Recent deep learning methods for vessel trajectory prediction are able to learn complex maritime patterns from historical automatic identification system (AIS) data and accurately predict sequences of future vessel positions with a prediction horizon of several hours. However, in maritime surveillance applications, reliably quantifying the prediction uncertainty can be as important as obtaining high accuracy. This article extends deep learning frameworks for trajectory prediction tasks by exploring how recurrent encoder-decoder neural networks can be tasked not only to predict but also to yield a corresponding prediction uncertainty via Bayesian modeling of aleatoric and epistemic uncertainties. We compare the prediction performance of two different models based on labeled or unlabeled input data to highlight how uncertainty quantification and accuracy can be improved by using, if available, additional information on the intention of the ship (e.g., its planned destination).

1. INTRODUCTION

Trajectory prediction—for collision avoidance, anomaly detection, and risk assessment—is a crucial functional component of intelligent maritime surveillance systems and next-generation autonomous ships. Maritime surveillance systems are increasingly relying on the huge amount of data made available by terrestrial and satellite networks of automatic identification system (AIS) receivers. The availability of maritime big data enables the automatic extraction of spatial-temporal mobility patterns that can be processed by modern deep learning networks to enhance trajectory forecasting.

Today, most commercial systems primarily rely on trajectory prediction methods based on the nearly constant velocity (NCV) model, since this linear model is simple, fast, and of practical operability to perform short-term predictions of straight-line trajectories [2]. However, the NCV model tends to overestimate the actual uncertainty as the prediction horizon increases. A novel linear model, based on the Ornstein–Uhlenbeck (OU) stochastic process, is becoming recognized as a reliable means to improve long-term predictions [3], with special focus on uncertainty reduction via estimation of current navigation settings. The OU model has also been adopted for maritime anomaly detection [4], [5] and unsupervised extraction of maritime patterns [6]. Although most maritime traffic is very regular, and thus, model-based methods can be easily applied, in the presence of maneuvering behaviors of the ship such models will tend to lack the desired prediction accuracy. In such cases, nonlinear and data-driven methods including adaptive kernel density estimation [7], nonlinear filtering [8], [9], nearest-neighbor search methods [10], [11], and machine learning techniques [12], may provide more suitable solutions. Furthermore, the latest advances in deep learning-based predictive models and the combined availability of large volumes of AIS data allow for enhanced vessel trajectory prediction and maritime surveillance. Recent approaches based on deep learning are documented in [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]. In [13], a variational recurrent neural network (RNN) architecture is proposed to extract knowledge on the hidden behavior of ships from AIS data streams for track reconstruction and anomaly detection. In [14], recurrent neural models are adopted to process sequential AIS data with the final goal of associating incoming messages with existing tracks. For vessel trajectory prediction, previous work on RNN-based methods proposed a single unidirectional long short-term memory (LSTM) [15], bidirectional LSTM [16], [17], and gated recurrent unit (GRU) [18] RNN to map a sequence of inputs to an output sequence of the same length. In [19], an autoencoder approach is used to predict entire vessel trajectories (with prediction horizons up to 30 min), instead of sequential states, naturally based on the requisite upstream clustering and classification steps. Further works addressed the problem of modeling the interactions between
vessels, e.g., by using a generative adversarial network with interaction module [20] or multigraph convolutional neural networks [21]. In addition, spatial grid-based approaches have been proposed [22], [23], where the transition of vessel states is defined between cells on the grid, rather than between time intervals of AIS observations, and deep learning is applied to predict vessel inflow and outflow [22], or maritime trajectories with estimated destination and arrival time [23]. Recently, encoder–decoder models [24], [25] have proven to be an effective approach for training neural networks to map an input sequence of AIS positions to an output sequence which, differently from standard RNN models [15], [16], [17], [18], is not necessarily of the same length. This is due to the fact that encoder–decoder models use a first RNN to encode the input sequence into a semantic summary or context, which is given as input to a second RNN to generate (decode) the output sequence step-by-step, conditioned on the encodings. The context is generated to convey only the essence of the input to the decoder, and aggregation functions, such as max and average pooling over time or the attention mechanism [25], can be introduced to learn how to associate elements of the context to elements of the output sequence. Besides the ability to handle variable-length input and output sequences, another key benefit of encoder–decoder models is the ability to train a single end-to-end model, where the encoder and decoder RNNs are trained jointly to maximize the prediction performance. However, the aforementioned deep learning models cannot provide predictive uncertainty, hence no quantification of the confidence with which the prediction outputs can be trusted is available.

