorithm that approximates the conditional expectations of SHAP values using a selection of background samples.

Additionally, there is a notable difference in look-back periods across the studies. For instance, a length of 60 days is proposed by Kanamoto et al. [2019], justified on the basis of average voyage lengths. Conversely, both Næss [2018] and Århus and Salen [2018] employ a sequence length of 20 days. Given these divergences, it is reasonable to assume that the choice of sequence length warrants experimentation, which similarly extends to the choice of an appropriate forecast horizon.

AIS Representation

The presented AIS-based financial forecasting studies universally employ a methodology that consolidates AIS attributes into multiple fixed regions. While this approach has seen success and keeps the number of features relatively low, depending on the number of areas and attributes, it is reasonable to assume that it may lack the granularity and accuracy to depict more complex trade patterns. Kanamoto et al. [2019] attempted to address this by applying a course filter to segregate traffic going to import ports, but this approach also seems limited.

Furthermore, consultations with the domain experts indicate that such a spatial approach to AIS features would not generate sufficiently meaningful interpretations from feature attribution methodologies. While it could support existing knowledge, it was thought to be insufficient in extracting new insight. Consequently, the selection of a better and more appropriate feature format that can be used with feature-attribution methods warrants consideration.

3.5.2 Suitable Models

Standard Deep Neural Network

Kanamoto et al. [2019] argue that advanced sequence models are limited by their high data requirements for effectively modeling sequences. With a rather limiting dataset, they instead employed a fully-connected DNN architecture comprising an input layer, 9 hidden layers, and an output layer, where the entire sequences were supplied to the input layer. The results demonstrated that the model saw improved performance when utilizing AIS data, suggesting its ability to leverage it effectively. Although the authors did not compare the model against more sophisticated sequence models, their straightforward implementation makes it conducive to further comparisons with other appropriate models.

Advanced Recurrent Neural Networks

Many of the explored studies, both for AIS-based and non-AIS forecasting tasks, often see superior results using recurrent models, with a particular inclination towards the LSTM variation. At the same time, the relatively short sequence lengths proposed by relevant studies for AIS-based financial forecasting might not warrant the use of the complex variations capable of extensive long-term dependency modeling. However, modern machine-learning frameworks facilitate a relatively seamless transition between different variations, thereby simplifying and motivating experimentation. Moreover, numerous studies have indeed observed better performance when employing LSTMs over regular RNNs, even in scenarios involving these shorter sequences.

Convolution-Based Models

There are several convolution-based architectures that could facilitate effective learning of AIS data. Most notably, the studies explored have highlighted the effectiveness of TCNs as sequence models, surpassing the performance of recurrent models across diverse benchmarks commonly employed for evaluating recurrent networks. Additionally, both Spadon et al. [2022] and Syed and Ahmed [2023] saw improved performance when incorporating convolutional layers preceding the recurrent layers.

Transformers

Recent academic investigations have observed a considerable surge in the application of Transformer models which, when tailored to time series forecasting, have demonstrated the ability to outperform a majority of existing state-of-the-art models. The Transformer architecture has already seen success within the field of AIS-based trajectory forecasting, as well as tabular time series forecasting.

Transformer models are yet to be applied to the specific domain of AIS-based financial forecasting, and their recency and complexity present a certain degree of implementation challenges. That being said, they are considered the new state-of-the-art for sequence modeling, and it would be of academic interest to contrast them against the current models used within the specific domain explored in this thesis.

3.5.3 Explanations from AIS Sequence Data

Several of the models presented allow for sequence-to-sequence modeling. This approach can provide a more comprehensive understanding of the future dynamics of financial instruments, moving beyond the simple prediction of a single future value. Despite its benefits, the intricacy inherent in explaining an output sequence may introduce considerable visualization challenges. For instance, the spatial-temporal heatmap explanations generated by the ST-LRP method proposed by Li et al. [2022a] would require an additional dimension to represent the relevance attributed to each item in the output sequence with respect to the input features. While such visualization may be possible to implement, it is reasonable to assume that a sequence-to-one approach is more likely to produce satisfactory and comprehensible explanations for domain experts, as it simply adds a singular numeric relevance score instead of a vector to each feature.

Additionally, multiple comparative XAI studies point to SHAP as domain experts' preferred means of explanation. However, as a feature attribution method, SHAP requires the features to be in a meaningful format that facilitates a good explanation medium.

Chapter 4

Model and Data Engineering

This chapter serves as a bridge between state-of-the-art research and the concrete models and methods employed in the empirical experiments of the thesis. It will first outline the data foundation as provided by Maritime Optima, the preprocessing applied, followed by the feature engineering that turns the data into a suitable feature format for both predictive capacity and explainability when using feature attribution methods. The chapter then presents the various models as adopted from the relevant state-of-the-art research. Lastly, the chapter will present the XAI methods used to interpret the models.

4.1 Data Foundation and Processing

While the basic messages of the AIS discussed in Section 2.1 form the basis for the thesis' data foundation, the final dataset incorporates refinements, abstractions, and various transformations that necessitate clarification and justification.

4.1.1 AIS Data

As a collaborative company for this thesis, Maritime Optima provided access to AIS data between December 2019 and March 2023, yielding approximately 1200 days of diverse maritime vessel activities conducted across the global geographical expanse. Given the study's focus on global commercial trade, the dataset was refined to exclude data associated with vessels uninvolved in this sector. Therefore, vessels engaged in activities such as fishing, oil service, cruising, and leisure were omitted[1]. Despite the potentially restrictive time frame of the dataset, it encapsulates several significant global events that influenced the maritime landscape, such as the COVID-19 pandemic [Millefiori et al., 2021; Yazir et al., 2020], the 2021 Suez Canal obstruction [Gerson, 2023], and the subsequent 2022 invasion of Ukraine [Rožić et al., 2022]. Consequently, the dataset was anticipated to encompass a wide spectrum of variations in maritime trade dynamics, including more extreme scenarios. These scenarios serve as intricate exemplars of cause-effect relationships within the maritime sector, offering an enriched learning environment for the models. This added complexity could not only bolster the system's understanding of the intricacies of maritime trade but also augment its predictive expertise in addressing novel or extraordinary circumstances.

[1] The activities cited represent only a subset of those excluded; in actuality, Maritime Optima's entire *Other* segment was excluded, which as of writing includes 28 different vessel segments, including the ones cited.

AIS Type 1-3 Position Reports

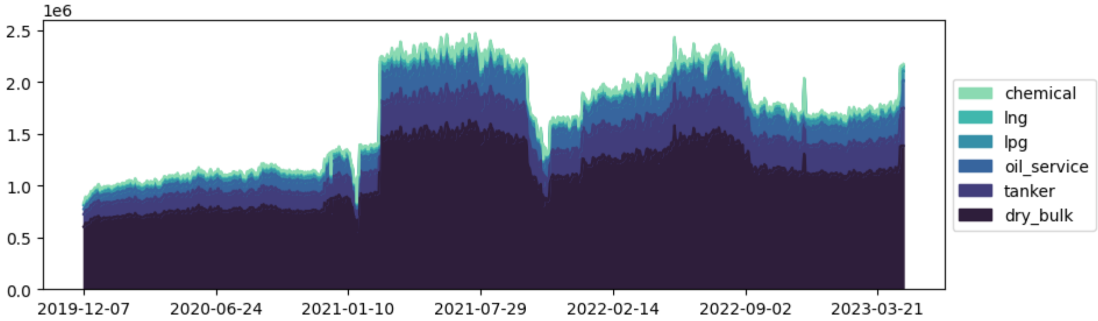


Figure 4.1: Number of types 1-3 AIS reports over time stacked by segment.

Over the examined period of 1200 days, roughly *2 billion* individual position reports were provided for the relevant vessels. Figure 4.1 shows the number of reports produced over this period. The reduced number of reports from the beginning of the period until the start of 2021 is due to the initiation of Maritime Optima's collaboration with new AIS providers, resulting in an apparent surge in the volume and frequency of reports around that time.

Additionally, there is a significant decrease in the number of reported AIS positions towards the end of 2021, with a gradual recovery within the summer of 2022. The underlying factors contributing to these fluctuations remain speculative, but some plausible explanations may be associated with China's late 2021 termination of access to data collected by terrestrial stations within its jurisdiction, as well as the market response to escalating tensions between Russia and Ukraine.

AIS Type 5 Static Reports

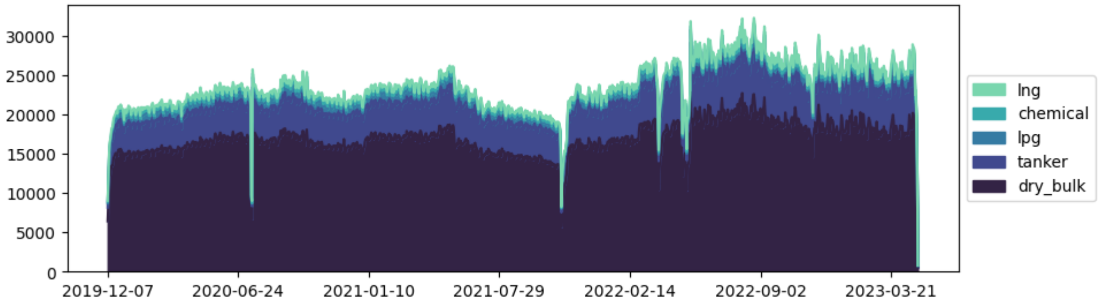


Figure 4.2: Coverage of type 5 AIS reports over time stacked by segment.

Approximately *23 million* static reports were provided. It is worth noting that this number does not represent the actual quantity of individual static reports, as Maritime Optima prunes sequences of identical static reports into aggregated reports that instead provide a time range for when the data is applicable. Hence, Figure 4.2 serves as a historical representation of static report coverage instead of the number of daily reports, delineating the quantity of applicable static reports available on a daily basis.

Voyages

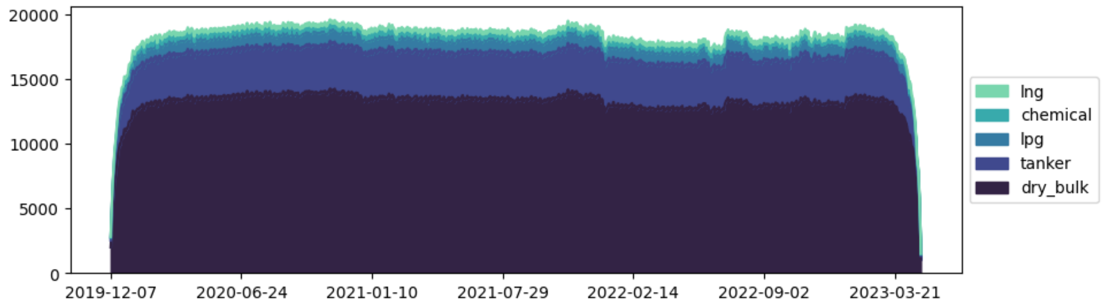


Figure 4.3: Active voyages over time stacked by segment.

A voyage represents a vessel's operation between two ports in which the vessel travels from port A to port B in a given time frame. These voyages are abstracted by Maritime Optima through a complex processing pipeline using the underlying AIS reports and data about ports, anchorages, and other mechanisms defining a vessel's voyage. Approximately 3.4 million voyages were provided, containing the information found in Table 4.1.

Property	Description
IMO number	Unique 7-digit vessel ID number assigned during construction
Distance	The distance of the voyage
Departure port	The port from which the vessel departed
Departure time	The time at which the vessel started the voyage
Arrival port	The destination port of the voyage
Arrival time	The time at which the vessel arrived at the destination

Table 4.1: Structure of a voyage as provided by Maritime Optima

It is important to note that voyages necessitate both a destination and an arrival port. As such, voyages currently underway that have not reached their destination are not included in the data, leading to a decrease in the voyages towards the end of the period when vessels are between the departure and arrival ports. Similarly, at the beginning of the data period, there will be vessels whose departure port is defined in unavailable past data. These phenomena are evident in Figure 4.3, where there are gradual slopes at each end of the data period. More specifically, the data appear to achieve stability approximately 40 days from each end, implying that voyage data within these 40-day spans likely is unreliable for use. Moreover, similarly to static reports, the figure shows the number of active voyages each day instead of the number of new entries each day.

Vessel Association Mapping

Maritime Optima provided an association mapping for vessels and their MMSI for the period. This mapping is paramount in ensuring the correct association of AIS reports with each respective vessel, as vessels can change their MMSI for a variety of reasons, such as switching or mounting new AIS equipment. Moreover, only MMSI is transmitted in position reports, and the IMO in static reports can often be wrong or omitted.

The mapping contains the corresponding IMO number for each MMSI identity for various periods, enabling efficient querying about the specific vessel associated with a given AIS report on any particular date.

4.1.2 Vessel Specifications

In addition to AIS-related data, Maritime Optima supplied comprehensive specification data for the commercially operative vessels across the relevant segments. The specifications were provided as *JSON* files. The data scope of each document is similar to that of standardized *vessel specification questionnaires*, containing several key features of each ship, such as:

IMO number: A seven-digit vessel ID number, unique to each vessel and assigned at the time of construction, stays with the vessel throughout its lifetime, irrespective of any changes in the vessel's name, ownership, or flag. This provides an immutable identifier that helps establish reliable associations between AIS reports and the corresponding vessel specifications.

Segment: A categorization of vessels based on their design and the nature of cargo they carry, for example, tankers, dry bulk carriers, container ships, etc. This is pivotal in understanding the vessel's functionality and the types of goods it transports. See Chapter 2 Section 2.2.3 for more information.

Deadweight Tonnage (DWT): Represents the total mass that a ship can safely transport, including the weight of the cargo, fuel, freshwater, crew, and belongings. It is a crucial measure of the cargo-carrying capacity of a vessel, directly impacting its operational efficiency and potential revenue.

Length Overall: The maximum length of the vessel, which is essential for determining berthing requirements and maneuverability in ports.

These characteristics, when combined with the AIS data, could greatly enrich the informational value of AIS messages. Each AIS message serves as a data point denoting the position, speed, course, and other navigational details. While these messages, in their raw form, depict the spatiotemporal behavior of vessels, they lack the context of the vessel's intrinsic properties. By combining the vessel specifications with the AIS data, we can infuse these AIS messages with additional layers of context, converting them into high-dimensional data points that carry comprehensive information about each vessel's behavior, capabilities, and operational patterns. Moreover, this combination allows for a more targeted approach when it comes to predictive modeling. When forecasting certain types of financial instruments, vessels with specific intrinsic properties may prove irrelevant or introduce unnecessary noise into the analysis. By having detailed vessel specifications at our disposal, we can systematically filter out these vessels and focus our predictive models on the most relevant data subset, thereby increasing the accuracy of our predictions.

4.1.3 Financial Data

The acquisition of relevant financial data is a crucial aspect of our research, enabling the formation of models that utilize maritime trade patterns to predict the future value of financial instruments. This data was sourced from Yahoo Finance, a widely recognized and publicly accessible platform that offers comprehensive financial data for an array of instruments, including stocks, bonds, ETFs, and currencies.

Yahoo Finance caters to diverse data requirements by providing real-time updates as well as historical data, facilitating a thorough analysis of past trends and market behavior. For our research, we utilized historical data to establish patterns and draw correlations with maritime trade dynamics. Yahoo Finance conveniently allows users to download this historical data in a CSV (Comma Separated Values) format, a universal format that can easily be imported into various data processing and programming languages, such as Python, for further manipulation and analysis.

The available financial data in the Yahoo Finance CSV download encapsulates several key metrics, each providing a unique viewpoint into the performance and volatility of the instrument. Table 4.2 elucidates the nature of these metrics.

Column	**Description**
Date	The date of which this entry was recorded (YYYY-MM-DD)
Open	The price when the market opened
High	The highest price during the day
Low	The lowest price during the day
Close	The price when the market closed
Adj Close	The close price adjusted for corporate actions (splits, dividends, etc.)
Volume	The number of traded shares during the day

Table 4.2: The columns making up Yahoo Finance's downloadable CSV data

The choice of financial instruments for analysis is centered on those that are intrinsically linked to the maritime trade industry. However, further details about the specific instruments chosen for forecasting will be delineated per experiment in Chapter 5. The ultimate intention is to unearth the influences of maritime trade on these financial instruments and to harness this knowledge for the prediction of future trends and values.

4.1.4 Preprocessing

Prior to handover, the data was subjected to pre-processing and cleaning through the company's proprietary AIS cleaning pipeline, reducing the impact of the limitations identified in Section 2.1.2 and rendering it suitable for the analytical requirements of this thesis.

Mitigating AIS Gaps

As highlighted in Section 2.1.2, a considerable challenge related to AIS data is its prevalent gaps in coverage. The static reports and voyage abstractions provided by Maritime Optima mitigate some of these issues, but the temporal discontinuity inherent in position reports persists as an obstacle; irregular data points, such as a vessel disappearing and reappearing every other day, can introduce substantial noise and bias into ML models, potentially leading to inaccurate pattern inferences for attributes such as trajectory or speed.

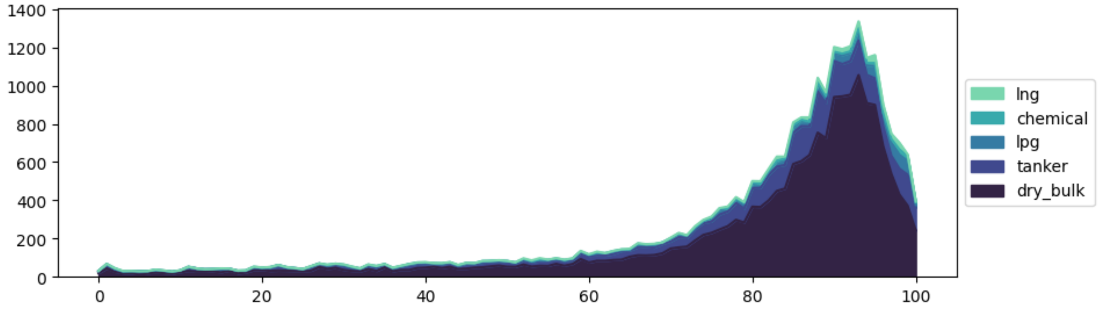


Figure 4.4: Distribution of vessels by the proportion of period with available *AIS type 1-3 position* data.

Figure 4.4 illustrates the distribution of daily AIS position report coverage over the period. The x-axis represents the proportion of the total period for which each vessel provided position reports, while the y-axis denotes the number of vessels corresponding to each percentage value. The majority of vessels have transmitted position reports every 80 to 100% of the days, with a minimal proportion of vessels transmitting position reports less than 60% of the days. It is reasonable to assume that vessels with suboptimal coverage will contribute negatively to the study, and while this coverage analysis does not account for the duration of these coverage gaps, **data for vessels with less than 60% overall coverage were omitted from the dataset**. This percentile is admittedly speculative, but it provides a plausible range that includes the majority of vessels whilst excluding those with the most restricted coverage.

Nevertheless, the remaining vessels that still have a rather substandard and discontinued coverage, albeit a minority, are likely to inject some noisy volatility into the data. To offset these discontinuities, an approach was employed inspired by the rasterization process employed by Chen et al. [2020] in Section 3.3.1 for interpolating position data in their spatial CNN implementation. However, the adopted approach does not interpolate between the data points; instead, it performs a simple forwarding of the last received data over any subsequent days without coverage. This approach is illustrated in Figure 4.5, showing three vessels over ten days and the consequent forwarding of the last status for each vessel. Although this approach might not be ideal in scenarios where positional data is needed, it minimizes the amount of lost data, requires fewer computational resources, and is considered sufficient for the requirements of this research.

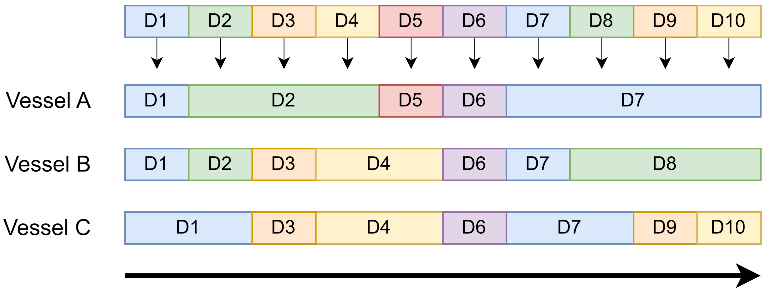


Figure 4.5: The adopted forwarding process to compensate for vessels' data dropout over ten days (left to right).

4.1.5 Feature Representations and Final Dataset

The feature representation adopted in this study is grounded in the theoretical principles that expound the relationship between the market and fleet behavior, as outlined in Section 2.2.2. Previous studies predicting financial instruments with AIS data corroborate these principles, demonstrating the effectiveness of key shipping variables, such as speed and draft. While the majority of these studies aggregated the variables across general maritime regions, Kanamoto et al. [2019] focused on demand traffic by filtering vessels heading for import ports through a course filter for each region. This approach introduces a supplementary dimension to the data, accounting for the dynamics of demand.

Port Relations

The voyage data presented in Section 4.1.1 provides an avenue for capturing supply-demand dynamics in a more detailed context centered around port relationships, moving beyond broad region-based analyses. Furthermore, the domain experts agree that feature-attribution explanations using the fidelity of port-port features are much better than the region-based explanations.

For a port p, voyages between p and a destination port p_d or an origin port p_o can be queried. Incorporating relevant shipping variables into these voyages can produce comprehensive descriptions of trade patterns across the various port pairs, further referred to as port relations. By basing the features on port relations, the models can use effective shipping variables, but in a more granular way that facilitates the modeling of supply-demand dynamics. In addition, port relation-based features will facilitate better feature-attributed explanations, as they will be tied to prominent maritime routes rather than general regions, enabling a more precise medium of explanation.

However, a foreseeable challenge of using these granular port relations over general regions is the expected growth in feature dimensionality attributed to the higher number of relations needed to model global trade appropriately. This will culminate in a larger set of features to interpret, an increased risk of overfitting, and the requirement for more computational resources. Nevertheless, the novelty and potential advantages of relation-centric features compellingly offset this challenge; the improved exactness of trade patterns and the ability to generate more precise explanations establish this as the representation used in this study.

It is reasonable to assume that high-traffic ports have a stronger connection to the maritime market. To generate a robust and encompassing feature set that effectively captures the underlying dynamics of maritime trade, the most active ports[2], defined as an ordered set P, serves as the foundation for establishing the port relations. For demonstration purposes, consider a scenario where $P = \{A, B, C\}$, where A is the most active port followed by B. For each port in P, the most frequented inbound and outbound ports are identified and put into two distinct ordered sets of relations reflecting the flow of vessels to and from each port. The inbound ports for a given port, say A, can be represented as $A_{in} = \{B, C, D\}$, signifying the ports with the highest number of voyages culminating at A. Similarly, the outbound ports for A could be represented as $A_{out} = \{C, D, E\}$, denoting the most frequent destinations for voyages originating from A. Following the identification of inbound and outbound ports, port relations are constructed. For each port in P, relations are established with its respective inbound and outbound ports, creating directed links that capture the major traffic flows. For instance, for port A, relations would be formed from all ports in A_{in} to A, and from A to all ports in A_{out}, yielding the relations: $[BA, CA, DA, AC, AD, AE]$. This process is applied to all ports in P, and the resulting relations for each port are concatenated to form the overall port relations denoted as PR

[2] The most active ports are defined as ports with the biggest number of total visits over the period. This is highly segment-specific and varies with the set of vessels in the dataset.

Duplicate relations arising from bidirectional traffic between ports, which are prevalent due to larger transshipment ports (export and import) like Singapore and Rotterdam, are resolved by prioritizing the relation from the more active port. Ports within set P are processed sequentially in descending order of activity level, allowing for a systematic extension of PR after each port's relations are considered and added. If a new relation already exists within PR, it is skipped in favor of the subsequent most trafficked relation, maintaining an exhaustive, non-redundant representation of port relations.

Shipping Variables

The term *shipping variables*, as utilized in the scope of this thesis, pertains to the various factors that describe the characteristics and dynamics of shipping operations, such as speed and draft. These variables form the foundation for current knowledge about the relationship between fleet behavior and the maritime market, some of which are found or can be derived from the available AIS data. These shipping variables are aggregated per port relation per timestep, and several variables should be included.

Traffic Traffic, quantified as the number of vessels currently en route between the ports, indicates the volume of activity within a specific port relation for a given day. High traffic levels may point towards an elevated demand and supply, while low traffic levels could suggest sub-optimal market conditions.

Speed Speed denotes the average speed of all vessels currently en route between the ports and is known to be a fundamental indicator of market conditions. To repeat from Chapter 2, market changes often compel ship owners to alter their vessels' speed in response to the freight rates; during market downturns, ship owners may resort to slow-steaming or drifting to minimize fuel expenses, which is primarily due to the fact that speed affects fuel consumption, constituting the dominant cost of sea freight, while in profitable market conditions, ship owners often increase vessel speeds to maximize profits.

Load Load is the load factor of the vessels, and substitutes draft as it maintains a normalized variable across vessel sizes; given the varying sizes of ships, and the predominance of smaller vessels, using draft might distort the understanding of load efficiency. It is derived by dividing the current draft of each vessel by the vessel's maximum draft. This results in a value between 0 and 1, with lower values indicating less loaded vessels. This variable is thought to identify situations where vessels are competing for sub-optimal cargo sizes that do not permit the vessels to be fully loaded, a pattern often observed in competitive market scenarios.

Duration Duration refers to the length of time taken for a vessel to complete its journey between two ports. This variable encapsulates several traveling aspects, such as the distance between ports and broader logistical aspects like weather conditions and potential delays caused by port congestion or navigation through high-traffic sea routes.

Time Series Representation

The resulting time series data comprises a sequence of daily observations, each containing a feature vector and the target value n days into the future, which is illustrated for a timestep t in Figure 4.6. Note that the figure does not show multiple time steps, but rather *one* timestep t and the corresponding feature vector $X_t = \{x_1, x_2, \ldots, x_{num\_features}\}$. The feature vectors are consistent across all timesteps, ensuring that feature x_1, which in the illustration is the number of ongoing voyages between port A and B, consistently represents this value for all observations in the time series.

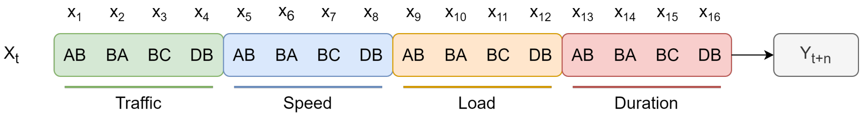

X_t
x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13 x14 x15 x16
AB BA BC DB AB BA BC DB AB BA BC DB AB BA BC DB
Y t+n
Traffic
Speed
Load
Duration

Figure 4.6: Representation of a feature vector and the corresponding future target value at a timestep t.

Data Normalization

Data normalization ensures that each feature, irrespective of its original scale or numerical range, has an equitable chance to influence the model's predictions. This also helps mitigate instability during training, as features with larger numerical ranges could dominate the learning process simply due to their larger values, not because they are necessarily more informative. In the case of the shipping variables selected for the dataset, there are significant differences between them. For instance, draft values might vary between 8 and 20 meters, traffic between 0 and several hundred, and load between 0.5 and 1. Moreover, an equitable contribution is fundamental for the correct understanding and interpretation of the features' impact on the output variable, which is particularly important when using methods such as SHAP for feature attribution.

Therefore, normalization was conducted on a feature-wise basis across all time steps. Each feature's minimum and maximum values were identified across all observations in the time series. These minimum and maximum values were then used to transform the feature values to a scale between 0 and 1. This ensures that draft values between some port A and B are normalized separately from draft values between A and C. The normalization transformation is formulated in Equation 4.1, where x_i is the original value of a feature at index (not timestep) i, x'_i is the normalized value of the feature, and $\min(x_i)$ and $\max(x_i)$ represent the minimum and maximum values of the feature x_i across all time steps, respectively.

$$x'_i = \frac{x_i - min(x_i)}{max(x_i) - min(x_i)} \tag{4.1}$$

Sliding-Window Sampling

To structure the time series in a way that is conducive to the requirements of ML models, a sliding window approach was employed. In this sliding window approach, for every observation in the time series, the label was set to be the target value from n timesteps into the future. Conversely, the input for each observation was designed to be a sequence of feature vectors $\{X_{t-w}, \ldots, X_t\}$ from the previous w timesteps, creating a *look-back period.* The sliding window approach essentially transforms the time series into a supervised learning problem, where past observations are used to predict future ones. However, determining an optimal look-back period size w is dependent on both the scale of the dataset and the specific requirements of the domain. The process of defining this size will be explored and experimented upon in Chapter 5. .

4.2 Models

This section bridges the gap between the findings in state-of-the-art research and the empirical applications pursued within this thesis. It introduces the various models, each derived from the forefront of current research, which will be employed for the problem and empirically evaluated in the rigorous experimentation outlined in Chapter 5.

As explored in Chapter 3, the relevant research outlines a multitude of models, architectures, and variations that warrant exploration. The models introduced in this chapter were all implemented using the TensorFlow/Keras API, an open-source library for high-performance numerical computation with less boilerplate than similar frameworks.

4.2.1 Fully Connected Deep Neural Network

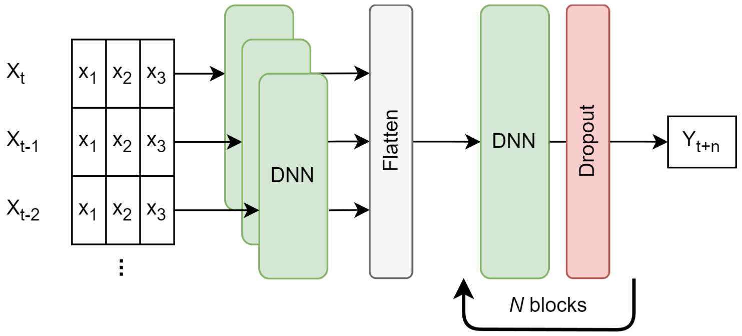


Figure 4.7: The architecture for the DNN model employed in this thesis.

Drawing inspiration from the study conducted by Kanamoto et al. [2019], which demonstrated satisfactory results utilizing a DNN on a dataset of comparable scale, a fully-connected DNN with a similar architecture to the one illustrated in Figure 3.2 was implemented.

The proposed model incorporates a *Dense* input layer that acts as a feature projector for each timestep in the input sequence. It reduces the dimensionality of the feature vectors while maintaining and compressing their original information content, effectively creating a sort of "embedding" vector for each timestep. This approach was adopted to control the neuron count following the subsequent flattening operation, which can explode when dealing with lengthy feature vectors. The resulting input is then directed through a series of hidden *Dense* layers, each of which uses a ReLU activation and is followed by a *Dropout* regularization. The final hidden layer feeds its output into a single neuron, constituting the model's output layer. For the concrete implementation details of the model, refer to Appendix E.1.

4.2.2 Recurrent Neural Network

Recurrent Neural Networks form a crucial part of this study, given their capability to handle sequential data, which is inherent in the overall research goals. Furthermore, an extensive body of current research, both directly related to and associated with similar forecasting problems, demonstrates significant success with these architectures. In line with these findings, two variations of RNNs were deployed: the standard simple recurrent network, and the LSTM cell variation. While there are additional variations of RNNs, LSTM appears to be the most popular

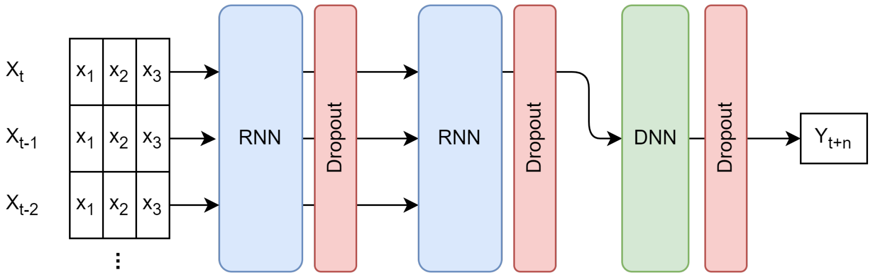


Figure 4.8: The architecture for the recurrent models employed in this thesis.

one in the current literature. As for their architecture as implemented in this thesis, they share the same architecture but have distinctive recurrent cells, illustrated in Figure 4.8, where the "RNN" can be exchanged for either the TensorFlow *SimpleRNN* or *LSTM* variation.

Each model starts with a recurrent layer which is succeeded by a Dropout layer. The first recurrent layer returns its sequences, ensuring the comprehensive sequence of hidden states propagates forward. Subsequently, a second recurrent layer mirrors the first without returning the sequence, transmitting only the final state. This is followed by another Dropout layer. The result of the second recurrent layer is then forwarded to a fully-connected *Dense* layer with a ReLU activation followed by another Dropout layer. The output from the dense layer is then fed into the output neuron. For the concrete implementation details of the model, refer to Appendix E.2.

4.2.3 Convolutional Neural Networks

There are two different architectures employed in this thesis that utilizes convolution mechanisms: a regular CNN model and an LSTM with a convolutional embedding layer.

CNN

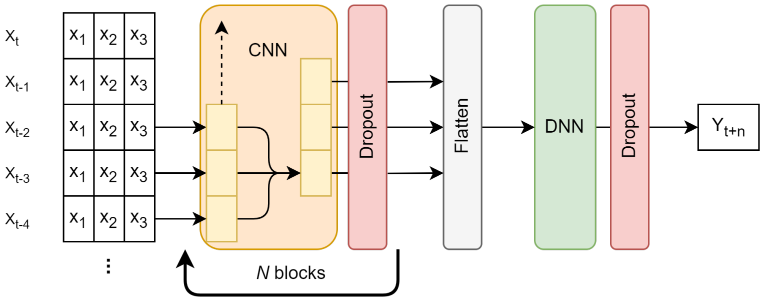


Figure 4.9: The architecture for the CNN model employed in this thesis.

CNNs have been successfully applied to time series tasks, exemplified by the promising results presented in various research studies. Nevertheless, the model deviates from some established architectures specific to time series, such as the TCN proposed by Bai et al. [2018]. While their TCN model delivers a sequence-to-sequence model deploying causal and dilated convolutions, the sequence-to-one implementations in this study allow the use of a simpler standard CNN network architecture, transforming an input sequence into a single prediction for the future.

The model consists of a sequence of convolutional blocks, each composed of a 1-dimensional convolution operation and a dropout operation for regularization. The convolutional layers use a ReLU activation function. The convolutions operate along the temporal dimension to model local temporal correlations in the data, generating a set of high-level features toward the end of the block sequence. The 2D output of the convolution layers is subsequently flattened into a 1D dimensional vector, allowing it to be fed into a final dense layer with ReLU activation and a Dropout layer, before entering the output neuron. For the concrete implementation details of the model, refer to Appendix E.3.

CNN-LSTM

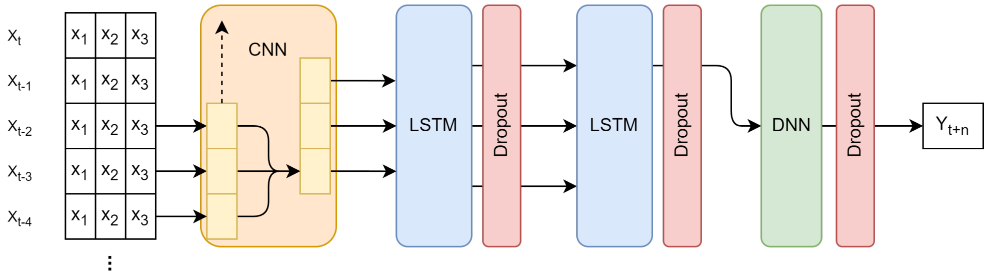


Figure 4.10: The architecture for the CNN-LSTM model employed in this thesis.

Building upon the successful applications of recurrent networks and the improved results reported by Spadon et al. [2022] and Syed and Ahmed [2023] using CNN-LSTM architectures, a CNN-LSTM is adopted in this study. However, in contrast to the aforementioned studies, the CNN layer within the adapted model does not convolve over each feature vector in the sequence. Given the non-spatial connectedness of the features in this thesis, it instead convolves over the temporal dimension, generating temporal embeddings as a form of additional temporal abstraction, potentially providing an enriched data representation for the subsequent recurrent layers.

The architecture of the CNN-LSTM model, illustrated in Figure 4.10, is essentially the same as the RNN model presented in Section 4.2.2, but adds a CNN layer before the first recurrent layer, performing a temporal convolution projection. This layer convolves over the temporal dimension without padding, functioning like an embedding mechanism that incorporates the context from surrounding timesteps. For the concrete implementation details of the model, refer to Appendix E.4.

4.2.4 Transformer

Amongst the models employed within this study, the Transformer architecture stands as a pivotal entity. Predicated on the transformer models' successful application in various time series forecasting tasks, such as in the work of Zerveas et al. [2021], the inclusion of this architecture is justified. Moreover, the model's introduction to this specific problem of forecasting financial instruments with AIS data broadens the empirical landscape, contributing to the exploration of diverse model architectures within this domain. Figure 4.11 depicts the implemented Transformer architecture, which adopts the structure delineated by Zerveas et al. [2021].

