GNN for asynchronous Spatio-Temporal Data Classification

Matthis, Leicht[1,*], Nadia Burkart[1], Rashmi Ambale Gopalakrishna[1] and Felix Bächle[1]

[1]Das Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung IOSB, Fraunhoferstraße 1, 76131 Karlsruhe, Germany

Abstract

This paper presents and explores a novel approach for solving classification tasks on asynchronous spatio-temporal data by encoding both spatial and temporal dependencies into a single graph structure. Time-series data are represented as chains of nodes connected by directed edges, capturing the sequential nature of the data. Additional edges are introduced to model interactions between time series, enabling the method to account for dependencies and relationships across multiple entities. This approach is applied to an example from the Maritime Situational Awareness domain, where the goal is to classify vessel types using tracks from the Automatic Identification System (AIS). Over the AIS maritime vessels transmit real-time information, including their position, speed, course, and general vessel details, for collision prevention and identification purposes. The method considers not only the movement patterns of individual vessels but also their interactions with other vessels and their tracks, which are critical for capturing complex maritime behaviors. This shall lay the basis for training Graph Neural Networks (GNN) with Graph Convolutions to perform classification tasks. This is the first step to model interactions and dependencies in spatio-temporal data, providing a promising framework for advancing classification tasks in dynamic, multi-entity systems.

Keywords

Spatio-Temporal Graph Neural Networks (STGNN), Time-series data representation, Maritime Situational Awareness

1. Introduction

This paper presents a novel approach to representing spatio-temporal data as directed graphs and using GNN for inference. This serves as the foundation for a framework to model interactions between correlated time series, aiming to be robust against data heterogeneity and asynchronicity. It demonstrates the approach using a use case from the maritime domain. For this purpose, an explicit graph structure is defined, a suitable GNN architectures discussed, and preliminary results presented. Based on these findings, outline directions for future research.

An example use case for demonstrating this framework originates from the field of Maritime Situational Awareness (MSA). It aims to support maritime surveillance operators who are tasked with identifying illegal or unsafe situations at sea. Due to the high volume of vessel traffic, maritime surveillance operators often face information overload. Automated systems that detect Situations of Interest (SoI) can assist them in their task. SoI could include illegal activities such as illegal fishing, smuggling, sabotage, or traffic violations. These situations are modeled using a variety of data sources, including vessel movement data, weather and oceanographic conditions, and information about stationary maritime infrastructure.

For this example, the focus lies on vessel movement patterns stemming from the *Automatic Identification System* [1]. This system allows maritime vessels to transmit kinematic information regularly, such as position, speed, course, and a timestamp, as well as static information like a unique identifier, ship type, size, and destination. This information is used for collision prevention and surveillance. The machine learning task will be to classify the ship type based on these movement patterns. The main focus is to create a test problem to evaluate the framework, but operational use is not implausible.

6th International Conference Recent Trends and Applications in Computer Science RTA-CSIT, May 22–24, 2025, Tirana, Albania

*Corresponding author.
matthis.leicht@iosb.fraunhofer.de (M. Leicht)
0009-0000-6657-8022 (M. Leicht); 0000-0001-7116-9338 (N. Burkart)

It is common for actors attempting to conceal their activities to modify their transmitted AIS data, a behavior known as *spoofing*. Thus, it is conceivable that a vessel disguises itself as another vessel of a different type. It is also plausible that information is not only present in the movement patterns of a single vessel but also in the combination of movement patterns of multiple vessels that operate in relative proximity and, therefore, can be expected to interact with each other. This highlights the potential of an approach that can model these interactions and make them accessible for machine learning methods.

