A Transformer-based Method for Vessel Traffic Flow Forecasting

Petros Mandalis[1], Eva Chondrodima[2*], Yannis Kontoulis[2], Nikos Pelekis[1], Yannis Theodoridis[2]

[1]Department of Statistics and Insurance Science, University of Piraeus, Piraeus, Greece.
[2*]Department of Informatics, University of Piraeus, Piraeus, Greece.

*Corresponding author(s). E-mail(s): evachon@unipi.gr;
Contributing authors: pmandalis@unipi.gr; ikontoulis@unipi.gr; npelekis@unipi.gr; ytheod@unipi.gr;

Abstract

In recent years, the maritime domain has experienced tremendous growth due to the exploitation of big traffic data. Particular emphasis has been given to deep learning methodologies for decision-making. Accurate Vessel Traffic Flow Forecasting (VTFF) is essential for optimizing navigation efficiency and proactively managing maritime operations. In this work, we present a distributed Unified Approach for VTFF (dUA-VTFF), which employs Transformer models and leverages the Apache Spark big data distributed processing framework to learn from historical maritime data and predict future traffic flows in a time horizon of up to 30 min. Particularly, dUA-VTFF leverages vessel timestamped locations along with future vessel locations produced by a Vessel Route Forecasting model. These data are arranged into a spatiotemporal grid to formulate the traffic flows. Subsequently, through the Apache Spark, each grid cell is allocated to a computing node, where appropriately designed Transformer-based models forecast traffic flows in a distributed framework. Experimental evaluations conducted on real Automatic Identification System (AIS) datasets demonstrate the improved efficiency of the dUA-VTFF when compared to state-of-the-art traffic flow forecasting methods.

Keywords: Big maritime data, Machine Learning, Neural Networks, Vessel Route Forecasting, Vessel Traffic Flow Forecasting

1 Introduction

In the realm of maritime transportation, the shipping industry remains the most efficient type of bulk transportation of goods. In this domain, it is vital to improve maritime transport systems to ensure efficient navigation and, more crucially, safe sailing [1]. Facilitated by the wide adoption of tracking systems [2], such as the Automatic Identification System (AIS), a tremendous amount of maritime data is available. Considerable research efforts have been directed towards harnessing this wealth of maritime data through data-driven methods, with a particular emphasis on machine learning and, more recently, deep learning techniques. These approaches are instrumental in advancing maritime safety management.

At the heart of maritime safety management lies Vessel Traffic Flow Forecasting (VTFF), an indispensable tool that plays a pivotal role in contributing to the management of maritime operations, for example, vessel monitoring, ensuring maritime safety, collision avoidance, etc. [3]. The traffic flow forecasting challenge is to predict the evolution of traffic in a specific region at a future time step based on historical traffic flow data [4]. In traffic flow data, is quite common to assume the form of spatiotemporal raster [4–7]. Raster data are grid-based data referring to observations of a continuous spatiotemporal field represented at fixed locations-regions in space and time. However, capturing the intricate dynamics of vessel movements in the real world presents a formidable challenge [8]. Thus, the VTFF problem is a challenging task characterized by intricated spatiotemporal correlations [9].

The research community has devoted significant attention to tackle the VTFF problem by initially employing traditional Machine Learning (ML) techniques [5], which effectively model nonlinear dynamics [10, 11]. Subsequently, deep learning methods emerged as a powerful alternative, capable of capturing complex nonlinear features within high-dimensional data, surpassing the predictive capabilities of traditional ML methods [4, 9].

In [12], a deep Neural Network (NN)-based scheme, named Unified Approach for VTFF (UA-VTFF), was proposed to tackle the VTFF problem by unifying the two most well-known VTFF perspectives: a) indirect - by estimating the future traffic based on future vessel locations produced by VRF algorithms, and b) direct - by predicting the future traffic based on sequence analysis of historical traffic flow. The UA-VTFF scheme processes historical AIS data and then applies a Vessel Route Forecasting (VRF) method to produce future vessel locations. Subsequently, the processed historical data, along with the produced points of the VRF, are arranged into a spatiotemporal grid to formulate the traffic flows. These traffic flows then undergo feature vector analysis to extract insights into traffic characteristics within both local and neighbor-surrounding grid cells, encompassing present and adjacent time intervals. These observations are then fed into a Long Short-Term Memory (LSTM) [13] -based model for predicting the future evolution of vessel traffic, with a forecasting horizon of up to 30 min.

While UA-VTFF proved its superiority over other baseline methods, there is a need for more accurate predictions within shorter computational times, while accommodating the vast volumes of traffic data processed by UA-VTFF. According to [14], attention-based models like Transformers [15] have the capability to efficiently capture the overall information of entire time series [14], even surpassing Recurrent Neural Networks (RNNs), such as LSTM, in performance. Transformer models [15], originally developed for natural language processing tasks, have shown a remarkable aptitude for capturing long-term dependencies and modeling sequential data [16]. The key component of the Transformer model is its self-attention mechanism, which assigns differential weights to input features. This mechanism enables the model to identify pivotal historical events that significantly influence future outcomes. Transformers have achieved superior results in vessel trajectory prediction problem [8, 17] and in the general case of traffic forecasting problem [18, 19].

The UA-VTFF scheme is a grid-based method that, through appropriate modifications and the utilization of efficient distributed processing, can reduce problem complexity and demand less computational time. Numerous big data platforms are available, with Apache Spark standing out as a widely adopted solution in both academia and industry. It is the latest open-source platform tailored for processing large datasets, particularly suited for machine learning tasks [20]. Furthermore, Apache Spark has been extensively applied in various studies for managing maritime data [21, 22], as well as in addressing general traffic forecasting problems [23]."

To address the need for accurate predictions in short computational times, our work introduces a novel VTFF methodology, the distributed Unified Approach for VTFF (dUA-VTFF). dUA-VTFF achieves superior prediction accuracy through the meticulous design of a Transformer-based model, showcasing its careful construction for optimal performance. Additionally, it leverages the Apache Spark big data distributed processing framework, enabling distributed traffic flow processing.

Our experimental evaluations, conducted using real AIS datasets, demonstrate that the dUA-VTFF exhibits improved prediction accuracy compared to baseline and state-of-the-art traffic flow forecasting methods, including [12, 24, 25]. Furthermore, the dUA-VTFF method outperforms the performance of [12] in terms of execution times, encompassing both learning and prediction times. It's also worth noting that, to the best of our knowledge, this is the first time that Transformer models have been used for addressing both the VTFF and grid-based traffic forecasting problems.

Overall, the main goal of this work is to design a specially crafted method for VTFF based on Transformers and Apache Spark. This involves building upon previous works [12][26] and substantially improving them in terms of both accuracy and execution times. More specifically, a distributed pipeline for accurate VTFF prediction is introduced based on Transformer models and Apache Spark. The proposed approach to VTFF, conducted in parallel across nodes, leverages the distributed nature of our framework, ensuring both efficiency and accuracy in forecasting. The incorporation of Big Data technology and distributed processing in the dUA-VTFF framework yields substantial gains in execution times and a scalable architecture for maritime applications. It should be noted that preliminary scalability testing was conducted on a single node.

The rest of this paper is organized as follows: Section 2 discusses related work; Section 3 provides background and preliminary terms; Section 4 presents the proposed dUA-VTFF methodology; Section 5 describes the available AIS data, presents the experimental setup, and discusses the results of our experimental study; Section 6 concludes the paper and discusses future extensions.

2 Related Work

According to [14], traffic forecasting methods can be categorized into three groups: statistical methods, traditional machine learning methods, and deep learning methods. The first category is often ill-suited for handling complex and dynamic time series data, while the second category can model more complex data but may not be optimal for capturing intricate and dynamic traffic data [14]. On the other hand, deep learning methods have emerged with the capability to capture complex nonlinear features within high-dimensional data, surpassing the predictive capabilities of traditional ML methods [4, 9]. Consequently, traffic forecasting has gradually shifted towards adopting deep learning methods.

Based on the literature review presented in [12, 26], numerous works have addressed the VTFF problem using machine learning or deep learning methods. However, many of these studies focus on specific maritime regions of interest or relatively large areas, which may not be conducive to effectively managing vessel traffic. In contrast, our work is capable of predicting traffic flow in open seas as well as in smaller maritime areas.

Several studies addressing the VTFF problem have employed RNNs. For example, Li and Ren [27] introduced a multi-step vessel traffic flow prediction method based on an LSTM Encoder-Decoder architecture, while in [28], Gated Recurrent Units (GRU) were employed for vessel traffic prediction. In recent studies [14], attention-based models like Transformers [15] have shown the ability to efficiently capture the overall information within entire time series [14], often outperforming RNN-based methods in terms of performance. In this study, we briefly present the most recent relevant works that have applied Transformer models to tackle traffic forecasting problems. To the best of our knowledge, Transformer models have not been adopted for addressing the traffic forecasting problems in the maritime domain.

In [18], different deep-learning time-series forecasting methods were examined, including, among others, Transformer models for making short-term predictions regarding traffic flows to ensure smart mobility. Cai et.al. [29] argued that Transformer cannot be directly applied to traffic forecasting and proposed a hybrid encoder–decoder architecture to capture the continuity and periodicity of time series and models spatial dependency. Fang et.al.[30] proposed a novel model for traffic forecasting, called Locality-aware spatio-temporal joint Transformer (Lastjormer), which incorporates spatio-temporal joint attention in the Transformer architecture to capture all dynamic dependencies in the traffic data. In [31], a method called "detrending" was employed to improve traffic prediction using a spatial-temporal Transformer encoder-decoder architecture.