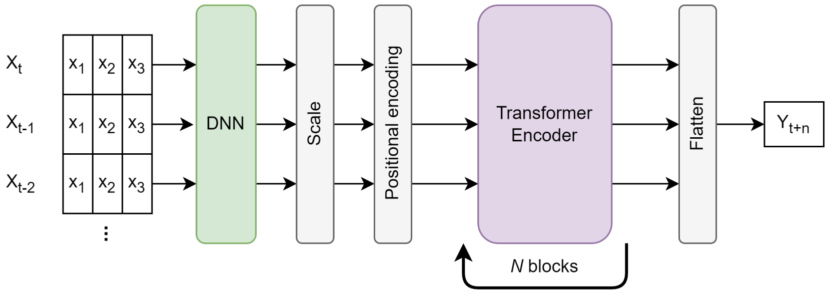


Figure 4.11: The architecture for the Transformer model employed in this thesis.

Moreover, the Transformer Encoder from the *Tensorflow Model Garden* library was utilized when implementing the Transformer model. This is an official repository and package containing a large number of state-of-the-art models and modeling solutions.

The initial layer of the model is a Dense layer, intended to project the sequences' feature vectors down to the encoders' head dimension. This projection is then scaled by the square root of the heads as suggested by Vaswani et al. [2017] that aids in managing the magnitude of input values, maintaining the stability of gradients during backpropagation. Subsequent to scaling, positional encoding is embedded into the input. While the original paper proposed a fixed sinusoidal and cosine positional encoding, several implementations, including Zerveas et al. [2021], have seen improved performance from learnable position encodings, which will be explored during the experimentation. The encoded input will then pass through a stack of transformer encoder blocks, facilitating the modeling of long-range dependencies within the sequences. The output from the encoders is subsequently flattened and linearly transformed to the output neuron. For the concrete implementation details of the model, refer to Appendix E.5.

4.3 Explanation Methods

This section will describe and elaborate on the implemented approaches to generating explanations for the various models presented in this chapter.

4.3.1 Interpretation for Non-Technical Users

An underlying motivation behind the selection of the explanation methods stems from the collaboration with domain experts, requiring the explanations to be accessible and comprehensible to those without technical background. Consequently, exceedingly model-specific explanations, which demand an in-depth comprehension of the foundational principles intrinsic to the diversity of models, are impracticable. The structure of the features comprises port relations and shipping variables, both of which are areas that maritime domain experts typically comprehend. Therefore, feature attribution methods have the potential to offer a medium of explanation that is well-aligned with the expectations and terminology of maritime domain experts.

4.3.2 SHAP

SHAP stands out as the logical choice to meet the needs outlined for this study. Its rich capability to perform feature attribution matches the guiding motivation to render intricate models comprehensible for non-technical domain experts. Moreover, as a model-agnostic method, it can be applied to any of the derived models, allowing comparisons of explanations. SHAP's method of attributing the influence of each feature to the outcome of the prediction facilitates an understanding of how various factors contribute to the model's decision. This approach coincides with the understanding of maritime domain experts who are already conversant with the variables in play and their potential impacts.

A prevalent *SHAP* Python library is an increasingly popular framework for explaining the behavior of various deep learning models. It integrates with both TensorFlow and PyTorch, enhancing its utility across various computational contexts. The library utilizes several of the methods discussed in Chapter 3. For instance, *GradientExplainer* combines ideas from Integrated Gradients (IG), SHAP, and SmoothGrad into one expected value equation, and the *DeepExplainer* presents a high-speed approximation algorithm for SHAP values in deep learning models inspired by DeepLIFT. However, their *DeepExplainer* was incompatible with the more complex models employed; it functioned as expected for the DNN, but not for the rest of the models. Nevertheless, the *GradientExplainer* worked for all models and was adopted to generate explanations for all models.

4.3.3 Custom SHAP Explanations

Regrettably, due to the utilization of Python scripts instead of Notebooks, the SHAP library did not have the ability to produce multi-sample explanations using Matplotlib. Consequently, alternative visualizations and methodologies had to be developed.

Feature Heatmaps

Feature heatmaps were created in order to visualize the relationship between temporal and feature dimensions, inspired by the work of Li et al. [2022a]. Although the features in this context are not spatially connected, the adaptation of this method by separating the visualization per shipping variables offers a clearer understanding of each feature's role and its temporal dependencies.

The SHAP values are used to construct the heatmaps, where for overall shipping variables like speed, all port relations have to be averaged to produce the plots. Moreover, when these feature heatmaps are produced for global explanations, the SHAP values also have to be averaged across the explanation period.

The resulting plots display the SHAP value on the vertical axis, the look-back periods on the horizontal axis, and higher feature values are colored red, whereas lower feature values are colored blue. The plot implementation can be found in Appendix G.1.

Violin Plots

Violin plots were not directly generatable through the SHAP library, necessitating a custom solution employing Seaborn's *kdeplot* function. In this workaround, numerous subset bins were generated for a subset of the feature values, yielding a binned distribution plot that bears a resemblance to traditional violin plots, albeit with reduced fidelity.

These custom violin plots operate fundamentally the same as their conventional counterparts. The horizontal axis represents the SHAP value, the distribution of occurrences is represented on the vertical axis, and the bin color signifies the mean value of the values enclosed within that specific bin. The color scheme is the same as for the feature heatmaps. The plot implementation can be found in Appendix G.2.

4.3.4 TimeSHAP

KernelSHAP, an explainer from the SHAP library capable of modeling any function, is not straightforward to apply to time series data. Therefore, a *TimeSHAP* implementation exists, as proposed by Bento et al. [2021], to introduce the KernelSHAP explainer to time series data.

However, while the library is readily available via pip, it was not compatible with the specific versions of Tensorflow and Tensorflow Models used in the thesis due to its usage of deprecated types in NumPy. Therefore, **TimeSHAP was not implemented in the study**.

Chapter 5

Experiments and Results

This chapter outlines the structured sequence of experiments designed to address the empirical research questions of the study, along with presenting the results obtained from each experiment. These experiments aim to provide insights beyond the existing literature, investigating the efficacy of using AIS data for financial instrument modeling within the maritime industry and studying the decision-making processes of the models.

To uphold the principles of rigorous scientific inquiry, the chapter also details the experimentation plan and setup, facilitating academic reproducibility.

5.1 Experimentation Plan

This section details the employed experimentation plan, which was designed to address the empirical research questions of the study. The focus was twofold: first, to evaluate the performance of AIS data and machine learning in forecasting financial instruments, and second, to acquire insight into their decision-making processes through XAI methods. The plan was structured around the following empirical research questions:

Research question 1.3 *To what extent can AIS data be used to model financial instruments in the maritime industry successfully?*

Research question 2.2 *Do the AIS-driven forecasting models provide meaningful and interpretable explanations as assessed by a domain expert within maritime trade analytics?*

Research question 2.3 *Do the explanations provided by the models resonate with established knowledge within the maritime trade analytics field?*

Research question 2.4 *Do the models yield novel insights into the interplay between the world fleet's behavior and the maritime trade market?*

To address these research questions, a series of experiments were conducted on a variety of financial instruments related to the maritime sector, each subject to different degrees of external influence. The selected instruments ranged from freight rate indices, which are closely associated with the operational performance of vessels, to Exchange-Traded Funds (ETFs) and stock prices, which incorporate higher levels of speculative and investment influence. This breadth of coverage was intended to test the versatility and robustness of AIS data under diverse scenarios, ranging from those tightly linked to real-world ship performance to those affected more by external market speculation. Additionally, each model was evaluated against various configurations of AIS complexities and look-back periods, as further detailed in Section 5.2.1, to further assess the effective extent of the AIS.

The majority of the evaluated instruments were linked to the dry bulk segment, as preliminary AIS data analysis indicated that this segment constitutes a significant portion of AIS data. However, one experiment was conducted on an instrument within the Tanker segment to provide a comparison between the two largest segments contained in the AIS data.

The models were assessed based on several metrics and benchmarked against baseline models. These baselines, chosen for their relevancy and wide usage in the field, would provide a clear standard against which the advanced models' performances were measured.

Upon obtaining the results, the explanations from the most promising models and data configurations were assessed with assistance from the domain experts to rationalize the decision-making process of these models. To maintain a reasonable scope for this thesis, explanations through XAI methods were limited to the best-performing models and configurations.

5.1.1 Baselines

Baseline models, being fundamental or conventional algorithms, help establish an initial comparison metric for evaluating the performance of more advanced models. They are generally used to establish a 'baseline' performance level, which these advanced models should ideally exceed. In the context of this study, two baseline models were employed, namely a Linear Regression model and an ARIMA model.

Linear Regression

Linear Regression, an uncomplicated, efficient, and interpretable model, served as one of the baseline models in this research. Despite its simplicity, Linear Regression is extensively applied in predictive analytics due to its easy interpretation and minimal computational overhead. While its assumption of linear relationships may not be useful in complex scenarios, its history as a scientific goto for robust modeling and interpretation makes it a reasonable candidate. The linear regression model from the *Scikit-learn* ML library was employed, and its implementation within this study involved flattening the temporal dimension of the data, given that linear regression models are not inherently designed to handle temporal information.

ARIMA

As the default forecasting algorithm for years, ARIMA positioned itself as a natural baseline in this study. However, unlike the evaluated models, ARIMA explicitly models temporal dependencies of the target value itself. In this study, ARIMA forecasts the instruments based on their historical values, differing from the other models that solely leverage AIS data. Consequently, **ARIMA's results are not directly comparable to those of the other models**. However, its inclusion in this study offers a comparative measure of the performance of AIS-only prediction models against a traditional time-series forecasting method. The AIS-only models were not expected to outperform ARIMA, but the comparison would likely offer valuable insights. The ARIMA model from the *Statsmodels* ML library was employed.

5.2 Experimental Setup

This section outlines the methodological blueprint for the conducted experiments, incorporating the pertinent data and parameters, with the purpose of providing a comprehensive overview that would allow researchers to reproduce the experimental procedures. The section first elaborates on the various employed AIS complexities before detailing the parameters used in each model, as well as the baselines. It will then elaborate on the training process and the various evaluations of both the models' performances and explanations.

5.2.1 Data Configurations

Various distinct sets of data configurations were intended to assess the impact of various combinations of data complexity and look-back periods in the experiments. These configurations were devised to align with available data and computational resource constraints, while simultaneously aiming to optimize the results. The goal was to examine a broad range of possibilities within set parameters to determine a suitable data representation for modeling maritime financial instruments using AIS data. **A consistent forecast horizon of 14 days was adopted**, offering practical short-term predictions. This fixed forecast horizon reduced an additional dimension from the array of configurations and provided valuable and timely forecasts, which is important in the volatile context of the maritime industry.

Three different data complexities were explored, where each complexity is tied to the number of port relations per shipping variable:

Low complexity The low-complexity configurations set $|P| = |P_{in}| = |P_{out}| = \mathbf{3}$, meaning that the three overall most active ports were selected. For each of these high-activity ports, relations were created to the three ports with the highest outflux traffic flow, along with the three ports with the highest inbound flow. This resulted in $3 \cdot 3 + 3 \cdot 3 = \mathbf{18}$ port relations, and subsequently $18 \cdot 4 = \mathbf{72}$ total features when multiplied by the shipping variables.

Medium complexity The medium-complexity configurations set $|P| = |P_{in}| = |P_{out}| = \mathbf{6}$, meaning that the six overall most active ports were selected. For each of these high-activity ports, relations were created to the six ports with the highest outflux traffic flow, along with the six ports with the highest inbound flow. This resulted in $6 \cdot 6 + 6 \cdot 6 = \mathbf{72}$ port relations, and subsequently $72 \cdot 4 = \mathbf{288}$ total features when multiplied by the shipping variables.

High complexity The high-complexity configurations set $|P| = |P_{in}| = |P_{out}| = \mathbf{9}$, meaning that the nine overall most active ports were selected. For each of these high-activity ports, relations were created to the nine ports with the highest outflux traffic flow, along with the nine ports with the highest inbound flow. This resulted in $9 \cdot 9 + 9 \cdot 9 = \mathbf{162}$ port relations, and subsequently $162 \cdot 4 = \mathbf{648}$ total features when multiplied by the shipping variables.

For the look-back periods, three different scenarios were explored: $t_{-n}, \ldots, t, \quad n \in \{20, 30, 40\}$. The selection of these look-back intervals balanced the constraints of the available data and the need to incorporate a comprehensive historical context into the model.

These factors yielded a comprehensive permutation of data configurations tested in each experiment, facilitating a systematic comparison across various setups. The configurations are referred to like *low/medium/high complexity 20/30/40*, where a *medium complexity 30* would refer to a configuration of medium complexity using 30 days as a look-back period.

5.2.2 Model Parameters

This section clarifies the parameters employed in each model utilized in the experiments. Each model's configuration is detailed, facilitating a comprehensive understanding of the experiments carried out, and laying the foundation for replication of results.

Moreover, owing to the large-scale nature of the experiments and computational constraints, the model parameters will be kept constant throughout the experimentation phase, using a rough set of manually-defined parameters that have proven reasonably effective.

Baselines

For the linear model, all parameters available in the *Scikit-learn* linear regression model were set to the default parameters.

For the ARIMA model, the parameters for the autoregressive terms, differentiations, and moving average components were taken directly from the works of Sarantopoulos [2021], which found an optimal set of parameters for predicting the Baltic Dry Index. The number of autoregressive terms was set to 3, the number of differentiations was set to 1, and no moving average was set. In the *Statsmodels* ARIMA model, this would mean the parameter *order = (3,1,0).* The remaining parameters were set to their default values. While the various experiments could see better results from the ARIMA model using a distinct set of per-experiment parameters, these parameters were used for all experiments.

Fully Connected Deep Neural Network

The Fully Connected Deep Neural Network (referred to as *DNN* in subsequent experiments) incorporated an initial layer that is 10% the size of the feature vectors, resulting in a layer of shape $(sequence\_length, 0.1 * num\_features)$. After flattening this input layer, the model employed 8 hidden layers with 512 neurons, each with ReLU activation and accompanied by a dropout rate of 20%.

Recurrent Neural Network

The recurrent networks tested in this study, both RNN and LSTM, featured 128 units in the recurrent layers and 128 neurons in the final dense layer. After every layer, a dropout of 20% was applied. Two recurrent models were examined with RNN and LSTM cells, respectively.

CNN

The CNN model incorporated 5 sequential convolutional layers, each of which had 128 filters, a kernel size of 3, ReLU activation, and a dropout of 20%. The final dense layer after the flatten operation had 128 neurons, ReLU activation, and a dropout rate of 20

CNN-LSTM

The CNN-LSTM model incorporated an initial convolutional layer with a filter size equating to 10% of the number of features and a kernel size of 5. The remaining architecture and parameters mirrored that of the previously outlined recurrent models.

Transformer

The Transformer model (referred to as *TF* in subsequent experiments) employed 2 encoder blocks, each with a head dimension of 300 over 4 heads. The first dense layer of the encoder's two-layer feedforward network incorporated 256 units, ReLU activation, and a 10% dropout. Additional dropouts for the encoders' attention mechanisms and output layers were also set to 10%, including the dropout layer following the input embedding. All other parameters followed their default values, including the *TransformerEncoderBlock* from the TensorFlow Model Garden library.

5.2.3 Training Process

The overall training process was designed to ensure the best possible model results for each permutation of the data configurations. Every model was created through the development of five candidate models to reduce the chances of model instability. Among the candidates, the one yielding the lowest MSE loss on the validation dataset was selected as the representative for each respective model.

Computational Resources

The training was conducted using Google Cloud Compute Engine on an *n1-highmem-4 machine*, equipped with four cores (Intel Broadwell CPU), 26 GB of memory, and an Nvidia T4 GPU. The image "c1-deeplearning-tf-2-10-cu113-v20230501-debian-10-py37" was employed to facilitate a preinstall of various libraries and CUDA/cuDNN drivers, which are crucial for GPU-accelerated deep learning tasks. Moreover, to accommodate specific library requirements, Python 3.9.9 was manually installed. A list of the libraries and their corresponding versions can be found in Appendix F.

Hyperparameters and Optimization

All models were optimized using the MSE loss metric and the *Adam* optimizer - an adaptive, robust, and efficient optimizer especially aimed towards datasets with high-dimensional parameter spaces [Kingma and Ba, 2014]. Given the limited size and the high complexity of the data, a smaller batch size of 16 with a learning rate of $5e-5$ was used to train the models. The other Adam parameters were set to their default values as provided by Tensorflow. The models were trained for a maximum of 500 epochs, and a custom early-stopping algorithm was implemented to ensure training efficiency and optimal model performance.

Early Stopping Mechanism

In order to streamline training efficiency, optimize performance, and avoid overfitting, a custom early-stopping mechanism was implemented. These mechanisms typically terminate the training process once the model begins to overfit and subsequently restore the weights to the optimal state. The mechanism implemented was designed to terminate the training if there was not a new lowest validation loss observed for a span of 40 successive epochs.

However, due to the absence of both a testing and validation set, and considering the occurrence of pre-emptively low validation loss, a constraint was imposed to ensure that the training loss was approximately equivalent to or less than the validation loss. This constraint ensured that the best model also demonstrated proficiency in modeling the training data, rather than achieving fortunate results with the validation data alone.

Should 40 epochs elapse without an improvement in the validation loss when adhering to this constraint, the training process would be halted, and the weights would revert to those which yielded the best validation loss. This strategy of early stopping is elaborated in detail in Algorithm 2.

Algorithm 2 Early stopping logic

```
$bestloss \leftarrow \infty$
$bestmodel \leftarrow$ copy of initialized model
$e_{max} \leftarrow$ max epochs without improvement

After each epoch:
$L_t \leftarrow$ training loss, $L_v \leftarrow$ validation loss
if $0.8 \cdot L_t < L_v < bestloss$ then                    ▷ Improvement
    $bestloss \leftarrow L_v$
    $bestmodel \leftarrow$ clone of model
else if no improvement for $e_{max}$ epochs then
    Stop training process
end if
```

5.2.4 Performance Evaluation

The typical training-validation-testing split was impractical due to the data's temporal nature and limited size. Therefore, only a training set and testing set were utilized, with the latter also functioning as the validation set during training. The performance of the models was evaluated using several standard metrics and additional metrics pertinent to the time series prediction task.

Training and Test Sets

Following the best practices for time series forecasting derived from Hyndman and Athanasopoulos, which underscores the imperativeness of training sets being temporally before test sets, the last portion of the training data was used as the test set for the experiments [Hyndman and Athanasopoulos, 2014, p. 50–52]. They also mention a cross-validation procedure called *rolling forecasting origin*, which iteratively rolls a test set through the time series using the past data to train. While this generates a rigorous foundation for testing, it was, in this case, unfeasible given the limited dataset size and the already exhaustive permutation of models and data configurations. Therefore, a test set at the end of the time series was used for validation, consisting of the last **120 days** between the 13th of October 2022 and the 21st of March 2023.

Additionally, the look-back periods were included in the test sets. Consequently, in a data configuration utilizing a 30-day look-back period, the total size of the test set would be $120+30 =$ **150**, but 120 samples would be evaluated. This was done to avoid contamination between the training and test sets, ensuring that none of the same time steps were included in the look-back periods of both training and evaluation.

Performance Metrics

Performance metrics are quantitative measures used to evaluate the quality of predictions generated by ML models, providing a comprehensive and multifaceted foundation for comparing the various models and data configurations. The metrics were applied to the models' predictions on the test data, gauging how well they performed on unseen data. In this work, a combination of both standard and custom metrics was employed to evaluate the forecasting models' performance from various perspectives. Some metrics facilitated the evaluation of the models' proximity to the overall target - being particularly important as the models did not have direct access to the financial instrument when predicting, relying exclusively on the AIS data. Further, some metrics assessed how effectively the models could predict the value change of instruments from time t to t_{+14}.

Mean Squared Error Mean Squared Error (MSE) is one of the more popular metrics for regression tasks. It quantifies the average of the squared differences between the predicted and target values. A lower MSE is indicative of better model performance, demonstrating that the model's predictions align more closely with the targets.

Same-Sign Proportion A Same-Sign Proportion (Sign) metric was employed to evaluate the model's ability to predict the directional change between time t and the predicted t_{+14}. The Sign metric calculates the proportion of test samples where the model correctly anticipates the sign of this change, i.e., if the price will go up or down over the next 14 days. A higher Sign value is therefore desirable, indicating that the model is better able to predict the direction of the instrument's movement. More specifically, values over 0.5 indicate that the model predicts the correct directional change for more than half of the validation samples.

Concordance Correlation Coefficient The Concordance Correlation Coefficient (CCC) extends the concept behind the Same-Sign Proportion metric. CCC evaluates the degree of agreement between the predicted and actual changes, encompassing both directions *and magnitude*. A high CCC value implies a stronger agreement between the predicted and actual changes, which is critical for accurate financial forecasting. For instance, a forecasted change of -100, when the actual change is -1, will retain the same sign; however, it would result in a low CCC due to the considerable difference in magnitude.

5.2.5 Explanation Evaluation

An objective quantitative evaluation of explanations is generally challenging. While the overall evaluation properties from Molnar [2022] are explained in Chapter 1, there were still challenges in applying and quantifying them. Moreover, the complexity and temporal nature of the features makes several of the properties very hard to evaluate. For example, stability and fidelity are hard to measure as permutations to the features in a time series are never random, and it does not make sense to permute them randomly, making it harder to find good and realistic permutations. Moreover, permuting data with numerous features is slow and impractical.

As a result, **the evaluation of explanations was conducted in collaboration with domain experts**. Some quantitative metrics were implemented, but not used as a means of comparison, and are referenced found in the performance matrices presented in Appendix H:

Variance Variance measures the variance of the SHAP values for each model. A lower variance means that a model gives more consistent explanations. This could be viewed as a measure of explanation certainty; however, it assumes that features should maintain a stable contribution, which is not necessarily the case with the complex and temporal nature of the data.

Importance Importance measures the mean of the absolute SHAP values, quantifying the importance of each value. This allows for comparisons of each model, signifying how sensitive the models are to changes in the features.

5.3 Experiments

5.3.1 Experiment 1 - Baltic Dry Index

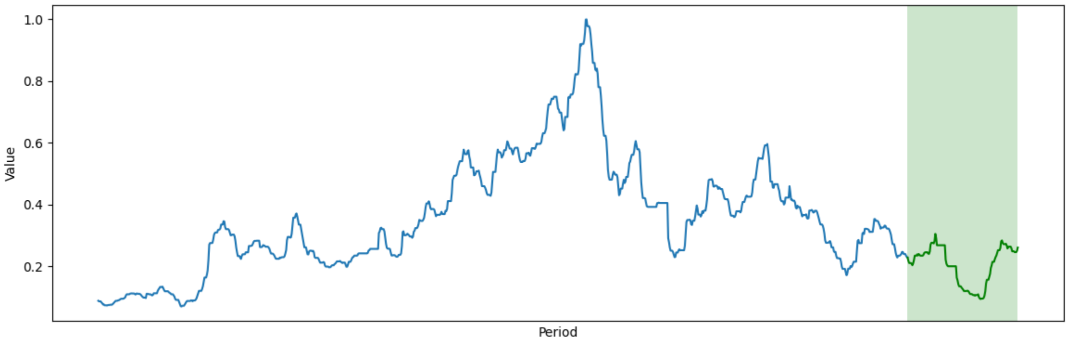


Figure 5.1: The normalized time series data for the Baltic Dry Index, with the test set marked in green.

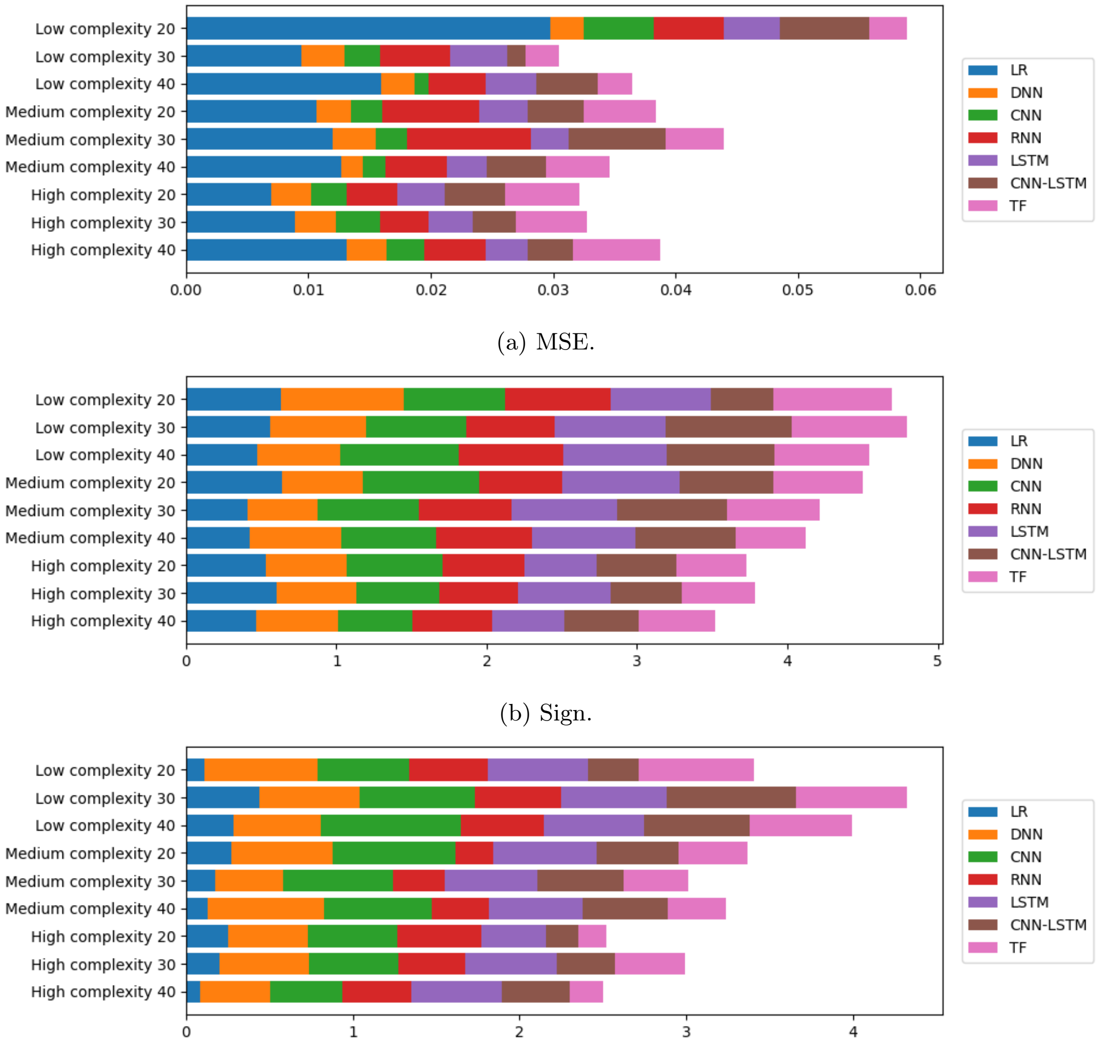
The Baltic Dry Index (BDI) is an economic indicator compiled by the Baltic Exchange that quantifies the cost of transporting various raw materials by sea. These commodities, typically transported in bulk, include items, including iron ore, coal, and grains. The BDI is calculated daily by a panel of ship brokers and their view on the current freight cost on various freight routes. This protects the index from speculative influence and roots it as a robust, demand-driven, and non-tradable index. Therefore, the BDI is considered an important economic indicator as it can provide insights into global supply and demand trends; high levels of the BDI might suggest that demand for raw materials is high, which can be a sign that economies are expanding and production is increasing. Conversely, low levels of the BDI can suggest a slowdown in global economic activity. Moreover, the correlation between the BDI and freight rates suggests a close tie to the macroeconomic conditions and international trade patterns, rendering the index an optimal candidate for prediction models using AIS data.

The index is a composite of 40% Capesize, 30% Panamax, and 30% Supramax trades. As these vessels are the primary vessels used for transporting dry bulk commodities, the AIS data used to forecast the BDI included dry bulk vessels above or at the size of the Supramax segment (40,000+ DWT) - a decision supported by experts at Maritime Optima.

Figure 5.1 shows the value of the BDI as employed in the thesis from the start of 2020 to the end of March 2023, with the green box indicating the period used for testing and validation. The test set displays a pronounced "U-shaped" trajectory, positioning itself as a more rigorous assessment of the models than that of a simpler, more linear trend; it demands not just the ability to fit a general trend, but also to capture dynamic fluctuations in the data and adjust forecasts accordingly.

BDI Model Performances

A comprehensive comparison of the various data configurations and models across the performance metrics is presented in Figure 5.2, and exact numbers can be found in Appendix H.1. Given the distinct trend in the test set, the CCC and Sign metrics hold high relevance.



(a) MSE.

(b) Sign.

(c) CCC.

Figure 5.2: Comparison of data configuration and models for the Baltic Dry Bulk Index across the performance metrics.

The findings suggest that most configurations provide the necessary conditions for the models to achieve a low MSE, except for the low complexity configuration using 20 days for the look-back period, mostly due to the poor performance exhibited by the linear regression baseline. In the case of the sign proportion and CCC, the ability of models to capture the 14-day changes decreases when the complexity of the datasets increases. Moreover, the linear regression baseline model was consistently outperformed across all metrics and all data configurations by other models, with the ARIMA model also failing to compete effectively, yielding a Sign of 0.6833 and a CCC of only 0.1495.

The best results were achieved by the CNN model on the low-complexity configuration using a 40-day look-back period, reaching an MSE of **0.0025**, a CCC of **0.8389**, and a Sign of **0.7917**. The training progression of the CNN model is presented in Figure 5.3, showing an initial sharp reduction in both the training and validation loss, followed by a steady descent until around the 80th epoch, after which signs of overfitting were observed, terminating the training process.

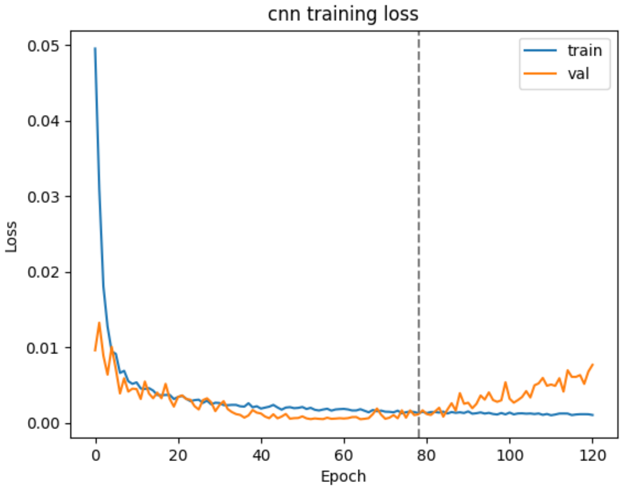
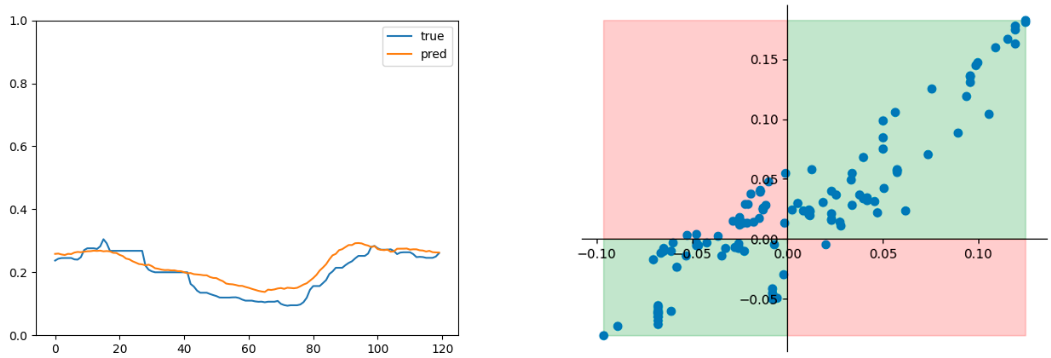


Figure 5.3: The training progression of the CNN model on the *low complexity 40* configuration for the Baltic Dry Index, with the early stopping marked as a stapled line.

The CNN model demonstrated exceptional proficiency in modeling the test set of the BDI. The model was not only able to accurately predict the overall magnitude of the new unseen period of the index, but also convincingly modeled the U-shaped trajectory. The model's predictions are shown in Figure 5.4, which compares the model's absolute predictions (5.4a) and the correlation of the $t \longrightarrow t_{+14}$ changes (5.4b) against the truth. A clear linear trend is evident between the predicted and actual changes, with some infrequent instances of directional disagreement corresponding to insignificant changes.

Nevertheless, while the CNN model achieved the best result, it is important to note that most models showed highly competent modeling ability of the BDI. The predictions of each model on the test set are presented in Appendix I.1, but on the *low complexity 30* configuration, as this yielded the best overall performance from all models.



(a) The predicted vs. actual absolute values.

(b) The predicted vs. actual 14-day changes.

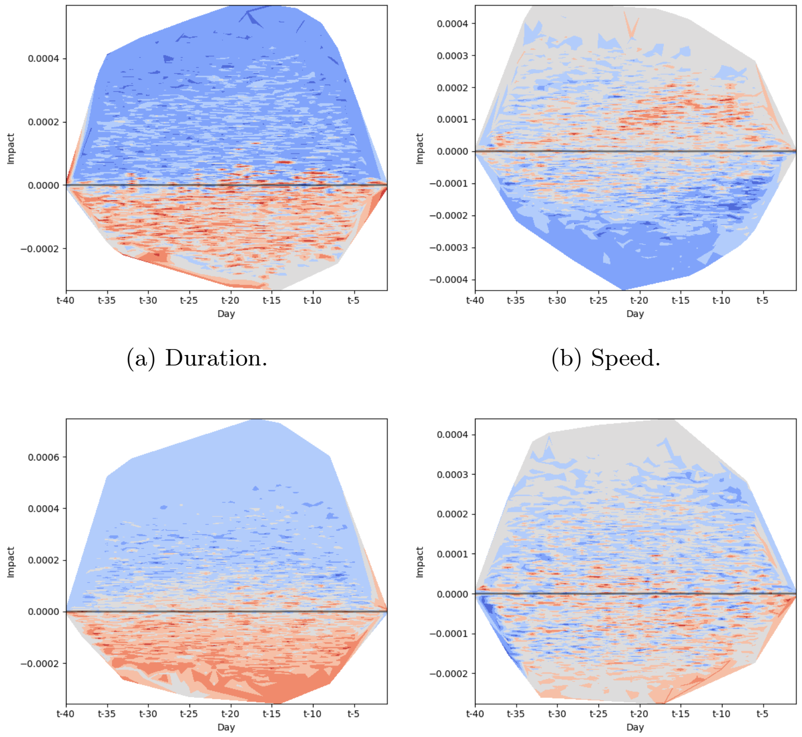
Figure 5.4: The predictions made by the CNN model on the Baltic Dry Index test set using the *low complexity 40* configuration, contrasting them against the truth.

BDI Model Explanations (Training Set)

Overall, the low-complexity 40 CNN model demonstrated decision-making principles consistent with established knowledge, as depicted in Figure 5.5, showcasing the average impact of each shipping variable during the look-back periods across the entire BDI training set.

The figure indicates that the model shows some understanding of how extended travel time between ports typically corresponds to a lower market condition (5.5a), as vessels typically reduce their speed to conserve fuel. Conversely, when the duration of voyages between ports is reduced, the model appears to associate that with higher market conditions, consistent with how vessels adjust their speed. This is also in line with the contributions of the speed values (5.5b), where the model assigns higher market values to higher speeds between ports, although this divide is not as clear. In the case of the load factor, the model delineates a clear difference between higher and lower load factors and their effect on the market; the model assigns a higher BDI value when vessels are transporting less load than usual, and vice versa for high loads. Traffic, i.e., the current number of vessels en route, manifests greater complexity, indicating that traffic might behave variably across different ports and diverse timeframes, hinting at a more nuanced interaction with the market dynamics. Consequently, the apparent arbitrariness in high and low traffic patterns across ports and periods could highlight the limitations of averaging traffic data for explanations, emphasizing the need for a more nuanced, contextual interpretation of traffic metrics in relation to market conditions. That being said, there seems to be a slightly higher emphasis on that higher traffic overall contributes to a lower BDI.

Figure 5.6 shows the five most important, i.e., the features with the most impact on the BDI, features across the training set. An extended plot of the top features can be found in Appendix J.1, together with plots for each shipping variable. The majority of the features are attributed in line with principles from established knowledge. However, the duration time from Changzhou, China (CNCZX) to Port Hedland, Australia (AUPHE) deviates from the overall principles, suggesting that longer travel times occur when the market is high or is growing. While this might seem like an anomaly, several situations can explain the pattern and will be discussed in Chapter 6.