2. Background and related Work

Methods for anomaly detection in the maritime domain, as described by Ribeiro et al., is a field that has seen numerous approaches. This includes data-driven models like DBSCAN and Artificial Neural Networks, as well as knowledge-based systems like Dynamic Bayesian Networks. Hybrid approaches that combine data-driven models with knowledge-based systems are also discussed. The survey highlights challenges associated with incomplete and inconsistently updated data.
In contrast, Graph Neural Networks and Spatio-Temporal Graph Neural Networks (STGNN) have attracted significant attention in recent research shown by Corradini et al.. STGNN are designed to operate on spatio-temporal graphs, which simultaneously encode spatial and temporal dependencies. Corradini et al. define a spatio-temporal graph as a structure that evolves over time in discrete steps. They used a graph structure to model spatial interactions and the sequential nature of the data. Such graphs are used as input for STGNN, which can be categorized into two types: those employing separate modules for spatial and temporal reasoning, and those processing the graph as a unified structure. STGNN have already been applied to vessel traffic prediction tasks in the maritime domain, as demonstrated in [4] and [5]. Zhang et al. created a Gated Spatio-Temporal Graph Aggregation Network, which uses GNN to capture spatial information from multiple vessels at a single time step. The output for multiple time steps is then used as input for a Multihead Attention Network for temporal aggregation. This is subsequently combined with a Temporal Convolution Network along with the output of a Ship Temporal Gating Encoder, which takes the time series of individual vessels as input to predict future vessel information. In contrast, Feng et al. defines a Spatio-Temporal Graph Convolutional Network based on a Graph Convolution in the spatial domain and a time convolution in the temporal domain.
This work focuses on encoding time series data into a unified graph structure that eliminates the need for separate spatial and temporal processing modules. This graph structure will utilize a definition for a spatio-temporal graph that relaxes the definition from Corradini et al.. This can be seen as an expansion of the work by Kapoor et al., where edges were added between nodes of successive subgraphs of a spatio-temporal graph.

3. Methodology

In this section the base idea for a graph structure and a possible architecture is outlined.

3.1. Spatio-temporal graph for vessel movement

Definition 1. *As a basis for modeling vessel movements and interactions, a directed inhomogeneous graph is defined as follows. Let*

- V *be a set of nodes,*
- $E_{intra}, E_{inter} \subseteq (x, y) \in V \times V | x \neq y$ *be two sets of directed edges of types* ***intra*** *and* ***inter****,*
- *and* $A \in \mathbb{R}^{|V| \times N_V}$ *denote the node-attribute matrix.*

Here, N_V *denotes the number of attributes of a node. Let* $G = (V, E_{intra}, E_{inter}, A)$ *be the directed inhomogeneous graph with node attributes.*

A graph to model vessel movement and ship interaction will be constructed based on the definition above. For every AIS message, a node is added to the graph. The node-attribute matrix will be of the form $A = [X, Y]$. This matrix A contains the kinematic features (e.g., speed, course, position) in X, while Y represents the vessel type, e. g. tanker, cargo ship, fishing vessel or passenger ship. Thus, for every node $v \in V$, A has a corresponding row vector $A_v = (X_v, Y_v)$, where X_v holds the kinematic data and the timestamp from the corresponding AIS message, and Y_v indicates the ship type as the target.
Next, the connectivity of the graph is established. The motivation is to connect nodes of the graph to model movement patterns for individual vessels, as well as for inter-vessel interactions when they are in proximity to each other. Therefore, this is a two-step process.

1. A consecutive sequence of a single vessel is a time series. To convert the time series into a graph structure, let $V_s \subseteq V$ be the set of nodes belonging to a specific vessel s. Obviously, V_s has a natural order defined by the timestamps t_v for all $v \in V_s$, and therefore a sequence of nodes exists: $(v_1, \dots, v_{|V_s|})$ with $v_n \in V_s$. Under the assumption that there are no two messages for one vessel at the same time, let this sequence now be connected pairwise following their order, such that $e_n = (v_n, v_{n+1}) \in E_{intra}, \forall n \in 1, \dots, |V_s|$.
   The directed structure preserves the time series nature of the data, and if this process is done for each vessel, G now holds the information for all these vessels.
2. To encode the interactions between vessels, a second step is necessary. Under the assumption that vessels interact with each other in some way, such as through collision avoidance, it makes sense to base the connection on spatial and temporal proximity between two vessels. Therefore, let δ be the time threshold and ϵ the spatial threshold. As the position is given in coordinates, the Haversine distance is used as a measure of spatial proximity. $D(x, y)$ denotes the Haversine distance, which is the angular distance on a great circle between two points on a sphere, given by

$$D(x, y) = 2 \arcsin \sqrt{\sin^2\left(\frac{(x_{lat} - y_{lat})}{2}\right) + \cos x_{lat} \cos y_{lat} \sin^2\left(\frac{(x_{lon} - y_{lon})}{2}\right)}.$$

   To approximate the real distance between two points, the formula

$$d = D(x, y) * r$$

   can be used, where r is the mean radius of the Earth. With these preconditions, the edges that encode inter-vessel interactions are added according to the following rule. For every $v \in V$, an edge $e = (w, v)$ will be added to E_{inter} when:

$$s_v \neq s_w,\ t_w > t_v,\ t_w = \max_{v_i \in s_w \wedge t_i < t_v} t_i,\ t_w - t_v < \delta,\ D(v, w) < \epsilon.$$

   Here, s_v and s_w denote the vessels to which v and w belong. In words, a node w is connected to a reference point v when it belongs to a different vessel than v, is the last node of w before v, and is within the limits of the spatial and temporal thresholds. This choice was made to ensure that connections always point from a node earlier in time to a node later in time. Additionally, the connectivity between the subgraphs of two vessels is limited.