Notably, none of the aforementioned studies considered the big data nature of traffic flow data or thoroughly investigated learning and prediction times, which are paramount in the VTFF problem in real applications. In contrast, our proposed method leverages the Apache Spark big data distributed processing framework, enabling distributed traffic flow processing.

Lv et.al. [24] employed a deep learning model using Stacked Auto Encoders (SAES) to learn generic traffic flow features. In their approach, a Logistic Regression algorithm is applied on top of the SAE to facilitate traffic flow predictions. The network layers are trained as Auto Encoders using a greedy layer-wise unsupervised learning method and fine-tuned using Back Propagation. In experiments, SAEs demonstrated superior performance compared to Support Vector Machines (SVMs), the Random Walk Forecast method, Feed-forward NNs, and Radial Basis Function NNs. Additionally, in [24] LSTM and GRU networks were employed to predict traffic volume measured by sensors on freeways. Results indicate that GRU and LSTM outperform the Auto Regressive Integrated Moving Average model, with GRU NNs exhibiting slightly better performance than LSTM and generally converging faster than LSTM.

According to [6], traffic data structures have four types of spatiotemporal data in the real world: event data, trajectory data, point reference data, and raster data (grid-based data representing observations of a continuous spatiotemporal field). Particularly, event data involves discrete events occurring at specific point locations and times, where each event may include non-spatiotemporal variables known as marked variables. Trajectory data captures the paths of moving bodies over time, including series of locations and additional marked variables of the moving body. Additionally, point reference data measure a continuous spatiotemporal field at moving reference sites, providing observations at discrete points for reconstructing the field using data-driven or physics-based methods. Finally, raster data, are quite a common representation in real-world traffic applications [4–7].

Furthermore, two types of spatial structures for traffic data exist in the real world: grid and graph [14], with maritime traffic knowledge mining methodologies categorized as grid-based, vector-based, and statistics-based methods [5]. Grid-based methodologies leverage the spatial structure of a grid, accommodating the representation of maritime traffic in a structured and organized manner. Particularly, grid-based approaches organize raw traffic data within a set of indexed grids, facilitating the reduction of problem scale [5]. Vector-based methods, on the other hand, involve the use of vectors to represent spatial relationships, while statistics-based approaches rely on analytical and statistical techniques to derive insights from maritime traffic data. This categorization showcases the diversity of methodologies employed for extracting meaningful knowledge from maritime traffic datasets. In our study, we focus on maritime traffic data, which takes the form of raster data categorized as grid-based.

3 Background and Definitions

In this section, background and preliminary terms are provided. The main definitions employed in this paper are as follows: Consider a maritime dataset D composed of S vessels, where the s-th vessel consists of m_s trajectories and the j-th trajectory is comprised of n_j^s timestamped positions (sampled at asynchronous time intervals) and can be represented as follows:

$$\mathbf{P}_j^s = \left[\mathbf{p}_j^s(1), ..., \mathbf{p}_j^s(n_j^s)\right] = \left[[t_j^s(1),\ \mathbf{o}_j^s(1)], ..., [t_j^s(n_j^s),\ \mathbf{o}_j^s(n_j^s)]\right], \\ s = 1, ..., S, \quad j = 1, ..., m_s \tag{1}$$

where $\mathbf{p}_j^s$ is a timestamped position, which consists of timestamp t and location $\mathbf{o}(x, y)$ in the Universal Transverse Mercator (UTM) system.

Definition 1. ***Route Forecasting***:

- **Given**:
  - a set of vessel trajectories D,
  - a vessel's trajectory $[\mathbf{p}_j^s(1), \ldots, \mathbf{p}_j^s(k')]$ consisting of k' consecutive points,
  - a time duration (prediction horizon) Δt,
  - a number of transitions r

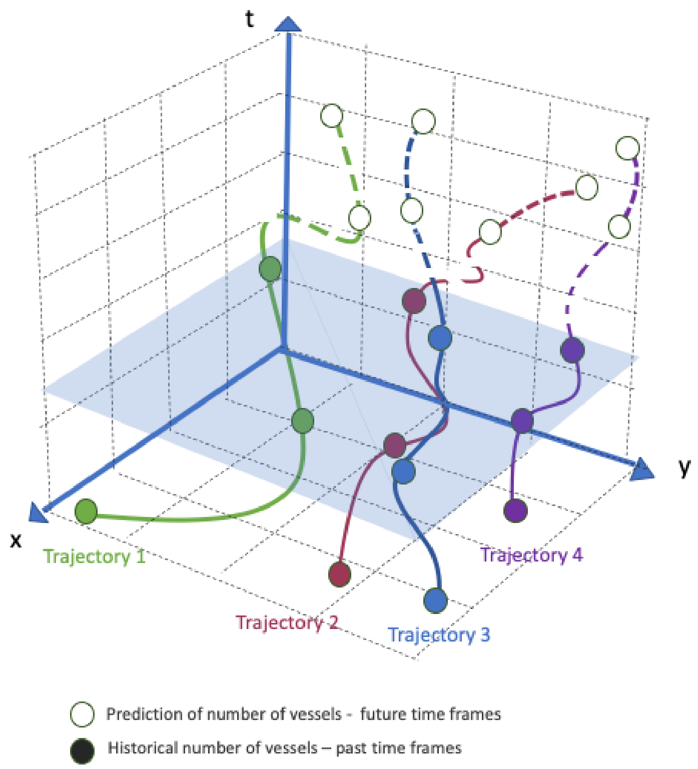


Fig. 1: Example of 4 vessel trajectories in a spatiotemporal grid of 5 temporal frames and 4 × 4 space resolution.

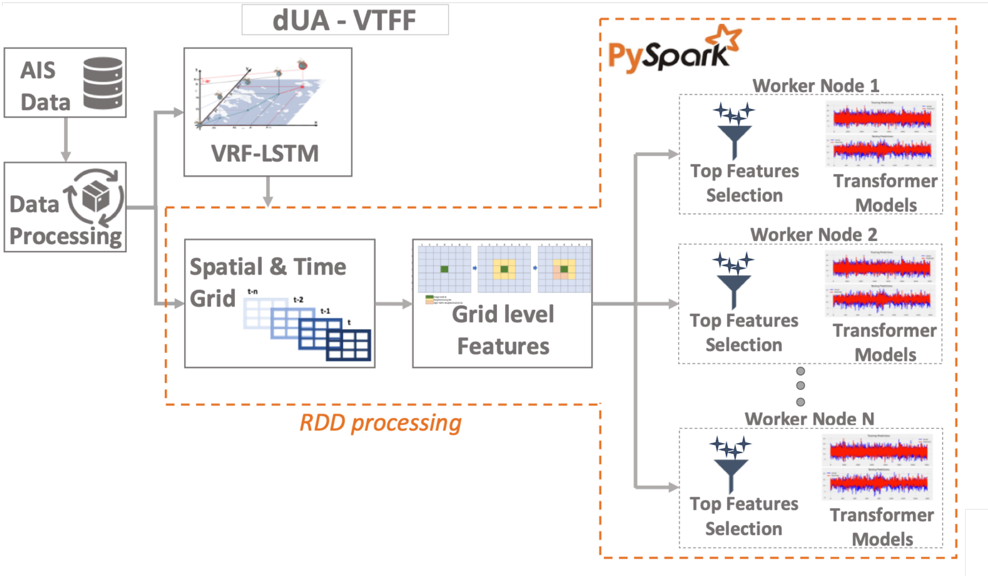


Fig. 2: Overview of the dUA-VTFF methodology.

- **Predict**: each vessel's future trajectory up to Δt, consisting of a total of r transitions with fixed sampling rate (i.e. $t^s_j(k'+1) = t^s_j(k') + \Delta t/r$, $t^s_j(k'+2) = t^s_j(k') + 2 * \Delta t/r$, ..., $t^s_j(k'+r) = t^s_j(k') + \Delta t$):

$$\left[\mathbf{o}^s_j(k'+1),\ \ldots,\ \mathbf{o}^s_j(k'+r)\right] \tag{2}$$

Definition 2. ***Traffic Flow Forecasting***:
- **Given**:
  - a time duration (prediction horizon) Δt,
  - a number of temporal transitions r,
  - a set of vessel trajectories D spanning in Ds (minimum bounding box of locations) in space and D_T in time,
  - a set of future vessel trajectories D_P spanning in Ds and Δt,
  - a spatiotemporal (i.e., 3D) grid that splits a) Ds into grid cells of resolution $G \times G$, and b) $D_T \cup \Delta t$ into time frames of duration $\Delta t/r$
- **Predict**: the volume of vessels in each cell of the spatiotemporal grid during Δt.

Fig. 1 presents an example of a spatiotemporal grid of 5 time frames and 4×4 space resolution. There are 4 trajectories evolving in 3 time frames, and the goal is to forecast the vessels' volume in each grid cell during the 2 time frames in the future.