To accompany results from deep learning by their associated confidence levels has recently arisen to be of heightened importance, and naturally there has been significant research attention toward that goal. Data-driven methods for uncertainty quantification have been recently proposed to estimate model and data uncertainty based on ensemble [26] or Bayesian [27], [28], [29], [30], [31] learning. Bayesian deep learning methods apply Bayesian modeling and variational inference to neural networks, leading to Bayesian neural networks (BNNs) that treat the network parameters as random variables instead of deterministic unknowns to represent the model uncertainty on its predictions. BNNs can capture the uncertainty within the learning model, while maintaining the flexibility of deep neural networks, and hence they are particularly appealing for safety-critical applications (e.g., autonomous transportation systems, robotics, medical and space systems), where uncertainty estimates can be propagated in an end-to-end neural architecture to enable improved decision-making. In particular, Gal and Ghahramani [27] showed that dropout, a well-known regularization technique to prevent neural network overfitting [32], can be used as a variational Bayesian approximation of the predictive uncertainty in existing deep learning models trained with standard dropout, by performing Monte Carlo (MC) sampling with dropout at test time, i.e., by sampling the network with random omission of units representing MC samples obtained from the posterior distribution over models. In [28], a Bayesian deep learning approach for the combined quantification of both data and model uncertainty, extracted from BNNs, is proposed for computer vision applications. In addition, a neural network architecture based on LSTM with uncertainty modeling is proposed in [33] to incorporate non-Markovian dynamic models in the prediction step of a standard Kalman filter for target tracking.

In this article, we present a vessel trajectory learning and prediction framework to generate future vessel trajectory samples given the sequence of the latest AIS observations. The proposed method is built upon the LSTM encoder–decoder architecture presented in [24] and [25], which has emerged as an effective and scalable model for sequence-to-sequence learning of maritime trajectories. We extend [24] and [25] by providing a practical quantification of the predictive uncertainty via Bayesian learning tools based on MC dropout [27]. Preliminary work on trajectory prediction with uncertainty quantification applied to the maritime domain was presented in [34].

In this work, we extend [34] by providing a comprehensive description of the prediction uncertainty modeling, a detailed introduction of the variational LSTM model used to implement our encoder–decoder architecture, and by proposing an alternative decoder scheme with a novel regularization method of the information about the vessel’s intention, which may be available in the maritime surveillance data. Moreover, we present novel results on the estimated predicted variance, and a performance comparison demonstrating the gain in state prediction accuracy with respect to [34]. Fig. 1 shows an example of how the proposed
model is able to predict future trajectories and prediction uncertainty given input sequences from past data.

To summarize, the main contributions of this work are as follows:

1) A model to quantify the predictive uncertainty of trajectory future states provided by encoder–decoder RNNs using Bayesian deep learning tools.
2) A novel regularization method for the decoding phase to prevent complex coadaptations between the encoded data and the high-level information about the vessel’s intention.
3) Experimental results on real-world AIS data showing the effectiveness of the proposed encoder–decoder architecture with uncertainty modeling in learning trajectory predictions with well-quantified uncertainty estimates using labeled and unlabeled data.

The rest of this article is organized as follows. In Section II, we introduce the vessel trajectory prediction problem. In Section III, we present the proposed attention-based encoder–decoder framework for trajectory prediction with uncertainty. Experimental results on a real-world AIS dataset are presented and discussed in Section IV. Finally, Section V concludes this article.

II. PROBLEM FORMULATION

We formulate the vessel trajectory prediction problem as a supervised learning process by following a sequence-to-sequence deep learning approach to directly generate the distribution of an output sequence of future states given an input sequence of past AIS observations. From a probabilistic perspective, the objective is to determine the following predictive distribution:

$$p(y^*_{1:h} | x^*_{1:t}, D),$$

which represents the probability of an output sequence $y^*_{1:h} \triangleq \{s_i^h\}_{k=1}^h \in \mathbb{R}^{d \times h}$ of $h \geq 1$ future $d$-dimensional states of the vessel given a new input sequence $x^*_{1:t} \triangleq \{s_i^l\}_{k=1}^l \in \mathbb{R}^{d \times l}$ of $l \geq 1$ observed states, and the available training data $D = \{X^i, Y^i, \Psi^i\}_{i=1}^N$ containing the two sets $X^i = \{x^i_{1:l}\}_{l=1}^N$ and $Y^i = \{y^i_{1:h}\}_{h=1}^N$ of training input and, respectively, output sequences. Note that each sequence element $s^i_k \triangleq s^i(t_k) \in \mathbb{R}^d$ represents the vessel’s position in latitude and longitude coordinates, taken from the available time-stamped AIS messages. In many practical cases, it is also possible to exploit some additional information available from AIS data, such as the vessel destination. The destination port is an example of voyage-related information provided by the AIS that may be relevant to anticipate a vessel trajectory. We denote this (possibly available) input feature, which may be salient to predict the $i$th output sequence, by $\Psi^i = \{\psi^i\}_{i=1}^N$, where $\psi^i \in \{0, 1\}^v$ is the $v$-way categorical feature for each trajectory $i$ representing the class label of the specific motion pattern encoded into a one-hot vector of size $v$ [25]. For example, $\psi^i = [0, 0, 1]$ would mean that this particular vessel followed a trajectory that has been labeled #3 based on three possible destinations.

A. Modeling Prediction Uncertainty

Uncertainty on the prediction estimates can be captured with recently developed Bayesian tools, which offer a practical framework for representing uncertainty in deep learning models [28], [29], [30]. In the context of supervised learning, two forms of uncertainty, i.e., aleatoric and epistemic uncertainty are considered, where aleatoric is the irreducible and epistemic the reducible part of uncertainty with respect to the available training data [28]. Aleatoric (or data) uncertainty captures noise inherent in the observations, whereas epistemic (or model) uncertainty accounts for uncertainty in the neural network model parameters. Epistemic uncertainty is a particular concern for neural networks given their many free parameters, and can be large for data that is significantly different from the training set. Thus, for any real-world application of neural network uncertainty estimation, it is critical that it be taken into account. We follow a combined aleatoric–epistemic model [28] to capture both aleatoric and epistemic uncertainty in our prediction model.

Following a Bayesian framework [35] with prior distributions placed over the parameters of the NN, epistemic uncertainty can be captured by learning a distribution of NN models $p(F | D)$ representing the posterior distribution over the space of functions $y_{1:h} = F(x_{1:t})$ that are likely to have generated our dataset $D$. The predictive probability (1) is then obtained by marginalizing over the implied posterior distribution of models, i.e.,

$$p(y^*_{1:h} | x^*_{1:t}, D) = \int p(y^*_{1:h} | x^*_{1:t}, F) p(F | D) dF. \quad (2)$$

In our case, $F$ are RNN encoder–decoder models [25] assumed to be described by a finite set of parameters $\theta$, such that

$$p(y^*_{1:h} | x^*_{1:t}, D) = \int p(y^*_{1:h} | x^*_{1:t}, \theta) p(\theta | D) d\theta. \quad (3)$$

Since $p(\theta | D)$ cannot be obtained analytically, it can be approximated by using variational inference with approximating distribution $q(\theta)$, which allows for efficient sampling. This results in the approximate predictive distribution

$$q(y^*_{1:h} | x^*_{1:t}, D) = \int p(y^*_{1:h} | x^*_{1:t}, \theta) q(\theta) d\theta, \quad (4)$$

which can be kept as close as possible to the original distribution by minimizing the Kullback–Leibler (KL) divergence between $q(\theta)$ and the true posterior $p(\theta | D)$ during the training stage.

Let us consider a generic NN architecture with $L$ transformation layers, and denote by $W_l$ the weight matrix of size $K_l \times K_{l-1}$ for each layer $l = 1, \ldots, L$. Then, following [36], by setting the set of weight matrices\(^1\) of the NN architecture as the set of parameters, i.e., $\theta = \{W_l\}_{l=1}^L$, and by using a Bernoulli approximating variational distribution, it is possible to relate variational inference in Bayesian NNs