A possible resulting graph for a vessel dataset is visualized in fig. 1 and fig. 2. Two vessels are part of this graph, distinguishable by color. Figure 1 shows the positional component of the graph, which means the position of the nodes in a 2D plane, while fig. 2 depicts the arrangement in time. As shown, the nodes of a vessel form a chain connected by inter-vessel edges. When two vessels come into close proximity to each other, additional interaction edges between the chains are shown in red. What this achieves is placing the data in the context of other points that might influence a decision made based on this data. On one hand, data stemming from one vessel is contextualized with its own past and future. On the other hand, the interaction edges expand the context to data from other vessels and their past and future data points. This is combined in one data structure, which can be used by specialized methods. This method is also robust against the number of present vessels, asynchronicity, and sampling rate, and to some degree, missing data points, which is another benefit.

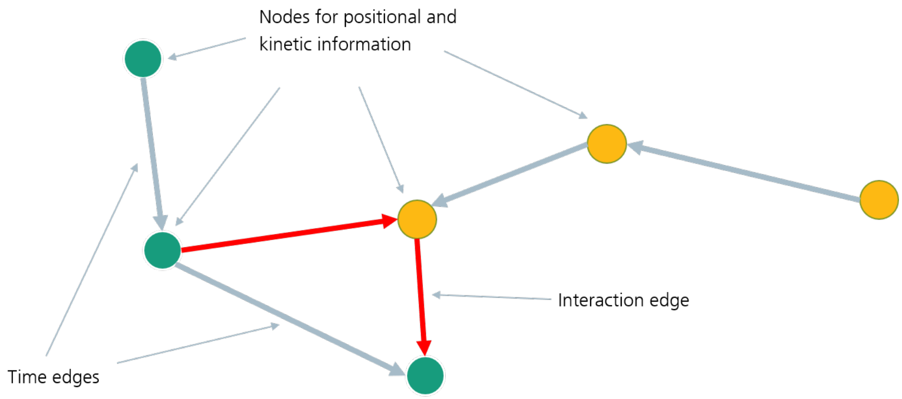

Figure 1: Spatial visualization of the spatio-temporal graph.

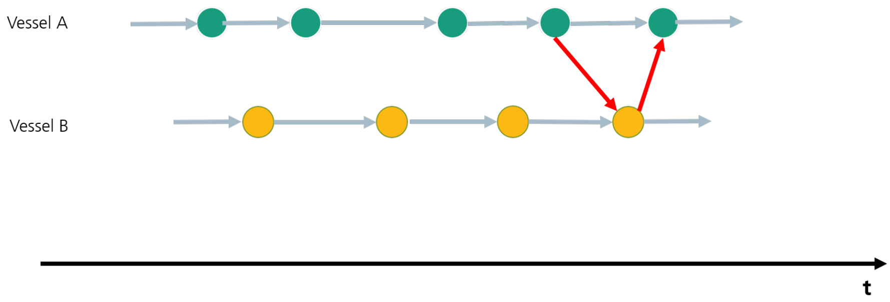

Figure 2: Temporal visualization of the spatio-temporal graph.

3.2. Model architecture

Message Passing Layers for Graph Neural Networks conduct an update step on graphs, for example, for each node. They utilize information not only from their own node attributes but also from adjacent nodes. This makes these models suitable for solving problems on the graphs created above. For directed graphs, Rossi et al. have developed a method that works on this type of graph. The core idea of a Message Passing Layer for undirected graphs is to update its activation using a combination of its own features along with an aggregation of those from its direct neighbors. Both the combination and aggregation functions are learnable. The paper expands this idea to directed graphs by using separate aggregations that distinguish the influence of nodes based on the direction of the edges between them. An abstraction for Message Passing Layers for directed graphs is given by

$$m_{v,\leftarrow}^{(l)} = AGG_{\leftarrow}^{(l)}\left(\{(x_w^{(l-1)}, x_v^{(l-1)}) : (w,v) \in E\}\right),$$
$$m_{v,\rightarrow}^{(l)} = AGG_{\rightarrow}^{(l)}\left(\{(x_v^{(l-1)}, x_w^{(l-1)}) : (v,w) \in E\}\right),$$
$$x_v^{(l)} = COM^{(l)}\left(x_v^{(l-1)}, m_{v,\leftarrow}^{(l)}, m_{v,\rightarrow}^{(l)}\right).$$