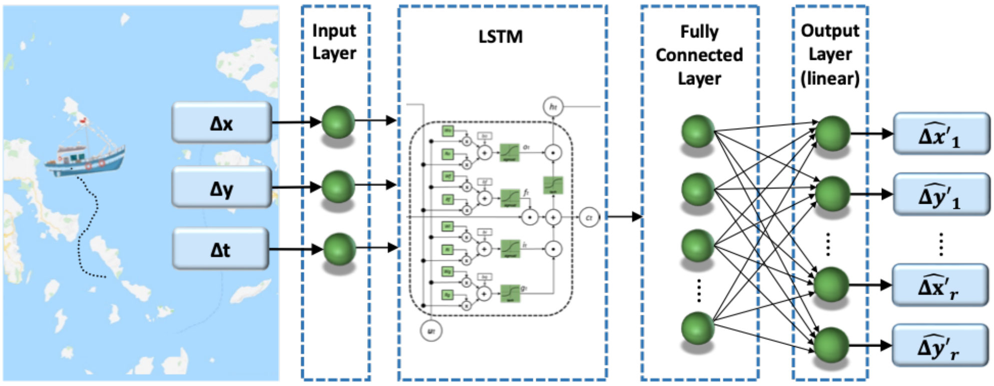


Fig. 3: Proposed VRF-LSTM network architecture, with one input layer, one LSTM layer, one fully-connected layer, and one output layer.

4 Methodology

In this section, our approach is presented including the VRF-LSTM method and the dUA-VTFF methodology. An overview of the proposed methodology is illustrated in Fig. 2. More specifically, first, historical AIS data pass through typical AIS data preprocessing. Then the VRF-LSTM method is applied. The processed historical data along with the produced points of the VRF are converted into Resilient Distributed Datasets (RDDs) and are arranged into a spatiotemporal grid, where each RDD representing traffic flows. Subsequently, the resulting traffic flows pass through a feature vector analysis, to a big data distributed processing framework, where each node in the distributed system is allocated a specific subset of grid cells. Finally, Transformer models are being employed for VTFF prediction.

4.1 VRF-LSTM method

In this study, we employ the VRF-LSTM method introduced in [12], adopting identical data preprocessing, NN structure, methods for update, optimization, validation, model selection, and dataset split criteria, due to the following findings: The VRF-LSTM model was originally introduced in [32] for effective location forecasting. It was validated using three real-word datasets of vessels moving in different sea areas and compared against various ML methods, demonstrating a prediction accuracy exceeding 30%.

A variant of the model in [32] was introduced in [33] operating as a multipoint location forecasting model, where the NN model needs to be executed r times to predict r future points. In [33] popular ML methods were investigated to address the VRF problem through an experimental testbed based on real vessel surveillance data, indicating that LSTM-based models efficiently forecasted vessel trajectories, outperforming all their rivals.

Another variant of the VRF-LSTM model was introduced in [12]. This variant approaches the VRF problem by treating it as a direct trajectory forecasting application, but it cannot produce predictions for every point (as in [32, 33]) due to its dedicated preprocessing procedure analyzed in [26, 33]. On the other hand, in contrast to [33], the variant in [12] can predict the desired future points in a single execution of the trained model, which is vital in real applications to manage vessel traffic effectively. Thus, in this study, we employ the VRF-LSTM method introduced in [12].

The overall process of producing a trained VRF-LSTM model, involves the training of the NN-based model, followed by the phases of validation and model selection [12, 32, 33]. First, the available data undergo two crucial preprocessing steps essential for the VRF-LSTM model, as discussed in [32]: i) simplifying stationary data (excluding records with speeds less than one knot) and eliminating insignificant trajectories (those with fewer than ten points), and ii) segmenting trajectories into sub-trajectories when the time interval between two consecutive points of the same trajectory exceeds a specific threshold (in this work set at 30 min.). Subsequently the differences in time and space between consecutive timestamped positions are computed.

The VRF-LSTM model is based on an NN model composed of an input layer of three neurons, an LSTM hidden layer, a fully-connected hidden layer and an output layer of $r * 2$ neurons. The NN predicts the future trajectories composed of 2D points of r transitions, by using as input timestamped locations. Consequently, every pair of output neurons yields a forecast for each transition in the trajectory. To facilitate model training, the output data are interpolated along the trajectory's r transitions. Following the training process, a single execution of the NN model is adequate to generate the desired predictions.

Regarding the input information, three features are provided, encompassing details about time and two spatial coordinates. These features specifically denote the temporal and spatial differences between successive timestamped positions. More specifically, for the s-th vessel, each timestep $k = 1, \ldots, n_{j-1}$, where n_j represents the length of the j-th trajectory, the NN is supplied with the following input vector:

$$\begin{gathered}\Delta \mathbf{u}_j^s(k) = \left[\Delta x_j^s(k), \Delta y_j^s(k), \Delta t_j^s(k)\right] = \\ \left[\ x_j^s(k) - x_j^s(k-1),\ y_j^s(k) - y_j^s(k),\ t_j^s(k) - t_j^s(k)\ \right].\end{gathered} \tag{3}$$

The desired output of the model, which corresponds to the s-th vessel's future trajectory consisting of a total of r transitions interpolated with fixed sampling rate equal to $\Delta t/r$ is the following:

$$\begin{gathered}\left[\ \Delta \mathbf{o'}_j^s(k+1),\ \Delta \mathbf{o'}_j^s(k+2),\ ...,\ \Delta \mathbf{o'}_j^s(k+r)\ \right] = \\ \left[\ \left(\ \Delta x'^s_j(k+1), \Delta y'^s_j(k+1)\ \right),\ ...,\ \left(\ \Delta x'^s_j(k+r), \Delta y'^s_j(k+r)\ \right)\ \right]\end{gathered} \tag{4}$$

where $\Delta x'$ and $\Delta y'$ are the intervals of the interpolated positions of easting and northing, respectively. Once the prediction procedure has been completed, the NN model has predicted for the s-th vessel the easting and northing intervals $\hat{\Delta x'}$ and $\hat{\Delta y'}$, respectively, of the r consecutive points with fixed sampling rate equal to $\Delta t/r$, which is defined as:

$$\begin{aligned}\left[\ \hat{\Delta\mathbf{o}'}^{s}_{j}(k+1),\ \hat{\Delta\mathbf{o}'}^{s}_{j}(k+2),\ \ldots,\ \hat{\Delta\mathbf{o}'}^{s}_{j}(k+r)\ \right] = \\ \left[\ \left(\ \hat{\Delta x'}^{s}_{j}(k+1), \hat{\Delta y'}^{s}_{j}(k+1)\ \right),\ \ldots,\ \left(\ \hat{\Delta x'}^{s}_{j}(k+r), \hat{\Delta y'}^{s}_{j}(k+r)\ \right)\ \right]\end{aligned} \tag{5}$$

Subsequently, the model output is transformed from differences to actual values. The training goal is the same as in [12, 32, 33], i.e. to minimize the loss function, which is defined as the Euclidean distance between the desired positions and the predicted positions. Details for the Backward Propagation Through Time (BPTT) algorithm, which was employed for updating the network parameters, can be found in [34]. Additionally, details for the Adam approach, which was employed for optimizing the synaptic weights on the training set, can be found in [35].

The training phase of the model is followed by the phases of validation and model selection [12, 32, 33]. The inclusion of a validation set is crucial for assessing the generalization ability of the model. In this context, the NN that exhibits superior performance over a specific number of epochs on the validation set is ultimately chosen. In simpler terms, the training concludes when the error fails to decrease for a specified number of epochs on the validation set.

4.2 The distributed Unified Approach for VTFF (dUA-VTFF) based on Transformer Networks

In this section the centralised and distributed versions of the Unified Approach for VTFF methods are presented.

4.2.1 The centralised Unified Approach for VTFF (UA-VTFF) method

Due to transceiver errors, AIS records may exhibit inaccuracies [36]. Consequently, the centralised UA-VTFF method [12] starts by applying a standard data cleansing preprocessing procedure to the available AIS data. This involves deduplication of records and the removal of noisy entries (particularly those associated with speeds exceeding 50 knots).

Subsequently, the centralised UA-VTFF method [12] proceeds by giving the vessels' historical trajectories into the proposed VRF-LSTM method (described in the previous section) in order to predict the future trajectories. Subsequently, the vessels' historical and future trajectories are assigned into the spatiotemporal grid, namely the calculation of the number of vessels that are placed in every grid cell is performed. The resulting numbers indicate the volume of the historical vessels n and the predicted vessels η (produced by the proposed VRF-LSTM model), which represent the traffic sequence in a specific cell (region) and time frame. Then, these traffic sequences for every grid cell b are enhanced with additional features, such as the volume of vessels in surrounding cells. Finally, the produced sequences are fed to an ML algorithm for predicting the future traffic flow in the grids, i.e. the vessels' number in a particular cell in future time frame Δt.