(a) Duration.

(b) Speed.

(c) Load.

(d) Traffic.

Figure 5.5: Average contribution of shipping variables in the look-back periods across the Baltic Dry Index training set. Orange colors indicate higher feature values and blue colors indicate lower feature values.

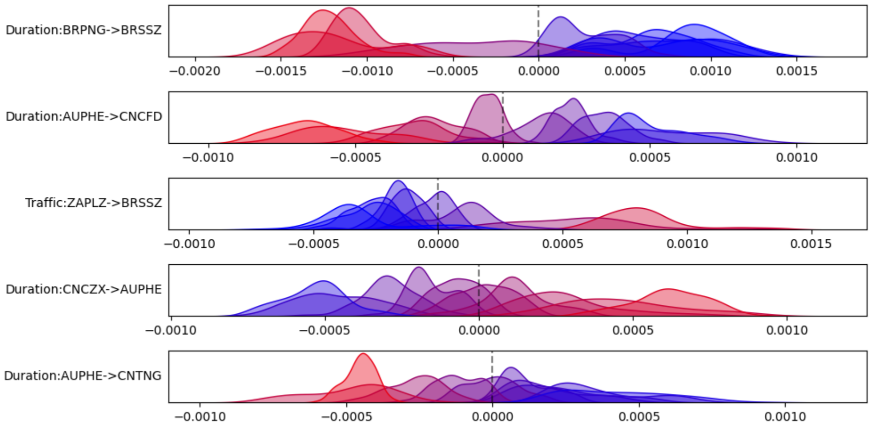
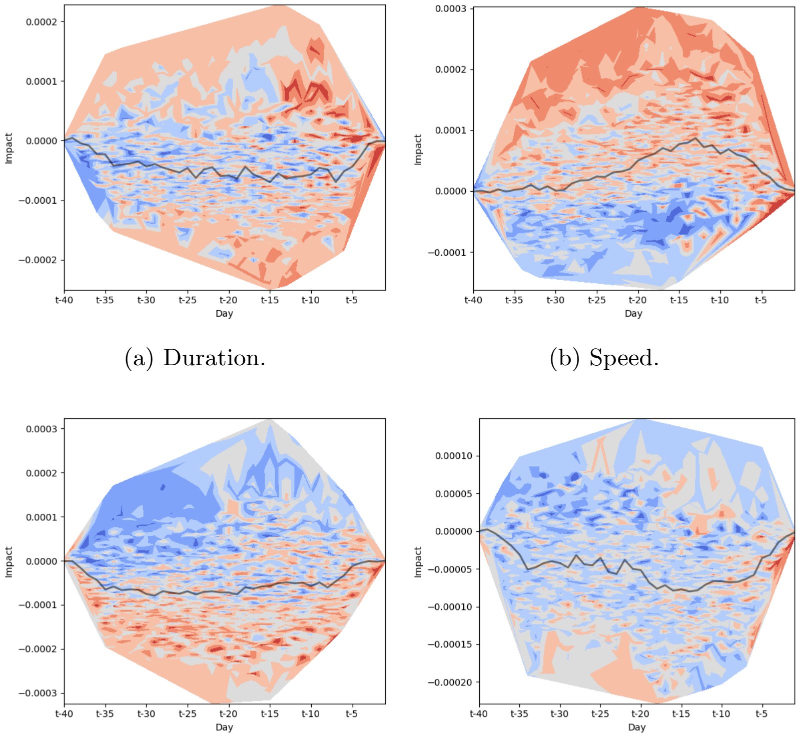


Figure 5.6: The five most impactful features of the Baltic Dry Index training set according to the best model.

BDI Model Explanations (Test Set)

Many of the model's decision-making principles remain overall consistent between the training and test sets, as displayed in Figure 5.7, where the training set was used as background distribution. Many of the same patterns reappear, especially for speed, which indicates that higher speeds contribute to a higher BDI. Additionally, the overall load contributions maintain consistency with the pattern observed for the training set, where lesser loads contribute to a higher BDI. The contribution of traffic remains quite complex even on the smaller test set, further emphasizing the possibility of traffic volume being either port-dependent or spuriously correlated. However, the duration values do not follow the same clear divide observed for the training set. Still, several possible explanations make this pattern reasonable, warranting an investigation into the various port relations.

Figure 5.8 presents the five most important features across the test period. Duration remains the most influential shipping variable of the top features, and patterns from the training set can be identified in the test set. Notably, the majority of the duration features still indicate that shorter travel times correspond to a higher BDI value, with the exception of travel times to Port Hedland (AUPHE), which continues to indicate higher BDI values when vessels spend longer time on their way to that port. The intricacies of the traffic variable remain complex, whilst the load variable continues to exhibit the established pattern whereby lower vessel load corresponds to a more buoyant market condition. The seeming anomaly load pattern observed for Port Hedland (AUPHE), being the world's preeminent iron ore export port, seems counterintuitive and warrants further discussion, which will be addressed in Chapter 6.



(a) Duration.

(b) Speed.

(c) Load.

(d) Traffic.

Figure 5.7: Average contribution of shipping variables in the look-back periods across the Baltic Dry Index test set. Orange colors indicate higher feature values and blue colors indicate lower feature values.

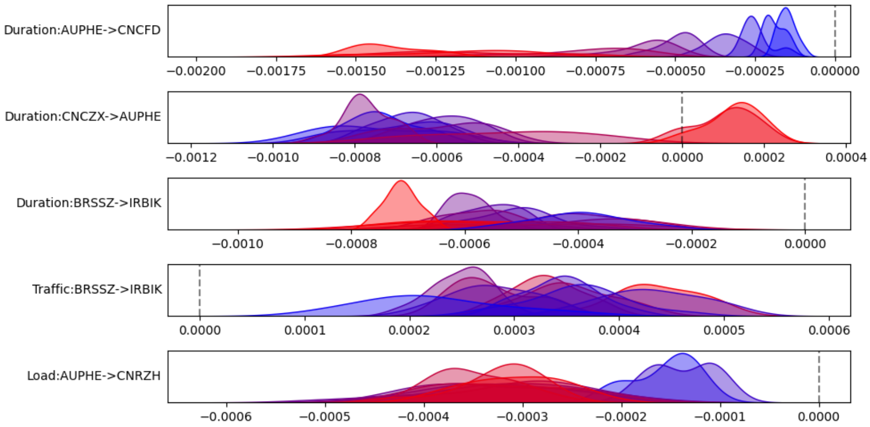


Figure 5.8: The five most impactful features of the Baltic Dry Index test set according to the best model.

5.3.2 Experiment 2 - Breakwave Dry Bulk Shipping ETF

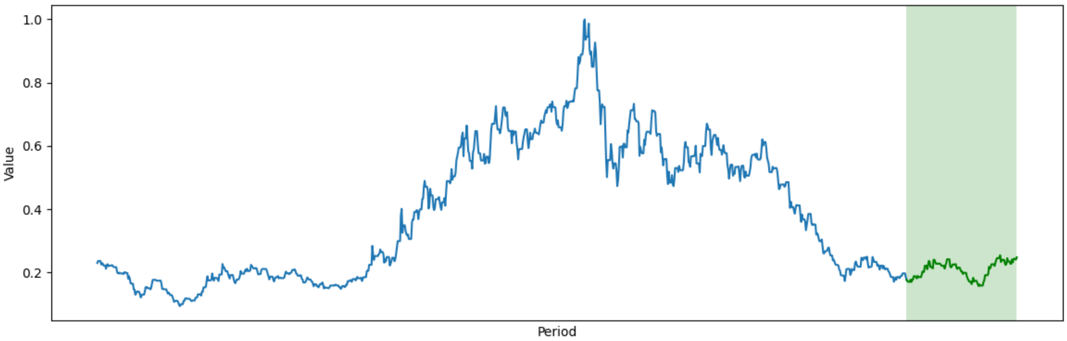


Figure 5.9: The normalized time series data for the Breakwave Dry Bulk Shipping ETF, with the test set marked in green.

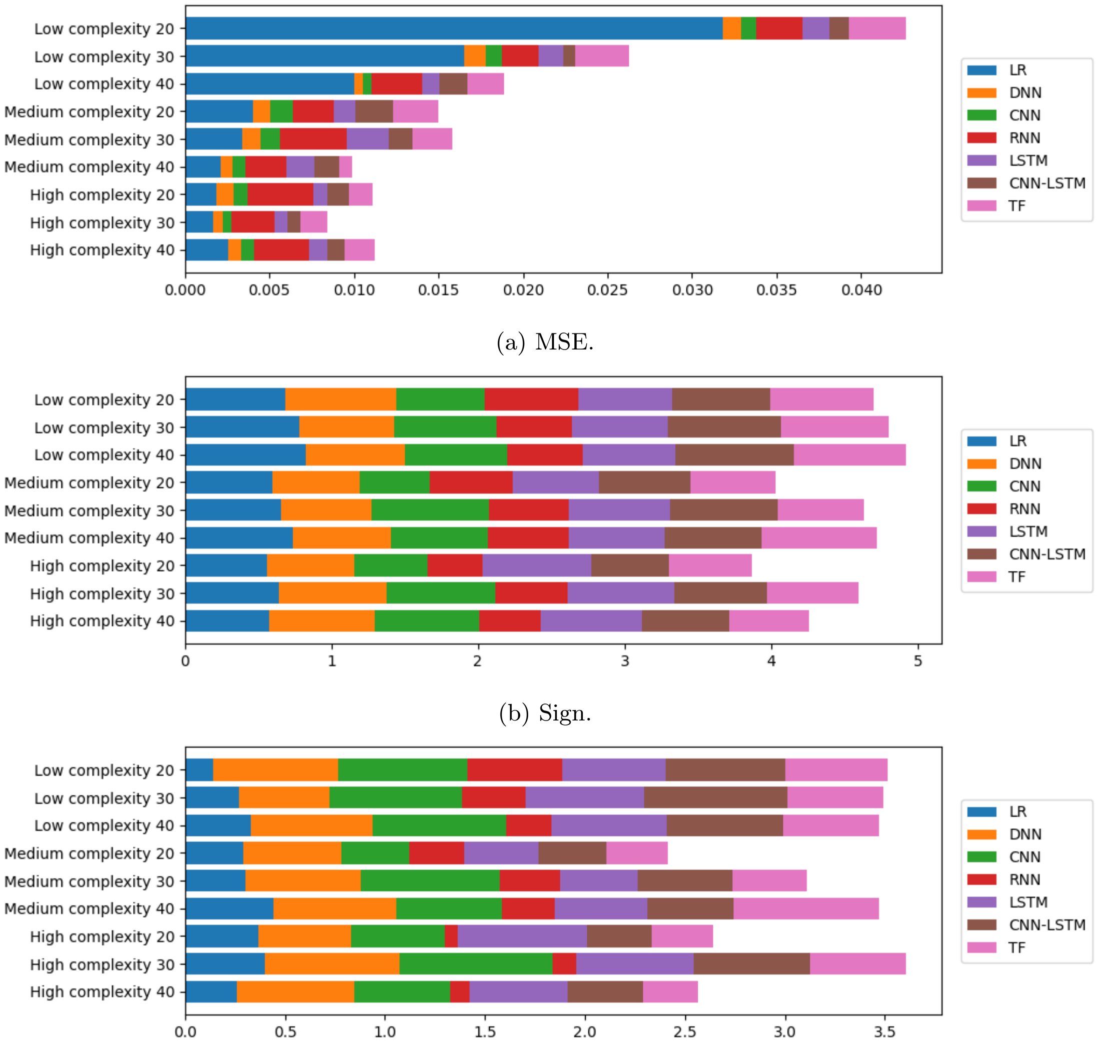
Subsequent to the models demonstrating robust abilities in forecasting the Baltic Dry Index using AIS data exclusively, a second experiment was conducted to incorporate more speculation into the instrument to further evaluate the effectiveness of the AIS.

The Breakwave Dry Bulk ETF (BDRY) is a financial instrument by Breakwave Advisors to provide investors with exposure to the dry bulk shipping market. Contrary to the BDI, BDRY is a tradable instrument that replicates the daily performance of shipping freight futures, specifically, a portfolio of near-dated dry bulk freight futures. The ETF's underlying contracts are Capesize, Panamax, and Supramax vessel futures contracts. The BDRY is influenced not only by the actual supply-demand dynamics of shipping dry bulk commodities but also by market participants' expectations and speculations regarding the future state of the shipping industry. This introduces a degree of volatility and speculation absent in the BDI, rendering the BDRY a practical subject for further assessment of the AIS's capabilities. Also, unlike BDI, BDRY allows a certain amount of tradability, which introduces an additional layer of market dynamics into the equation. The BDRY comprises the same vessels as the BDI, namely Capesize, Panamax, and Supramax trades. Therefore, the AIS data used to predict the BDRY included the same vessels as for experiment 1. Although the input features are the same, the weightage could differ based on the near-dated futures contracts it reflects.

Figure 5.9 illustrates the normalized time series data for the Breakwave Dry Bulk ETF, ranging from the start of 2020 to the end of March 2023. The resulting test set exhibits a similar U-shaped pattern to that found for the BDI, which is to be expected, albeit with a marginally diminished magnitude. Overall, the BDRY time series is very similar to that of the BDI, which is expected considering the overlapping influential factors of both instruments. Nevertheless, there are noticeable differences, rendering it intriguing to examine if the models are capable of effectively modeling these differences with the same input data.

BDRY Model Performances

A comprehensive comparison of the various data configurations and models across the performance metrics is presented in Figure 5.10, and exact numbers can be found in Appendix H.2. While the trend found in the test set is of less magnitude than for the BDI, the CCC and Sign metrics are still of relevance, although the changes they measure are relatively small.



(a) MSE.

(b) Sign.

(c) CCC.

Figure 5.10: Comparison of data configuration and models for the Breakwave Dry Bulk Shipping ETF across the performance metrics.

Similarly to the BDI, most configurations facilitated patterns that the models were able to use to model the BDRY. The linear regression baseline performed fairly poorly on the MSE metric when using less complex configurations, but improved with higher complexities and longer look-back periods, even being sufficiently competitive at the more complex configurations. As for the ARIMA baseline, it continued to perform poorly, yielding a Sign of 0.379 and a negative CCC of -0.0028. Overall, consistently better Sign and CCC results were achieved at lower complexities, but the overall best CCC for all models where achieved for the high complexity 30 configuration. The predictions of all models for this configuration are presented in Appendix I.2.

The best predictions for the BDRY were produced by the CNN model on the high complexity 30, the same model as for the BDI but on a more complex data configuration. The CNN reached an MSE of **0.0005**, a CCC of **0.7648**, and a Sign of **0.7417**. The training progression of the model is shown in Figure 5.11. The training process was terminated before showing clear signs of overfitting, which is likely due to the $train\_loss < val\_loss$ condition being triggered at around the 60-80th epoch, where the training loss was close to the validation loss and remained higher than the validation loss for the next 40 epochs.

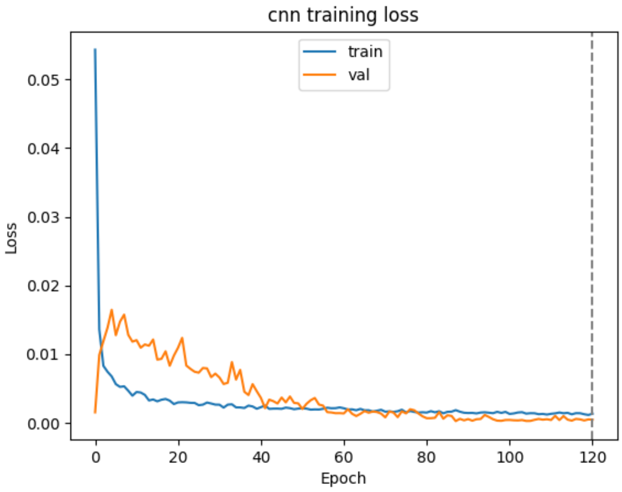
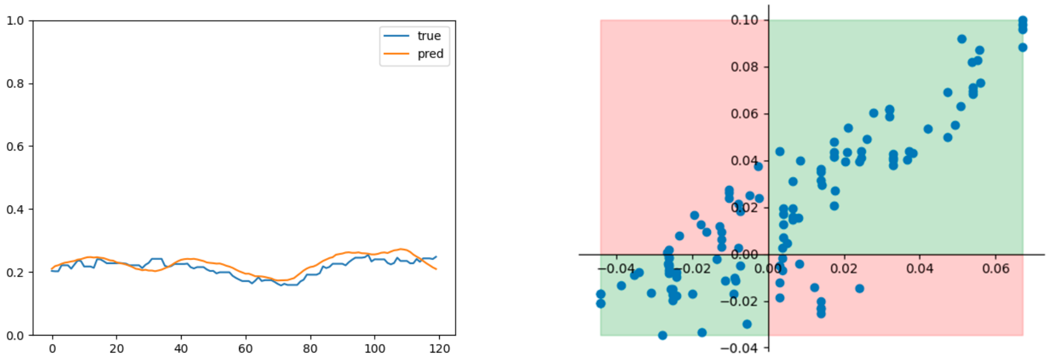


Figure 5.11: The training progression of the CNN model on the *high complexity 30* configuration for the Breakwave Dry Bulk Shipping ETF, with the early stopping marked as a stapled line.

Again, as for the BDI, the CNN model shows superior modeling ability. Impressively, while the overall magnitude of change in the test set is relatively small, the model seems to be able to model it well. Moreover, the model is able to find the precise magnitude of the ETF's value despite only being exposed to AIS data. The model's predictions are shown in Figure 5.4, which compares the model's absolute predictions (5.4a) and the correlation of the $t \longrightarrow t_{+14}$ changes (5.4b) against the truth. As with the BDI, a clear linear trend is seen between the predicted and actual changes, with a few instances of directional disagreement corresponding to very small changes.



(a) The predicted vs. actual absolute values.

(b) The predicted vs. actual 14-day changes.

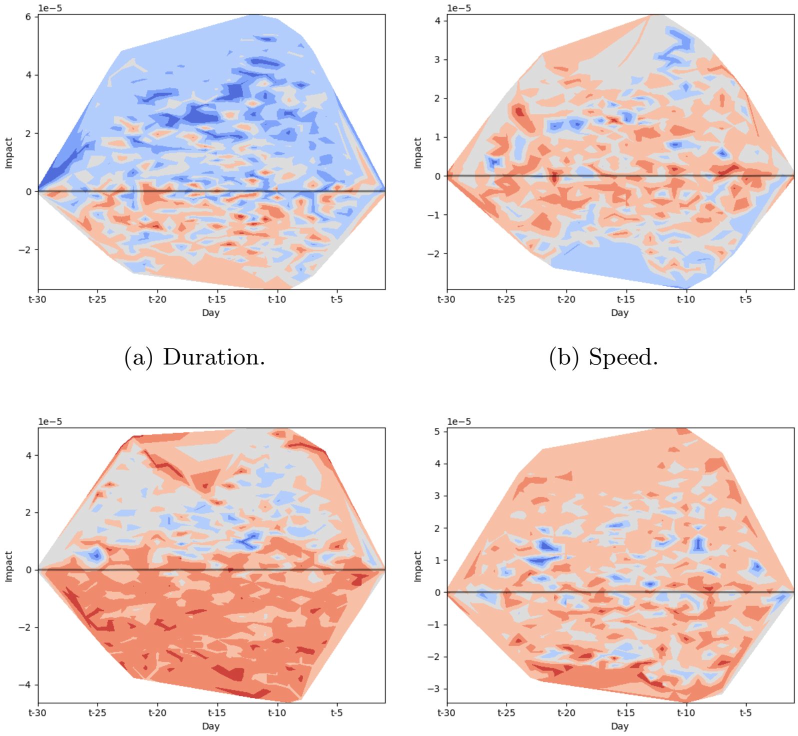
Figure 5.12: The predictions made by the CNN model on the Breakwave Dry Bulk Shipping ETF test set using the *high complexity 30* configuration, contrasting them against the truth.

BDRY Model Explanations (Training Set)

As one might anticipate, configurations of higher complexity introduce additional nuances and intricacies, which results in convoluted summaries when aggregated over the training samples, as displayed in Figure 5.13

Generally, the attributions for the duration variable align consistently with findings from the first experiment, seeing a distinct separation between shorter and longer travel times for higher and lower instrument values, with some outliers of higher instrument values attributed to longer travel duration. Interestingly, the closely related speed variable displays more complicated attributions, making it difficult to discern a general pattern in a global plot shown in the figure. The load variable attributions also remain consistent with that of the BDI experiment, suggesting that the instrument's value is typically higher when vessels have less load. As for traffic, the attributions maintain a complex relationship, similar to the first experiment; however, there appears to be a slightly higher distribution of high traffic with contributions to lower instrument value.

Figure 5.14 shows the five most important features across the training set. An extended plot of the top features can be found in Appendix J.2, together with plots for each shipping variable. It is important to acknowledge that this figure merely represents the top five determinative features out of a total of 648, situating them as components of a broader analytical challenge that necessitates a holistic view for proper interpretation. Nevertheless, the majority of the top features pertain to the duration variable, suggesting that various port relations have inconsistencies regarding how travel time impacts the instrument's value. However, for the two port relations indicating that increased travel time corresponds to the elevated value for the BDRY, there seems to be a highly skewed distribution, indicating that there might have been outliers or extreme cases somewhere in the period. Nevertheless, this is contrary to the other two relations, where lower travel time is inversely associated with the BDRY, and the distribution appears more uniformly dispersed. In terms of traffic, it suggests that a higher traffic volume from Caofeidian to Qinhuangdao in China generally contributes to a higher BDRY value in some cases.



(a) Duration.

(b) Speed.

(c) Load.

(d) Traffic.

Figure 5.13: Average contribution of shipping variables in the look-back periods across the Breakwave Dry Bulk Shipping ETF training set. Orange colors indicate higher feature values and blue colors indicate lower feature values.

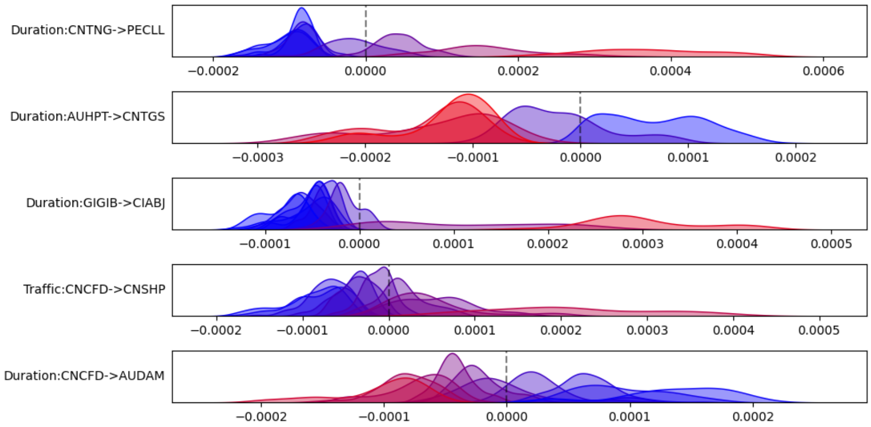
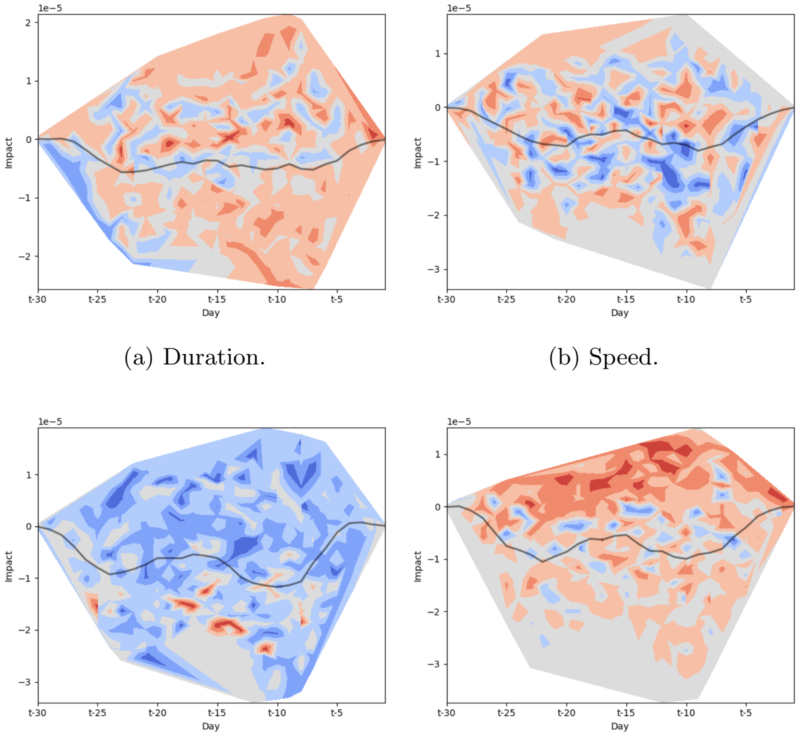


Figure 5.14: The five most impactful features of the Breakwave Dry Bulk Shipping ETF training set according to the best model.

BDRY Model Explanations (Test Set)

For the new unseen data in the test set, the overall decision-making principles of the model do not seem consistent with that of the training set. As seen in Figure 5.15, the attributions of the duration variable are now also seemingly random. Moreover, the distinct separation in the load variable is also gone, with some scattered areas of high load values being present for lower BDRY values. The only value with a clear principle is the traffic variable, which over the period, has attributed higher values of the BDI to increased traffic between ports.

Looking at Figure 5.16, none of the same port relations found at the top in the training set are present at the top for the test set. This shows that the contributions to the BDRY do not lie in some larger contributing features, but is spread out over many.



(a) Duration.

(b) Speed.

(c) Load.

(d) Traffic.

Figure 5.15: Average contribution of shipping variables in the look-back periods across the Breakwave Dry Bulk Shipping ETF test set. Orange colors indicate higher feature values and blue colors indicate lower feature values.

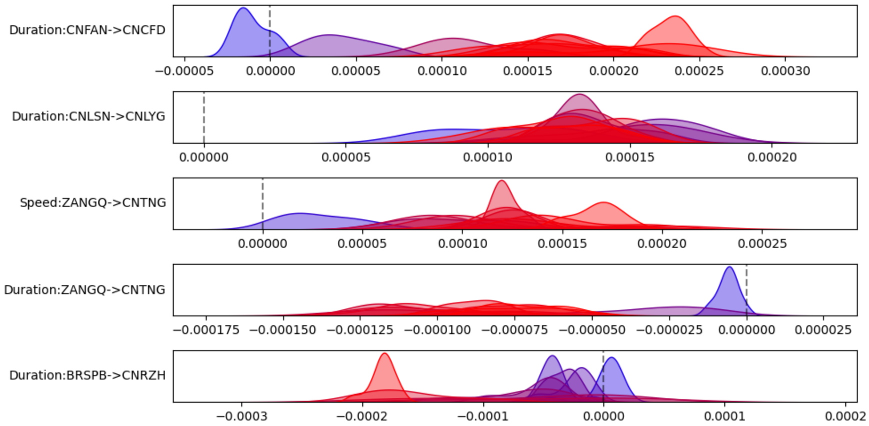


Figure 5.16: The five most impactful features of the Breakwave Dry Bulk Shipping ETF test set according to the best model.

5.3.3 Experiment 3 - Golden Ocean Group Ltd.

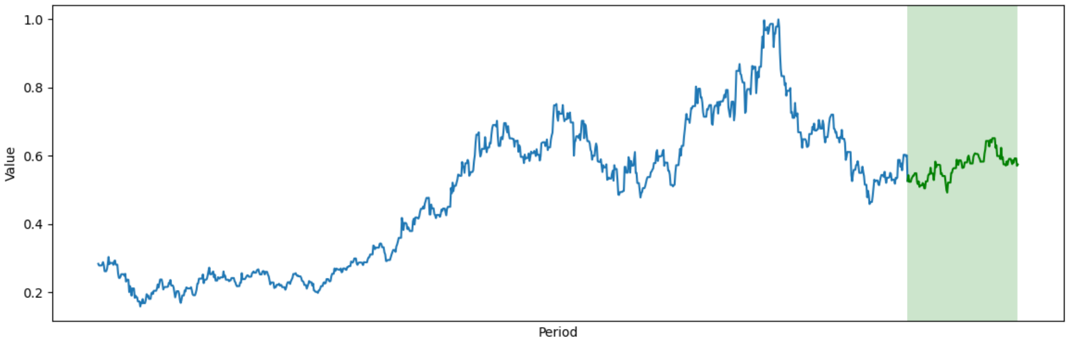


Figure 5.17: The normalized time series data for Golden Ocean Group Ltd., with the test set marked in green.

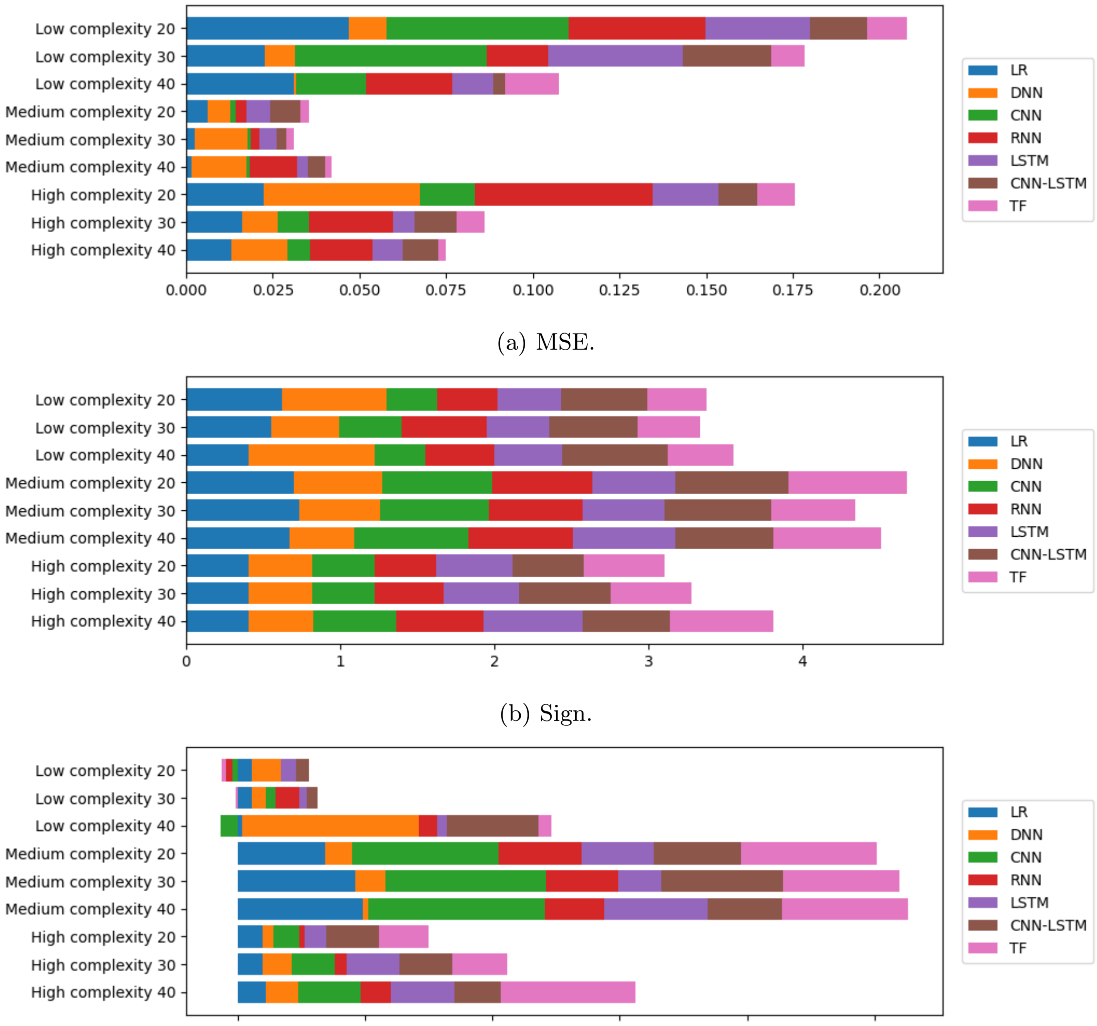
This experiment presented an opportunity to explore the predictive potential of AIS in a real-world corporate context, which introduces more complex influences and dynamics compared to the largely sector-driven factors in the BDI and the speculation-influenced dynamics of the BDRY. The subject of this experiment was the stock price of Golden Ocean Group Ltd. (GOGL), a prominent global entity in the dry bulk shipping industry. Unlike the BDI and BDRY, GOGL's stock price is influenced by an array of factors that extend beyond global maritime trade patterns and market speculations. The financial stability, strategic choices, and operational efficacy of the company play significant roles, adding further complexities and external determinants which pose a challenge to the use of AIS exclusively. Nevertheless, considering that GOGL's revenue and evaluation are strongly tied to their fleet's performance, it is an interesting proposition that there might be certain correlations, particularly given the promising outcomes of the previous two experiments.

The Golden Ocean Group operates a sizable fleet of Capesize, Panamax, and Supramax vessels, mirroring the composition of the BDI and BDRY. This homogeneity between the instruments facilitates an insightful comparison with the outcomes of previous experiments and allows the application of the same AIS data employed in the earlier experiments.

Figure 5.17 displays the normalized time series data for the Golden Ocean Group Ltd stock price, spanning from the start of 2020 to the end of March 2023. The test set displays a rather slow upwards trend overall, hovering at around 50% of the peak values observed in the training set, scattered with some erratic fluctuations. The test set does also not include the distinctive "U-shaped" trajectory found for the previous instruments. The overall time series is vastly different than that of the BDI and BDRY - although it shares the overall slow rise at the start of the period and the distinct drop at the end of the period.

GOGL Model Performances

A comprehensive comparison of the various data configurations and models across the performance metrics is presented in Figure 5.18, and exact numbers can be found in Appendix H.3. As the test set consists of a relatively slow and stable increase in value with scattered fluctuations, the Sign and CCC metrics are of reduced relevance given the rather minor changes they would measure overall.



(a) MSE.

(b) Sign.

(c) CCC.

Figure 5.18: Comparison of data configuration and models for Golden Ocean Group Ltd. across the performance metrics.

The medium complexity configuration, regardless of the look-back period, seems to form a better foundation for the models, with all models seeing better performance across all metrics for the medium complexity configurations. Among the look-back periods examined, a period of 40 days appeared to yield the best overall scores for the CCC and Sign metrics, while a period of 30 days produced the best MSE scores. Moreover, the linear regression baseline model demonstrated competitive performance relative to the other, more advanced models, seeing an MSE of 0.0016, a CCC of 0.4913, and a Sign of 0.6750. ARIMA once again displayed rather poor results, yielding a CCC of 0.0050 and a Sign of 0.4916.

The absolute best result was yet again achieved by the CNN model, on the medium complexity dataset with 40 days for the look-back period, achieving an MSE of **0.0009**, a CCC of **0.6960**, and a Sign of **0.7417**. The training progression of the CNN model is presented in Figure 5.19, showing an initial sharp reduction in the training loss. Interestingly. the validation loss had a slower and more erratic descent until it reached the same levels as that of the training loss at around the 200th epoch, before becoming more unstable. This could be a sign of the model overfitting to the data, but at the same time, the data used for validation is different from that of the training. That being said, there was a period at the same levels seen in the test set, which could indicate that the model overfits this section. Either that or the AIS data is similar for these two periods, which could make sense.

As a parenthetical note, it is acknowledged that the DNN model demonstrated substantial performance when tested on the low-complexity 40 configuration. However, this result stands in stark contrast with the previously observed performance of the DNN model, its performance on other data configurations within this specific experiment, and its performance on identical configurations utilizing 20 and 30 days for the look-back period. Consequently, the outcome derived from the application of the DNN on the low-complexity 40 configuration is dismissed as an outlier and not given further consideration in this analysis. Instead, the aforementioned CNN model will be used.