Hereby $v \in V$ mark the node, which is updated, $AGG_{\leftarrow}^{(l)}$ and $AGG_{\rightarrow}^{(l)}$ denote the in- and outgoing aggregation mappings and $COM^{(l)}$ the combination mapping. Directed Graph Convolution Networks (Dir-GCN) were introduced in Rossi et al. as an extension of Graph Convolutional Layers from Kipf and Welling for directed graphs. The propagation rule is given by

$$H^{(l)} = \sigma\left(S_{\rightarrow}H^{(l-1)}W_{\rightarrow}^{(l)} + S_{\rightarrow}^{T}H^{(l-1)}W_{\leftarrow}^{(l)}\right).$$

Here, $H^{(l)}$ represents the node activations of the l-th layer, $W_{\rightarrow}^{(l)}$ and $W_{\leftarrow}^{(l)}$ are trainable channel-mixing maps for nodes connected by ingoing and outgoing edges, respectively, and σ is a point-wise nonlinearity.

The normalized message passing function $S_{\rightarrow}$ is given by

$$S_{\rightarrow} = D_{\rightarrow}^{-1\frac{1}{2}} E D_{\leftarrow}^{-1\frac{1}{2}},$$

where E is the adjacency matrix of the input graph, and $D_{\rightarrow}$ and $D_{\leftarrow}$ are the out-degree and in-degree matrices of the input graph, respectively. The update function can be implemented in such a way that it places more emphasis on information received from nodes connected by incoming or outgoing edges. As the layers are single-hop layers, this means that in one layer, information is only passed on to directly neighboring nodes. This makes it clear that the layer depth is connected to the time horizon the model considers when applied to spatio-temporal graphs, as defined in sec. 3.1. Another aspect of applying these layers to these graphs is the different handling of incoming and outgoing connections. This can be interpreted as information being processed differently in classification depending on whether it originates from before or after a certain point in time.

4. First experiments and results

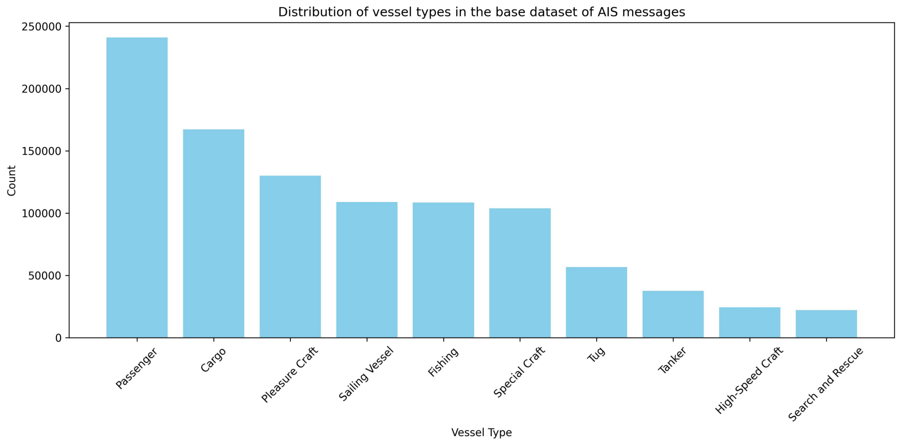

Figure 3: Distribution of AIS messages over vessel classes.