The UA-VTFF method can predict the number of vessels in the future time frame $t+1$ in the b-th grid cell composed of l total cells. For this, a model should be constructed with output $\hat{n^b}_{t+1}$ and the corresponding input N^b_{t+1} composed of the following features:

- $n^b_{t-l}, ..., n^b_{t-1}, n^b_t$: the vessels' number n in time frame t in the b-th cell grid
- $n^{b'}_{t-l}, ..., n^{b'}_{t-1}, n^{b'}_t$: the vessels' number n in time frame t in the neighbor-surrounding cell grids b' of b-th cell grid
- $n^{b''}_{t-l}, ..., n^{b''}_{t-1}, n^{b''}_t$: the vessels' number n in time frame t in the neighbor-surrounding cell grids with high traffic b'' of b-th cell grid
- η^b_{t+1}: the vessels' number produced by the VRF-LSTM model in the future time frame $t+1$ in the b-th cell grid
- $\eta^{b'}_{t+1}$: the vessels' number produced by the VRF-LSTM model in the future time frame $t+1$ in the neighbor-surrounding cell grids b' of b-th cell grid
- $\eta^{b''}_{t+1}$: the vessels' number produced by the VRF-LSTM model in the future time frame $t+1$ in the neighbor-surrounding cell grids with high traffic b'' of b-th cell grid
- $md^b_{t-l}, ..., md^b_{t-1}, md^b_t$: the day of the month md in each time frame t in the b-th cell grid
- $dw^b_{t-l}, ..., dw^b_{t-1}, dw^b_t$: the day of week dw in each time frame t in the b-th cell grid
- $td^b_{t-l}, ..., td^b_{t-1}, td^b_t$: the time of day td in each time frame t in the b-th cell grid

Regarding the calculation of the neighbor-surrounding grid cells and those with high traffic, the neighbor cells correspond to those cells that surround the center-targeted. The surrounding grid cells with the highest traffic are those with the highest correlation with the center-targeted and are defined based on statistical analysis and taking into account that these cells should correspond to regions areas associated with a high risk of accidents [37]. To clarify the above discussion, Fig. 4 illustrates a grid with nearby cells only in space of 7×7. The green cell illustrates the central-targeted cell for which a prediction will take place. The yellow cells correspond to the neighborhood cells, while the pink cells present the neighborhood cells with high traffic and high correlation with the center-targeted.

4.2.2 Transformer Architecture

Several ML techniques [12] can be applied to the produced spatiotemporal grid-based traffic sequences, enhanced by additional features, to predict future vessel traffic. In this study, we utilize the Transformers architecture, harnessing its ability to capture long-term dependencies in sequential data. For the VTFF problem, we adapt the Transformer model [15] to sequence-to-sequence forecasting. The model architecture consists of an encoder layer, a Self-Attention Mechanism, Feed-forward NNs, and a Decoder with Linear Layer as presented in Fig. 5. Particularly, a brief description of each employed layer is given below:

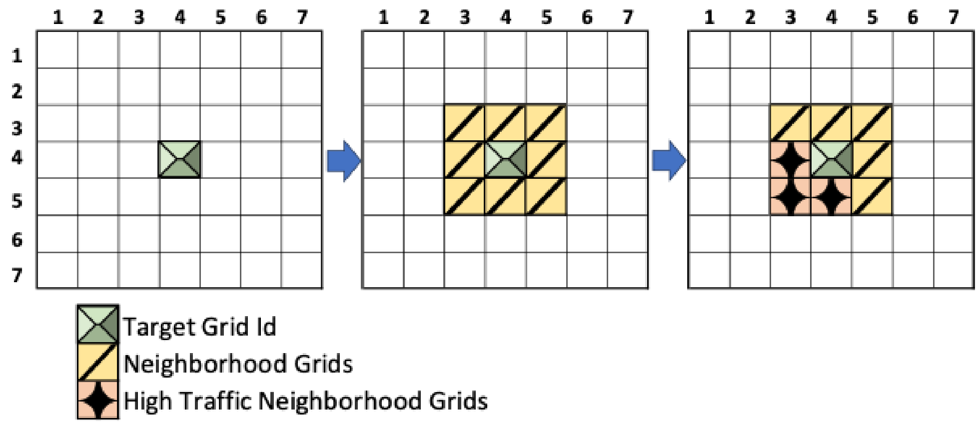


Fig. 4: Example of 7×7 neighbor-surrounding traffic cells of a spatial grid.

- Encoder Layer: The encoder layer processes the input sequence. For the VTFF application, this layer learns to understand and encode the patterns in the historical time series data of each grid cell.
- Self-Attention Mechanism: Self-attention allows the model to weigh the importance of different points in the time series. For forecasting problems, this is particularly useful as it helps the model to focus on the most relevant parts of the past data when predicting future values.
- Feed-forward NNs: These networks, placed within each Transformer block, allow for the learning of complex relationships. In the context of VTFF application, they can help in capturing non-linear dependencies, which are often present in maritime data. Our solution includes 4 layers of Self-Attention and Feed-forward NNs.
- Decoder with Linear Layer: The decoder is responsible for generating the output sequence. In the VTFF case, this represents the future values of the time series. The linear layer at the end of the decoder transforms the decoder's output into the forecasting future values format, configured with 6 * 2 neurons. This specific design is aligned with the goal of predicting the next 6 observations, each observation capturing 2D spatial data for a 5-minute interval. The linear layer effectively maps the decoder's output into this 6-interval forecast format.

Regarding model training, we employed Adam as the optimizer [35]. Also, we optimized the Transformers by fine-tuning the hidden layer sizes, the learning rate determining the number of epochs, and configuring the batch size. Additionally, we implemented an early stopping procedure to prevent the network from overfitting by monitoring its performance on a validation set.

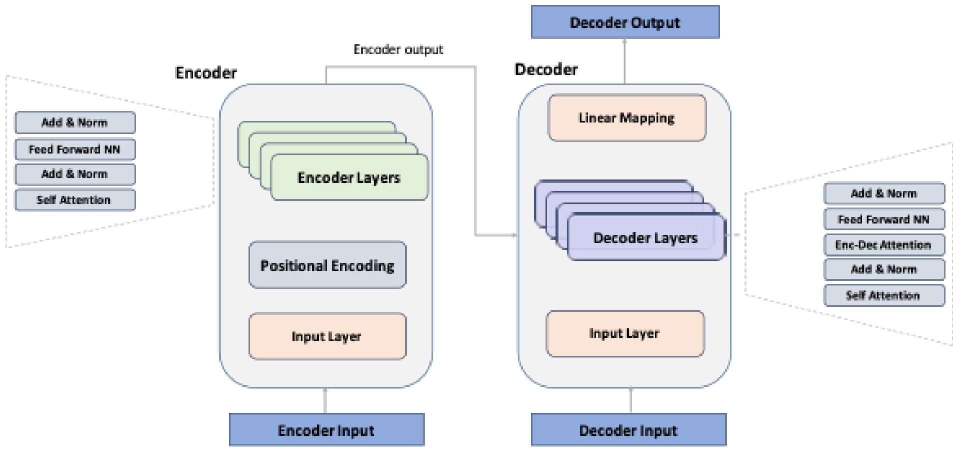


Fig. 5: Network architecture, with an encoder layer, a Self-Attention Mechanism, Feed-forward NNs, and a Decoder with Linear Layer.

4.2.3 The distributed Unified Approach for VTFF (dUA-VTFF) method

The large amount of maritime data in the VTFF problem necessitates the deployment of a robust and scalable big data framework. The dUA-VTFF methodology harnesses the combined strengths of Apache Spark and Transformer models to ensure efficient processing of large datasets and accurate predictions. The advantage of incorporating Big Data technology and distributed processing in dUA-VTFF framework are significant gains in execution times described in section 5.3.2 and scalable architecture for maritime applications.

In our framework we leverage Apache Spark, a distributed computing system, that has the ability to process massive amounts of data seamlessly across clusters. The starting point for the Spark-based application is the establishment of a Spark session that enables the distributed processing capabilities of Spark.

The configuration of the Spark session is pivotal to the application's performance, a necessity underscored by the VTFF problem's demands for processing extensive datasets. To tailor the Spark environment to the demands of this specific problem, various parameters during the session's creation, such as the memory configuration that ensures that the available resources are used efficiently preventing out-of-memory errors and reducing garbage collection.

The distributed version of the UA-VTFF, namely dUA-VTFF pipeline is illustrated in Fig. 2. The dUA-VTFF starts by giving the available vessels' historical trajectories into the proposed VRF-LSTM algorithm to predict the future trajectories. Subsequently, the vessels' historical and future trajectories data are consolidated and converted into RDDs, laying the groundwork for efficient distributed processing across the computational framework, i.e. data preprocessing is conducted using Spark's RDD mechanism. RDD is the primary concept that Spark offers, and it is a collection of elements partitioned across the cluster nodes that can be operated on in parallel [38].

A crucial step in this process is the formation of a spatiotemporal grid constructed by aggregating the RDDs based on time and space dimensions. The available data (the vessels' historical and future trajectories), once preprocessed, are assigned into this spatiotemporal grid. Specifically, the available data are partitioned based on the unique spatial grid cells, allowing each to be treated as an independent unit for training.

This dual segmentation facilitates a more granular and precise analysis of traffic flow patterns. Following the grid creation, the calculation of the number of vessels (in the form of RDDs) that are placed in every grid cell is performed. Then, each RDD representing traffic flows undergoes a comprehensive feature vector analysis. This analysis is pivotal in generating a rich set of features for each grid cell. The features encapsulate various aspects, including the influence of neighboring grid cells, historical traffic patterns, and seasonality aspects. This multi-faceted approach ensures a thorough understanding of the dynamics at play within each grid.

The final phase of the process involves a distributed approach to VTFF prediction. Each node in the distributed system is allocated a specific subset of grid cells. Within each node, the process unfolds in two stages for each grid cell. Initially, a top feature selection is conducted to identify the most relevant features that influence traffic flow and for every grid cell b the produced traffic sequences are enhanced with additional features. Subsequently, Transformer models are applied to these enhanced traffic sequences for VTFF prediction. This two-stage process, conducted in parallel across nodes, leverages the distributed nature of our framework, ensuring efficient and accurate forecasting.