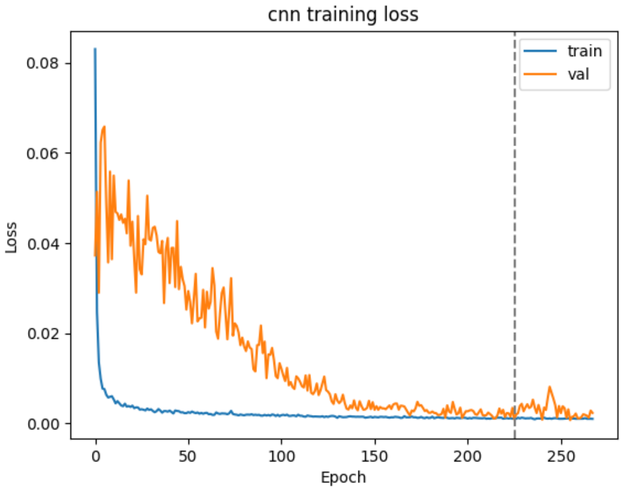
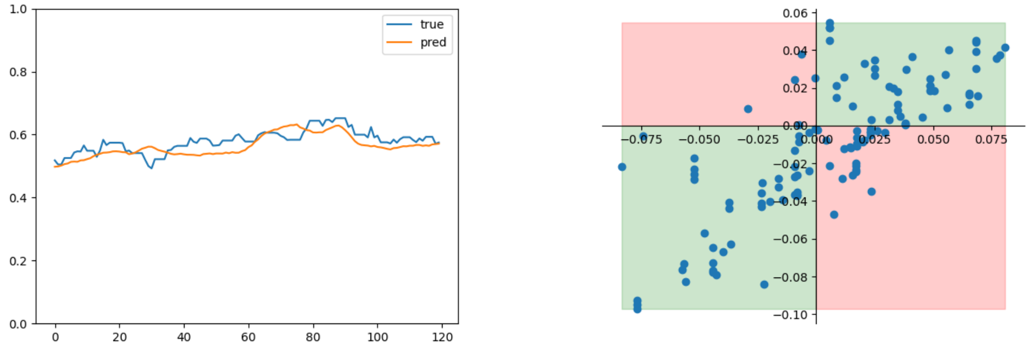


Figure 5.19: The training progression of the CNN model on the *medium complexity 40* configuration for Golden Ocean Group Ltd., with the early stopping marked as a stapled line.



(a) The predicted vs. actual absolute values.

(b) The predicted vs. actual 14-day changes.

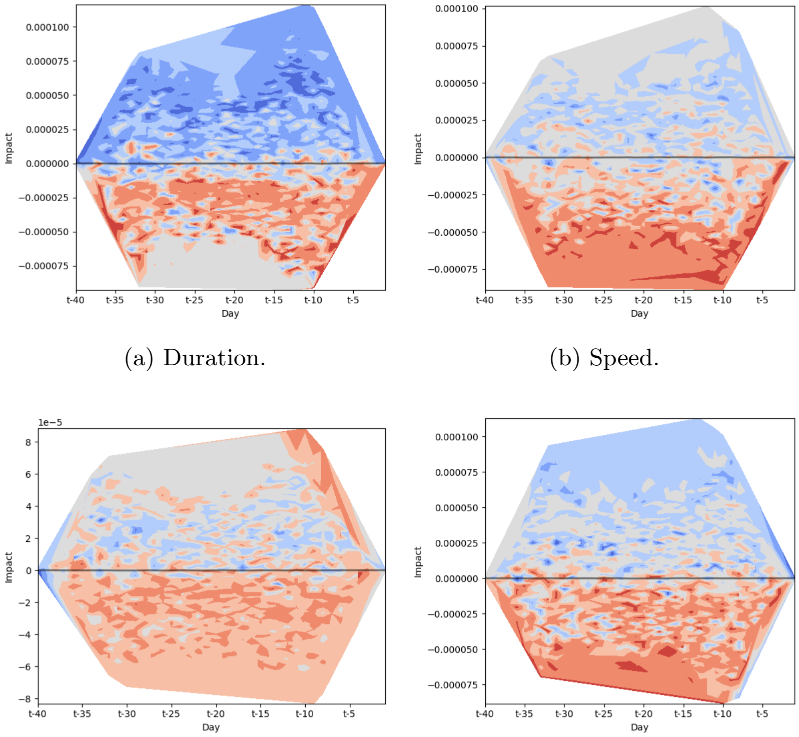
Figure 5.20: The predictions made by the CNN model on the Golden Ocean Group Ltd. test set using the *medium complexity 40* configuration, contrasting them against the truth.

GOGL Model Explanations (Training Set)

Figure 5.21 presents the average impact of each shipping variable during the look-back periods across the entire GOGL training set for the medium-complexity 40 CNN model.

Similar decision principles for the duration variable are seen for the GOGL instrument as for the BDI and BDRY, showing a distinct separation; longer travel times indicate a lower value for the GOGL instrument, and vice versa for shorter travel times. Strangely, the inverse observation is made for the speed variable, where the model attributes a higher instrument value to lower speeds, contradicting the inherent relationship between duration and speed. That being said, the model attributes more importance to duration, as presented in Appendix J.3. The attributions for the load variable are mostly consistent with those from earlier experiments, where higher average load factors are attributed to a lower instrument value. The figure also indicates some instances where high load factors over the past 10 days have been attributed to a higher GOGL value. The traffic variables have a much more pronounced boundary for the GOGL instrument, indicating that the model attributes higher values of the stock to lower volumes of traffic, and vice versa.

Figure 5.22 shows the five most important features across the training set. An extended plot of the top features can be found in Appendix J.3, together with plots for each shipping variable. The majority of the top features are comprised of the duration variable, accounting for three out of the five features. The most impactful feature, the duration between Hay Point, Australia (AUHPT) and Jingtang, China (CHTGS), adheres to established principles and the overall decision principles of the models (as depicted in Figure 5.21). However, an inversion is found for the two other duration-based features, although the distribution of contradicting occurrences seemingly being low, which could suggest the possibility of specific and exceptional circumstances. The top traffic-based feature aligns with the model's overarching principle of higher instrument values being associated with lower traffic, which is also true for the top load feature.



(a) Duration.

(b) Speed.

(c) Load.

(d) Traffic.

Figure 5.21: Average contribution of shipping variables in the look-back periods across the Golden Ocean Group Ltd. training set. Orange colors indicate higher feature values and blue colors indicate lower feature values.

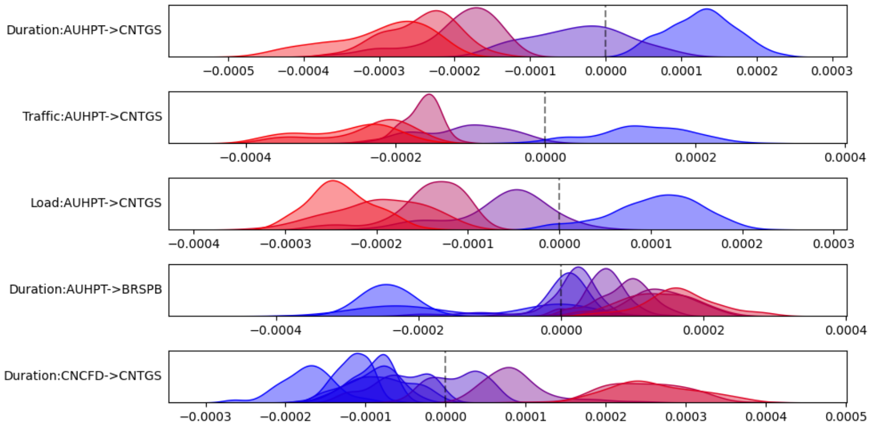
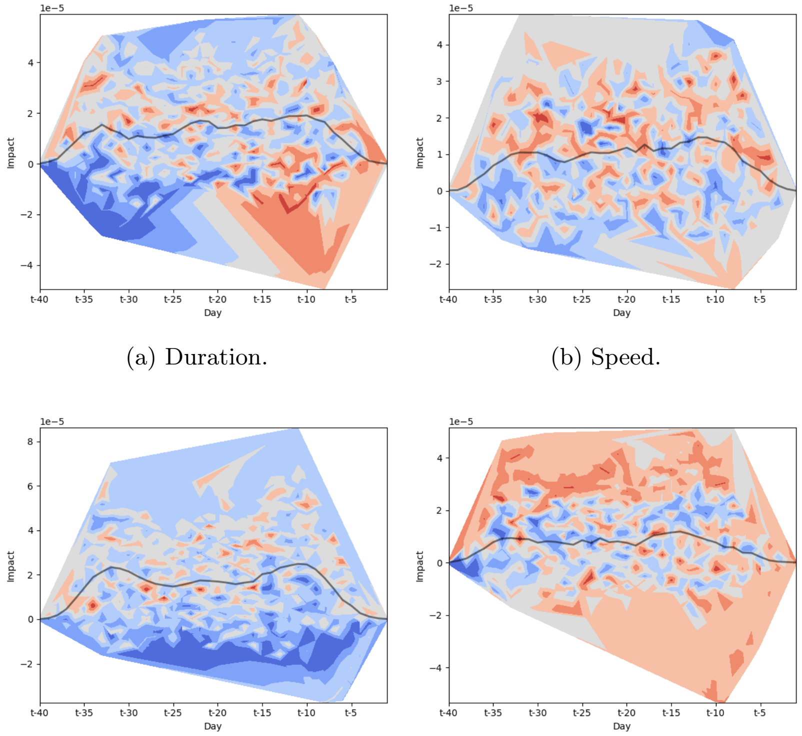


Figure 5.22: The five most impactful features of the Golden Ocean Group Ltd. training set according to the best model.

Model Explanations (Test Set)

The characteristic decision principles established for the training set do not carry over to the test set. The overall attributions seem highly stochastic, showing similar apparent random patterns as was seen for the BDRY. Figure 5.24 shows the five most important contributing features for the test set. As for the high-complexity configuration used with BDRY, it appears that the model does not weigh one particular port relation more than others, as the distribution of port relations for the test set is vastly different than that of the training set. Moreover, the figure reveals weird behavior from the SHAP values, as several of the features show empty contribution plots. It is unknown if these errors are a result of weird SHAP values when aggregated or the code that produced the plot.



(a) Duration.

(b) Speed.

(c) Load.

(d) Traffic.

Figure 5.23: Average contribution of shipping variables in the look-back periods across the Golden Ocean Group Ltd. training set. Orange colors indicate higher feature values and blue colors indicate lower feature values.

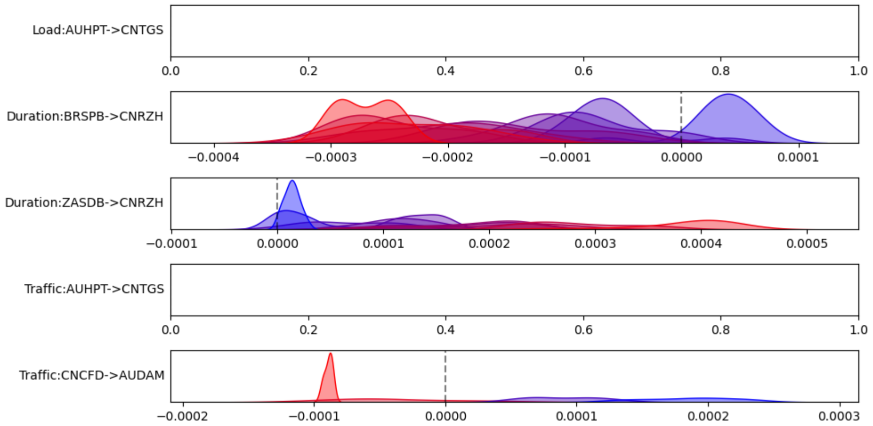


Figure 5.24: The five most impactful features of the Golden Ocean Group Ltd. test set according to the best model.

5.3.4 Experiment 4 - Frontline Ltd.

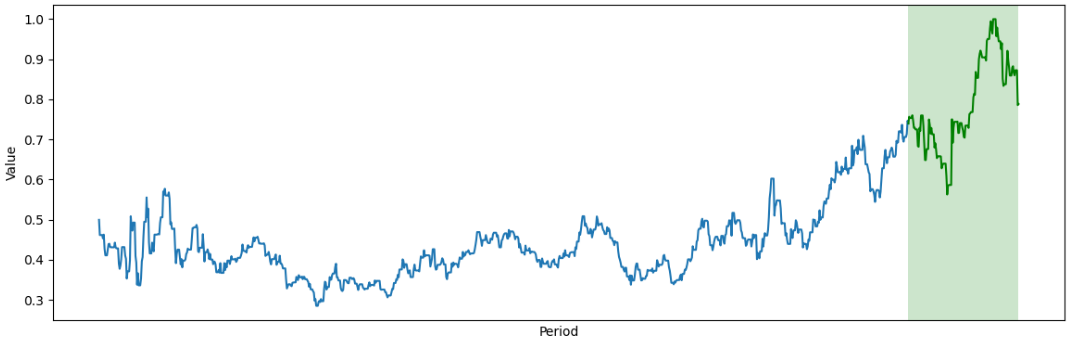


Figure 5.25: The normalized time series data for Frontline Ltd., with the test set marked in green.

Given the demonstrated efficacy of models to represent instruments associated with the dry bulk shipping industry, even under speculative and external influencing conditions, an experiment was conducted to evaluate the viability of AIS within the Tanker segment. Obtaining objectively biased instrumental data emerged as a challenge in this context, with the publicly available instruments being considerably shorter than the period for which we had AIS data. Consequently, the study focused on the stock price of Frontline Ltd. (FRO), an industry leader in the international maritime transportation of crude oil, anticipating that the modeling capabilities previously exhibited for Golden Ocean Ltd. could translate into the tanker market context.

Frontline operates among the largest vessels in the crude oil market. However, due to the limited availability of AIS data for the Tanker segment and the fidelity of the features, only low and medium-complexity datasets could be generated. This limitation was a result of the numerous 'NaN' values in the high-complexity dataset, likely attributable to the normalization of certain relationships lacking data. Therefore, the dataset includes all Tanker vessels exceeding 10,000 DWT, yielding low and medium-complexity data configurations.

Figure 5.25 shows the value of the FRO as employed in the thesis from the start of 2020 to the end of March 2023, with the green box indicating the period used for testing and validation. The test set captures the peak value of the dataset, commencing with an initial downward trend, followed by a steep ascent to the highest plateau, after which a drop to approximately 80% of this plateau concludes the test set.

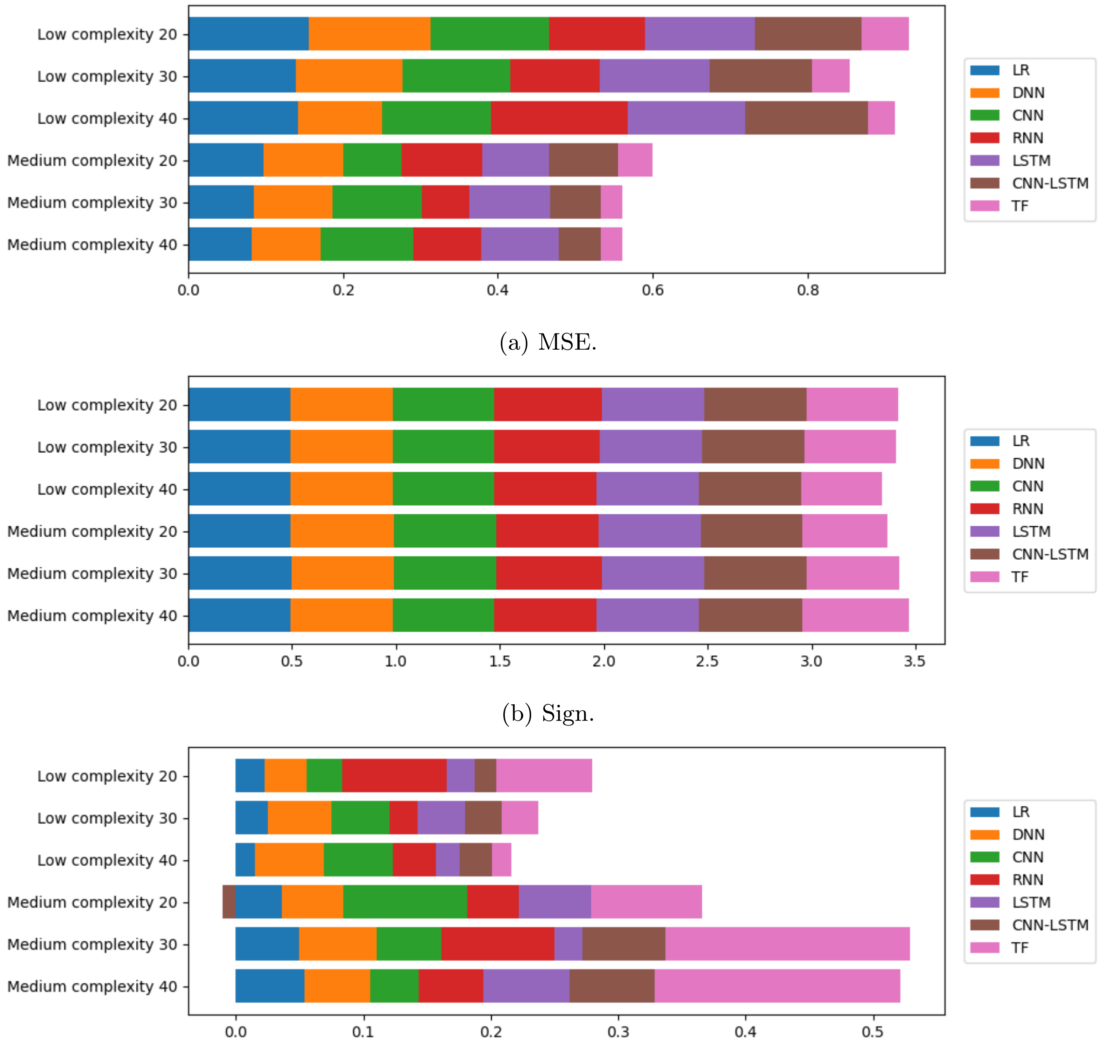
FRO Model Performances

A comprehensive comparison of the various data configurations and models across the performance metrics is presented in Figure 5.26, and exact numbers can be found in Appendix H.4. The CCC and Sign metrics are of high relevance, given the substantial motions in the test set.

In general, all models performed poorly. The medium complexity configurations saw better MSE and CCC scores, with Sign staying mostly the same, attributed to the that the majority of models produced estimates below the truth. This is why the Transformer model performed better; it produced estimates at approximately the same magnitude as the instrument's value.

However, although the Transformer saw the best results, it does not appear to have sufficiently modeled the FRO.

The ARIMA baseline also performed poorly, with an MSE of 0.0078, CCC of -0.0018, and a Sign of 0.4833. In contrast, the best Transformer model achieved an MSE of **0.0274**, a CCC of **0.1935**, and a Sign of **0.5083**. As a result, no further analysis was conducted for the FRO, given the poor performance exhibited by the models.



(a) MSE.

(b) Sign.

(c) CCC.

Figure 5.26: Comparison of data configuration and models for Frontline Ltd. across the performance metrics.

Chapter 6

Evaluation and Conclusion

6.1 Evaluation

The use of explainable AI systems and AIS data to forecast and explain financial instruments in the maritime industry yielded meaningful insights. Four experiments were conducted in total: three for the dry bulk segment and one for the tanker segment. The dry bulk experiments focused on forecasting three distinct financial instruments tied to the performance of the dry bulk fleet, each with varying levels of external subjective and speculative influence. Remarkably, the models demonstrated outstanding proficiency in predicting the financial instruments of the dry bulk market for new unseen data. Across all dry bulk experiments, the best models consistently surpassed the baselines. These results highlight the promising potential of AIS data in forecasting financial instruments associated with the dry bulk segment. Nevertheless, the results differed for the tanker segment, where AIS data proved inadequate for even approximate forecasting of the Frontline Ltd. stock, and hence, explanations for the Tanker segment were not considered.

To maintain the scope of the experiments within feasible boundaries for this thesis, the interpretations were confined to the best-performing data configurations and models. For the Baltic Dry Index, the best model expressed decision-making largely consistent with established knowledge within the field of maritime trade analysis. However, some decision-making nuances diverged from general principles, many of which could be explained through speculation by domain experts. This introduces an element of ambiguity to the explanations, where the inverse of an explanation could also be logically valid, which will be discussed more in Section 6.2.3. However, it is completely possible that these inconsistencies could be attributed to spurious correlations or model inaccuracies, but this is hard to gauge given this ambiguity. The incorporation of the Breakwave Dry Bulk Shipping ETF and the Golden Ocean Group Ltd. stock price introduced additional stochastic variability to the models. Despite the models' surprising proficiency in forecasting these instruments, certain decision-making principles did not convincingly align with the established knowledge. However, it is important to note that the established knowledge primarily pertains to the market as a whole and not these specific instruments. Moreover, it appears that the increasing complexity of features yielded increasingly complex explanations.

The CNN model demonstrated superior performance across all experiments, outperforming the long-standing recurrent neural networks and the recently dominant Transformer model. Nonetheless, the study's scope did not extensively include rigorous model optimization. Consequently, the comparison of the various models merit a cautious interpretation, considering that the employed parameters and architectures were chosen based on their anticipated adequacy in performance rather than through comprehensive optimization methods.

The feature composition combining shipping variables and port relations provided an explanatory medium that was highly comprehensible to domain experts through SHAP feature attribution. However, making them feasible to understand and visually representable within the context of a master's thesis necessitated them to be averaged across temporal dimensions, over either the look-back periods or the datasets as a whole. Given the various influential historical events in the dataset, this presumably led to a notable loss of important temporal information and made some aggregated explanations complex; despite the simplicity of the explanations, their interpretation remained challenging.

As a result of ambiguous explanations and the time limitation of both the thesis and available domain experts, the thesis regrettably did not conclude any novel insights into the interplay between the world fleet's behavior and the maritime trade market. However, it did demonstrate that the models, through decision-making principles contesting existing knowledge, can adeptly predict unseen data, highlighting the potential existence of yet undiscovered patterns.

6.2 Discussion

An overwhelming amount of interesting topics, achievements, and limitations for discussion arose during the research project, which will be addressed in this section.

6.2.1 The Forecasting-Power of the AIS

Although many of the explanations deviated from established knowledge, the models expressed an unexpectedly competent ability to model the various dry bulk instruments using the AIS exclusively, despite the larger influence of non-AIS factors present in the stock prices and ETFs. When first formulating the research questions, the models were deemed successful if they at all produced proximate estimates. However, the outcomes surpassed these initial expectations for all dry bulk-related instruments, leading to contemplation regarding potential information leakage between the training and test sets, or an inadvertent inclusion of the instrument in the features. After examination, these assumptions were dismissed.

An alternative explanation for the observed results could be that the models capitalized on similar patterns within the training and test sets; an overfitting situation within the training set might have extended to the test set, and considering the scarce number of data samples, this hypothesis could be plausible. Moreover, considering that the test sets only included 120 days, the test sets might not have contained sufficient diversity to fully challenge the models on new data that is not also found in the training set. However, it is important to note that the BDI model demonstrated clear indications of starting to overfit when early stopping, and in the case of the BDRY instrument, the model also stopped before reaching a stage of observable overfitting. Additionally, these models demonstrated an ability to identify broader decision-making principles drawn from established knowledge. This suggests a possibility that they have achieved the necessary generalization to predict the test sets, albeit the scarcity of samples instills a certain degree of skepticism.

As for the Frontline Ltd. stock price, Maritime Optima domain experts agree that its valuation can be more closely tied to oil prices than to vessel movements. Additionally, the considerably lower representation of AIS data for these vessels seemed inadequately integrated into the feature structure of the models. Consequently, this study suggests that AIS data pertaining to tanker vessels alone is insufficient to model such instruments efficiently with the proposed feature format.

Additionally, most models were able to beat the ARIMA model in finding 14-day changes in the instruments. Although the ARIMA model did produce a better MSE metric, this outcome is primarily attributable to the model's direct access to the value in question, and even a no-change baseline would yield a comparable MSE score. It is entirely possible that the ARIMA model would be able to produce better results by using more than the three last values of the instrument and different configurations. However, the ARIMA model employed the parameters as found in another research study, with the comparisons against ARIMA serving more as exploratory tests than serious comparisons. It is nonetheless noteworthy that AIS-based models outshined ARIMA models in terms of 14-day changes.

6.2.2 Data Limitations and Feature Engineering

The data foundation facilitated billions of AIS type 1-3 position reports and several hundred million AIS type 5 static reports. However, the temporal scope of the data was limited, spanning from the beginning of 2020 until the execution of the experiments. Consequently, this temporal limitation resulted in a dataset constituting 1161 samples, given that the AIS data was consolidated into a daily format. This significant reduction profoundly impacted the available choices for look-back periods and the length of the test set, establishing a challenging balance between sufficient sample sizes for training and testing. Additionally, the features turned out to be highly complex, and the ratio between feature sizes and the number of samples was quite skewed. Moreover, the period for which the data spanned was marked with highly influential and rare global events, and the inclusion of data from earlier periods might have facilitated a more comprehensive understanding of the subjects under typical circumstances.

Another critical aspect is the accuracy of the arrival detection system from Maritime Optima, which serves as the basis for voyage abstraction. This problem is challenging given the inherent limitations of the AIS, and the detection system, while functionally proficient, is not without shortcomings. Therefore, some voyages might skip ports due to arrivals not being detected, introducing erroneous data, particularly much longer durations. While this could be mitigated through further pre-processing, the data from the voyages were used as they were.

In hindsight, using a draft-based load factor as a feature is fundamentally flawed for its semantic meaning in this study. Initially, it was posited that ships not carrying cargo to their maximum load capacity could indicate competition for less desirable cargo. However, this assumption overlooked that different types of cargo have varying densities. A vessel fully loaded with dense materials like iron will have a high draft and, thus, a higher load. Conversely, the same vessel loaded with less dense materials, such as coal, may reach full load at a lower draft due to the vessel's volume capacity. Furthermore, after conducting the experiments, it became apparent that a low load factor does not necessarily imply an empty vessel. Each feature was normalized with respect to its own values over the entire dataset, thereby signifying that a load value of 0.9 would still be normalized to 0 if it represented the lowest load value for a specific port relation. Although the load plots still adhere to the semantic understanding that blue values correspond to lower load values, it does not necessarily signify an empty vessel. Consequently, the current format of the plots complicates their interpretation. Another limitation is that some vessels perform parcel freights, meaning that they might load or unload the same cargo in several ports, between which the vessel would travel with half the load.

Duration and speed, while inherently intertwined, showed a high degree of independence for the various experiments, especially for experiment 3, looking into the Golden Ocean Group Ltd. stock. That being said, the duration variable is likely more accurate, as it is detected through the positions and timestamps rather than the speed value as indicated by vessels. This could be the reason why the model favored duration so much in contrast to the speed variables.

6.2.3 Model Interpretation and Explanation Ambiguity

Overall, the domain experts from Maritime Optima sufficiently comprehended the explanations. One minor challenge with the explanations themselves was that port LOCODEs were not as recognizable as the name of ports. Regardless, the LOCODEs provide a mechanism for concise explanations, and additional information about each port is readily available through online searches. Moreover. the experts were definitely more interested in the violin plots per port relation per shipping variable rather than the aggregated heatmaps, as they provided a more nuanced form of interpretation. The explanations were still complicated due to the aggregation over the entire dataset, and it was discussed that changing the magnitude of the violins for an axis representing time in the dataset would be better.

Despite the clarity of the explanations, their interpretation posed a challenge primarily due to inherent ambiguities. This ambiguity arises from the fact that in the intricate sphere of maritime trading, the inverse of an explanation could also potentially hold true. The speed of vessels between certain ports, for instance, could indicate both high and low market conditions, depending on the context. In situations of high demand, high speeds and shorter travel times could signify an elevated market, as it allows vessels to conduct as many trades and carry as much cargo as possible to take advantage of heightened costs. Yet, as suggested by the domain experts, there are circumstances, particularly concerning larger ports, where vessels might slow down due to port congestion, i.e., the number of idle vessels outside a port waiting to be let in. This is reasonable, as port congestion may be elevated during times of high demand, and, thus, a higher volume of vessels. Moreover, given that vessels may have to wait, regardless of arriving early, they often slow down to conserve fuel. A case in point is the varying market values observed in relation to travel times to Port Hedland and Hay Point, both in Australia. It was suggested that Port Hedland, being the world's largest port, experiences more congestion than Hay Point, thus causing longer travel times due.

As for traffic, one would expect higher traffic to indicate a higher market, but this was not seen in the results. More often than not, the model attributed higher markets to lower volumes of traffic. Again, the domain experts speculated about several situations in which this could be the case. However, without also knowing the supply side, i.e., what and how much cargo is available at each port, no concrete conclusions can be drawn.

The more surprising results were the models' constant attribution of lower load factors to higher instrument values. As already discussed in Section 6.2.2, the load factor was somewhat flawed, but it is interesting that it maintains the same distinct overall principle for all experiments. Again, this goes against the overall principle that vessels are usually fully loaded in high markets. One could argue that it is because of spurious correlations or that the load factor values somehow have been inverted. In any case, it is an overwhelmingly distinct and consistent observation. Also, without any information about the supply side and cargo information, this remains inconclusive.

6.3 Contributions

The work conducted in this thesis contributes to the research field of machine learning and explainable AI, specifically in the domain of high-dimensionality multivariable sequence forecasting, and to the analytical field of maritime trade, providing extensive material for further analysis.

The thesis has compiled several state-of-the-art research studies pertaining to the goals of this thesis, presenting work for both advanced sequence forecasting models and applicable XAI methods. Furthermore, the literature review has presented current approaches to forecasting various financial instruments on the basis of AIS data.

Moreover, the thesis has empirically evaluated and compared several state-of-the-art sequential models using high-dimensional AIS-based tabular data, yielding concrete performance results using several performance metrics measuring both how close the predictions are to the truth, but also how well the models are able to model 14-day changes of various financial instruments. This, in turn, has contributed to findings that AIS data has been effective in modeling several financial instruments in the maritime industry on data between 2020 and 2023.

The thesis proposes an explanation format that is easily understandable for domain experts through various aggregations of SHAP values. It has also highlighted several of the limitations arising from the interpretation of these explanations.

Furthermore, the thesis presents findings and results indicating that the best models can exhibit decision-making principles consistent with that of established knowledge for various financial instruments using AIS data. Additionally, the model shows some complex decision-making that contests established knowledge when fed a higher complexity of features. However, the complexity of interpreting the more intricate and ambiguous nature of the explanations poses a considerable challenge in ascertaining the novelty and accuracy of these deviations. Nonetheless, the thesis contributes an extensive collection of explanations to the research community, encompassing explanations derived from the best-performing models for each financial instrument where satisfactory modeling capabilities were identified.

Lastly, this thesis augments the body of research that leverages AI in deriving new knowledge in respective domains. The developed models have demonstrated decision-making capabilities that both rival and follow established knowledge within the realm of maritime trade analysis. These findings suggest the possibility of latent knowledge inherent in these models, as evidenced by their impressive forecasting outcomes, hence pushing the boundaries of our existing understanding. While this thesis did not provide any concrete new knowledge, it establishes the groundwork for further exploration in both the maritime domain and domains characterized by lesser ambiguity.

6.4 Future Work

This thesis serves as a robust foundation, setting the stage for several potential extensions and explorations. This section aims to illustrate and detail the various paths for the extension and adaptation of this work, outlining prospective research directions that could enrich the field and further contribute to our understanding. While several possible extensions are proposed in the subsequent subsections, it is also hoped that this thesis will serve as a catalyst for further innovation and inquiry.

6.4.1 Improve Model Performance and Data Foundation

Despite the apparent competency of the presented models in handling various instruments, there's always potential for further improvement. An extended dataset encompassing a longer historical period is likely to enhance the performance of the models as well as allow for a more rigorous evaluation. Additionally, an extended dataset allows for longer look-back periods, which could benefit both the models' performances as well as explanations. Integrating additional relevant non-AIS data is thought to improve the model. This includes the instrument's value itself, bunker pricing information, anchorage data to better model congestion, various macroeconomic variables, and cargo data from multiple ports to more adeptly model supply and help mitigate the problem of the load variable, as discussed in Section 6.2.2. Moreover, consideration of different vessel sizes, each exhibiting unique behavioral patterns, could yield more nuanced insights. A reassessment of the selection criteria for ports in port relations, possibly shifting focus from visit frequency to trade volume using draft values, could yield different, possibly more informative results. Furthermore, the models employed in this thesis are relatively simple, and additional effort can go into the various models to improve their architecture. For instance, the consistently best-performing CNN model could be improved by testing out residual connections, dilution, larger kernel sizes, more filters, etc.

6.4.2 Extend Beyond Financial Instruments

While this thesis centers around financial instruments, there is ample room to extend the scope of predicted variables, as well as additional financial instruments. For instance, the absence of an accessible objective index for the Tanker segment in this study suggests an area that requires further exploration to achieve a more definitive result on the efficacy of AIS in modeling the Tanker market. Furthermore, the forecasting of freight rates presents a natural and relevant expansion. Moving beyond the scope of financial instruments, traffic flow or congestion could serve as promising non-financial variables to forecast.

6.4.3 Explore Additional Explanation Mechanisms

A natural recommendation for further research is the exploration of additional XAI methodologies for high-dimensional time series data, as this study has leaned on SHAP and feature attribution. Moreover, the implementation of TimeSHAP encountered obstacles due to compatibility issues with the library dependencies necessary for the models. This urges exploration of the TimeSHAP library, resolving the aforementioned compatibility challenges, to derive potentially additional insightful explanations.

In addition, explanations for each port relation per shipping variable, as presented in Appendix J, would benefit from incorporating the temporal dimension of the entire datasets. A proposal is the construction of a two-dimensional heatmap for each port relation, with one dimension signifying the SHAP value and the other the time in the dataset. This approach offers an alternative to compressing these dimensions into a single distribution for the entire dataset. as seen in the violin plots in this thesis. A more nuanced exploration of this sort could help clarify explanations and potentially account for certain anomalous periods in the dataset.

Moreover, expanding upon the exploration of XAI methodologies for high-dimensional time series data, further research could delve into model-specific explanation mechanisms. While the study utilized feature attribution through SHAP values for global explanations, the different models have unique architectural characteristics that would be better explained and interpreted using techniques specific to them, such as the attention mechanism in the transformer model.

6.4.4 Improve XAI Methods for Sequence Models

There is a compelling opportunity for the advancement and enrichment of XAI methodologies specifically tailored to sequence models. A central limitation identified in this thesis lies in the aggregation and averaging of temporal dimensions for global explanations. This practice likely suffers data loss and compromises the explanatory capacity of the models, rendering the process less insightful and beneficial. Future work should innovate techniques that maintain temporal granularity or adopt advanced temporal aggregation methods to better capture temporal patterns and dependencies.

6.4.5 Derive Robust Qualitative Explanation Metrics

The variance and importance metrics implemented in this thesis, while quantitative, fall short of providing a holistic measure of explanation quality in the context of AIS-based high-dimensional time series data. While they measure the range and consistency of models' explanations, these metrics fail to capture the qualitative aspects that ultimately determine the utility.

6.4.6 In-depth Analysis of Model Results

The ambiguous and dual nature of the explanations calls for a deeper examination. A deeper and more nuanced study of the port-specific explanations for the various shipping variables could help derive more insight. The suggested additional explanations from the previous subsection could be instrumental in this pursuit.

6.4.7 Interdisciplinary Application of Methodologies

This research stresses the transformative potential of explainable AI in deriving novel knowledge. The effective modeling of financial instruments in the maritime trade domain is a testament to this potential. and while the explanations derived in this research might be multifaceted, there might be opportunities for more definite answers in other domains. This interdisciplinary application of AI and XAI not only extends the reach of these powerful tools but also enriches scientific understanding in multiple fields. Therefore, it is highly recommended that the application of AI and XAI be expanded to further domains, driving a new era of knowledge discovery.

Bibliography

Adland, R., Jia, H., and Strandenes, S. P. (2018). The determinants of vessel capacity utilization: The case of brazilian iron ore exports. *Transportation Research Part A: Policy and Practice*, 110:191–201.

Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. (2017). Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*.

Århus, G. H. and Salen, S. R. (2018). Predicting shipping freight rate movements using recurrent neural networks and ais data-on the tanker route between the arabian gulf and singapore. Master's thesis, NTNU.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. (2010). How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831.

Bai, S. (2022). Boston house price prediction: machine learning. In *2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)*, pages 1678–1684. IEEE.

Bai, S., Kolter, J. Z., and Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Bellingham, J. G., Zhang, Y., Kerwin, J. E., Erikson, J., Hobson, B., Kieft, B., Godin, M., McEwen, R., Hoover, T., Paul, J., et al. (2010). Efficient propulsion for the tethys long-range autonomous underwater vehicle. In *2010 IEEE/OES Autonomous Underwater Vehicles*, pages 1–7. IEEE.

Bento, J. a., Saleiro, P., Cruz, A. F., Figueiredo, M. A., and Bizarro, P. (2021). Timeshap: Explaining recurrent models through sequence perturbations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 2565–2573, New York, NY, USA. Association for Computing Machinery.

Boshoff, W. H. and Fourie, J. (2010). The significance of the cape trade route to economic activity in the cape colony: a medium-term business cycle analysis. *European Review of Economic History*, 14(3):469–503.

Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Bükk, C. and Hoang, L. (2022). Explainable ai for the transformer model used on chemical language.