To evaluate the method, a straightforward experiment was designed using a Dir-GCN model with a layer depth of 20. As model size is connected to the considered time horizon, the data has to be prepared accordingly. To generate the model input data, 1 million consecutive AIS messages from the bay near Amsterdam, the *IJsselmeer*, and the *Markermeer* were used, stemming from the AIS-hub network [9]. As can be observed in Fig. 3, the overall dataset is biased regarding vessel classes. In particular, the class "Passenger Craft" dominates. The dataset is partitioned in a $60\%/20\%/20\%$ ratio without shuffling to create the training, test, and validation datasets, thus preventing overlap between the sets.
The next step is the creation of base graphs following the concept presented in Sec. 3.1, each for the train, test, and validation datasets. The chosen parameters defining the connectivity of the resulting graph were defined as $\delta = 300s$ and $\epsilon = 1km$. In Tab. 1, base metrics of the training base graph are provided. The high number of edges in relation to the number of nodes and the average in- and out-degree shows that a high number of *inter*-vessel edges must exist in the graph. As each node is only connected to at most one node of the same vessel before and after it, the rest must stem from edges to nodes of other vessels. This is also reflected in the average clustering coefficient, which shows relatively high local connectivity. On a global scale, the graph is very sparsely connected, which can be explained by the limitations on the introduction of edges between nodes imposed by parameters δ and ϵ. From these base graphs, subgraphs are sampled as input for training. To maximize the information for the model, neighbors up to the order of 20 are sampled from the base graph, moving backwards against the flow of the graph concerning a reference node. This process is applied to all nodes in the

Metric	Value
Number of nodes	600,000
Number of edges	9,206,039
Average in-degree	15.3434
Average out-degree	15.3434
Density	2.56×10^{-5}
Clustering coefficient	0.5726

Table 1
Graph metrics summary.

base set. The resulting subgraphs are filtered to retain those containing more than 18 nodes and are normalized at the graph level as described in Chen et al.. Variance is added to each node as additional features to provide the model with information about the spatial extension of the graph. As the target, the class of the last node is used. It is important to note that the model for this trial ignored the *intra* and *inter* edge types for classification. As an implementation framework, the Python library PyTorch Geometric from Fey and Lenssen was used.
A test run using micro batches of size 128 demonstrated almost immediate convergence within one epoch. While a deeper examination of convergence across the mini batches was not conducted, the accuracy of predictions was approximately ~ 0.3. This is a better result than random choosing, but as shown in Fig. 4, the classifier simply fits to the most common class. Therefore, this first experiment was not successful as the performance is not result of the models ability to detect patterns in the graph structure but roots from the bias of the data.

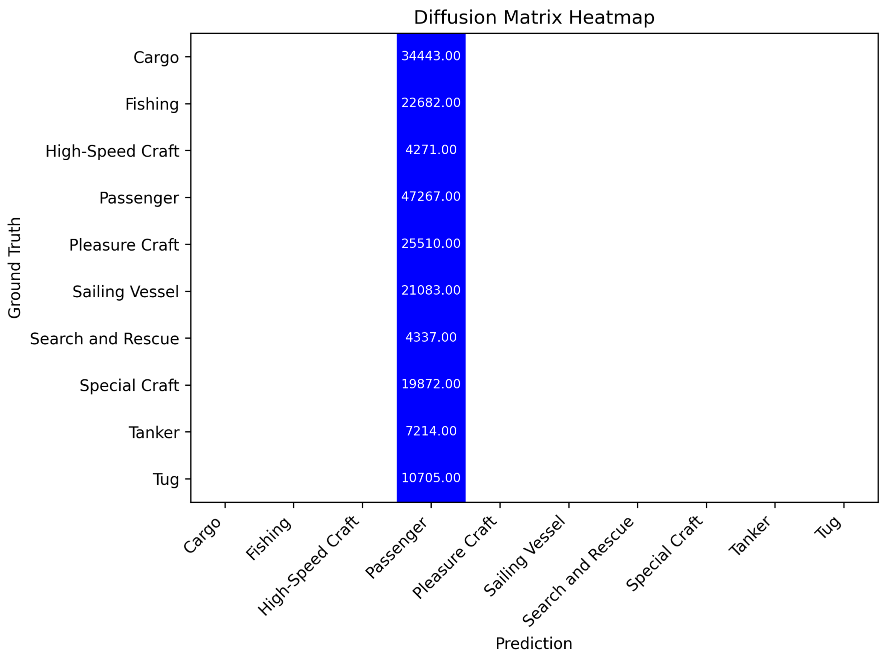

Figure 4: Heatmap of the GCN-classifier.

5. Conclusion

The first tests are indicating several problems with the current approach:

- The current model doesn't take edge types into account, so it can't distinguish between different vessels in a subgraph. This might lead to subpar performance, especially in combination with the next point.

- The current method for dataset creation and sampling leads to a high degree of overlap between different subgraphs. This is a source of error on its own, but in combination with the model choice, it could result in two very similar graphs belonging to different vessels, which may negatively affect performance.