The Transformer models, also leverage Spark's distributed features for training. The framework is designed to auto-detect available GPUs, distributing the computational workload of the model across them. This framework trains the Transformer model on each unique spatial grid cell simultaneously across a cluster, which is a component of Spark's RDD parallelization. This distributed approach ensures the framework's scalability, guaranteeing its readiness for ever-growing maritime data.

5 Experimental Study

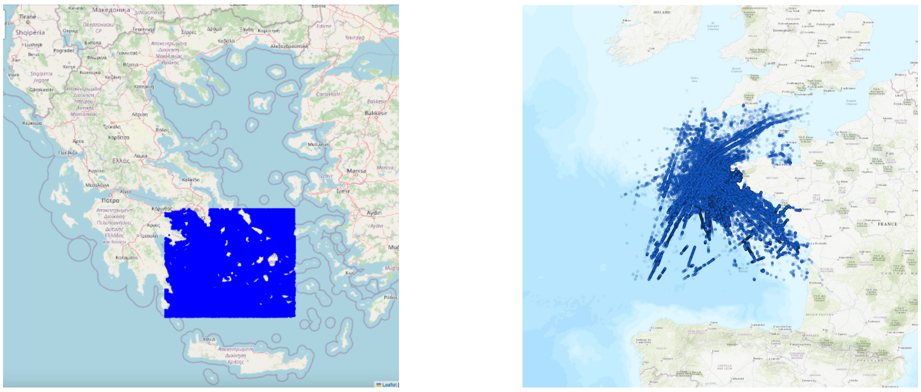
This section presents the experimental setup for evaluating dUA-VTFF method including various learning schemes for comparison.

5.1 Dataset description

Experiments were conducted using real-world AIS data collected from two distinct sea areas to assess the performance of the implemented models across various motion patterns.

5.1.1 Aegean-Cyclades Dataset

The dataset provided by MarineTraffic.com includes a total of 1,757,440 records of location related information derived from AIS messages of vessels sailing in the Aegean



(a) (b)

Fig. 6: Overview of the employed datasets: (a) Aegean-Cyclades, (b) Brest

Sea, Greece, during November, 2018. In detail, it corresponds to AIS activity transmitted by 2344 distinct vessels of various types and covers an area in the Aegean Sea that falls in the rectangle bounded by latitude in [36...38] and longitude in [23...26], as illustrated in Fig.6a. The sampling rate in this dataset ranges from less than 1 second to several days, with a median value of 2.5 min, whereas the number of AIS messages broadcasted by a vessel varies from 1 to 12801, with a median value of 354.

5.1.2 Brest Dataset

The dataset provided by [39] includes a total of 18,657,858 AIS records of location related information collected by the AIS receiver located in Brest, France, from October 1, 2015 to March 31, 2016. The AIS activity was transmitted by 5041 distinct vessels of various types covering a sea area that falls in the rectangle bounded by latitude in [-10...0] and longitude in [45...51], as illustrated in Fig.6b. More information about the original dataset can be found in [39].

5.2 Experimental setup

In this section, we present the experimental setup employed in our study. We provide insights into the hardware and software setup, along with the evaluation criteria adopted to assess the efficacy of the implemented algorithms. Furthermore, we detail the spatiotemporal grid configuration, and outline the time series generation process for analysis. Finally, we discuss the dataset splitting, as well as the model parameterization utilized in our experiments.

Hardware and Software Environment

All the methods were implemented using Python and the experiments were carried out on a workstation equipped with an Apple M1 Pro 10-Core CPU of 16 GB of RAM, and a 12-Core GPU. Also, PySpark was used for implementing dUA-VTFF, while the Transformer models were designed in PyTorch.

Evaluation Criteria

Moreover, all the implemented algorithms were evaluated according to the busy grid cells. These cells represent regions characterized by regular navigation patterns and heavy traffic, posing a high risk of accidents, as discussed in [37, 40]. Conversely, areas with low vessel density indicate a lower degree of traffic flow complexity, suggesting more regular and predictable vessel travel time sequences [41]. This observation, also, aligns with the findings of the experimental study conducted in [26], which showed that the ML model performed better when evaluated in both busy and non-busy regions (with ”non-busy regions” denoting areas with zero traffic), as opposed to the ML model exclusively tested in busy areas.

Performance Metric

Relative Absolute Error (RAE) [42], described by Eq. 6, was adopted for evaluating the performance of the models. RAE is calculated by taking into account the total absolute error and dividing it by the absolute difference between the mean and the actual value:

$$RAE = \sum_{i=1}^{N} \frac{|y_i - \hat{y}_i|}{|y_i - \overline{y}|} \tag{6}$$

where N is the number of records in a specific grid cell at a specific time frame in the spatiotemporal grid; y and $\hat{y}$ represent the actual and the predicted number of vessels, respectively; $\overline{y}$ is the mean of the actual number of vessels of the N records. RAE has a possible value between 0 and 1. Values near zero, with zero being the best value, implies a good model, and vice versa.

Grid Setup

Given the VTFF problem formulation of Section 3, the available data were organized into a spatiotemporal (3D) grid consisting of spatial grid cells with a resolution of $G = 2$km and time frames of duration $\Delta t/r = 30/6$ min, using a prediction horizon of $\Delta t = 30$min and a number of temporal transitions of $r = 6$. We resulted in the abovementioned parameters after experimenting with different values of spatial grid's resolution, number of temporal transitions and prediction horizon. More information can be found in [12].

Statistics & Time Series Generation

The traffic flow generated using the Aegean-Cyclades dataset and the abovementioned parameters is summarized as follows: A total of 14,340 cells were created, with traffic flow values ranging from a minimum of 0 to a maximum of 103,617 vessels. Also, the median and mean traffic flow was 25 and 120 vessels, respectively. Out of all the grid cells, only 768 cells exhibited a traffic flow exceeding 300 vessels throughout the entire observation period, while 4,000 cells had fewer than 10 vessels during the entire duration. The top 10 grid cells, on average, had more than 5 vessels every 5 minutes and included more than 300,000 vessels during the entire period.

The traffic flow generated using the Brest dataset and the abovementioned parameters is summarized as follows: A number of 17,693 cells were created, with the traffic flow of min, max, median and mean values equal to 1, 491478, 2, and 96 vessels, respectively. Of the whole grid cells, only 363 cells presented traffic flow of more than 300 vessels across the entire period, while 13,747 cells included less than 10 vessels in the whole period. The top 10 grid cells have more than 3 vessels on average every 5 minutes and include more than 1,000,000 vessels in the entire period.

As a result, for each maritime dataset, we generated 10 distinct chronologically ordered timeseries, each of the same length, corresponding to traffic volume. These time series were derived from the 10 busy grid cells and were sampled at a rate of 5 minutes. We selected a 5 min sampling rate, as it is commonly used in vessel traffic flow applications [43, 44]. Previous studies have also indicated that using a 5-minute traffic interval leads to more accurate predictions [25].

Dataset Splitting

The traffic volume records of each timeseries were, also, allocated into three sets, namely training, validation, and testing, using a 70%–5%–25% percent ratio, respectively. Particularly, due to the time dependence, we followed a chronological order to split into the three datasets. The training set was used to define the parameters for the models, while the validation set was used to choose the model. The performance of the selected model was evaluated on the testing set. Aside from training and validation sets, it is crucial to evaluate an ML model on an independent set due to overfitting that could occur with respect to the validation set.

Model Parameterization

Regarding the model parameterization, several aspects were taken into account and optimized accordingly. More specifically, the proposed transformer models (described in section 4.2.2) were optimised by using Adam algorithm and according to the following parameters:

- Learning Rate: Controls the size of the steps that the model takes during the optimisation process. Smaller values lead to smaller adjustments to the model which requires more iterations in order to have convergence. We explored values ranging from 0.001 to 0.01 in our grid search.
- Number of Layers: Each layer consists of self-attention and Feed-forward NNs, as illustrated in Fig. 5 depicts the model's architecture. The more the layers the higher the complexity of the model. We explored values ranging from 1-12 layers in our grid search.
- Dropout parameter: This parameter addresses the regularization of the algorithm and helps prevent overfitting. During training, a fraction of the nodes (set by the dropout rate) are randomly dropped, forcing the model to learn redundant representations and making it less sensitive to specific features of the training data. The higher the values the higher is the amount of regularization. We explored values ranging from 0.1 to 0.4 in our grid search.
- Batch Size: Represents the number of training observations that are used in one iteration of the models training. The larger the batch size the higher the requirements

of memory and computational power. We explored smaller values ranging from 64 - 256 and in larger values ranging from 1000 - 3000.

- Number of heads: Determines the number of parallel attention heads in each layer. By increasing this parameter, the model can attend to different parts of the input sequence, potentially capturing a richer set of patterns. We iterated on a range of 4 -16 heads.
- Number of Epochs: An epoch represents a complete pass over the entire training dataset. We iterated on a range of 30-200 epochs.