Chen, C., Liu, Y., Chen, L., and Zhang, C. (2022). Bidirectional spatial-temporal adaptive transformer for urban traffic flow forecasting. *IEEE Transactions on Neural Networks and Learning Systems.*

Chen, X., Liu, Y., Achuthan, K., and Zhang, X. (2020). A ship movement classification based on automatic identification system (ais) data using convolutional neural network. *Ocean Engineering*, 218:108182.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167.

Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. (2017). Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.*

Dikshit, A. and Pradhan, B. (2021). Interpretable and explainable ai (xai) model for spatial drought prediction. *Science of the Total Environment*, 801:149797.

Dos Santos, C. and Zadrozny, B. (2014). Learning character-level representations for part-of-speech tagging. In *International Conference on Machine Learning*, pages 1818–1826. PMLR.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608.*

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.

Emmens, T., Amrit, C., Abdi, A., and Ghosh, M. (2021). The promises and perils of automatic identification system data. *Expert Systems with Applications*, 178:114975.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2009). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–308.

García, M. V. and Aznarte, J. L. (2020). Shapley additive explanations for no2 forecasting. *Ecological Informatics*, 56:101039.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR.

Gerson, A. (2023). Stranding of the mega-ship ever given in the suez canal: Causes, consequences, and lessons to be learned. In *The Suez Canal: Past Lessons and Future Challenges*, pages 231–252. Springer.

Ghahramani, Z. (2004). Unsupervised learning. *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, pages 72–112.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning.* MIT press.

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., et al. (2022). A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110.

Harati-Mokhtari, A., Wall, A., Brooks, P., and Wang, J. (2007). Automatic identification system (ais): Data reliability and human error implications. *The Journal of Navigation*, 60(3):373–389.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Hyndman, R. J. and Athanasopoulos, G. (2014). *Forecasting: Principles and Practice*. OTexts, print edition.

International Maritime Organization (1974). *SOLAS: International Convention for the Safety of Life at Sea*. International Maritime Organization London.

Ivanovs, M., Kadikis, R., and Ozols, K. (2021). Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters*, 150:228–234.

Jordan, M. I. (1986). Serial order: A parallel distributed processing approach. In *Advances in psychology*, volume 121, pages 471–495. Elsevier.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589.

Jung, Y.-J., Han, S.-H., and Choi, H.-J. (2021). Explaining cnn and rnn using selective layer-wise relevance propagation. *IEEE Access*, 9:18670–18681.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Kanamoto, K., Wada, Y., and Shibasaki, R. (2019). Predicting a dry bulk freight index by deep learning with global vessel movement data. In *Transdisciplinary Engineering for Complex Socio-technical Systems*, pages 105–114. IOS Press.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kober, J., Bagnell, J. A., and Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274.

Lara-Benítez, P., Gallego-Ledesma, L., Carranza-García, M., and Luna-Romera, J. M. (2021). Evaluation of the transformer architecture for univariate time series forecasting. In *Advances in Artificial Intelligence: 19th Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2020/2021, Málaga, Spain, September 22–24, 2021, Proceedings 19*, pages 106–115. Springer.

Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D. (2017). Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, G., Li, F., Xu, C., and Fang, X. (2022a). A spatial-temporal layer-wise relevance propagation method for improving interpretability and prediction accuracy of lstm building energy prediction. *Energy and Buildings*, 271:112317.

Li, Y., Wang, Y., and Ma, K. (2022b). Integrating transformer and gcn for covid-19 forecasting. *Sustainability*, 14(16):10393.

Lindsay, G. W. (2021). Convolutional neural networks as a model of the visual system: Past, present, and future. *Journal of cognitive neuroscience*, 33(10):2017–2031.

L'Heureux, A., Grolinger, K., and Capretz, M. A. (2022). Transformer-based model for electrical load forecasting. *Energies*, 15(14):4993.

Maanum, M. O. and Selnes, H. P. (2015). Determinants of vessel speed in the vlcc market: Theory vs. practice. Master's thesis.

Makridakis, S., Spiliotis, E., Assimakopoulos, V., Semenoglou, A.-A., Mulder, G., and Nikolopoulos, K. (2022). Statistical, machine learning and deep learning forecasting methods: Comparisons and ways forward. *Journal of the Operational Research Society*, pages 1–20.

Martin Stopford (2008). *Maritime Economics.* Routledge, 3 edition.

Mehta, P., Pandya, S., and Kotecha, K. (2021). Harvesting social media sentiment analysis to enhance stock market prediction using deep learning. *PeerJ Computer Science*, 7:e476.

Millefiori, L. M., Braca, P., Zissis, D., Spiliopoulos, G., Marano, S., Willett, P. K., and Carniel, S. (2021). Covid-19 impact on global maritime mobility. *Scientific reports*, 11(1):1–16.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38.

Molnar, C. (2022). *Interpretable Machine Learning.* 2 edition.

Mordvintsev, A., Olah, C., and Tyka, M. (2015). Inceptionism: Going deeper into neural networks.

Næss, P. A. (2018). *Investigation of multivariate freight rate prediction using machine learning and ais data.* PhD thesis, Master Thesis, Norwegian University of Science and Technology.

Nguyen, D. and Fablet, R. (2021). Traisformer-a generative transformer for ais trajectory prediction. *arXiv preprint arXiv:2109.03958.*

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499.*

Ozyegen, O., Ilic, I., and Cevik, M. (2020). Evaluation of local explanation methods for multivariate time series forecasting. *arXiv preprint arXiv:2009.09092.*

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr.

Pirie, C., Suresh, M., Salimi, P., Palihawadana, C., and Nanayakkara, G. (2022). Explainable weather forecasts through an lstm-cbr twin system. CEUR Workshop Proceedings.

Qu, K., Si, G., Shan, Z., Kong, X., and Yang, X. (2022). Short-term forecasting for multiple wind farms based on transformer model. *Energy Reports*, 8:483–490.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1:81–106.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Reza, S., Ferreira, M. C., Machado, J., and Tavares, J. M. R. (2022). A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks. *Expert Systems with Applications*, 202:117275.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Rožić, T., Naletina, D., and Zając, M. (2022). Volatile freight rates in maritime container industry in times of crises. *Applied Sciences*, 12(17):8452.

Saluja, R., Malhi, A., Knapič, S., Främling, K., and Cavdar, C. (2021). Towards a rigorous evaluation of explainability for multivariate time series. *arXiv preprint arXiv:2104.04075*.

Samek, W. and Müller, K.-R. (2019). Towards explainable artificial intelligence. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 5–22.

Sarantopoulos, F. (2021). Modelling the dry bulk shipping market with the use of recurrent neural networks. Master's thesis, NTUA.

Shapley, L. S. et al. (1953). A value for n-person games.

Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR.

Siami-Namini, S., Tavakoli, N., and Namin, A. S. (2018). A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pages 1394–1401. IEEE.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.

Smagulova, K. and James, A. P. (2019). A survey on lstm memristive neural network architectures and applications. *The European Physical Journal Special Topics*, 228(10):2313–2324.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. (2017). Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Spadon, G., Ferreira, M. D., Soares, A., and Matwin, S. (2022). Unfolding collective ais transmission behavior for vessel movement modeling on irregular timing data using noise-robust neural networks. *arXiv preprint arXiv:2202.13867*.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Syed, M. A. B. and Ahmed, I. (2023). A cnn-lstm architecture for marine vessel track association using automatic identification system (ais) data. *arXiv preprint arXiv:2303.14068*.

Tang, Y., Song, Z., Zhu, Y., Yuan, H., Hou, M., Ji, J., Tang, C., and Li, J. (2022). A survey on machine learning models for financial time series forecasting. *Neurocomputing*, 512:363–380.

Tax, N., Teinemaa, I., and van Zelst, S. J. (2020). An interdisciplinary comparison of sequence modeling methods for next-element prediction. *Software and Systems Modeling*, 19:1345–1365.

Ullah, I., Rios, A., Gala, V., and Mckeever, S. (2021). Explaining deep learning models for tabular data using layer-wise relevance propagation. *Applied Sciences*, 12(1):136.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Vega García, M. (2019). Interpretable forecasts of no2 concentrations through deep shap.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., and Titov, I. (2019). Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. *IEEE transactions on acoustics, speech, and signal processing*, 37(3):328–339.

Wang, D., Zhang, C., Wang, B., Li, B., Wang, Q., Liu, D., Wang, H., Zhou, Y., Shi, L., Lan, F., et al. (2019). Optimized crispr guide rna design for two high-fidelity cas9 variants by deep learning. *Nature communications*, 10(1):4284.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Williams, R. J. and Zipser, D. (1995). Gradient-based learning algorithms for recurrent. *Backpropagation: Theory, architectures, and applications*, 433:17.

Wu, M. and Zhang, Z. (2010). Handwritten digit classification using the mnist data set. *Course project CSE802: Pattern Classification & Analysis*, 366.

Wu, N., Green, B., Ben, X., and O'Banion, S. (2020). Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv preprint arXiv:2001.08317*.

Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.-J., and Xiong, H. (2020). Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*.

Xu, X., Zuo, L., and Huang, Z. (2014). Reinforcement learning algorithms with function approximation: Recent advances and applications. *Information sciences*, 261:1–31.

Yan, J., Mu, L., Wang, L., Ranjan, R., and Zomaya, A. Y. (2020). Temporal convolutional networks for the advance prediction of enso. *Scientific reports*, 10(1):1–15.

Yazir, D., Şahin, B., Yip, T. L., and Tseng, P.-H. (2020). Effects of covid-19 on maritime industry: a review. *International maritime health*, 71(4):253–264.

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., and Eickhoff, C. (2021). A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2114–2124.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Zhang, Y., Liua, J., Lia, Y., Petrosian, O., and Krinkin, K. (2022). Forecasting and xai for applications usage in os.

Zhao, Z., Xia, C., Chi, L., Chang, X., Li, W., Yang, T., and Zomaya, A. Y. (2021). Short-term load forecasting based on the transformer model. *Information*, 12(12):516.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115.

Appendix

A AIS Message Specifications

A.1 Overview of AIS Messages

AIS message specification from the USCG - accessed February 1st, 2023.
`https://www.navcen.uscg.gov/ais-messages`

ID	Name	Description	Priority
1	Position report	Scheduled position report; Class A shipborne mobile equipment	1
2	Position report	Assigned scheduled position report; Class A shipborne mobile equipment	1
3	Position report	Special position report, response to interrogation; Class A shipborne mobile equipment	1
4	Base station report	Position, UTC, date and current slot number of base station	1
5	Static and voyage related data	Scheduled static and voyage related vessel data report, Class A shipborne mobile equipment	4
6	Binary addressed message	Binary data for addressed communication	4
7	Binary acknowledgement	Acknowledgement of received addressed binary data	1
8	Binary broadcast message	Binary data for broadcast communication	4
9	Standard SAR aircraft position report	Position report for airborne stations involved in SAR operations only	1
10	UTC/date inquiry	Request UTC and date	3
11	UTC/date response	Current UTC and date if available	3
12	Addressed safety related message	Safety related data for addressed communication	2
13	Safety related acknowledgement	Acknowledgement of received addressed safety related message	1
14	Safety related broadcast message	Safety related data for broadcast communication	2
15	Interrogation	Request for a specific message type can result in multiple responses from one or several stations	3
16	Assignment mode command	Assignment of a specific report behaviour by competent authority using a Base station	1
17	DGNSS broadcast binary message	DGNSS corrections provided by a base station	2
18	Standard Class B equipment position report	Standard position report for Class B shipborne mobile equipment to be used instead of Messages 1, 2, 3	1
19	Extended Class B equipment position report	No longer required. Extended position report for Class B shipborne mobile equipment; contains additional static information	1
20	Data link management message	Reserve slots for Base station(s)	1
21	Aids-to-navigation report	Position and status report for aids-to-navigation	1
22	Channel management	Management of channels and transceiver modes by a Base station	1
23	Group assignment command	Assignment of a specific report behaviour by competent authority using a Base station to a specific group of mobiles	1
24	Static data report	Additional data assigned to an MMSI Part A: Name Part B: Static Data	4
25	Single slot binary message	Short unscheduled binary data transmission Broadcast or addressed	4
26	Multiple slot binary message with Communications State	Scheduled binary data transmission Broadcast or addressed	4
27	Position report for long range applications	Class A and Class B "SO" shipborne mobile equipment outside base station coverage	1
28-63	Undefined; Reserved for future use	N/A	N/A

A.2 AIS Type 1-3 Position Report Specification

AIS position report specification from the USCG - accessed February 1st, 2023. `https://www.navcen.uscg.gov/ais-class-a-reports`

Parameter	**Description**
Message ID	Identifier for this message 1, 2 or 3
Repeat indicator	Used by the repeater to indicate how many times a message has been repeated. See Section 4.6.1, Annex 2; 0-3; 0 = default; 3 = do not repeat any more.
User ID	MMSI number
Navigational status	0 = under way using engine, 1 = at anchor, 2 = not under command, 3 = restricted maneuverability, 4 = constrained by her draught, 5 = moored, 6 = aground, 7 = engaged in fishing, 8 = under way sailing, 9 = reserved for future amendment of navigational status for ships carrying DG, HS, or MP, or IMO hazard or pollutant category C, high speed craft (HSC), 10 = reserved for future amendment of navigational status for ships carrying dangerous goods (DG), harmful substances (HS) or marine pollutants (MP), or IMO hazard or pollutant category A, wing in ground (WIG); 11 = power-driven vessel towing astern (regional use); 12 = power-driven vessel pushing ahead or towing alongside (regional use); 13 = reserved for future use, 14 = AIS-SART (active), MOB-AIS, EPIRB-AIS 15 = undefined = default (also used by AIS-SART, MOB-AIS and EPIRB-AIS under test)
Rate of turn ROTAIS	0 to +126 = turning right at up to 708 deg per min or higher 0 to -126 = turning left at up to 708 deg per min or higher Values between 0 and 708 deg per min coded by ROTAIS = 4.733 SQRT(ROTsensor) degrees per min where ROTsensor is the Rate of Turn as input by an external Rate of Turn Indicator (TI). ROTAIS is rounded to the nearest integer value. +127 = turning right at more than 5 deg per 30 s (No TI available) -127 = turning left at more than 5 deg per 30 s (No TI available) -128 (80 hex) indicates no turn information available (default). ROT data should not be derived from COG information.
SOG	Speed over ground in 1/10 knot steps (0-102.2 knots) 1 023 = not available, 1 022 = 102.2 knots or higher
Position accuracy	The position accuracy (PA) flag should be determined in accordance with the table below: 1 = high (<= 10 m) 0 = low (> 10 m) 0 = default
Longitude	Longitude in 1/10 000 min (+/-180 deg, East = positive (as per 2's complement), West = negative (as per 2's complement). 181= (6791AC0h) = not available = default)
Latitude	Latitude in 1/10 000 min (+/-90 deg, North = positive (as per 2's complement), South = negative (as per 2's complement). 91deg (3412140h) = not available = default)
COG	Course over ground in 1/10 = (0-3599). 3600 (E10h) = not available = default. 3 601-4 095 should not be used
True heading	Degrees (0-359) (511 indicates not available = default)
Time stamp	UTC second when the report was generated by the electronic position system (EPFS) (0-59, or 60 if time stamp is not available, which should also be the default value, or 61 if positioning system is in manual input mode, or 62 if electronic position fixing system operates in estimated (dead reckoning) mode, or 63 if the positioning system is inoperative)
special maneuvre indicator	0 = not available = default 1 = not engaged in special maneuver 2 = engaged in special maneuver (i.e.: regional passing arrangement on Inland Waterway)
Spare	Not used. Should be set to zero. Reserved for future use.
RAIM-flag	Receiver autonomous integrity monitoring (RAIM) flag of electronic position fixing device; 0 = RAIM not in use = default; 1 = RAIM in use. See Table
Communication state	See Rec. ITU-R M.1371-5 Table 49
Number of bits	

A.3 AIS Type 5 Static Report Specification

AIS static report specification from the USCG - accessed February 1st, 2023.
`https://www.navcen.uscg.gov/ais-class-a-static-voyage-message-5`

Parameter	**Description**
Message ID	Identifier for this Message
Repeat indicator	Used by the repeater to indicate how many times a message has been repeated. Refer to §?4.6.1, Annex 2; 0-3; 0 = default; 3 = do not repeat any more
User ID	MMSI number
AIS version indicator	0 = station compliant with Recommendation ITU-R M.1371-1 1 = station compliant with Recommendation ITU-R M.1371-3 (or later) 2 = station compliant with Recommendation ITU-R M.1371-5 (or later) 3 = station compliant with future editions
IMO number	0 = not available = default – Not applicable to SAR aircraft 0000000001-0000999999 not used 0001000000-0009999999 = valid IMO number; 0010000000-1073741823 = official flag state number.
Call sign	7?=?6 bit ASCII characters, @@@@@@@ = not available = default Craft associated with a parent vessel, should use "A" followed by the last 6 digits of the MMSI of the parent vessel. Examples of these craft include towed vessels, rescue boats, tenders, lifeboats and liferafts.
Name	Maximum 20 characters 6 bit ASCII "@@@@@@@@@@@@@@@@@@@@" = not available = default The Name should be as shown on the station radio license. For SAR aircraft, it should be set to "SAR AIRCRAFT NNNNNNN" where NNNNNNN equals the aircraft registration number.
Type of ship and cargo type	0 = not available or no ship = default 1-99 = as defined below 100-199 = reserved, for regional use 200-255 = reserved, for future use Not applicable to SAR aircraft
Overall dimension/ reference for position	Reference point for reported position. Also indicates the dimension of ship (m) (see below) For SAR aircraft, the use of this field may be decided by the responsible administration. If used it should indicate the maximum dimensions of the craft. As default should A = B = C = D be set to "0"
Type of electronic position fixing device	0 = undefined (default) 1 = GPS 2 = GLONASS 3 = combined GPS/GLONASS 4 = Loran-C 5 = Chayka 6 = integrated navigation system 7 = surveyed 8 = Galileo, 9-14 = not used 15 = internal GNSS
ETA	Estimated time of arrival; MMDDHHMM UTC Bits 19-16: month; 1-12; 0 = not available = default Bits 15-11: day; 1-31; 0 = not available = default Bits 10-6: hour; 0-23; 24 = not available = default Bits 5-0: minute; 0-59; 60 = not available = default For SAR aircraft, the use of this field may be decided by the responsible administration
Maximum present static draught	In 1/10 m, 255 = draught 25.5 m or greater, 0 = not available = default; in accordance with IMO Resolution A.851 Not applicable to SAR aircraft, should be set to 0
Destination	Maximum 20 characters using 6-bit ASCII; @@@@@@@@@@@@@@@@@@@@ = not available For SAR aircraft, the use of this field may be decided by the responsible administration
DTE	Data terminal equipment (DTE) ready (0 = available, 1 = not available = default)
Spare	Spare. Not used. Should be set to zero. Reserved for future use.
Number of bits	Occupies 2 slots

B Vessel Sub-segments

Sub-segments as defined by Maritime Optima.

Segment	Sub-segment	Range
Dry bulk	Mini bulkers 1	0 - 5,000 DWT
	Mini bulkers 2	5,000 - 10,000 DWT
	Mini bulkers 3	10,000 - 15,000 DWT
	Handysize	15,000 - 40,000 DWT
	Supramax	40,000 - 60,000 DWT
	Ultramax	60,000 - 67,000 DWT
	Panamax	67,000 - 78,000 DWT
	Kamsarmax	78,000 - 86,000 DWT
	Post Panamax	86,000 - 100,000 DWT
	Baby Cape	100,000 - 140,000 DWT
	Cape	140,000 - 200,000 DWT
	Newcastlemax	200,000 - 210,000 DWT
	Ultra Cape	210,000+ DWT
Tankers	Small	0 - 13,000 DWT
	Intermediate	13,000 - 20,000 DWT
	Flexy	20,000 - 30,000 DWT
	Handy	30,000 - 43,000 DWT
	Medium Range	43,000 - 50,000 DWT
	Panamax (LR 1)	50,000 - 80,000 DWT
	Panamax (LR 2)	80,000 - 125,000 DWT
	Suezmax	125,000 - 200,000 DWT
	VLCC	200,000+ DWT
Chemical	Small 1	0 - 5,000 DWT
	Small 2	5,000 - 10,000 DWT
	Intermediate	10,000 - 19,000 DWT
	Flexy	19,000 - 25,000 DWT
	Handy	25,000 - 30,000 DWT
	Medium Range	30,000 - 45,000 DWT
	Panamax	45,000 - 80,000 DWT
	80 000+	80,000+ DWT
LPG	Coaster	0 - 15,000 CBM
	Handy	15,000 - 25,000 CBM
	MGC	25,000 - 50,000 CBM
	LGC	50,000 - 70,000 CBM
	VLGC	70,000 - 120,000 CBM
LNG	Small	0 - 20,000 CBM
	Medium	20,000 - 100,000 CBM
	Large	100,000 - 200,000 CBM
	Very Large	200,000+ CBM

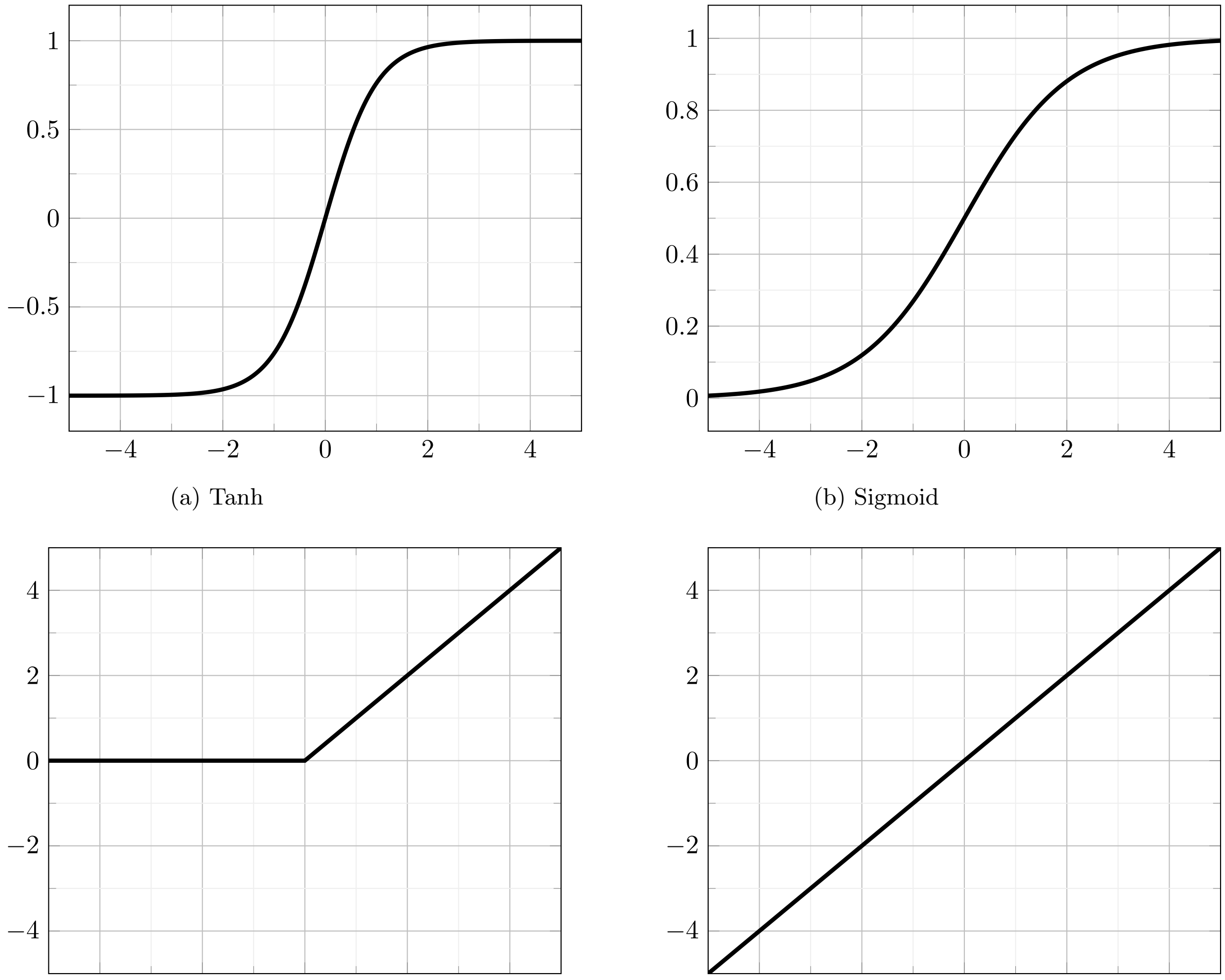
C Summary of Maritime Analytical Techniques

A summary of the four most popular forecasting techniques within the shipping industry. From Martin Stopford [2008] Table 17.2.

Analytical technique		**Main characteristic**
Opinion survey	Delphi technique	Discussion session in which a group of experts make a consensus forecast
	Opinion surveys	Send questionnaire to selection of experts and analyse results
Trend analysis	Naive	Simple rule e.g. ‘no change’, or ‘if earnings are more than twice OPEX they will fall’
	Trend extrapolation	Fit a trend using one of several methodologies and extrapolate forward
	Smoothing	Smooth out fluctuations to obtain average change, and project this
	Decomposition	Split out trend, seasonality, cyclicality and random fluctuations, and project each separately
	Filters	Forecasts are expressed as a linear combination of past actual values and/or errors
	Autoregressive (ARMA)	Forecasts expressed as a linear combination of past actual values
	Box-Jenkins model	Variant of the ARMA model, with rules to deal with the problem of stability
Mathematical model	Single regression	Estimated equation with one explanatory variable to predict target variable
	Multiple regression	Estimated equation with more than one independent variable to predict target variable
	Econometric models	System of regression equations to predict target variable
	Supply-demand models	Estimate supply and demand from their component parts and predict change in balance
	Sensitivity analysis	Examine the sensitivity of the forecast to different assumptions
Probability analysis	Monte Carlo	Probability analysis used to calculate the likelihood of a particular outcome occurring.

D Activation Functions

Graphical representations of relevant activation functions referenced throughout this thesis. These illustrations serve to provide a clear understanding of the functional characteristics and the input-output mapping that each activation function facilitates.



(a) Tanh

(b) Sigmoid

(c) ReLU

(d) Linear

E Model Implementations

The code implementation of the various models used for the experiments, employing the Tensorflow and Keras libraries.

E.1 Fully Connected Deep Neural Networks

```
model = Sequential()
model.add(Input(shape=(sequence_length, num_features)))

# Project and flatten
model.add(Dense(int(projection_proportion * num_features)))
model.add(Flatten())

# Hidden layers
for _ in range(num_hidden_layers):
    model.add(Dense(hidden_layer_size, activation="relu"))
    model.add(Dropout(dropout_rate))

# Output layer
model.add(Dense(1))
```

E.2 Recurrent Neural Networks

The subsequent architectural blueprint utilized two distinct variations of RNNs: the *SimpleRNN* and the *LSTM* cells from Tensorflow. The code implementation presented herein presents a generic "RNN" cell as a prototype. However, in the actual implementation, this prototypical cell was substituted with the specific cell types that corresponded to each variant of the RNN models.

```
model = Sequential([
    Input(shape=(sequence_length, num_features)),

    # 2 recurrent layers
    RNN(rnn_layer_size, return_sequences=True),
    Dropout(rnn_dropout_rate),
    RNN(rnn_layer_size, return_sequences=False),
    Dropout(rnn_dropout_rate),

    # Final dense and output layer
    Dense(dnn_layer_size, activation="relu"),
    Dropout(dnn_dropout_rate),
    Dense(1),
])
```

E.3 1D-CNN

```
model = Sequential()
model.add(Input(shape=(sequence_length, num_features)))

# CNN layers
for _ in range(num_cnn_blocks):
    model.add(Conv1D(
        num_cnn_filters,
        kernel_size=kernel_size,
        activation="relu"
    ))
    model.add(Dropout(cnn_dropout_rate))

# Flatten CNN output
model.add(Flatten())

# Final dense and output layer
model.add(Dense(dnn_layer_size, activation="relu"))
model.add(Dropout(dnn_dropout_rate))
model.add(Dense(1))
```

E.4 CNN-LSTM

```
model = Sequential([
    Input(shape=(sequence_length, num_features)),

    # Temporal convolution projection
    Conv1D(
        int(projection_proportion * num_features),
        kernel_size=kernel_size
    ),

    # 2 recurrent layers
    RNN(rnn_layer_size, return_sequences=True),
    Dropout(rnn_dropout_rate),
    RNN(rnn_layer_size, return_sequences=False),
    Dropout(rnn_dropout_rate),

    # Final dense and output layer
    Dense(dnn_layer_size, activation="relu"),
    Dropout(dnn_dropout_rate),
    Dense(1),
])
```

E.5 Transformer

```
# Input definition
inputs = Input(shape=(sequence_length, num_features))

# Projection
x = Dense(head_size)(inputs)

# Scaling
x = x * tf.math.sqrt(tf.cast(head_size, tf.float32))

# Positional encoding with dropout
positional_encoding = RelativePositionEmbedding(head_size)(x)
x = x + positional_encoding
x = Dropout(dropout)(x)

# Encoder stack
for _ in range(num_transformer_blocks):
    x = TransformerEncoderBlock(
        num_attention_heads=num_heads,
        inner_dim=encoder_nn_dim,
        inner_activation="relu",
        output_dropout=dropout,
        attention_dropout=dropout,
        inner_dropout=dropout,
    )(x)

x = Flatten()(x)
outputs = Dense(1)(x)

model = Model(inputs, outputs)
```

F Libraries and Versions

This appendix provides a comprehensive list of the libraries and versions utilized in the execution of the experiments within this thesis. These libraries constitute a mix of dependencies that were inherent to the Google Cloud VM, specifically the *c1-deeplearning-tf-2-10-cu113-v20230501-debian-10-py37* image, and additional necessary dependencies that were subsequently configured after a **fresh install of a compatible Python 3.9.9 version**. These libraries and their specific versions ensure the successful execution and replication of the experimental procedures.

```
absl-py==1.4.0
array-record==0.4.0
astunparse==1.6.3
cachetools==5.3.1
certifi==2023.5.7
charset-normalizer==3.1.0
click==8.1.3
cloudpickle==2.2.1
colorama==0.4.6
cycler==0.11.0
Cython==0.29.35
dm-tree==0.1.8
etils==1.3.0
flatbuffers==23.5.26
gast==0.4.0
gin-config==0.5.0
google-api-core==2.11.1
google-api-python-client==2.90.0
google-auth==2.20.0
google-auth-httplib2==0.1.0
google-auth-oauthlib==0.4.6
google-pasta==0.2.0
googleapis-common-protos==1.59.1
grpcio==1.56.0
h5py==3.9.0
httplib2==0.22.0
idna==3.4
immutabledict==2.2.4
importlib-metadata==6.7.0
importlib-resources==5.12.0
joblib==1.2.0
kaggle==1.5.13
keras==2.10.0
Keras-Preprocessing==1.1.2
kiwisolver==1.4.4
libclang==16.0.0
llvmlite==0.40.1
lxml==4.9.2
Markdown==3.4.3
MarkupSafe==2.1.3
matplotlib==3.3.3
matplotlib-inline==0.1.6
numba==0.57.1
numpy==1.23.5
oauth2client==4.1.3
oauthlib==3.2.2
opencv-python-headless==4.5.2.52
opt-einsum==3.3.0
packaging==23.1
pandas==2.0.2
patsy==0.5.3
```

```
Pillow==9.5.0
portalocker==2.7.0
promise==2.3
protobuf==3.19.6
psutil==5.9.5
py-cpuinfo==9.0.0
pyasn1==0.5.0
pyasn1-modules==0.3.0
pycocotools==2.0.6
pyparsing==3.1.0
python-dateutil==2.8.2
python-slugify==8.0.1
pytz==2023.3
PyYAML==5.4.1
regex==2023.6.3
requests==2.31.0
requests-oauthlib==1.3.1
rsa==4.9
sacrebleu==2.2.0
scikit-learn==1.2.2
scipy==1.10.1
seaborn==0.12.2
sentencepiece==0.1.99
seqeval==1.2.2
shap==0.41.0
six==1.16.0
slicer==0.0.7
statsmodels==0.14.0
tabulate==0.9.0
tensorboard==2.10.1
tensorboard-data-server==0.6.1
tensorboard-plugin-wit==1.8.1
tensorflow==2.10.0
tensorflow-addons==0.20.0
tensorflow-datasets==4.9.0
tensorflow-estimator==2.10.0
tensorflow-hub==0.13.0
tensorflow-io-gcs-filesystem==0.32.0
tensorflow-metadata==1.13.0
tensorflow-model-optimization==0.7.5
tensorflow-text==2.10.0
termcolor==2.3.0
text-unidecode==1.3
tf-models-official==2.10.0
tf-slim==1.1.0
threadpoolctl==3.1.0
toml==0.10.2
tqdm==4.65.0
traitlets==5.9.0
typeguard==2.13.3
typing_extensions==4.6.3
tzdata==2023.3
uritemplate==4.1.1
urllib3==1.26.16
Werkzeug==2.3.6
wrapt==1.15.0
zipp==3.15.0
```

G Custom SHAP Visualization Code

The appendix provides the code used for the custom SHAP visualizations.