\(^1\)All bias terms are omitted to simplify the notation.
to the dropout mechanism. In particular, the approximating distribution can be defined for each row $i$ of $W_i$ as
\[ q(w_i) = \gamma N(w_i; \theta, \rho^2 I) + (1 - \gamma) N(w_i; m_i, \rho^2 I), \tag{5} \]
where $m_i$ is the vector of variational parameters, $\rho$ the standard deviation of the Gaussian prior distribution placed over $w_i$, and $\gamma$ the Bernoulli probability used for dropout. Then, evaluating the model output $F_{\hat{\theta}}$ with $\hat{\theta} \sim q(\theta)$ corresponds to performing dropout by randomly masking rows in the weight matrix $W_i$ during the forward pass. Predictions can then be obtained by performing MC dropout [27] which consists of executing dropout at test time and averaging results for all $M$ samples, i.e., by approximating (4) as
\[ q(y_{1:h}^*|x_{1:k}, D) \approx \frac{1}{M} \sum_{j=1}^{M} p(y_{1:h}^*|x_{1:k}, \hat{\theta}_j), \tag{6} \]
with $\hat{\theta}_j \sim q(\theta)$. Note that in the case of MC dropout for RNNs, the same parameter realizations $\hat{\theta}_j$ are used for each time step of the input sequence [36].

As shown in [28], aleatoric uncertainty can be modeled together with epistemic uncertainty by estimating the sufficient statistics of a given distribution describing the measurement noise of data. By fixing a Gaussian likelihood to model aleatoric uncertainty, the predictive distribution of an output sequence $\hat{y}_{1:h}$ for a given input sequence $x_{1:k}$ can be approximated by a sequence of multivariate Gaussian distributions, where each output is a Gaussian $N(\hat{y}_k; \hat{\Sigma}_k, k = 1, \ldots, h$.

B. Aleatoric and Epistemic Uncertainty

The supervised learning task (1) can be recast as a sequence regression problem [37], which aims at training a neural network model $F$ to predict, given an input sequence $x_{1:k}$ of length $k$, the predictive mean $\hat{y}_{1:h}$ and the predictive covariance $\Sigma_{1:h}$, i.e.,
\[ \hat{y}_{1:h}, \Sigma_{1:h} = F_\theta(x_{1:k}), \tag{7} \]
where $F$ is a BNN parameterized by model weights $\theta$ drawn from the approximate dropout variational distribution $\theta \sim q(\theta)$. Note that a single network can be used to predict both $\hat{y}_{1:h}$ and $\Sigma_{1:h}$. By setting the distribution modeling aleatoric uncertainty as Gaussian, we induce a minimization objective of the loss function $L_{1:h} \triangleq L(y_{1:h})$, here defined via the negative log-likelihood
\[ L_{1:h} = \frac{1}{2} \sum_{k=1}^{h} (y_k - \hat{y}_k)^T \Sigma_k^{-1} (y_k - \hat{y}_k) + \frac{1}{2} \ln |\Sigma_k|, \tag{8} \]
which enables simultaneous training of both $\hat{y}_k$ and $\Sigma_k$ in an end-to-end optimization process of the entire vessel prediction and uncertainty estimation framework. Thus, for each state of the predicted sequence, the network outputs a Gaussian distribution parameterized by $\hat{y}_k$ and $\Sigma_k$ such that the negative log-likelihood in (8) of the ground truth vessel positions $y_k$, $k = 1, \ldots, h$ over all training data is as small as possible.

If the epistemic uncertainty can be estimated through MC dropout [27], [28] from $M$ samples $\{\hat{y}_1^j, \Sigma_1^j\}_{j=1}^{M}$, then the total predictive uncertainty for the output $\hat{y}_k$ can be approximated as follows:
\[ \Sigma_k = \frac{1}{M} \sum_{j=1}^{M} \hat{\Sigma}_k^j - \frac{1}{M} \sum_{j=1}^{M} \hat{y}_k^j \hat{\Sigma}_k^j \hat{y}_k^j \tag{9} \]
where the first two terms of the sum correspond to the epistemic uncertainty, and the third corresponds to the aleatoric uncertainty. Finally, we average the uncertainty across time steps $k = 1, \ldots, h$ to obtain the uncertainty estimate of the complete output sequence.

III. LEARNING TO PREDICT UNDER UNCERTAINTY

We extend the encoder–decoder architecture based on recurrent networks with attention proposed for vessel trajectory prediction in [24] and [25] by providing a practical quantification of the total (i.e., comprising both model and data: epistemic and aleatoric) predictive uncertainty following a Bayesian deep learning approach.