5.2.1 Comparison to state-of-the-art methods

To assess the effectiveness of the proposed dUA-VTFF method compared to state-of-the-art methodologies, we implemented the approaches introduced in [24, 25]. The former work marked a significant milestone in the history of deep learning-based short-term traffic forecasting [45]. Both methods in [24, 25] have been employed as baseline benchmarks in recent studies [46, 47], and [48, 49], respectively. Also, both studies [24, 25] applied their respective methods to the Caltrans Performance Measurement System (PeMS) database [24]. This dataset contains traffic flow data collected from sensors installed on freeways in California. Note that a constant sampling rate of 5 min is employed in both studies [24, 25].

In the case of [24], traffic data from different sensors were aggregated to calculate the average traffic flow volume for each freeway. Thus, the traffic volume data for each freeway constitutes a distinct timeseries. Subsequently, SAEs were applied to historical traffic volume data recorded at previous time intervals in each freeway, to predict the traffic flow of each freeway at a future time interval. The output dimension of the SAEs matched the number of freeways, and the input dimension was calculated as the output dimension multiplied by the number of previous time intervals. In this study, the traffic volume data of each freeway in the PeMS dataset corresponds to the traffic volume measured in each grid cell within the maritime dataset's sea area.

On the other hand, in [25], the authors considered only the traffic volume measured by each sensor, without taking into account the freeways. Consequently, the traffic volume data collected by each distinct sensor constituted a separate timeseries. Subsequently, a unique NN model was created for each of these traffic timeseries. Specifically, for each sensor GRU models were applied to historical traffic volume data recorded at previous time intervals to predict the traffic flow for that specific sensor at a future time interval. The output dimension of these models was set to one, and the input dimension was determined by the number of previous time intervals. In this study, the traffic volume data of each sensor in the PeMS dataset corresponds to the traffic volume measured in each grid cell within the maritime dataset's sea area.

Furthermore, in this study, the parameters for the SAE and GRU models were selected according to [24, 25], respectively. Specifically, the number of previous time intervals was set equal to 6, and the SAES' output dimension was configured to match the number of the top busy grid cells, which is equal to 10.

5.2.2 Big data distributed processing framework

To validate the efficiency of the dUA-VTFF method, a series of experiments were conducted aimed at benchmarking the framework's responsiveness to varying computational demands. The primary objective was to determine if the framework could dynamically scale and optimize its resources to meet the growing data processing requirements, a core attribute of big data solutions. These experiments conducted on the Aegean-Cyclades dataset.

More specifically, the first experiment focused on preliminary scalability testing conducted on a single node, where we incrementally increased the number of worker nodes involved in the data processing task. Specifically, the number of workers nodes took the values of 2, 4, 8, and 16. This setup simulated a real-world scenario where computational resources, such as CPU cores, GPU cores, or entire servers, can be dynamically added or removed based on the task demands. However, it's important to recognize that scalability in a distributed system could differ significantly from that of a single node.

Furthermore, the second experiment explored the framework's capability to handle vast amounts of data, a key feature of big data challenges. We progressively increased the dataset's size and assessed the framework's performance in terms of processing speed and efficiency.

Finally, the execution time for the dUA-VTFF and UA-VTFF methods was calculated in the Aegean-Cyclades dataset. This comprehensive analysis of execution times for both distributed and centralized approaches allows us to assess and compare the efficiency and performance of each method. Understanding the computational demands of these methods is crucial for making informed decisions regarding their applicability in real-world scenarios.

5.3 Results and Discussion

This section presents the results produced by the proposed dUA-VTFF method, along with a comparative analysis involving other learning schemes.

5.3.1 Evaluation of Models' Prediction Accuracy

Results for the two available datasets are summarized in Table 1 and Table 2, which depict the RAE metric between the original and the predicted number of vessels per grid cell and prediction time horizon. These results are provided for the testing set, for the top 10 grid cells, covering all the implemented methods.

Table 1: Prediction results (RAE) for different methods in the Aegean-Cyclades dataset

Method	Time prediction horizon (min)					
	5	10	15	20	25	30
dUA-VTFF	**0.5809**	**0.6278**	**0.7118**	**0.7897**	0.8090	**0.7916**
UA-VTFF (LSTM) [12]	0.7082	0.7058	0.7538	0.8162	0.8236	0.822
GRU [25]	0.6738	0.6635	0.7516	0.7952	**0.8054**	0.7957
SAE [24]	0.7271	0.721	0.7833	0.8692	0.8993	0.8976

Table 2: Prediction results (RAE) for different methods in the Brest dataset

Method	Time prediction horizon (min)					
	5	10	15	20	25	30
dUA-VTFF	**0.3043**	**0.3810**	**0.3978**	**0.4448**	**0.4655**	**0.5065**
UA-VTFF (LSTM) [12]	0.3318	0.4110	0.4441	0.4782	0.5264	0.5511
GRU [25]	0.3208	0.4053	0.4076	0.4556	0.4744	0.5073
SAE [24]	0.3487	0.4128	0.4328	0.4687	0.5291	0.5599

The proposed method dUA-VTFF consistently outperforms the other methods in terms of RAE across all prediction horizons in both datasets, except for the Aegean-Cyclades at the 25-min. prediction horizon. The effectiveness of dUA-VTFF is attributed to the carefully designed and proposed methodology, which relies on the utilization of Transformer neural networks. The attention mechanisms inherent in Transformers allow the model to capture intricate spatial and temporal dependencies, enhancing its predictive capabilities. This is particularly evident in shorter time horizons (5 to 15 min), where dUA-VTFF exhibits a significant performance advantage.

On the other hand, the GRU model [25] consistently outperforms the UA-VTFF (LSTM) [12]. This observation sheds light on the effectiveness of GRUs in the given context, possibly due to their ability to retain relevant information while mitigating vanishing gradient problems. Also, these results align with those reported in [24]. Additionally, the SAE method [24] yields the least favorable results, indicating limitations in its ability to capture the complexity of vessel dynamics in these scenarios compared to the other implemented techniques.

A critical consideration is the inherent differences in Transformer, LSTM, GRU, and SAE models. Transformers, with their self-attention mechanism, excel in capturing contextual dependencies across the entire sequence, showcasing their effectiveness in handling both sequential and spatial relationships. LSTM models, known for their memory cell and gating mechanisms, are proficient in capturing long-term dependencies but may face challenges in certain spatiotemporal contexts. GRUs, similar to LSTMs, offer advantages in capturing temporal dependencies but with a simplified structure, often proving effective in various sequence modeling tasks. In contrast, the SAE model relies on unsupervised learning through layer-wise pre-training and subsequent fine-tuning. While this method has shown success in certain applications, its inferior performance in our context suggests limitations in capturing the complexities of spatiotemporal vessel patterns.

As the prediction horizon extends from 20 to 30 min, dUA-VTFF's performance metrics become more comparable with those of the GRU model, suggesting considerations for task-specific time horizons. This shift emphasizes the importance of selecting models based on the specific characteristics and requirements of the prediction task. This implies a complex relationship between prediction accuracy and time horizon, necessitating further investigation into model behavior as temporal spans increase.

To assess the statistical significance of the performance differences observed among the various methodologies employed across the top 20 grids, we conducted a comprehensive statistical evaluation. Specifically, we compared the performance of dUA-VTFF against that of three alternative approaches: UA-VTFF, GRU, and SAE.

Table 3: Statistical Significance Tests for the Aegean-Cyclades dataset

Method compared with dUA-VTFF	T-statistic	p-value
UA-VTFF	-3.6752	0.0383
GRU	-3.3671	0.0442
SAE	-8.0712	0.0003

Table 3 presents the results of the conducted statistical significance analysis on the Aegean-Cyclades dataset.

As a preliminary step in our analysis, we conducted the Shapiro-Wilk test for normality on the performance differences observed between dUA-VTFF and each of the other methodologies. This test was chosen for its robustness in assessing the assumption of normal distribution in small to medium-sized datasets. Remarkably, in all three instances, the test failed to reject the null hypothesis, indicating that the differences in performance metrics were normally distributed.

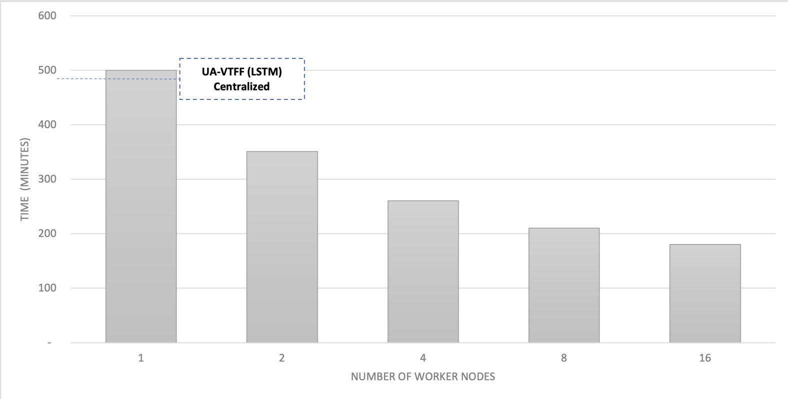
Given the confirmation of normal distribution, we proceeded with paired sample t-tests to directly compare the performance of dUA-VTFF against each of the other three methodologies. This statistical test was selected for its suitability in comparing the means of two related groups, where each observation in one group is uniquely paired with a corresponding observation in the other group. Such a pairing is particularly relevant in our context, where each grid serves as a unique scenario under which the performances of two methodologies are compared.