G.1 Custom SHAP Heatmap Plot Code

Both *shap_ values* and *feature_ values* should have the shape: (num_samples, sequence_length).

```
num_samples, num_timesteps = shap_values.shape
df = pd.DataFrame(
    {
        "shap_value": shap_values.flatten(),
        "feature_value": feature_values.flatten(),
        "sample": np.repeat(np.arange(num_samples), num_timesteps),
        "timestep": np.tile(np.arange(num_timesteps), num_samples),
    }
)

df["feature_value"] = MinMaxScaler().fit_transform(df[["feature_value"]])

cmap = ListedColormap(sns.color_palette("coolwarm", n_colors=256))
triangulation = tri.Triangulation(df["timestep"], df["shap_value"])
plt.tricontourf(triangulation, df["feature_value"], cmap=cmap)

plt.plot(
    df["timestep"].unique(),
    df.groupby("timestep")["shap_value"].mean().values,
    color="black",
    linewidth=2,
    alpha=0.5,
)

plt.gca().xaxis.set_major_formatter(
    plt.FuncFormatter(
        lambda value, tick_number: f"t-{str(num_timesteps - int(value))}"
        if int(value) < num_timesteps
        else "t"
    )
)

# (...)
```

G.2 Custom SHAP Violin Plot Code

Both variables *shap_ values* and *feature_ values* should have the shape: (num_ samples, sequence_ length, features), where one entry in the *features* constitues one violin. *max_ features* is the number of violins to show with the highest overall importance, and *feature_ names* is an array of the feature names as provided in the *shap_ values* and *feature_ values* variables.

```python
feature_importance = np.mean(np.abs(shap_values), axis=0)
top_indices = np.argsort(feature_importance)[-max_features:][::-1]

fig, axes = plt.subplots(max_features, 1, figsize=(10, max_features))
cmap = mcolors.LinearSegmentedColormap.from_list("n", ["blue", "red"])

for idx, i in enumerate(top_indices):
    feature_name = feature_names[i]
    feature_vals = feature_values[:, i]
    shap_vals = shap_values[:, i]

    df = pd.DataFrame({"feature_value": feature_vals, "shap_value": shap_vals})
    norm = mcolors.Normalize(vmin=feature_vals.min(), vmax=feature_vals.max())

    # Divide the data into bins based on the feature values
    bins = pd.qcut(df["feature_value"], q=10, duplicates="drop")
    for bin in bins.unique():
        subset = df[bins == bin]
        sns.kdeplot(
            subset["shap_value"],
            fill=True,
            label=str(bin),
            color=cmap(norm(subset["feature_value"].mean())),
            ax=axes[idx],
            alpha=0.4,
        )

    axes[idx].set_yticks([])
    axes[idx].set_xlabel("")
    axes[idx].axvline(0, color="k", linestyle="--", alpha=0.5)
    axes[idx].set_ylabel(
        feature_name, rotation=0, labelpad=5, ha="right", va="center"
    )

    if axes[idx].legend_ is not None:
        axes[idx].legend_.remove()

# (...)
```

H Model Performance Results

H.1 Experiment 1 - Baltic Dry Index

Low Complexity 20

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0298	**0.0027**	0.0057	0.0057	0.0045	0.0074	0.0030
CCC	0.1110	0.6789	0.5478	0.4761	0.5964	0.3045	**0.6913**
Sign	0.6333	**0.8167**	0.6750	0.7000	0.6667	0.4167	0.7833
Variance	0.5315	0.1163	0.0620	0.1300	0.0996	0.1290	0.1005
Importance	0.3307	0.0757	0.0332	0.0824	0.0589	0.0730	0.0627

Low Complexity 30

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0094	0.0035	0.0029	0.0057	0.0047	**0.0015**	0.0027
CCC	0.4393	0.6044	0.6898	0.5136	0.6346	**0.7749**	0.6651
Sign	0.5583	0.6417	0.6667	0.5833	0.7417	**0.8333**	0.7667
Variance	0.1710	0.0604	0.0414	0.0835	0.0724	0.1017	0.0775
Importance	0.1083	0.0385	0.0239	0.0477	0.0423	0.0609	0.0471

Low Complexity 40

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0159	0.0027	**0.0012**	0.0046	0.0042	0.0050	0.0028
CCC	0.2869	0.5230	**0.8389**	0.5011	0.6010	0.6273	0.6156
Sign	0.4750	0.5500	**0.7917**	0.6917	0.6917	0.7167	0.6250
Variance	0.1023	0.0607	0.0367	0.0804	0.0544	0.0866	0.0591
Importance	0.0636	0.0379	0.0208	0.0424	0.0317	0.0510	0.0365

Medium Complexity 20

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0107	0.0028	**0.0025**	0.0080	0.0039	0.0046	0.0060
CCC	0.2709	0.6104	**0.7342**	0.2275	0.6223	0.4879	0.4137
Sign	0.6417	0.5333	0.7750	0.5500	**0.7833**	0.6250	0.5917
Variance	0.0346	0.0313	0.0305	0.0512	0.0337	0.0409	0.0512
Importance	0.0209	0.0191	0.0157	0.0246	0.0193	0.0233	0.0314

Medium Complexity 30

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0120	0.0035	**0.0026**	0.0101	0.0031	0.0079	0.0048
CCC	0.1773	0.4080	**0.6574**	0.3124	0.5560	0.5131	0.3905
Sign	0.4083	0.4667	0.6750	0.6167	0.7000	**0.7333**	0.6167
Variance	0.0219	0.0171	0.0136	0.0334	0.0247	0.0318	0.0412
Importance	0.0131	0.0105	0.0076	0.0135	0.0128	0.0172	0.0256

Medium Complexity 40

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0127	**0.0018**	0.0019	0.0050	0.0033	0.0048	0.0051
CCC	0.1313	**0.6944**	0.6497	0.3406	0.5653	0.5090	0.3519
Sign	0.4250	0.6083	0.6333	0.6333	**0.6917**	0.6667	0.4667
Variance	0.0158	0.0138	0.0135	0.0303	0.0179	0.0244	0.0290
Importance	0.0094	0.0085	0.0076	0.0111	0.0083	0.0115	0.0179

High Complexity 20

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0070	0.0033	**0.0028**	0.0042	0.0039	0.0049	0.0060
CCC	0.2499	0.4841	**0.5310**	0.5094	0.3865	0.1904	0.1680
Sign	0.5333	0.5333	**0.6417**	0.5417	0.4833	0.5250	0.4667
Variance	0.0122	0.0113	0.0154	0.0206	0.0207	0.0173	0.0374
Importance	0.0072	0.0068	0.0080	0.0098	0.0114	0.0093	0.0226

High Complexity 30

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0089	**0.0034**	0.0036	0.0040	0.0035	0.0036	0.0058
CCC	0.2009	0.5341	0.5403	0.3995	**0.5496**	0.3471	0.4208
Sign	0.6000	0.5333	0.5500	0.5250	**0.6167**	0.4750	0.4833
Variance	0.0079	0.0087	0.0074	0.0278	0.0168	0.0126	0.0246
Importance	0.0046	0.0052	0.0039	0.0092	0.0077	0.0060	0.0148

High Complexity 40

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0131	0.0033	**0.0030**	0.0051	0.0034	0.0037	0.0071
CCC	0.0878	0.4203	0.4295	0.4156	**0.5434**	0.4032	0.2042
Sign	0.4667	**0.5417**	0.5000	0.5250	0.4833	0.4917	0.5083
Variance	0.0058	0.0084	0.0068	0.0199	0.0116	0.0090	0.0243
Importance	0.0033	0.0050	0.0037	0.0062	0.0047	0.0035	0.0147

H.2 Experiment 2 - Breakwave Dry Bulk Shipping ETF

Low Complexity 20

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0318	0.0011	**0.0009**	0.0027	0.0016	0.0011	0.0034
CCC	0.1431	0.6219	**0.6456**	0.4776	0.5176	0.5991	0.5088
Sign	0.6833	**0.7583**	0.6000	0.6417	0.6417	0.6667	0.7083
Variance	0.5544	0.1007	0.0642	0.1220	0.0982	0.1290	0.1201
Importance	0.3424	0.0631	0.0306	0.0784	0.0604	0.0766	0.0742

Low Complexity 30

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0165	0.0013	0.0010	0.0022	0.0014	**0.0007**	0.0031
CCC	0.2714	0.4524	0.6643	0.3134	0.5921	**0.7176**	0.4828
Sign	**0.7833**	0.6417	0.7000	0.5167	0.6500	0.7750	0.7333
Variance	0.1898	0.0675	0.0399	0.0757	0.0715	0.0754	0.0803
Importance	0.1189	0.0435	0.0215	0.0454	0.0440	0.0477	0.0499

Low Complexity 40

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0100	0.0006	**0.0005**	0.0030	0.0011	0.0016	0.0022
CCC	0.3298	0.6066	**0.6676**	0.2280	0.5763	0.5829	0.4781
Sign	**0.8250**	0.6750	0.7000	0.5167	0.6333	0.8083	0.7583
Variance	0.1118	0.0515	0.0318	0.0637	0.0490	0.0742	0.0591
Importance	0.0700	0.0324	0.0170	0.0361	0.0277	0.0424	0.0362

Medium Complexity 20

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0040	**0.0010**	0.0013	0.0024	0.0013	0.0022	0.0027
CCC	0.2920	**0.4875**	0.3444	0.2732	0.3699	0.3401	0.3083
Sign	0.6000	0.5917	0.4750	0.5667	0.5917	**0.6250**	0.5833
Variance	0.0421	0.0271	0.0192	0.0358	0.0270	0.0364	0.0435
Importance	0.0256	0.0171	0.0096	0.0187	0.0149	0.0204	0.0266

Medium Complexity 30

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0034	**0.0011**	0.0011	0.0040	0.0025	0.0014	0.0023
CCC	0.3044	0.5746	**0.6961**	0.2992	0.3911	0.4734	0.3695
Sign	0.6583	0.6167	**0.8000**	0.5417	0.6917	0.7333	0.5917
Variance	0.0259	0.0225	0.0134	0.0298	0.0264	0.0309	0.0345
Importance	0.0157	0.0140	0.0072	0.0134	0.0133	0.0161	0.0211

Medium Complexity 40

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0021	**0.0007**	0.0007	0.0024	0.0016	0.0015	0.0007
CCC	0.4414	0.6155	0.5274	0.2650	0.4628	0.4319	**0.7268**
Sign	0.7333	0.6750	0.6583	0.5500	0.6583	0.6583	**0.7917**
Variance	0.0184	0.0187	0.0093	0.0275	0.0189	0.0219	0.0227
Importance	0.0111	0.0117	0.0052	0.0103	0.0082	0.0100	0.0139

High Complexity 20

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0019	0.0010	0.0008	0.0039	**0.0008**	0.0013	0.0014
CCC	0.3664	0.4660	0.4661	0.0662	**0.6454**	0.3254	0.3073
Sign	0.5583	0.6000	0.5000	0.3750	**0.7417**	0.5250	0.5667
Variance	0.0152	0.0151	0.0109	0.0203	0.0133	0.0154	0.0327
Importance	0.0089	0.0091	0.0055	0.0092	0.0064	0.0079	0.0196

High Complexity 30

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0017	0.0006	**0.0005**	0.0026	0.0008	0.0007	0.0016
CCC	0.3992	0.6735	**0.7648**	0.1169	0.5904	0.5815	0.4768
Sign	0.6417	0.7333	**0.7417**	0.4917	0.7333	0.6333	0.6250
Variance	0.0099	0.0087	0.0068	0.0179	0.0120	0.0107	0.0206
Importance	0.0058	0.0051	0.0036	0.0070	0.0051	0.0050	0.0124

High Complexity 40

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0026	0.0008	**0.0007**	0.0033	0.0011	0.0010	0.0018
CCC	0.2582	**0.5877**	0.4805	0.0974	0.4903	0.3755	0.2769
Sign	0.5750	**0.7167**	**0.7167**	0.4167	0.6917	0.6000	0.5417
Variance	0.0072	0.0078	0.0054	0.0155	0.0098	0.0093	0.0163
Importance	0.0042	0.0047	0.0029	0.0049	0.0035	0.0033	0.0098

H.3 Experiment 3 - Golden Ocean Group Ltd

Low Complexity 20

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0471	**0.0109**	0.0523	0.0396	0.0302	0.0163	0.0113
CCC	0.0575	**0.1114**	-0.0207	-0.0269	0.0593	0.0516	-0.0137
Sign	0.6250	**0.6750**	0.3333	0.3917	0.4083	0.5583	0.3833
Variance	0.5334	0.1167	0.1240	0.2187	0.0719	0.1447	0.1680
Importance	0.3292	0.0703	0.0658	0.1160	0.0418	0.0870	0.1042

Low Complexity 30

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0228	**0.0085**	0.0554	0.0178	0.0386	0.0255	0.0096
CCC	0.0548	0.0560	0.0360	**0.0963**	0.0280	0.0423	-0.0078
Sign	0.5500	0.4417	0.4083	0.5500	0.4083	**0.5750**	0.4000
Variance	0.1933	0.0761	0.0746	0.0843	0.0515	0.1100	0.1060
Importance	0.1203	0.0470	0.0424	0.0518	0.0277	0.0640	0.0657

Low Complexity 40

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0312	**0.0006**	0.0201	0.0249	0.0118	0.0033	0.0157
CCC	0.0164	**0.6932**	-0.0681	0.0756	0.0376	0.3596	0.0485
Sign	0.4083	**0.8167**	0.3250	0.4500	0.4417	0.6833	0.4250
Variance	0.1101	0.0585	0.0326	0.0836	0.0648	0.0718	0.1091
Importance	0.0687	0.0370	0.0186	0.0437	0.0335	0.0432	0.0682

Medium Complexity 20

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0062	0.0065	**0.0016**	0.0032	0.0067	0.0088	0.0024
CCC	0.3422	0.1089	**0.5745**	0.3235	0.2830	0.3457	0.5305
Sign	0.7000	0.5750	0.7083	0.6500	0.5417	0.7333	**0.7667**
Variance	0.0381	0.0267	0.0209	0.0344	0.0277	0.0417	0.0416
Importance	0.0231	0.0166	0.0104	0.0174	0.0148	0.0236	0.0255

Medium Complexity 30

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0027	0.0150	**0.0010**	0.0024	0.0052	0.0027	0.0021
CCC	0.4638	0.1184	**0.6269**	0.2871	0.1686	0.4779	0.4553
Sign	**0.7333**	0.5250	0.7083	0.6083	0.5333	0.6917	0.5417
Variance	0.0239	0.0468	0.0130	0.0290	0.0213	0.0285	0.0327
Importance	0.0144	0.0290	0.0069	0.0122	0.0103	0.0155	0.0203

Medium Complexity 40

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0016	0.0157	**0.0009**	0.0138	0.0030	0.0052	0.0019
CCC	0.4913	0.0208	**0.6960**	0.2310	0.4056	0.2925	0.4954
Sign	0.6750	0.4167	**0.7417**	0.6750	0.6667	0.6333	0.7000
Variance	0.0168	0.0181	0.0097	0.0292	0.0196	0.0291	0.0252
Importance	0.0101	0.0113	0.0053	0.0102	0.0092	0.0137	0.0156

High Complexity 20

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0224	0.0451	0.0158	0.0514	0.0189	0.0110	**0.0110**
CCC	0.0986	0.0409	0.1040	0.0203	0.0839	**0.2076**	0.1939
Sign	0.4083	0.4083	0.4083	0.4000	0.4917	0.4667	**0.5250**
Variance	0.0138	0.0333	0.0138	0.0876	0.0251	0.0221	0.0373
Importance	0.0081	0.0202	0.0067	0.0386	0.0140	0.0124	0.0226

High Complexity 30

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0161	0.0102	0.0091	0.0243	**0.0063**	0.0121	0.0081
CCC	0.0970	0.1143	0.1710	0.0471	0.2045	0.2106	**0.2131**
Sign	0.4083	0.4083	0.4083	0.4500	0.4833	**0.6000**	0.5250
Variance	0.0089	0.0115	0.0081	0.0417	0.0191	0.0141	0.0294
Importance	0.0052	0.0070	0.0042	0.0143	0.0093	0.0069	0.0178

High Complexity 40

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0132	0.0160	0.0067	0.0180	0.0085	0.0103	**0.0022**
CCC	0.1107	0.1278	0.2431	0.1199	0.2491	0.1823	**0.5279**
Sign	0.4083	0.4167	0.5417	0.5667	0.6417	0.5667	**0.6667**
Variance	0.0064	0.0095	0.0066	0.0198	0.0145	0.0127	0.0180
Importance	0.0038	0.0058	0.0036	0.0061	0.0060	0.0055	0.0109

H.4 Experiment 4 - Frontline Ltd

Low Complexity 20

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.1563	0.1660	0.1546	0.1882	0.1476	0.1441	**0.0474**
CCC	0.0226	0.0126	0.0327	0.0250	0.0283	0.0152	**0.1788**
Sign	0.4917	0.4917	0.4917	**0.5250**	0.4917	0.4917	**0.5250**
Variance	0.2559	0.0454	0.0262	0.4892	0.0605	0.0744	0.2133
Importance	0.1550	0.0266	0.0127	0.2758	0.0363	0.0439	0.1315

Low Complexity 30

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.1393	0.1372	0.1392	0.1161	0.1415	0.1316	**0.0499**
CCC	0.0254	**0.0500**	0.0450	0.0220	0.0375	0.0287	0.0287
Sign	0.4917	0.4917	0.4917	**0.5083**	0.4917	0.4917	0.4417
Variance	0.1026	0.0293	0.0199	0.2951	0.0423	0.0537	0.1916
Importance	0.0641	0.0178	0.0098	0.1331	0.0224	0.0304	0.1202

Low Complexity 40

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.1426	0.1086	0.1404	0.1764	0.1511	0.1585	**0.0353**
CCC	0.0153	0.0540	**0.0540**	0.0342	0.0185	0.0253	0.0153
Sign	**0.4917**	**0.4917**	**0.4917**	**0.4917**	**0.4917**	**0.4917**	0.3917
Variance	0.0599	0.0254	0.0157	0.2055	0.0366	0.0382	0.1208
Importance	0.0376	0.0152	0.0082	0.0741	0.0176	0.0196	0.0755

Medium Complexity 20

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0969	0.1033	0.0747	0.1055	0.0859	0.0888	**0.0438**
CCC	0.0363	0.0479	**0.0972**	0.0408	0.0570	-0.0105	0.0867
Sign	0.4917	**0.5000**	0.4917	0.4917	0.4917	0.4917	0.4083
Variance	0.0239	0.0151	0.0345	0.0220	0.0165	0.0342	0.1116
Importance	0.0142	0.0090	0.0181	0.0095	0.0087	0.0190	0.0676

Medium Complexity 30

Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0847	0.1021	0.1153	0.0610	0.1047	0.0656	**0.0280**
CCC	0.0499	0.0604	0.0511	0.0888	0.0223	0.0651	**0.1916**
Sign	0.5000	0.4917	0.4917	**0.5083**	0.4917	0.4917	0.4500
Variance	0.0145	0.0107	0.0067	0.1411	0.0123	0.0244	0.0717
Importance	0.0086	0.0065	0.0035	0.0555	0.0054	0.0121	0.0433

Medium Complexity 40

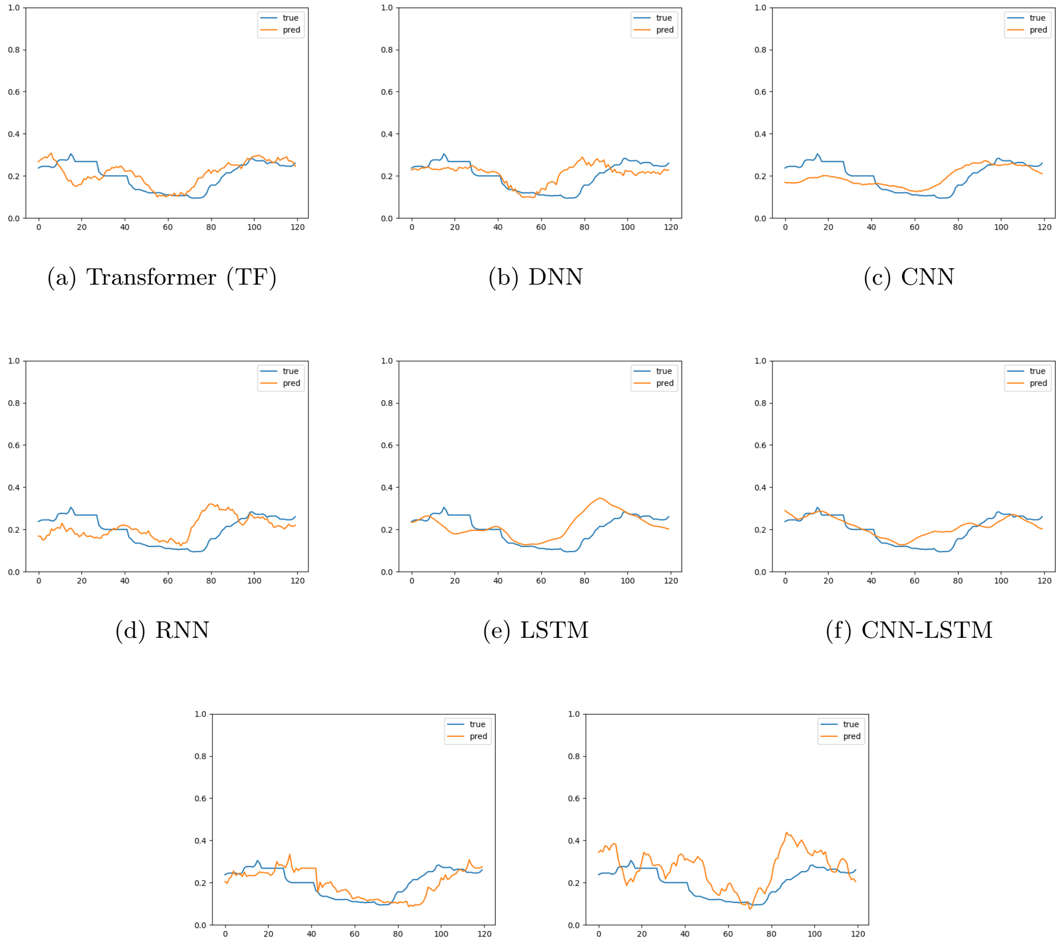
Metric	LR	DNN	CNN	RNN	LSTM	CNN-LSTM	TF
MSE	0.0825	0.0885	0.1203	0.0873	0.1004	0.0537	**0.0274**
CCC	0.0539	0.0517	0.0381	0.0506	0.0682	0.0660	**0.1935**
Sign	0.4917	0.4917	0.4917	0.4917	0.4917	0.5000	**0.5083**
Variance	0.0102	0.0093	0.0051	0.0234	0.0150	0.0194	0.0646
Importance	0.0060	0.0057	0.0026	0.0068	0.0063	0.0084	0.0389

I Model Predictions

I.1 Experiment 1 - Baltic Dry Index

This appendix shows all evaluated model's predictions on the *low complexity 30* configuration of the Baltic Dry Index, which saw the best overall performance across all models.

Absolute Value Plots



(a) Transformer (TF)

(b) DNN

(c) CNN

(d) RNN

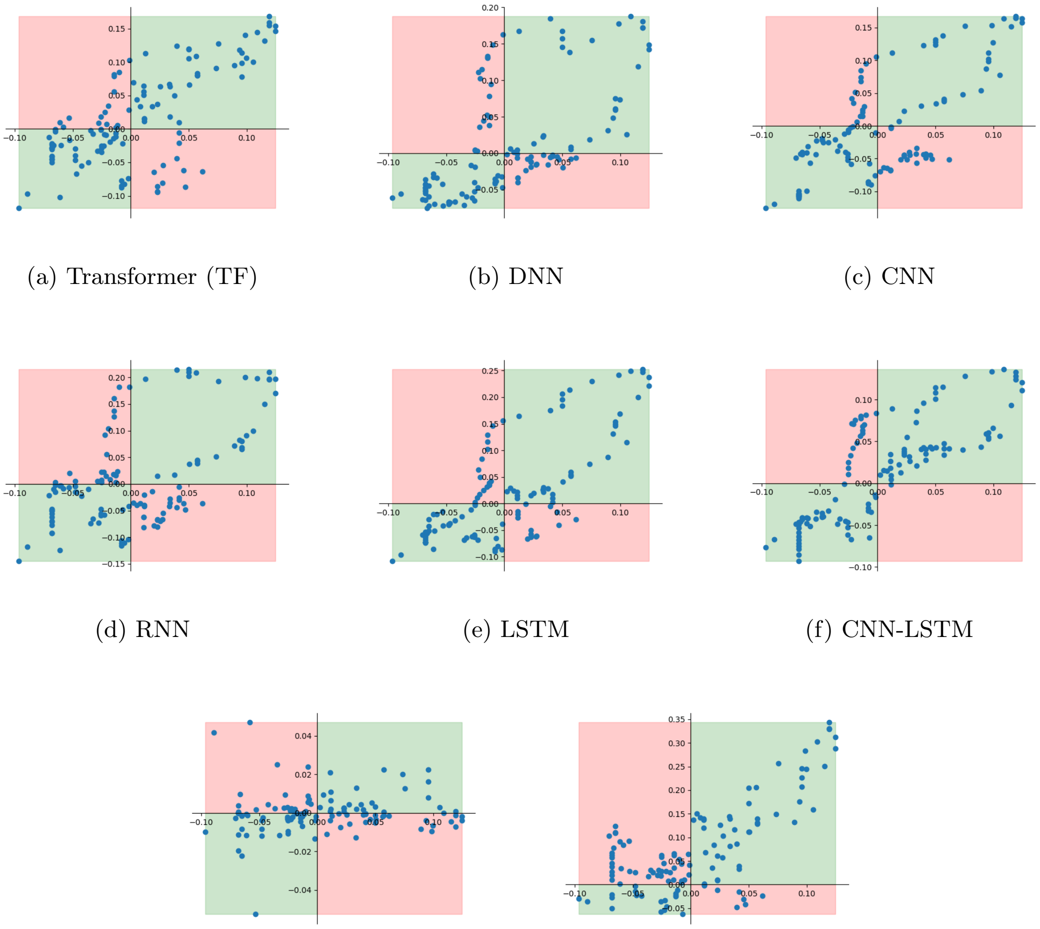
(e) LSTM

(f) CNN-LSTM

(g) ARIMA

(h) Linear (LR)

Change Correlation Plots



(a) Transformer (TF)

(b) DNN

(c) CNN

(d) RNN

(e) LSTM

(f) CNN-LSTM

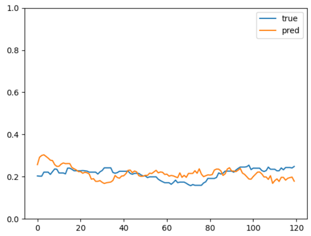
(g) ARIMA

(h) Linear (LR)

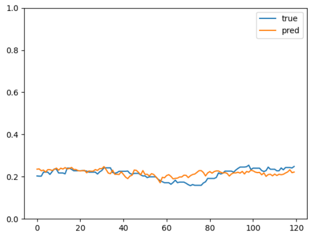
I.2 Experiment 2 - Breakwave Dry Bulk Shipping ETF

This appendix shows all evaluated model's predictions on the *high complexity 30* configuration of the Breakwave Dry Bulk Shipping ETF, which saw the best overall performance across all models.

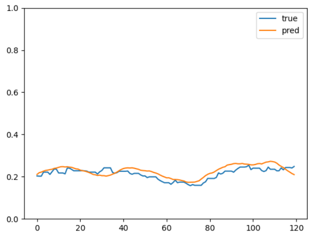
Absolute Value Plots



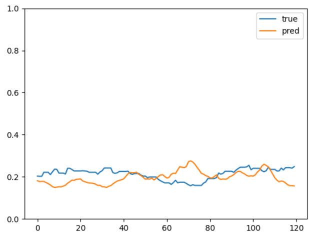
(a) Transformer (TF)



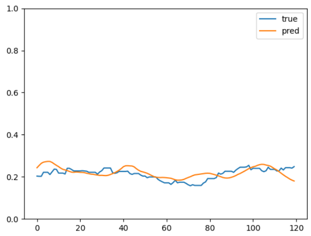
(b) DNN



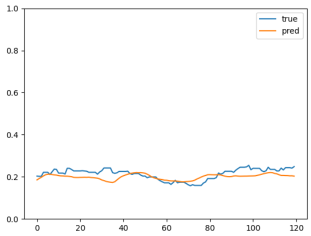
(c) CNN



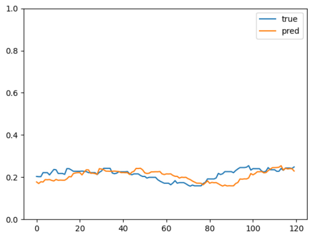
(d) RNN



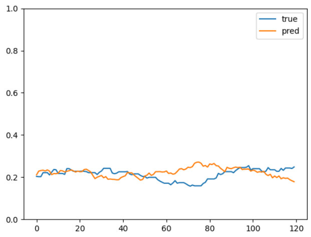
(e) LSTM



(f) CNN-LSTM

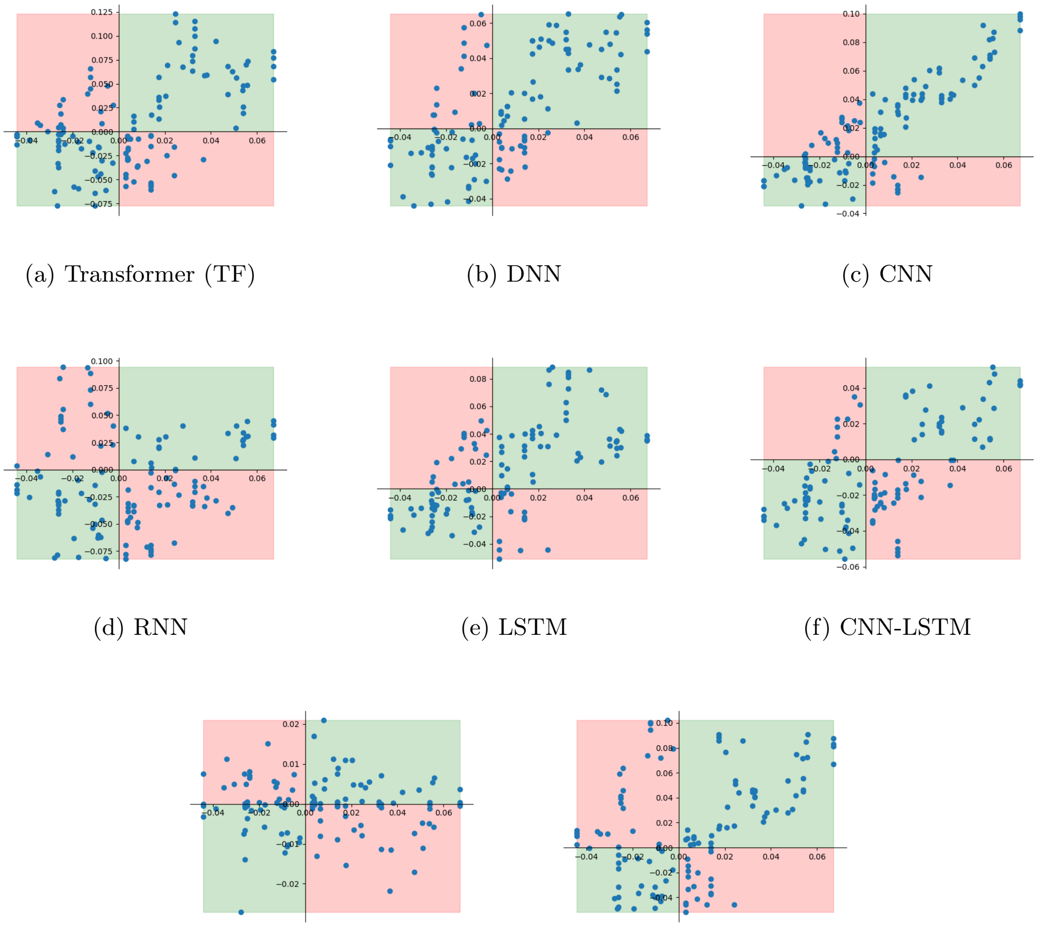


(g) ARIMA



(h) Linear (LR)

Change Correlation Plots



(a) Transformer (TF)

(b) DNN

(c) CNN

(d) RNN

(e) LSTM

(f) CNN-LSTM

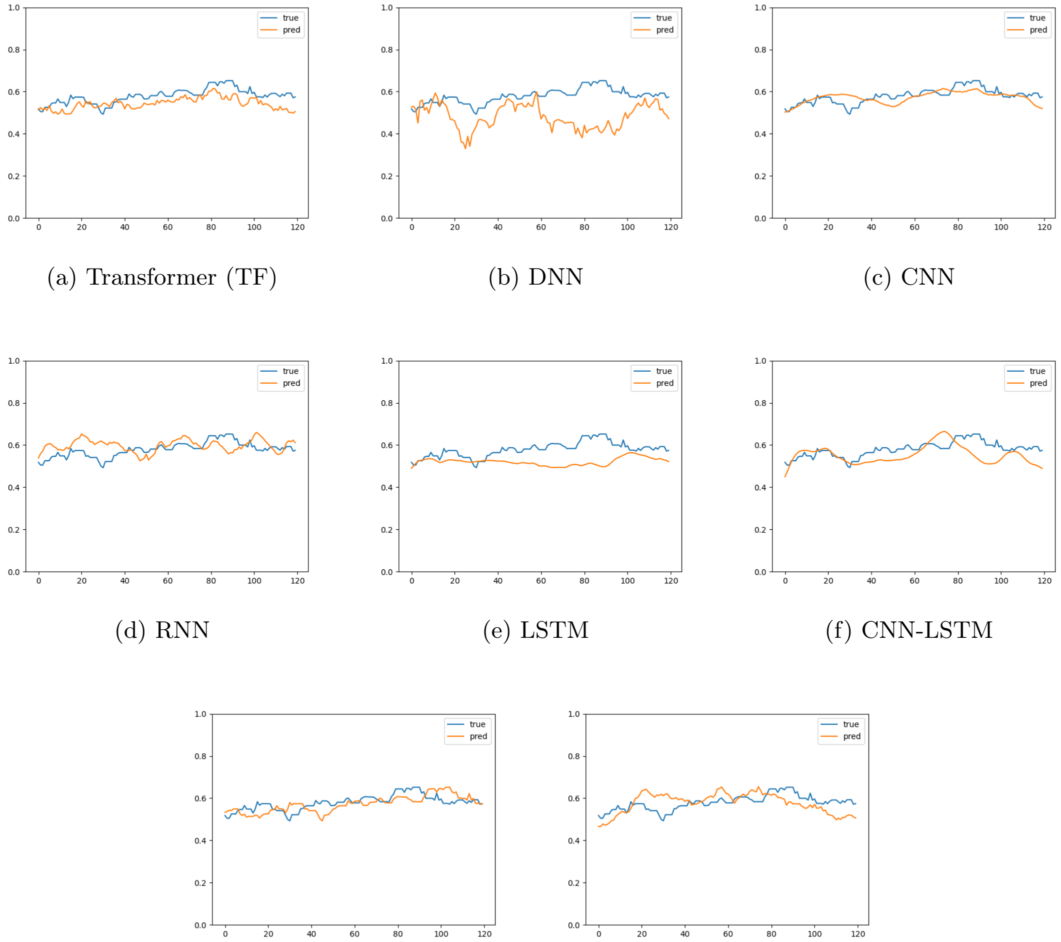
(g) ARIMA

(h) Linear (LR)

I.3 Experiment 3 - Golden Ocean Group Ltd.

This appendix shows all evaluated model's predictions on the *medium complexity 40* configuration of Golden Ocean Group Ltd., which saw the best overall performance across all models.

Absolute Value Plots



(a) Transformer (TF)

(b) DNN

(c) CNN

(d) RNN

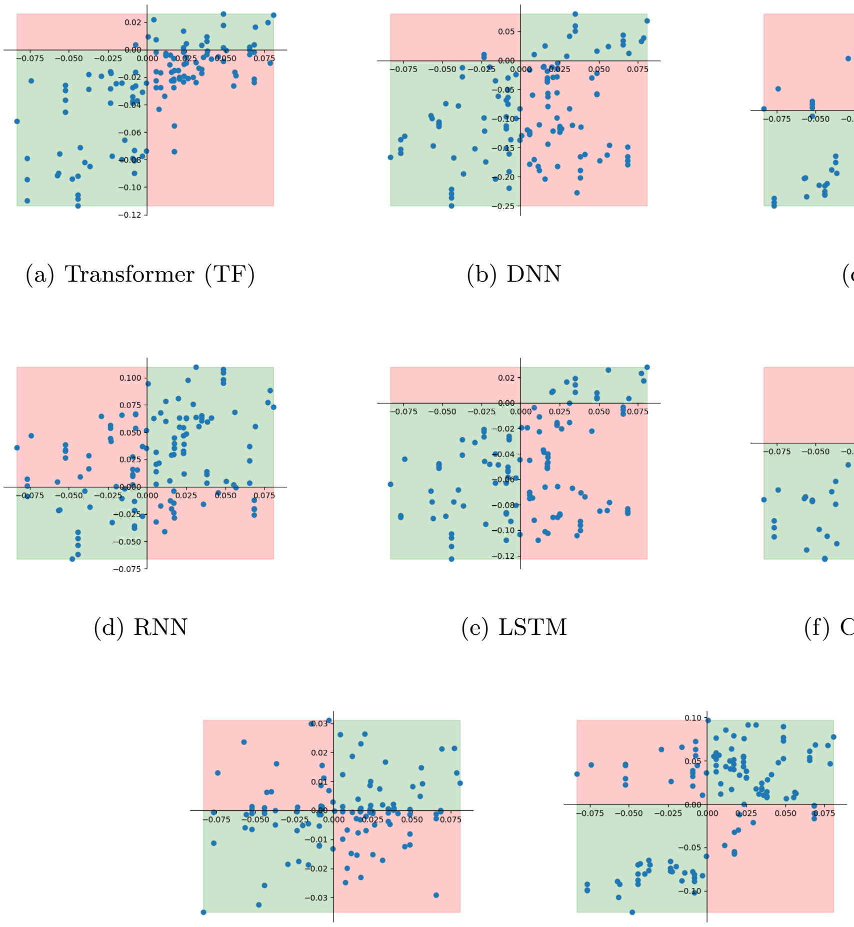
(e) LSTM

(f) CNN-LSTM

(g) ARIMA

(h) Linear (LR)

Change Correlation Plots



(a) Transformer (TF)

(b) DNN

(c) CNN

(d) RNN

(e) LSTM

(f) CNN-LSTM

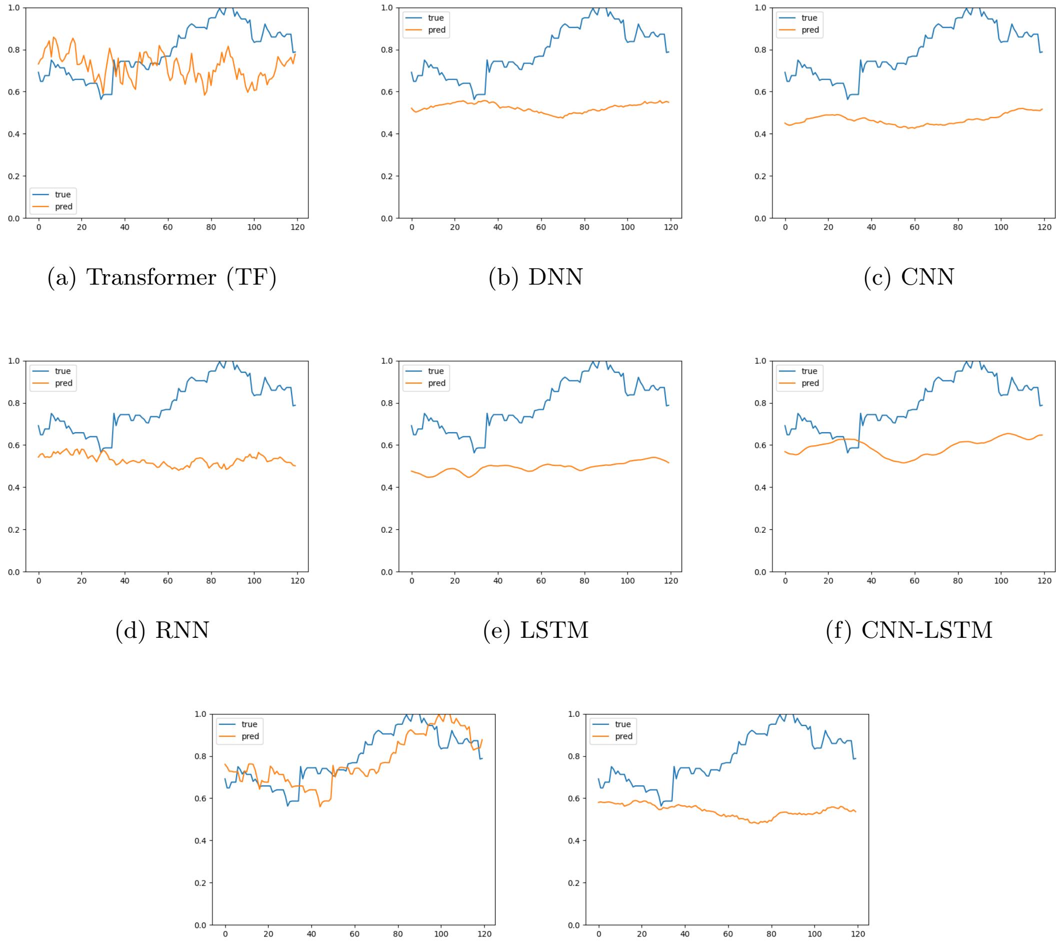
(g) ARIMA

(h) Linear (LR)

I.4 Experiment 4 - Frontline Ltd.