This will be addressed by using models that consider edge types and by creating datasets through sampling from a much larger time span in a way that prevents overlap.
Further interesting points for research will be targeted:

- This framework expands the space of hyperparameters. In addition to traditional choices like layer depth, channels, and learning rate, the creation of the base graph becomes crucial as it influences the internal function of the model. Obvious choices include the spatial and temporal thresholds for interaction edges, as well as the overall connectivity between different vessels. A thorough review of optimal hyperparameters will be conducted here.
- Also, the approach has to be compared to existing models in performance.
- The next aspect is exploring how alternative architectures can improve performance. An obvious candidate would be DIR-Graph Attention Networks (DIR-GATs) [12], which consider the similarity of nodes for message passing.
- Expanding on the first point, the modification of the base graph in a learnable fashion needs to be evaluated. This could include simply adding edges to the base graphs through a model or learning whole representations of graphs as a preprocessing step.

Overall, while the initial results not showed success, the framework still promises potential regarding its flexibility, and many interesting open questions remain to be explored. Work on this topic will continue, improving performance investigating applications in other fields where time series influence each other, such as production monitoring.

Declaration on Generative AI

During the preparation of this work, the authors used GPT 4o Mini in order to: Grammar and spelling check. Further, the authors used GPT GPT 4o Mini for the Abstract in order to: Drafting content. After using these tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] International Maritime Organization (IMO), AIS transponders, https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx, 2025. URL: https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx, Accessed:2025-04-11.

[2] C. V. Ribeiro, A. Paes, D. d. Oliveira, AIS-based maritime anomaly traffic detection: A review, Expert Systems with Applications 231 (2023) 120561. URL: https://www.sciencedirect.com/science/article/pii/S0957417423010631. doi:10.1016/j.eswa.2023.120561.

[3] F. Corradini, M. Gori, C. Lucheroni, M. Piangerelli, M. Zannotti, A Systematic Literature Review of Spatio-Temporal Graph Neural Network Models for Time Series Forecasting and Classification, 2024. URL: http://arxiv.org/abs/2410.22377. doi:10.48550/arXiv.2410.22377, arXiv:2410.22377 [cs] version: 1.

[4] X. Zhang, J. Liu, P. Gong, C. Chen, B. Han, Z. Wu, Trajectory prediction of seagoing ships in dynamic traffic scenes via a gated spatio-temporal graph aggregation network, Ocean Engineering 287 (2023) 115886. URL: https://linkinghub.elsevier.com/retrieve/pii/S0029801823022709. doi:10.1016/j.oceaneng.2023.115886.

[5] H. Feng, G. Cao, H. Xu, S. S. Ge, IS-STGCNN: An Improved Social spatial-temporal graph convolutional neural network for ship trajectory prediction, Ocean Engineering 266 (2022) 112960.

URL: https://linkinghub.elsevier.com/retrieve/pii/S0029801822022430. doi:10.1016/j.oceaneng.2022.112960.
[6] A. Kapoor, X. Ben, L. Liu, B. Perozzi, M. Barnes, M. Blais, S. O'Banion, Examining COVID-19 Forecasting using Spatio-Temporal Graph Neural Networks, 2020. URL: http://arxiv.org/abs/2007.03113. doi:10.48550/arXiv.2007.03113, arXiv:2007.03113 [cs].
[7] E. Rossi, B. Charpentier, F. D. Giovanni, F. Frasca, S. Günnemann, M. Bronstein, Edge Directionality Improves Learning on Heterophilic Graphs, 2023. URL: http://arxiv.org/abs/2305.10498. doi:10.48550/arXiv.2305.10498, arXiv:2305.10498 [cs].
[8] T. N. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, 2017. URL: http://arxiv.org/abs/1609.02907. doi:10.48550/arXiv.1609.02907, arXiv:1609.02907 [cs].
[9] AIS-HUB, Free AIS vessel tracking | AIS data exchange | JSON/XML ship positions, https://www.aishub.net/, 2025. URL: https://www.aishub.net/, Accessed:2025-04-11.
[10] Y. Chen, X. Tang, X. Qi, C.-G. Li, R. Xiao, Learning Graph Normalization for Graph Neural Networks, 2020. URL: http://arxiv.org/abs/2009.11746. doi:10.48550/arXiv.2009.11746, arXiv:2009.11746 [cs].
[11] M. Fey, J. E. Lenssen, Fast graph representation learning with PyTorch Geometric, in: ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.
[12] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph Attention Networks, 2018. URL: http://arxiv.org/abs/1710.10903, arXiv:1710.10903 [stat].