Across all comparisons -dUA-VTFF vs. UA-VTFF, dUA-VTFF vs. GRU, and dUA-VTFF vs. SAE-we observed statistically significant differences at the 0.05 significance level. In summary, the success of dUA-VTFF in surpassing other models lies in its ability to leverage Transformer NNs, offering improved accuracy in capturing complex spatiotemporal patterns, particularly in shorter prediction horizons. The contrasting performance of GRU, LSTM, and SAE models provides valuable insights into their suitability for this specific predictive task, contributing to a comprehensive understanding of their respective strengths and limitations.

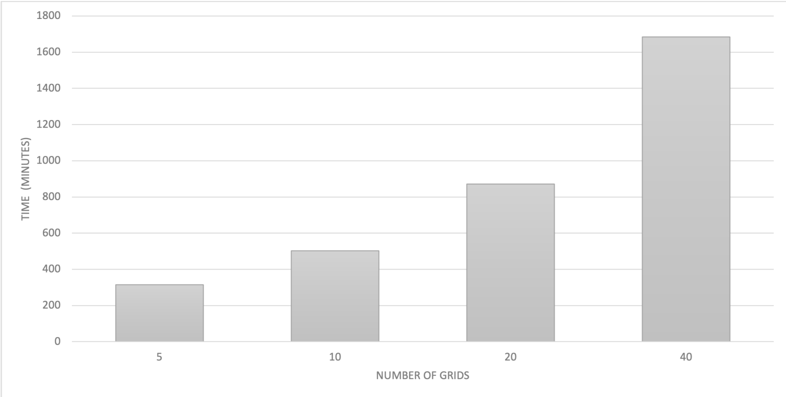
5.3.2 Evaluation of Models' Efficiency and Scalability

In relation to the scalability experiment, which involves investigating the framework's response to an increasing number of worker nodes, the results from Fig. 7a provide evidence of the system's scalability. However, it should be noted that preliminary scalability testing was conducted on a single node, whereas the scalability on a distributed system could be significantly different.

Starting with a single worker node, the processing time stood at 500 minutes. As the number of nodes doubled from 1 to 2, there was a noticeable reduction in processing time to 351 minutes. Further doubling the nodes to 4 led to a decrease in time to 261 minutes. This trend continued, with 8 nodes processing the data in 210 minutes and 16 nodes further reducing the time to 180 minutes. The diminishing returns on time saved as nodes increased confirmed the distributed framework's efficiency. In summary, initial results showed a significant decrease in processing times with an increase in the



(a)



(b)

Fig. 7: dUA-VTFF efficiency experiments in the Aegean-Cyclades dataset: (a) Scalability, (b) Increased data size

number of worker nodes, confirming the framework's ability to distribute the workload effectively and leverage parallel processing.

In relation to the dataset size experiment, which involves assessing the framework's performance with progressively larger datasets, the results from Fig.7b showcases the robustness of the dUA-VTFF method. A dataset size of 5 grids was processed in 320 minutes. However, as the dataset size doubled to 10 grids, the processing time also expanded to 500 minutes. This pattern of increasing processing times continued consistently with the growth of dataset sizes. Despite the inevitable increases in processing time with larger datasets, the framework exhibited resilience and consistent performance throughout. As anticipated, the distributed framework demonstrated its strength in managing large datasets resulting in consistent and efficient performance even with growing data sizes.

Furthermore, Table 4 presents the execution time in seconds for the dUA-VTFF and UA-VTFF methods in the Aegean dataset for the top 5 and 20 grids using

Table 4: Execution Time in the Aegean-Cyclades dataset

Method	Time (sec) Top 5 Grids (150 Epochs)	Top 20 Grids (150 Epochs)
dUA-VTFF	1252	4904
UA-VTFF	2086	8135

150 epochs. The dUA-VTFF approach demonstrates a gain in execution time of approximately 40% compared to the UA-VTFF technique.

In conclusion, these experiments highlighted the fundamental advantages of the big data distributed framework in addressing the VTFF problem. The dUA-VTFF method's competence in a distributed big data environment is evident since it not only scales effectively with increased computational resources but also manages vast data sizes with consistent performance.

6 Conclusion

Predicting the dynamic evolution of vessel traffic flow consistently and accurately is essential for supporting efficient vessel traffic management. This includes tasks such as collision avoidance, achieved by selecting routes that intersect areas with lower vessel density, among other critical functions. Inherently, the VTFF problem, which is characterized by complex spatiotemporal correlations, imposes high accuracy in prediction and short computational times. Towards this direction, we introduce the dUA-VTFF method, a novel approach to address the VTFF problem, learning from historical maritime data and predicting future traffic flows in a time horizon of up to 30 min. The dUA-VTFF approach effectively combines the power of the Transformer models and utilizes efficient distributed processing of the Apache Spark big data framework. Our dUA-VTFF takes as input historical timestamped vessel locations along with future vessel positions generated by a VRF model to create traffic flows. These data are arranged into a spatiotemporal grid and then each grid cell is allocated to a computing node through the Apache Spark, where Transformer models forecast traffic flows. Our experimental analyses, which were conducted using real AIS datasets, reveal that the dUA-VTFF method exhibits improved prediction accuracy compared to state-of-the-art traffic flow forecasting methodologies. Future work includes the investigation of a) weather impact on vessel traffic flow, b) tensor factorization analysis, and c) extreme traffic events to fine-tune the model. Moreover, we plan to extend our experiments to a distributed computing environment that more accurately reflects real-world distributed systems. This will involve employing multiple computational nodes, implementing distributed data management strategies, and considering network setups to simulate the complexities of a distributed system.

Acknowledgments. This work was supported by EU Horizon 2020 Programme VesselAI (Grant No 957237).

Data Availability. The Brest dataset analyzed during the current study is available in [39], https://doi.org/10.5281/zenodo.1167595. The Aegean-Cyclades dataset was provided by MarineTraffic within the scope of the VesselAI project (https://vessel-ai.eu/) and restrictions apply to the availability of these data, so are not publicly available.

Declarations. The authors declare that they have no competing interests.

References

[1] Papadimitrakis, M., Alexandridis, A.: A vessel propulsion controller based on economic model predictive control. In: 2023 American Control Conf., pp. 1825–1831 (2023). https://doi.org/10.23919/ACC55779.2023.10156575

[2] Chondrodima, E., Georgiou, H., Pelekis, N., Theodoridis, Y.: Particle swarm optimization and RBF neural networks for public transport arrival time prediction using gtfs data. Int. J. Inf. Manage. Data Ins. **2**(2), 100086 (2022) https://doi.org/10.1016/j.jjimei.2022.100086

[3] Papadimitrakis, M., Stogiannos, M., Sarimveis, H., Alexandridis, A.: Multi-ship control and collision avoidance using MPC and RBF-Based trajectory predictions. Sensors **21**(21), 6959 (2021) https://doi.org/10.3390/s21216959

[4] Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., Yin, B.: Deep learning on traffic prediction: Methods, analysis, and future directions. IEEE Trans. Intell. Trans. Syst. **23**(6), 4927–4943 (2022) https://doi.org/10.1109/TITS.2021.3054840

[5] Xiao, Z., Fu, X., Zhang, L., Goh, R.S.M.: Traffic pattern mining and forecasting technologies in maritime traffic service networks: A comprehensive survey. IEEE Trans. Intell. Transp. Syst. **21**(5), 1796–1825 (2020) https://doi.org/10.1109/TITS.2019.2908191

[6] Guo, S., Lin, Y., Li, S., Chen, Z., Wan, H.: Deep spatial–temporal 3d convolutional neural networks for traffic data forecasting. IEEE Trans. Intell. Trans. Syst. **20**(10), 3913–3926 (2019) https://doi.org/10.1109/TITS.2019.2906365

[7] Nguyen, D., Van, C.L., Ali, M.I.: Vessel trajectory prediction using sequence-to-sequence models over spatial grid. In: 12th ACM Int. Conf. DEBS, pp. 258–261 (2018)

[8] Li, H., Jiao, H., Yang, Z.: Ais data-driven ship trajectory prediction modelling and analysis based on machine learning and deep learning methods. Transp. Res. E: Logist. Transp. Rev. **175**, 103152 (2023)

[9] Dai, F., Huang, P., Mo, Q., Xu, X., Bilal, M., Song, H.: St-innet: Deep spatio-temporal inception networks for traffic flow prediction in smart cities. IEEE Trans. Intell. Transp. Syst. **23**(10), 19782–19794 (2022) https://doi.org/10.1109/TITS.2022.3179789

[10] Papadimitrakis, M., Alexandridis, A.: Active vehicle suspension control using road preview model predictive control and radial basis function networks. Appl. Soft Comput. **120**, 108646 (2022) https://doi.org/10.1016/j.asoc.2022.108646

[11] Alexandridis, A., Sarimveis, H., Ninos, K.: A Radial Basis Function network training algorithm using a non-symmetric partition of the input space–application to a model predictive control configuration. Adv. Eng. Softw. **42**(10), 830–837 (2011) https://doi.org/10.1016/j.advengsoft.2011.05.026

[12] Mandalis, P., Chondrodima, E., Kontoulis, Y., Pelekis, N., Theodoridis, Y.: Towards a unified vessel traffic flow forecasting framework. In: Proc. Workshop BMDA - EDBT/ICDT (2023)

[13] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)

[14] Gao, M., Wei, Y.: Traffic prediction based on a novel deep learning model: Tpbert. In: 2023 6th Int. Conf. ICAIBD, pp. 241–248 (2023). https://doi.org/10.1109/ICAIBD57115.2023.10206156