This appendix shows all evaluated model's predictions on the *medium complexity 40* configuration of Frontline Ltd., which saw the best overall performance across all models.

Absolute Value Plots



(a) Transformer (TF)

(b) DNN

(c) CNN

(d) RNN

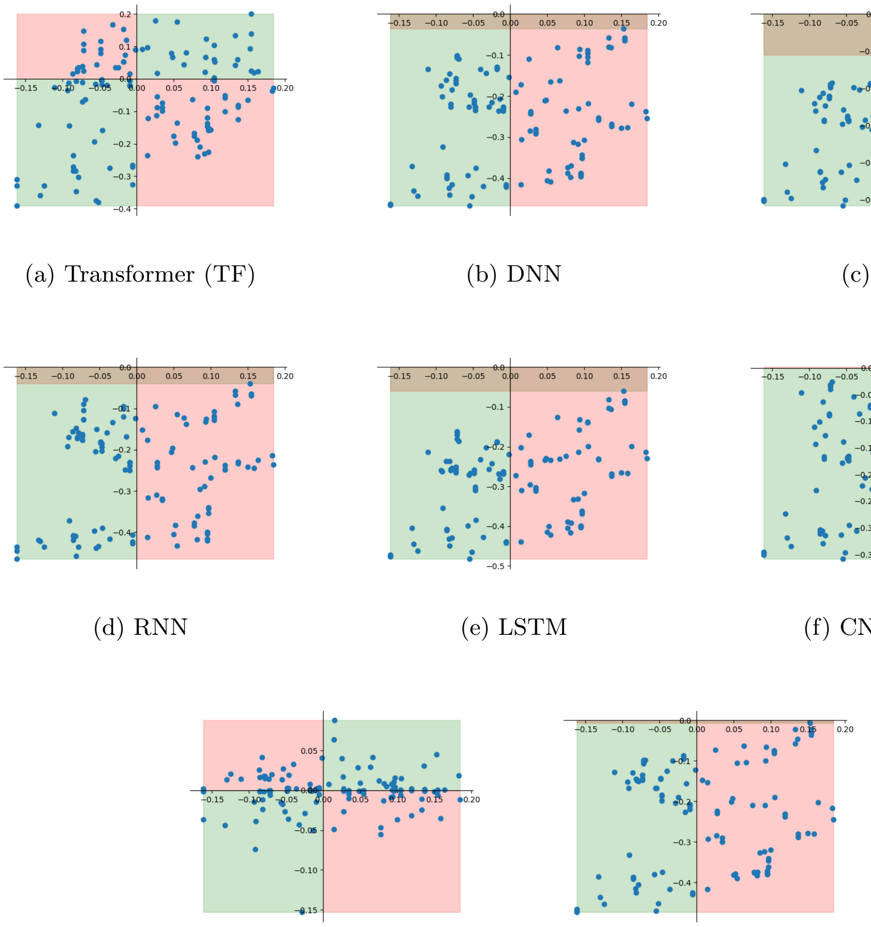
(e) LSTM

(f) CNN-LSTM

(g) ARIMA

(h) Linear (LR)

Change Correlation Plots



(a) Transformer (TF)

(b) DNN

(c) CNN

(d) RNN

(e) LSTM

(f) CNN-LSTM

(g) ARIMA

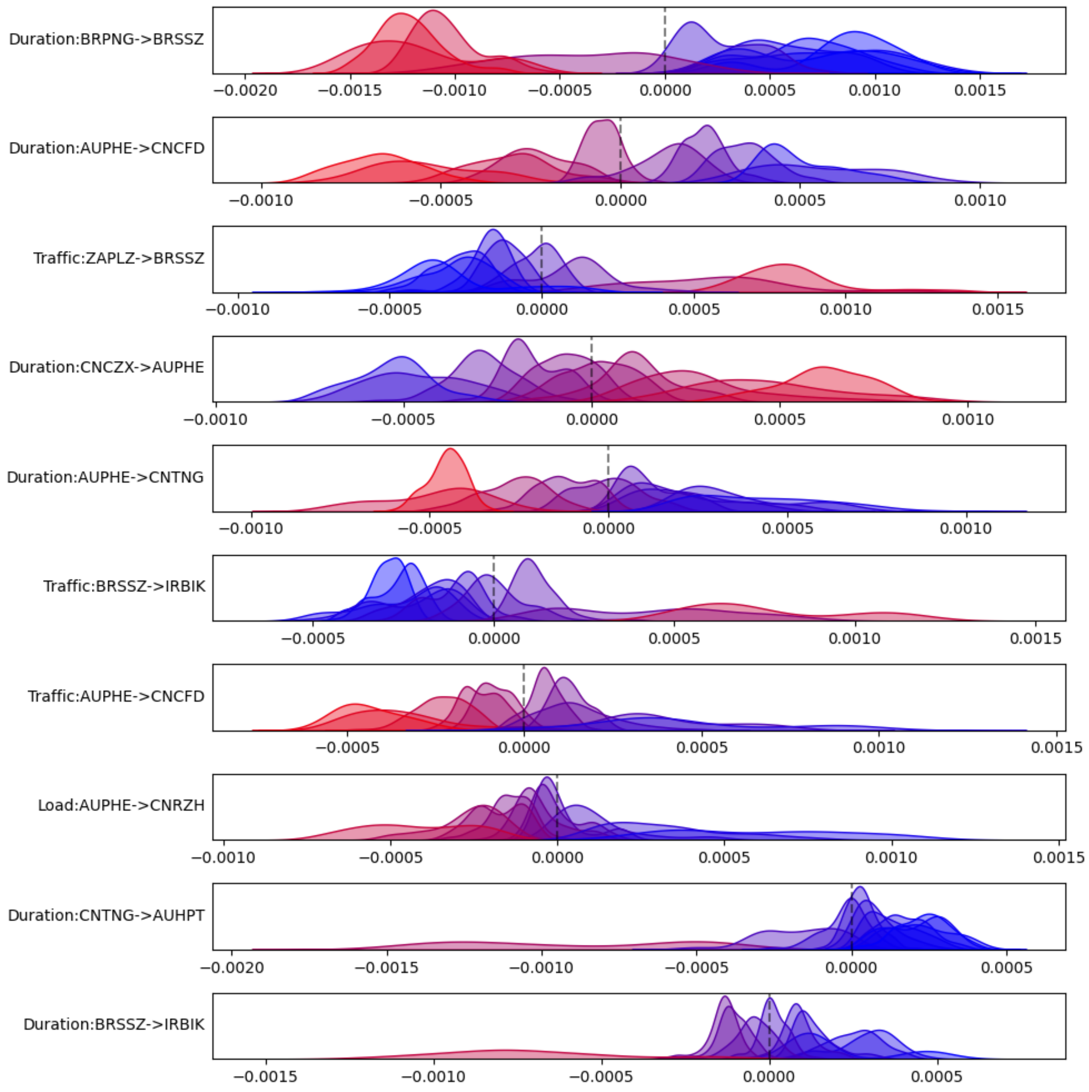
(h) Linear (LR)

J Model Explanations

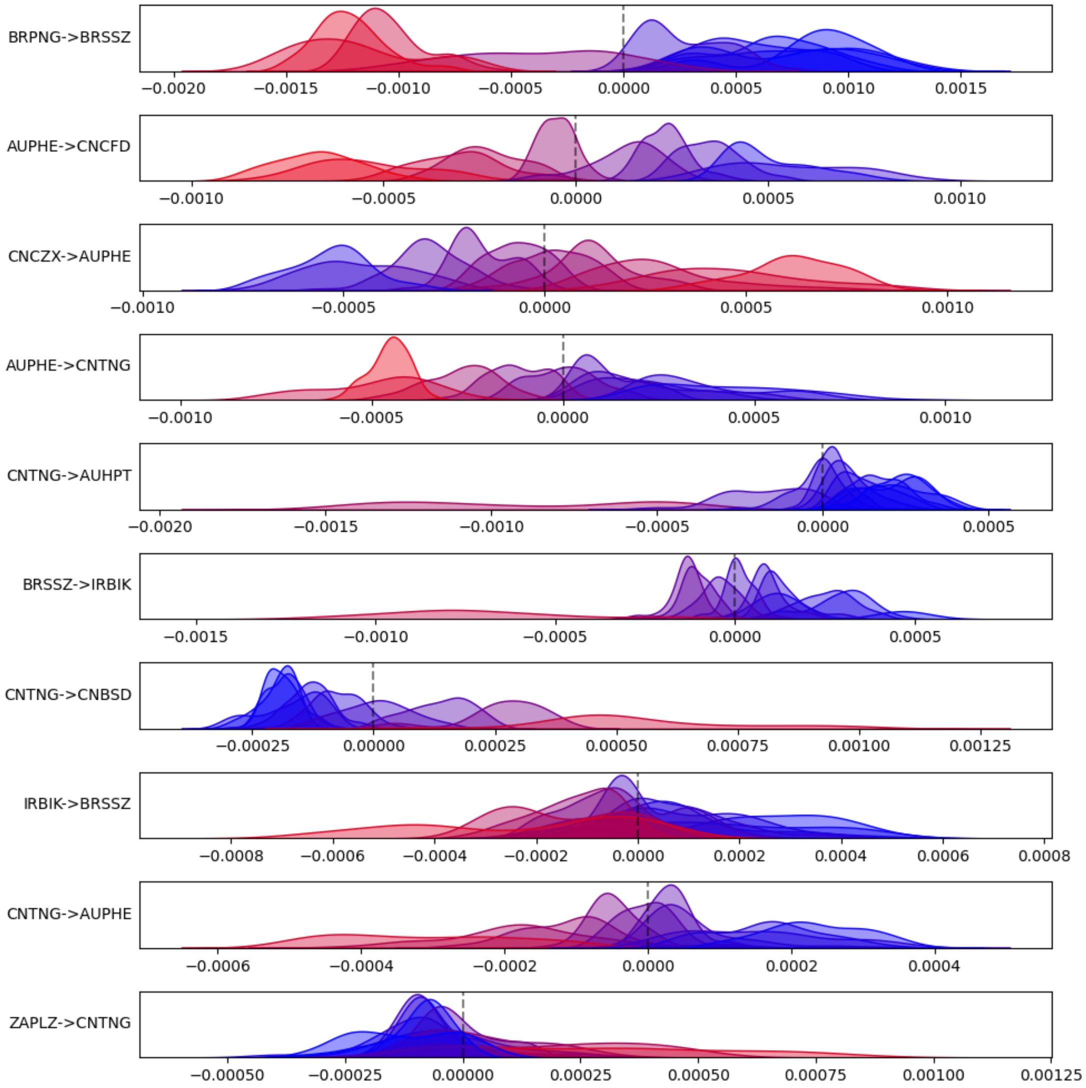
This appendix provides a supplementary set of relation-specific explanations pertaining to the best-performing model for the various financial instruments' train sets. The color red signifies higher feature values, whereas blue indicates lower feature values. The x-axis denotes the SHAP values, whereas the y-axis represents the distribution.

J.1 Experiment 1 - Baltic Dry Index (BDI)

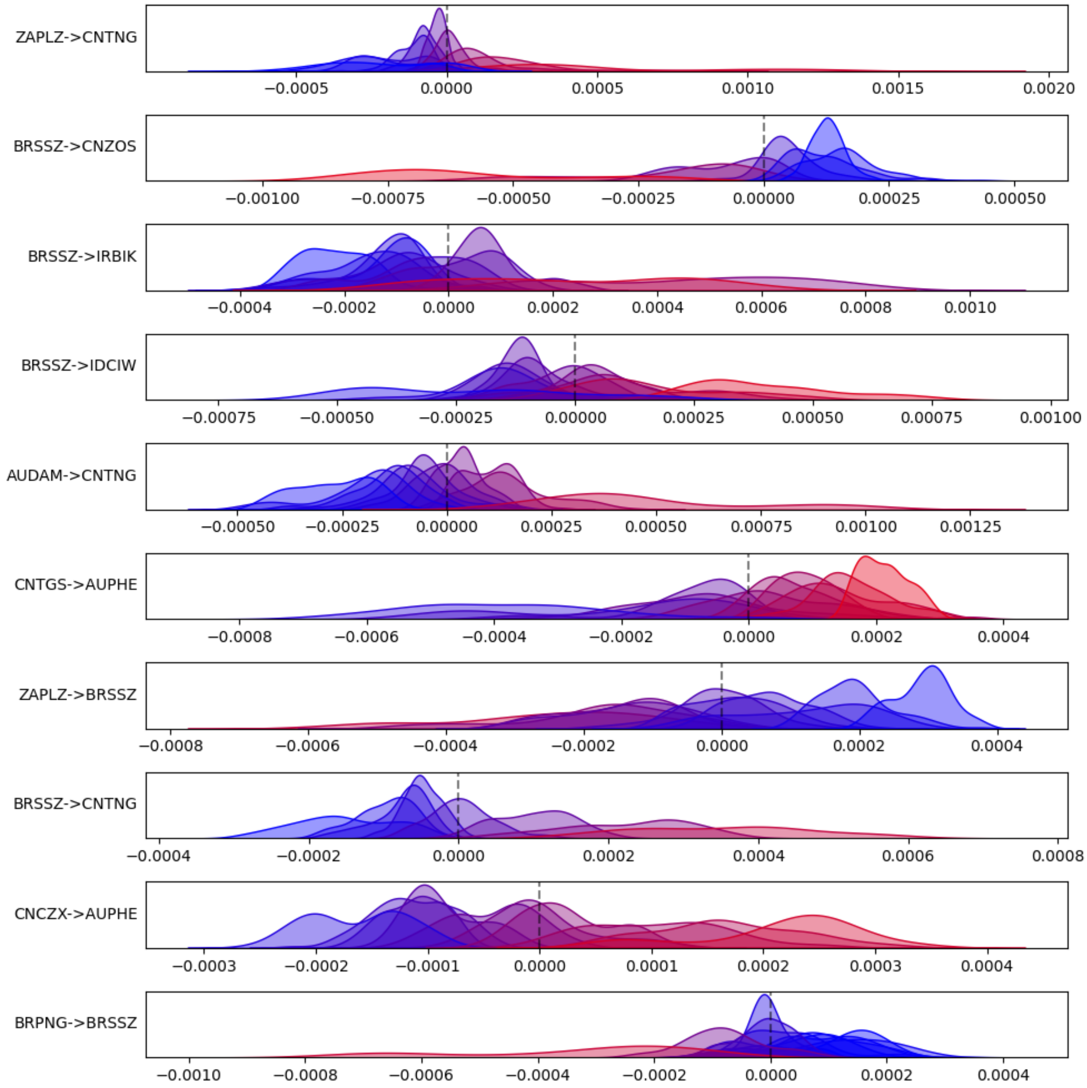
BDI - Top 10 Contributing Features by Importance



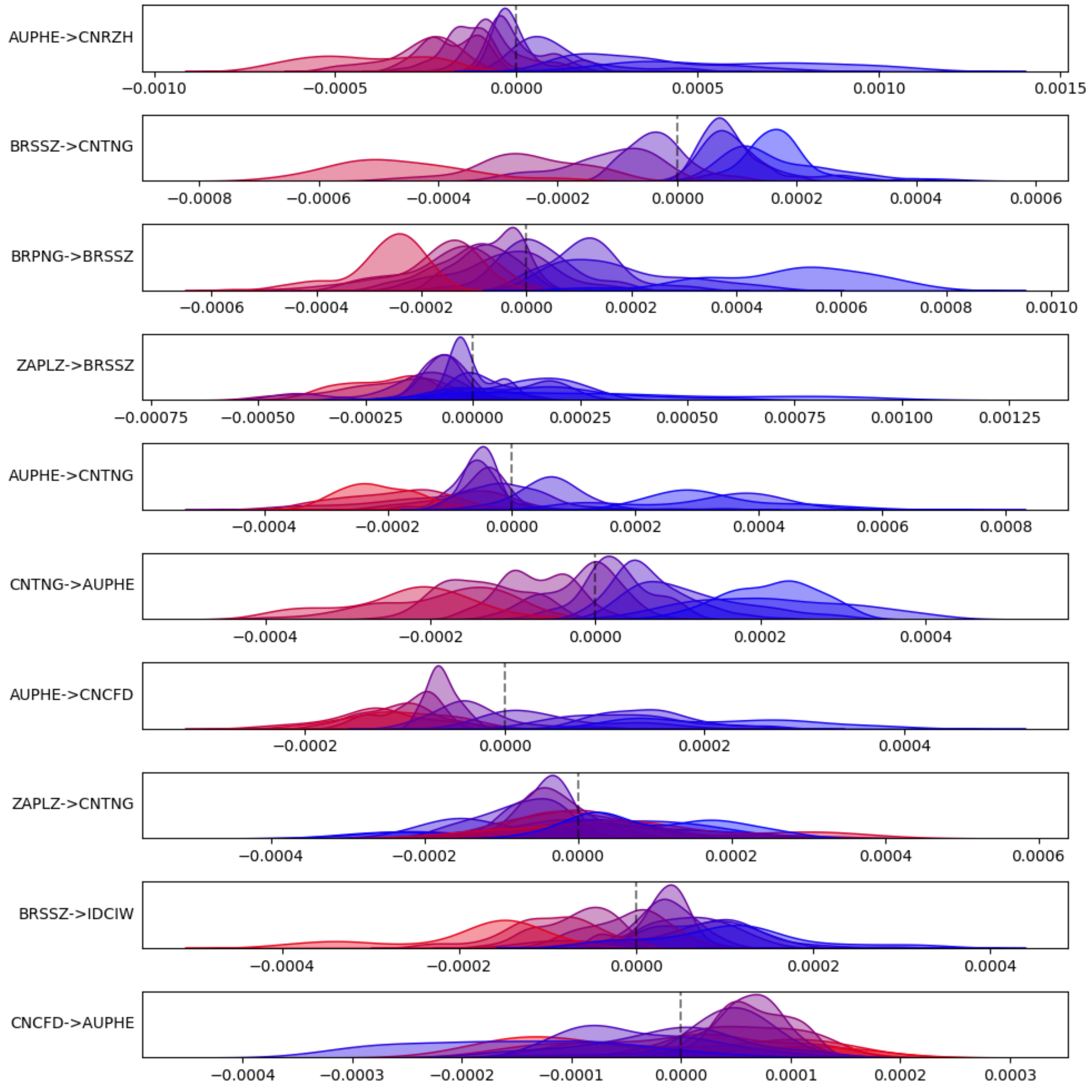
BDI - Top 10 Contributing Relations for Duration



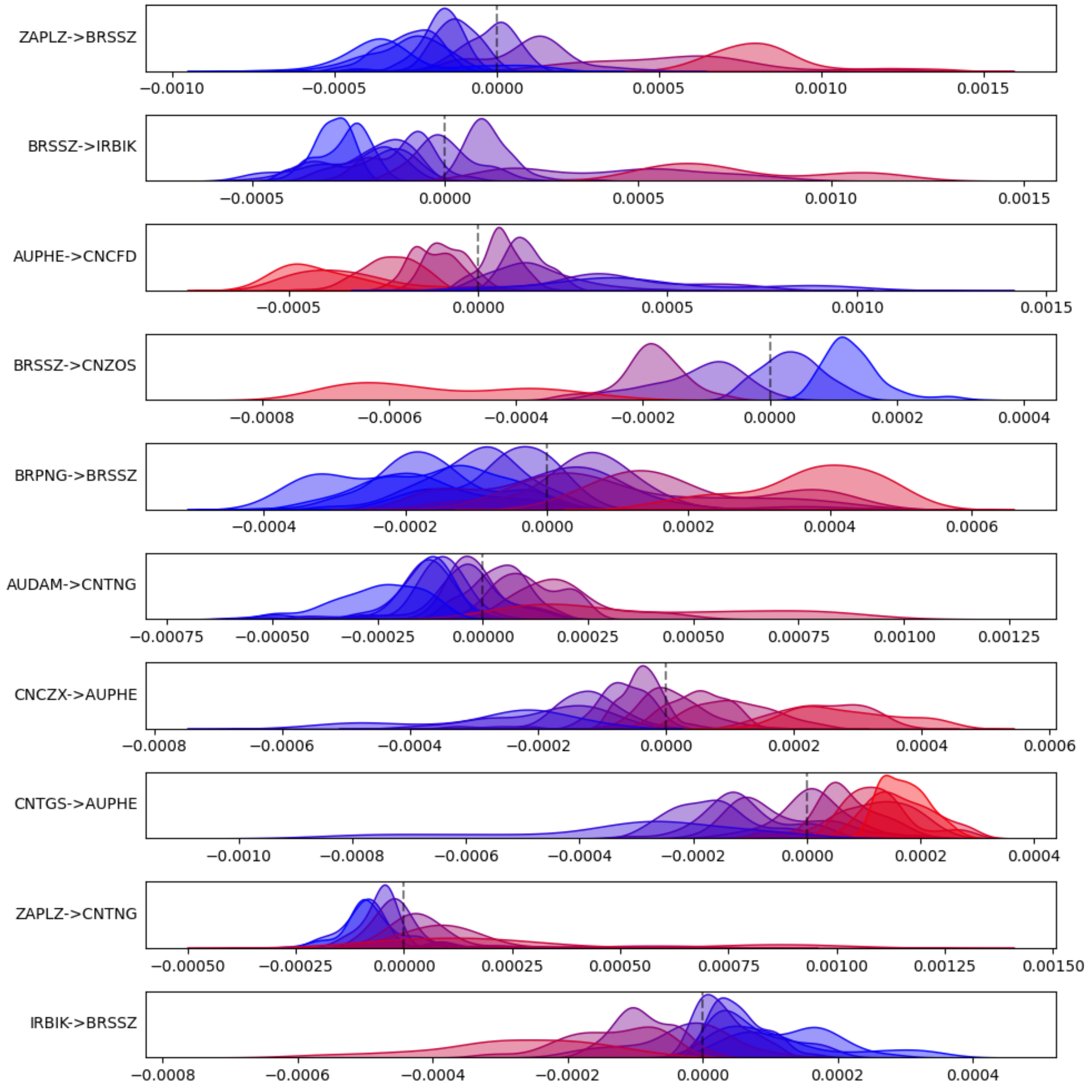
BDI - Top 10 Contributing Relations for Speed



BDI - Top 10 Contributing Relations for Load

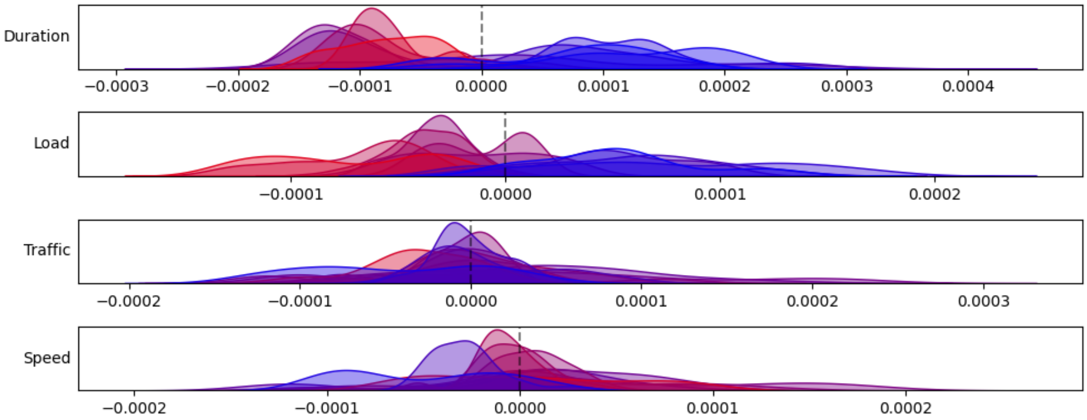


BDI - Top 10 Contributing Relations for Traffic



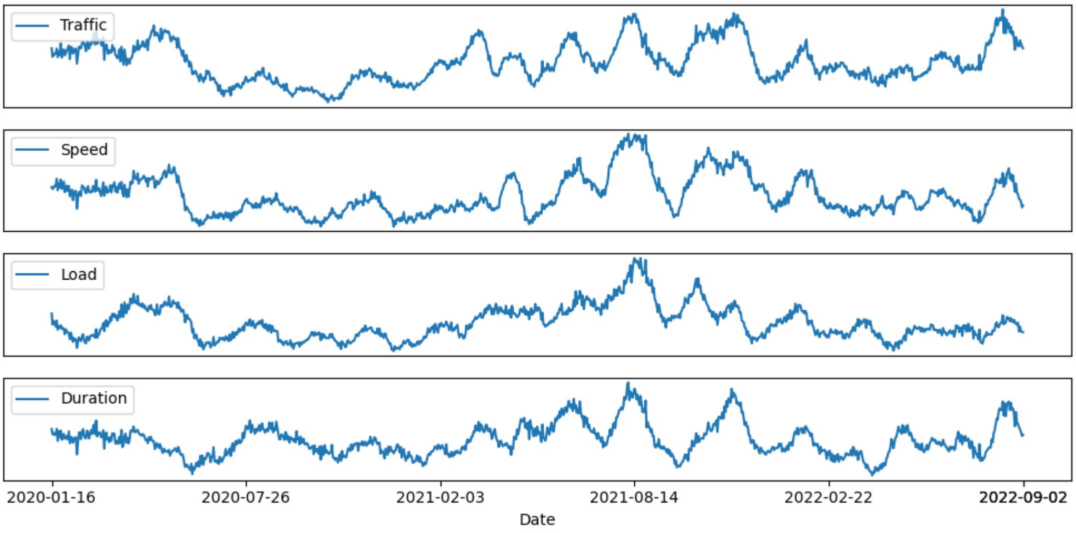
BDI - Shipping Variable Contribution

The SHAP values when aggregated across look-back periods, port relations, and the training dataset. Sorted by importance (most important first).



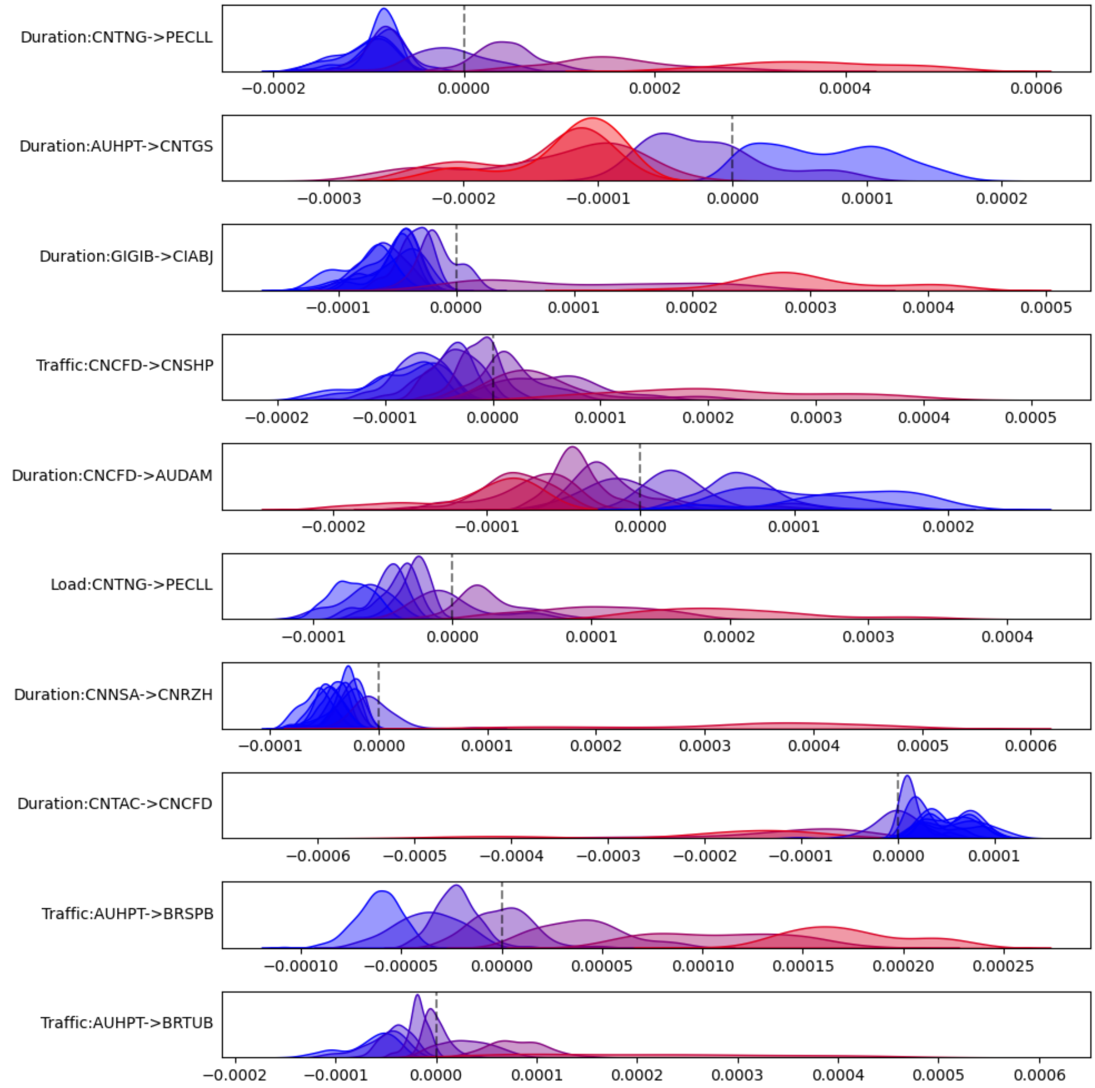
BDI - Shipping Variable Importance

The historical importance (absolute contribution) of each shipping variable when aggregated across look-back periods, port relations, and the training dataset.

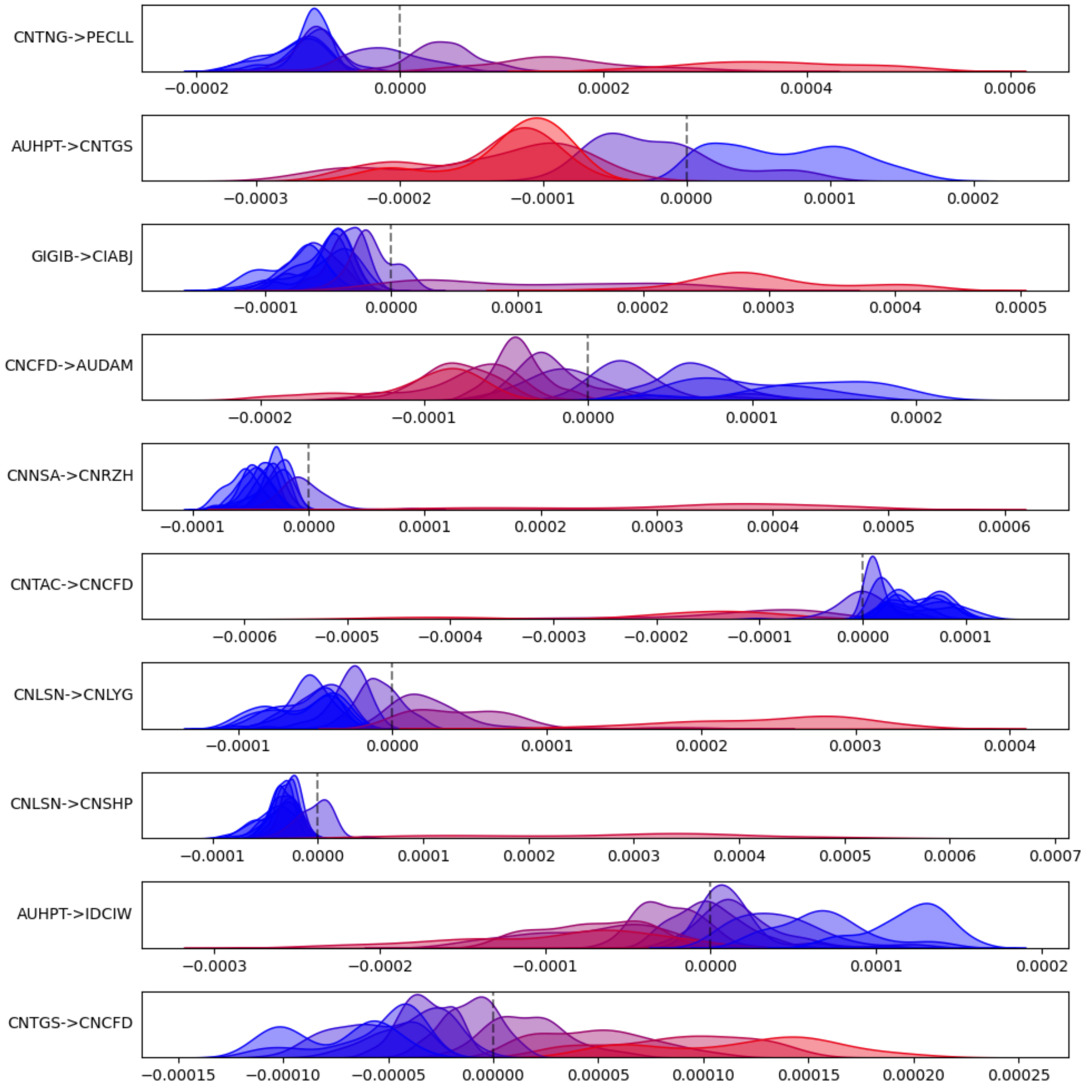


J.2 Experiment 2 - Breakwave Dry Bulk Shipping ETF (BDRY)

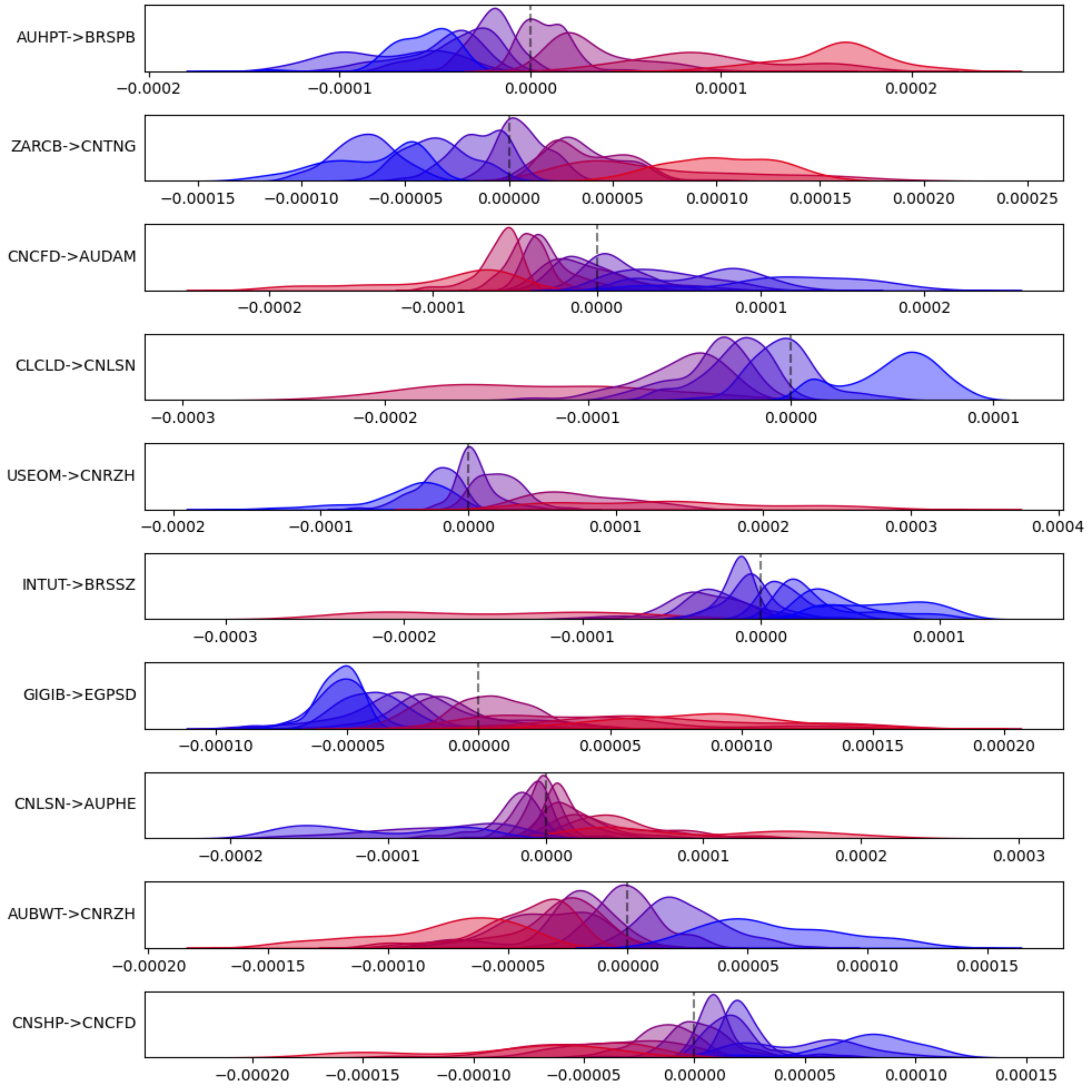
BDRY - Top 10 Contributing Features by Importance