[15] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30** (2017)

[16] Huang, P., Chen, Q., Wang, D., Wang, M., Wu, X., Huang, X.: Tripleconvtransformer: A deep learning vessel trajectory prediction method fusing discretized meteorological data. Front. Environ. Sci. **10** (2022)

[17] Jiang, D., Shi, G., Li, N., Ma, L., Li, W., Shi, J.: Trfm-ls: Transformer-based deep learning method for vessel trajectory prediction. J. Mar. Sci. Eng. **11**(4) (2023) https://doi.org/10.3390/jmse11040880

[18] Zaman, M., Saha, S., Abdelwahed, S.: Assessing the suitability of different machine learning approaches for smart traffic mobility. In: 2023 IEEE Transp. Electrific. Conf. & Expo, pp. 1–6 (2023). https://doi.org/10.1109/ITEC55900.2023.10186901

[19] Yan, H., Ma, X., Pu, Z.: Learning dynamic and hierarchical traffic spatiotemporal features with transformer. IEEE Trans. Intell. Transp. Syst. **23**(11), 22386–22399 (2022) https://doi.org/10.1109/TITS.2021.3102983

[20] Zhu, L., Yu, F.R., Wang, Y., Ning, B., Tang, T.: Big data analytics in intelligent transportation systems: A survey. IEEE Trans. Intell. Transp. Syst **20**(1), 383–398 (2019) https://doi.org/10.1109/TITS.2018.2815678

[21] Petrou, P., Nikitopoulos, P., Tampakis, P., Glenis, A., Koutroumanis, N., Santipantakis, G.M., Patroumpas, K., Vlachou, A., Georgiou, H., Chondrodima, E.,

et al.: Argo: a big data framework for online trajectory prediction. In: 16th Int. Symp. SSTD, pp. 194–197 (2019). https://doi.org/10.1145/3340964.334098

[22] Tampakis, P., Chondrodima, E., Pikrakis, A., Theodoridis, Y., Pristouris, K., Nakos, H., Petra, E., Dalamagas, T., Kandiros, A., Markakis, G., Maina, I., Kavadas, S.: Sea area monitoring and analysis of fishing vessels activity: The i4sea big data platform. In: 21st IEEE Int. Conf. MDM, pp. 275–280 (2020). https://doi.org/10.1109/MDM48529.2020.00063

[23] Agafonov, A., Yumaganov, A.: Short-term traffic flow forecasting using a distributed spatial-temporal k nearest neighbors model. In: 2018 IEEE Int. Conf. CSE, pp. 91–98 (2018). https://doi.org/10.1109/CSE.2018.00019

[24] Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.-Y.: Traffic flow prediction with big data: A deep learning approach. IEEE Trans. Intell. Transp. Syst **16**(2), 865–873 (2015) https://doi.org/10.1109/TITS.2014.2345663

[25] Fu, R., Zhang, Z., Li, L.: Using lstm and gru neural network methods for traffic flow prediction. In: 2016 31st Youth Acad. Annual Conf. Chinese Assoc. Autom., pp. 324–328 (2016). https://doi.org/10.1109/YAC.2016.7804912

[26] Mandalis, P., Chondrodima, E., Kontoulis, Y., Pelekis, N., Theodoridis, Y.: Machine learning models for vessel traffic flow forecasting: An experimental comparison. In: 23rd IEEE Int. Conf. MDM, pp. 431–436 (2022). https://doi.org/10.1109/MDM55031.2022.00094

[27] Li, Y., Ren, H.: Vessel traffic flow prediction using lstm encoder-decoder. In: 5th Int. Conf. SPML, pp. 1–7. Association for Computing Machinery, New York, NY, USA (2022)

[28] Xiao, H., Zhao, Y., Zhang, H.: Predict vessel traffic with weather conditions based on multimodal deep learning. J. Mar. Sci. Eng. **11**(1), 39 (2023)

[29] Cai, L., Janowicz, K., Mai, G., Yan, B., Zhu, R.: Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. Trans. GIS **24**(3), 736–755 (2020)

[30] Fang, Y., Zhao, F., Qin, Y., Luo, H., Wang, C.: Learning all dynamics: Traffic forecasting via locality-aware spatio-temporal joint transformer. IEEE Trans. Intell. Transp. Syst. **23**(12), 23433–23446 (2022) https://doi.org/10.1109/TITS.2022.3197640

[31] Huang, L., Zhu, F., Li, Z.: Trendformer: Trend adaptive transformer for traffic flow prediction. In: 2022 5th Int. Conf. DSIT, pp. 1–6 (2022). https://doi.org/10.1109/DSIT55514.2022.9943858

[32] Chondrodima, E., Pelekis, N., Pikrakis, A., Theodoridis, Y.: An efficient lstm

neural network-based framework for vessel location forecasting. IEEE Trans. Intell. Transp. Syst. **24**(5), 4872–4888 (2023) https://doi.org/10.1109/TITS.2023.3247993

[33] Chondrodima, E., Mandalis, P., Pelekis, N., Theodoridis, Y.: Machine learning models for vessel route forecasting: An experimental comparison. In: 23rd IEEE Int. Conf. MDM, pp. 262–269 (2022). https://doi.org/10.1109/MDM55031.2022.00056

[34] Werbos, P.J.: Backpropagation through time: what it does and how to do it. Proc. IEEE **78**(10) (1990)

[35] Kingma, P.D., Ba, J.: Adam: A method for stochastic optimization. In: Int. Conf. ICLR (2015)

[36] Troupiotis-Kapeliaris, A., Zissis, D., Bereta, K., Vodas, M., Spiliopoulos, G., Karantaidis, G.: The big picture: An improved method for mapping shipping activities. Remote Sensing **15**(21), 5080 (2023) https://doi.org/10.3390/rs15215080

[37] Wang, X., Li, J., Zhang, T.: A machine-learning model for zonal ship flow prediction using ais data: A case study in the south atlantic states region. J. Mar. Sci. Eng. **7**, 463 (2019)

[38] Apache Spark. Accessed on April 10, 2024. https://spark.apache.org/docs/2.1.3/programming-guide.html

[39] Ray, C., Dréo, R., Camossi, E., Jousselme, A.-L., Iphar, C.: Heterogeneous integrated dataset for maritime intelligence, surveillance, and reconnaissance. Data in Brief **25**, 104141 (2019)

[40] Stogiannos, M., Papadimitrakis, M., Sarimveis, H., Alexandridis, A.: Vessel trajectory prediction using radial basis function neural networks. In: IEEE EUROCON 19th Int. Conf. Smart Tech. (2021). https://doi.org/10.1109/EUROCON52738.2021.9535562

[41] Zhang, M., Zhang, D., Fu, S., Kujala, P., Hirdaris, S.: A predictive analytics method for maritime traffic flow complexity estimation in inland waterways. Reliab. Eng. Syst. Saf. **220**, 108317 (2022)

[42] Russo, E., Silvestri, C., Simeoni, M., Soares, A., Matwin, S.: Predicting fishing effort and catch using semantic trajectories and machine learning. In: Multiple-Aspect Analysis of Semantic Trajectories: First International Workshop, MASTER 2019, Held in Conjunction with ECML-PKDD 2019, Würzburg, Germany, September 16, 2019, Proceedings, p. 83 (2020). Springer Nature

[43] Xiao, Z., Ponnambalam, L., Fu, X., Zhang, W.: Maritime traffic probabilistic

forecasting based on vessels' waterway patterns and motion behaviors. IEEE Trans. Intell. Transp. Syst. **18**(11), 3122–3134 (2017) https://doi.org/10.1109/TITS.2017.2681810

[44] Qu, J., Liu, R.W., Guo, Y., Lu, Y., Su, J., Li, P.: Improving maritime traffic surveillance in inland waterways using the robust fusion of ais and visual data. Ocean Engineering **275**, 114198 (2023)

[45] Manibardo, E.L., Laña, I., Ser, J.D.: Deep learning for road traffic forecasting: Does it make a difference? IEEE Trans. Intell. Transp. Syst. **23**(7), 6164–6188 (2022) https://doi.org/10.1109/TITS.2021.3083957

[46] Zhang, S., Guo, Y., Zhao, P., Zheng, C., Chen, X.: A graph-based temporal attention framework for multi-sensor traffic flow forecasting. IEEE Transactions on Intelligent Transportation Systems **23**(7), 7743–7758 (2022) https://doi.org/10.1109/TITS.2021.3072118

[47] Varga, B., Pereira, M., Kulcsár, B., Pariota, L., Péni, T.: Data-driven distance metrics for kriging-short-term urban traffic state prediction. IEEE Trans. Intell. Transp. Syst. **24**(6), 6268–6279 (2023) https://doi.org/10.1109/TITS.2023.3251022

[48] Sun, H.: Traffic flow prediction model based on attention spatiotemporal graph convolutional network. In: 2023 3rd Int. Symp. ISCTIS, pp. 148–153 (2023). https://doi.org/10.1109/ISCTIS58954.2023.10213060

[49] Zhao, Y., Lin, Y., Wen, H., Wei, T., Jin, X., Wan, H.: Spatial-temporal position-aware graph convolution networks for traffic flow forecasting. IEEE Trans. Intell. Transp. Syst. **24**(8), 8650–8666 (2023) https://doi.org/10.1109/TITS.2022.3220089