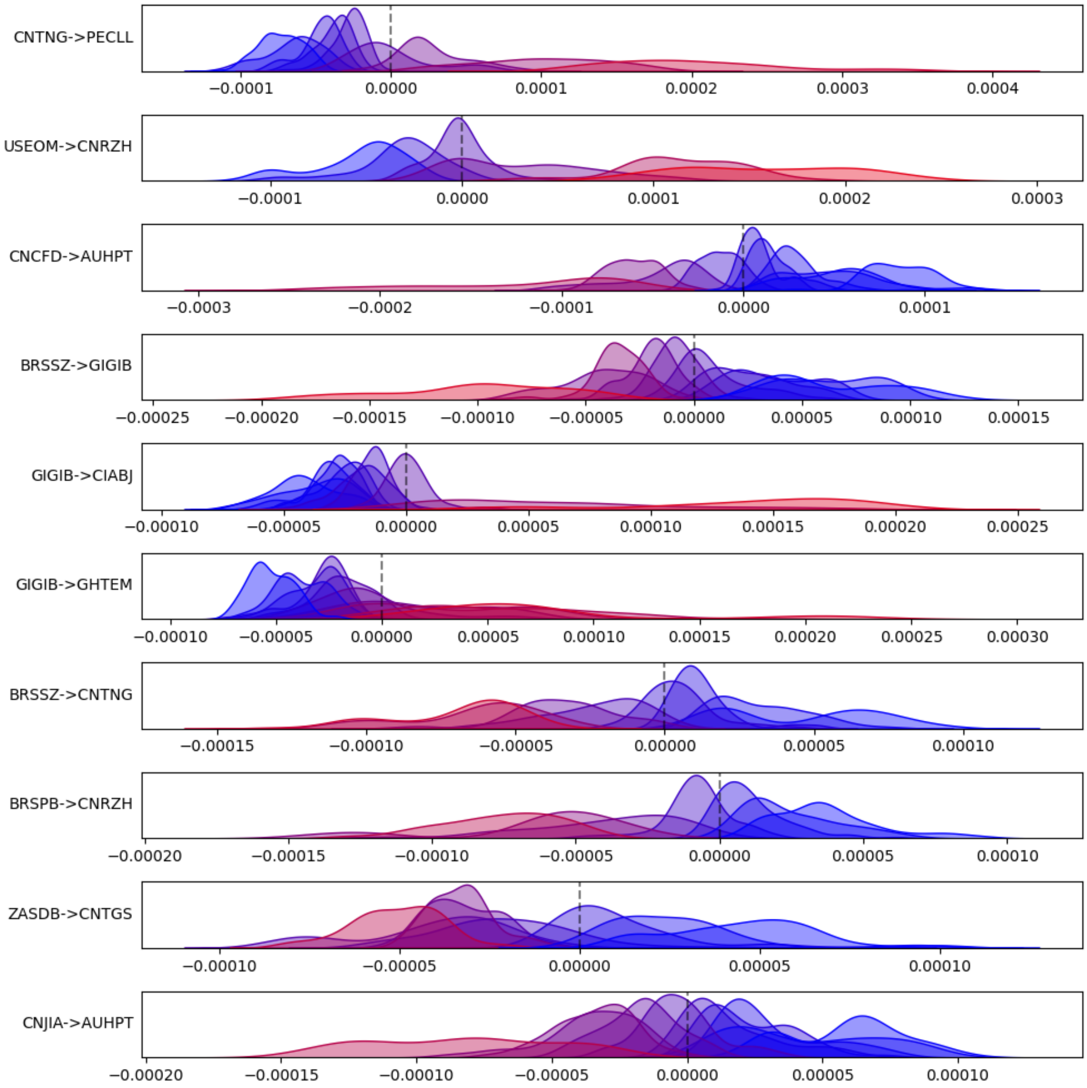
BDRY - Top 10 Contributing Relations for Duration



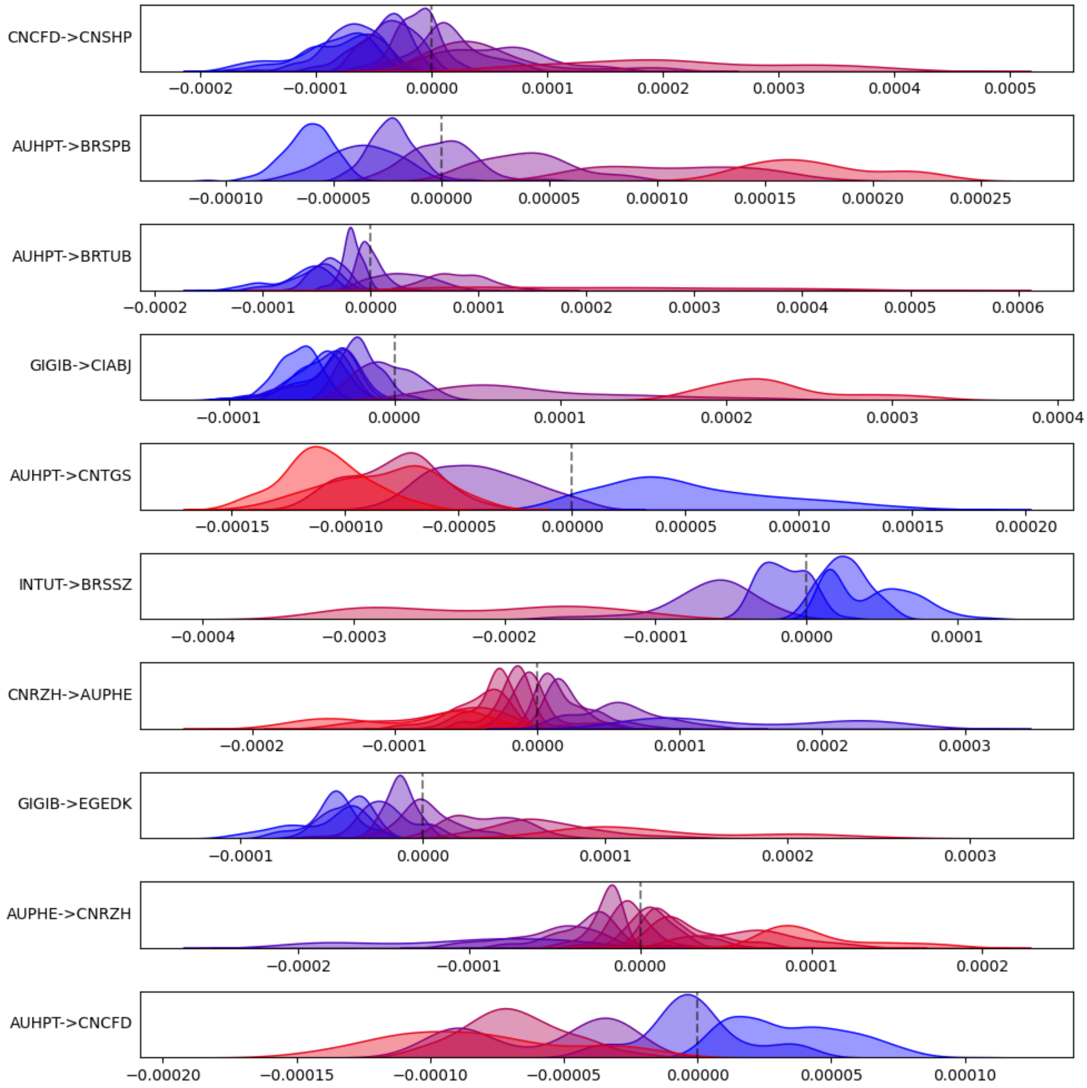
BDRY - Top 10 Contributing Relations for Speed



BDRY - Top 10 Contributing Relations for Load

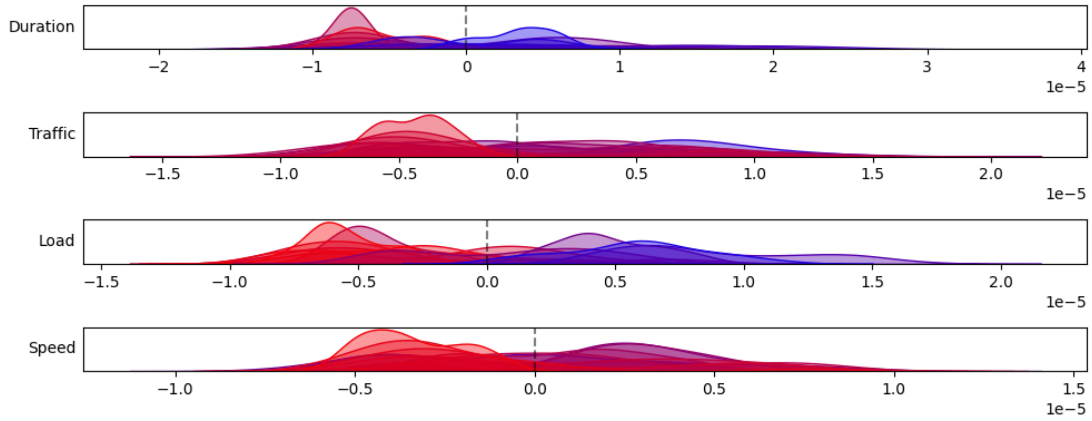


BDRY - Top 10 Contributing Relations for Traffic



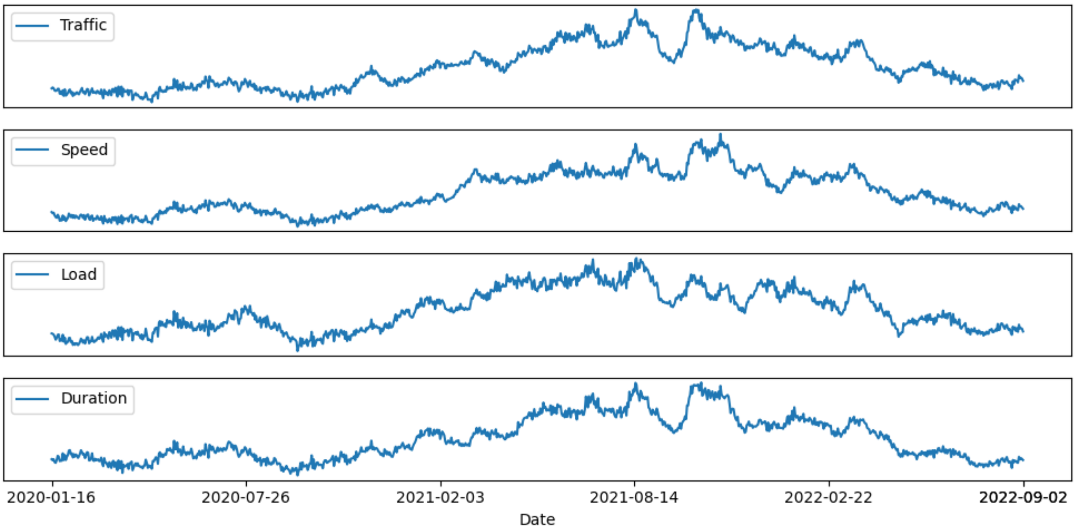
BDRY - Shipping Variable Contribution

The SHAP values when aggregated across look-back periods, port relations, and the training dataset. Sorted by importance (most important first).



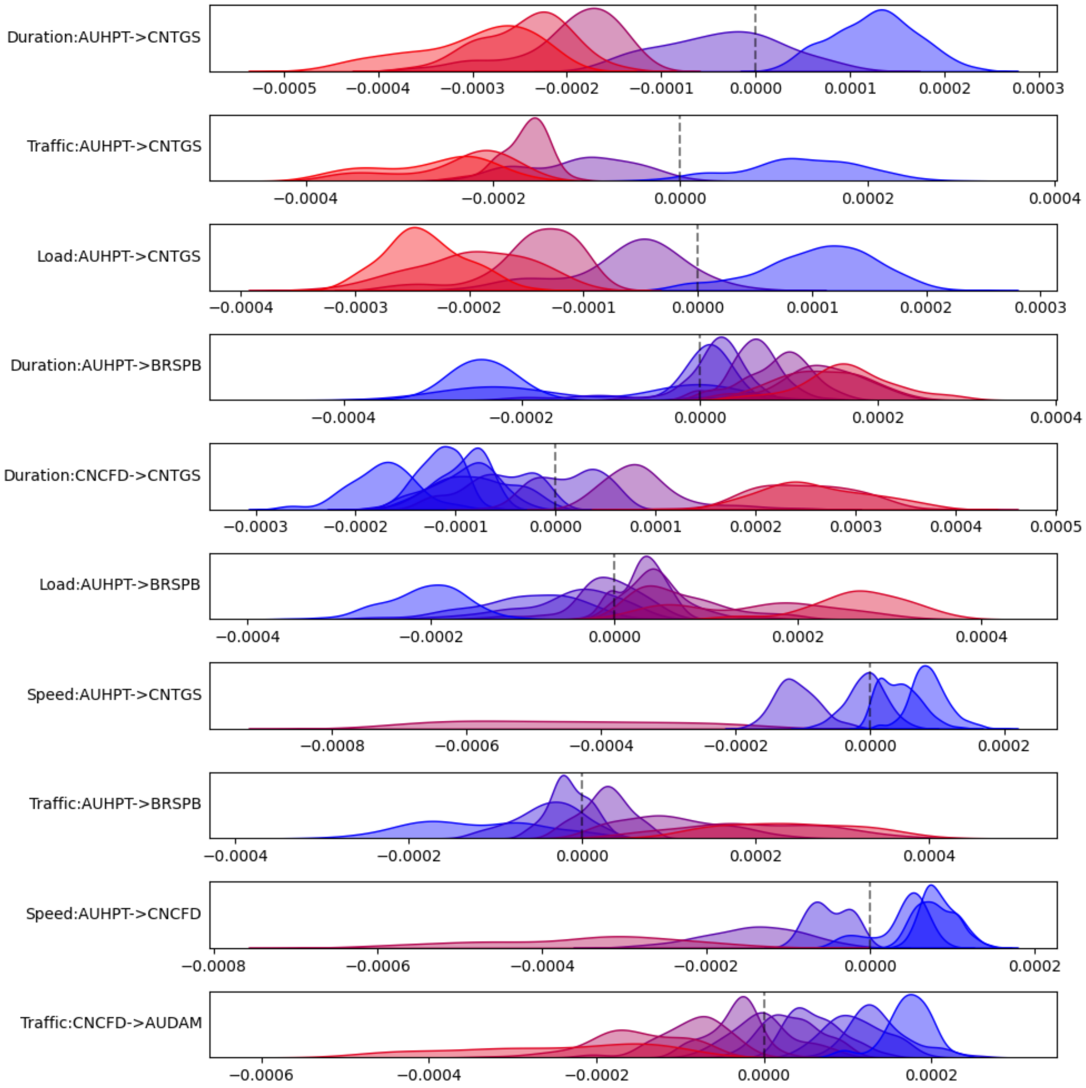
BDRY - Shipping Variable Importance

The historical importance (absolute contribution) of each shipping variable when aggregated across look-back periods, port relations, and the training dataset.

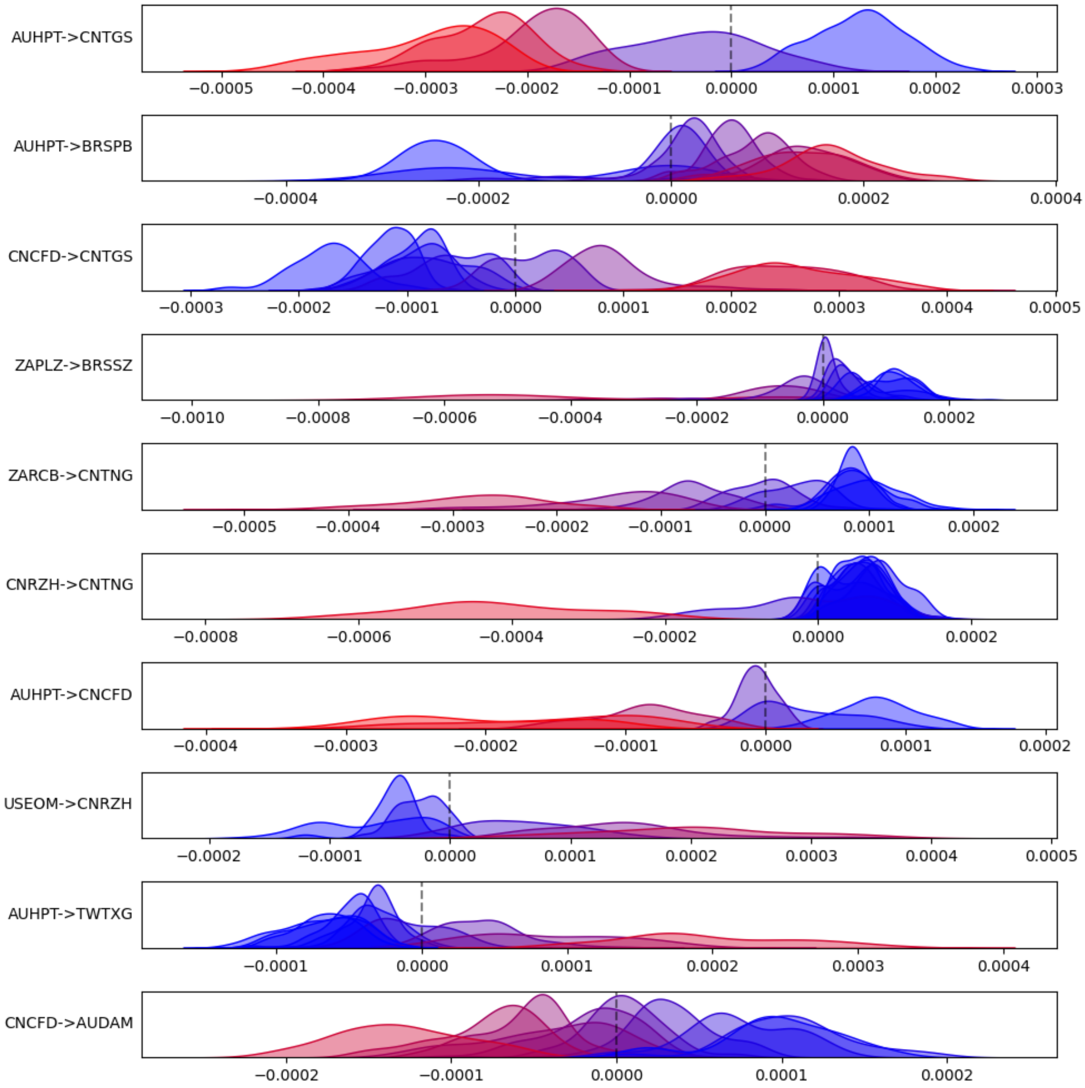


J.3 Experiment 3 - Golden Ocean Group Ltd. (GOGL)

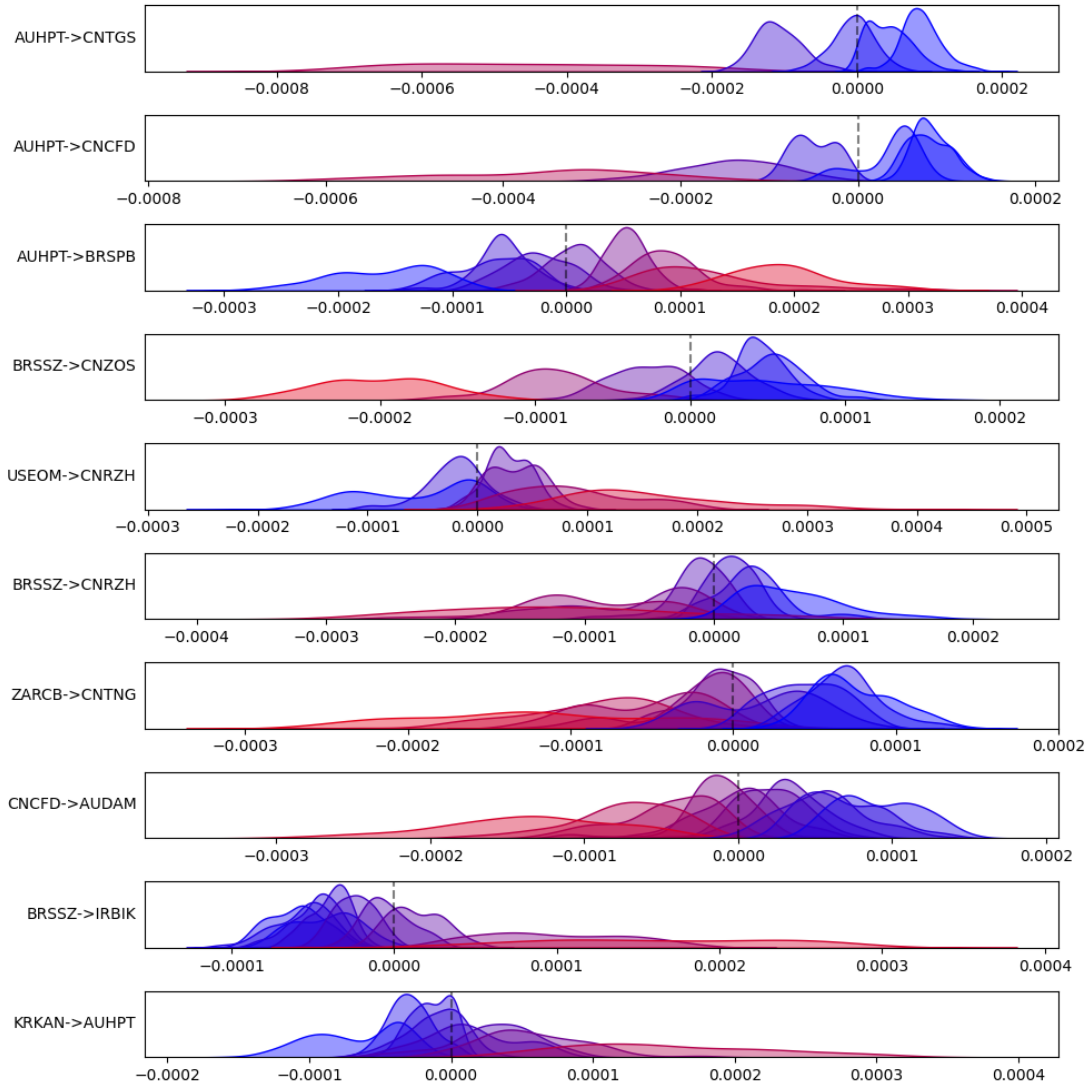
GOGL - Top 10 Contributing Features by Importance



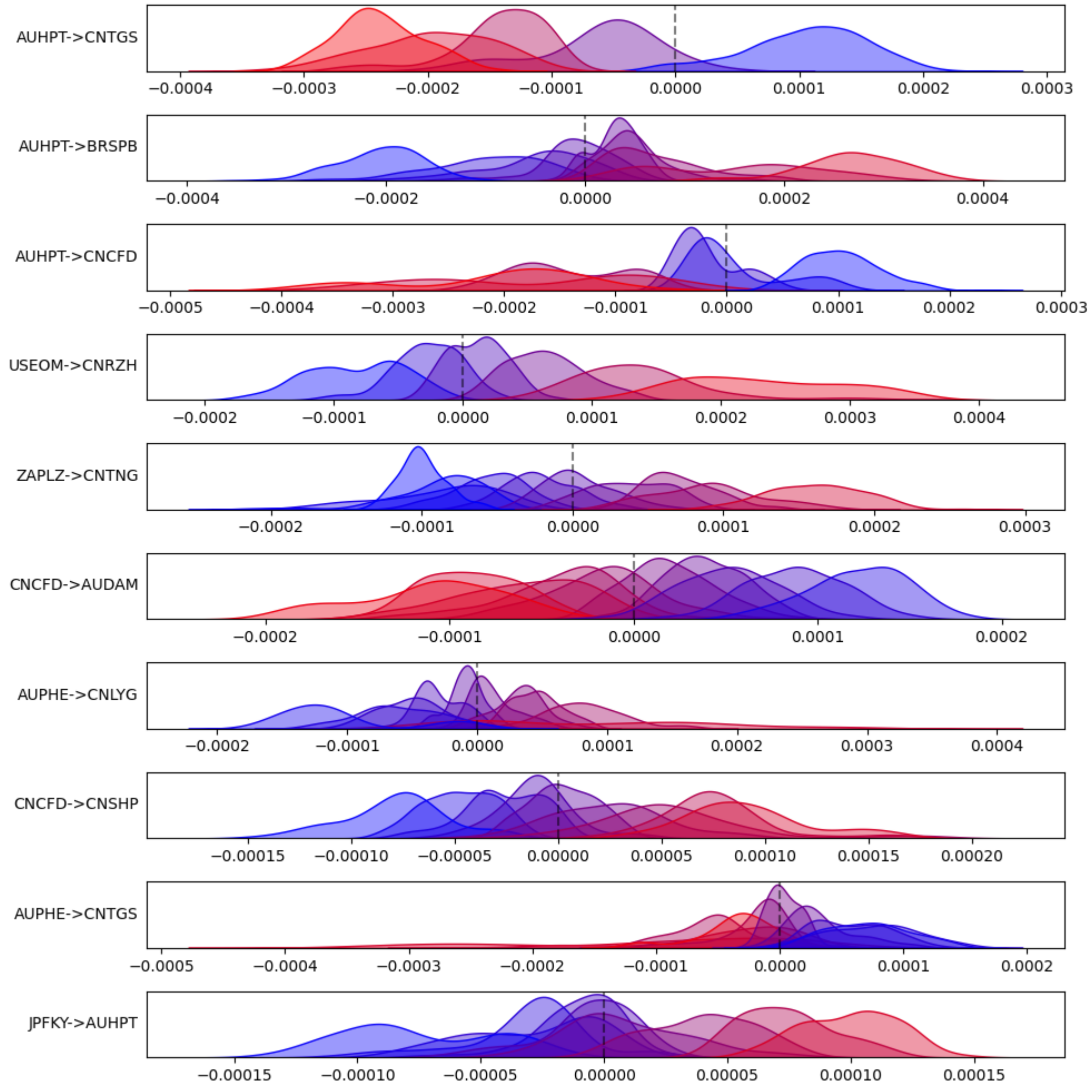
GOGL - Top 10 Contributing Relations for Duration



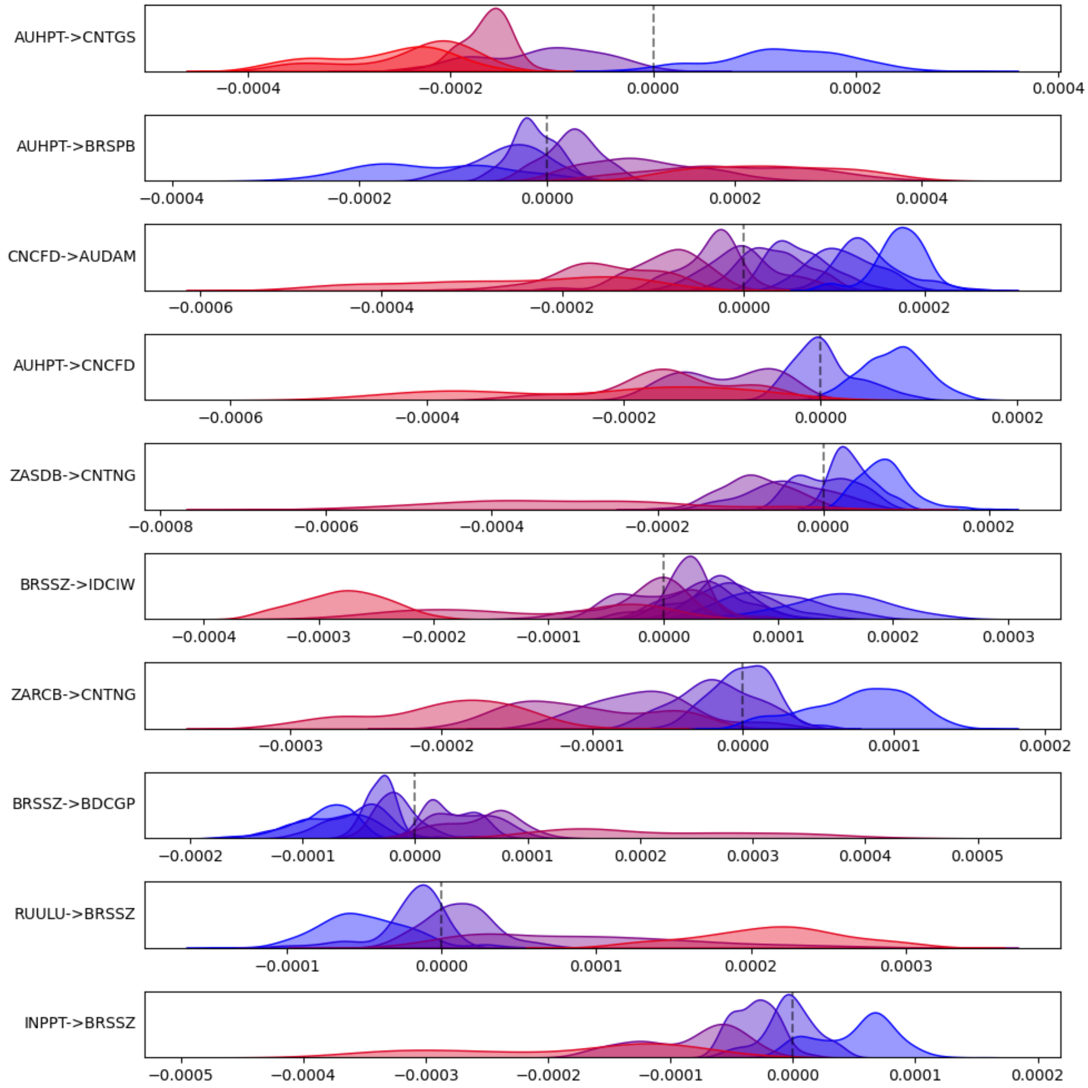
GOGL - Top 10 Contributing Relations for Speed



GOGL - Top 10 Contributing Relations for Load

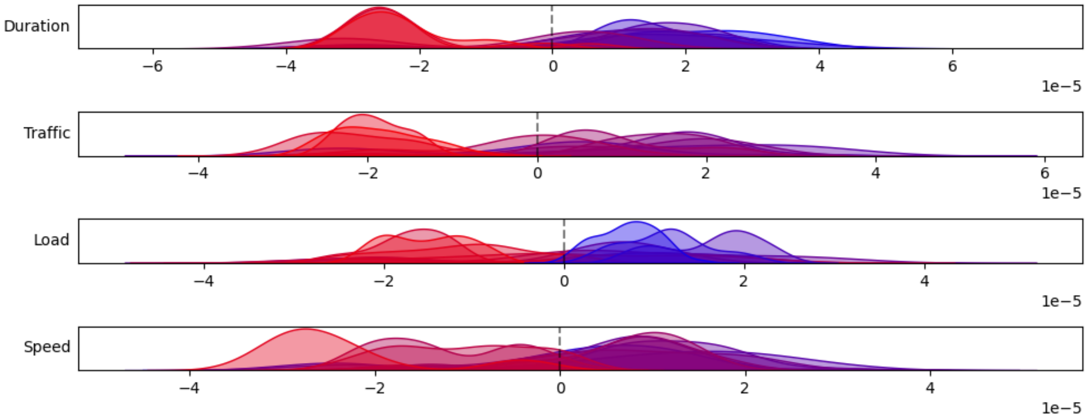


GOGL - Top 10 Contributing Relations for Traffic



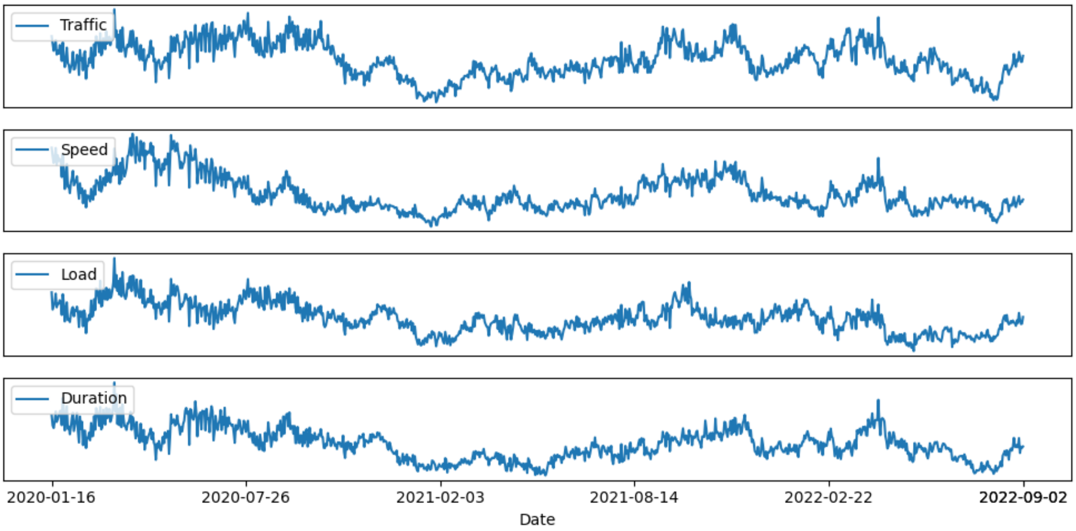
GOGL - Shipping Variable Contribution

The SHAP values when aggregated across look-back periods, port relations, and the training dataset. Sorted by importance (most important first).



GOGL - Shipping Variable Importance

The historical importance (absolute contribution) of each shipping variable when aggregated across look-back periods, port relations, and the training dataset.



K Ports and Locodes

This appendix provides easy access to some of the ports for which locodes are used in the explanations.

Locode	Port Name	Country
AUBWT	Burnie	Australia
AUDAM	Dampier	Australia
AUHPT	Hay Point	Australia
AUPHE	Port Hedland	Australia
BRPNG	Paranaguá	Brazil
BRSPD	Sao Pedro	Brazil
BRSSZ	Santos	Brazil
BRTUB	Tubarão	Brazil
CIABJ	Abidjan	Cote d'Ivoire
CLCLD	Caldera	Chile
CNBSD	Baoshan	China
CNCFD	Caofeidian	China
CNJIA	Jiangyin	China
CNLSN	Lanshan	China
CNLYG	Lianyungang	China
CNRZH	Rizhao	China
CNNSA	Nansha	China
CNSHP	Qinhuangdao	China
CNTAC	Taicang	China

Locode	Port Name	Country
CNTGS	Tangshan	China
CNTNG	Tianjin	China
CNCZX	Changzhou	China
CNZOS	Zhoushan	China
EGEDK	El Dekheila	Egypt
EGPSD	Port Said	Egypt
GHTEM	Tema	Ghana
GIGIB	Gibraltar	Gibraltar
IDCIW	Ciwandan	Indonesia
INTUT	Tuticorin	India
IRBIK	Tuticorin	Iran
JPFKY	Fukuyama	Japan
KRKAN	Gwangyang	South Korea
PECLL	Callao	Peru
TWTXG	Taichung	Taiwan
USEOM	Welcome	United States
ZAPLZ	Port Elizabeth	South Africa
ZARCB	Richards Bay	South Africa
ZASDB	Saldanha Bay	South Africa

NTNU

Norwegian University of
Science and Technology