As shown in Fig. 2, the proposed encoder–decoder architecture consists of a bidirectional encoder RNN that reads a sequence of vessel positions one state at a time encoding the input information into a sequence of hidden states, and a decoder RNN that generates an output sequence of future states step-by-step conditioned on a context representation of the input sequence generated from the encoder’s...
hidden states through an attention aggregation layer. Moreover, we show that, if available, additional information on the intention of the vessel can be exploited to generate the decoder’s outputs.

The RNN encoder–decoder architecture consists of an encoder model with \( \theta_E \) parameters and a decoder model with \( \theta_D \) parameters. Each model is implemented as a fully trainable RNN containing different weight matrices. We define the Bernoulli variational distribution \( q(\theta) \) over the union of all the weight matrices of our architecture, i.e., \( \theta = \{ \theta_E, \theta_D \} \). In the next sections we will see how the full model learns to predict future trajectories under uncertainty.

### A. Variational LSTM

LSTM networks are a special kind of RNN capable of capturing long temporal patterns in sequential data. They work well on a large variety of problems [24], [25], [38]. A Bayesian view of RNNs has been proposed in [36], where the authors interpret LSTM as probabilistic models considering the network weights as variables trainable by suitable likelihood functions. This has been shown to be equivalent to implementing a novel variant of dropout for RNN models to approximate the posterior distribution over the weights with a mixture of Gaussians leading to a tractable optimization objective.

Following [36], we extend the deterministic LSTM architecture implementing dropout on input or output, and recurrent connections, with the same network units dropped at each time step. In this article, the LSTM architecture maps the input sequence \( x_1, \ldots, x_t \) into a sequence of cell activation and hidden states by applying the \textit{tied-weights} LSTM parameterization in [36]

\[
\begin{bmatrix}
\tilde{i}_t \\
\tilde{f}_t \\
\tilde{o}_t \\
\tilde{g}_t
\end{bmatrix} =
\begin{bmatrix}
W_i & U_i \\
W_f & U_f \\
W_o & U_o \\
W_g & U_g
\end{bmatrix}
\begin{bmatrix}
x_t \\
h_{t-1}
\end{bmatrix}
= \begin{bmatrix}
W_i x_t + U_i h_{t-1} \\
W_f x_t + U_f h_{t-1} \\
W_o x_t + U_o h_{t-1} \\
W_g x_t + U_g h_{t-1}
\end{bmatrix},
\]

where \( W_i, W_f, W_o, W_g \) are all \( p \)-by-\( d \) weight matrices; and \( U_i, U_f, U_o, U_g \) have dimension \( p \) by \( p \); \( d \) is the number of input features and \( p \) is the dimension of the (unidirectional) hidden state, i.e., \( h_{t-1} \in \mathbb{R}^p \). The \textit{input}, \textit{forget}, \textit{output}, and \textit{input modulation} gates are \( i_t \), \( f_t \), \( o_t \), and \( g_t \), respectively, and can be computed as

\[
\begin{align*}
i_t &= \text{sigm}(\tilde{i}_t), \\
f_t &= \text{sigm}(\tilde{f}_t) \\
o_t &= \text{sigm}(\tilde{o}_t), \\
g_t &= \tanh(\tilde{g}_t).
\end{align*}
\]

Finally, the cell activation state and hidden state vectors, respectively, \( c_t \) and \( h_t \), are

\[
\begin{align*}
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t),
\end{align*}
\]

where \( \odot \) denotes the Hadamard product. Note that the matrix \( W_R \in \mathbb{R}^{(4p) \times (d+p)} \) is a compact representation of all the weight matrices of the four LSTM gates.

The LSTM model can be viewed as a probabilistic model by treating the network weights \( W_R \) as random variables. Then, to perform approximate variational inference in the probabilistic LSTM model (referred to as \textit{Variational LSTM}), we apply the dropout mechanism with Bernoulli approximating variational distribution (5) over the weights \( W_R \) of the LSTM parameterization (10) as

\[
\begin{bmatrix}
\tilde{i}_t \\
\tilde{f}_t \\
\tilde{o}_t \\
\tilde{g}_t
\end{bmatrix} =
\begin{bmatrix}
W_R \circ x_t, d_i \\
W_R \circ h_{t-1}, d_f \\
W_R \circ h_{t-1}, d_o \\
W_R \circ h_{t-1}, d_g
\end{bmatrix},
\]

which corresponds to randomly zeroing rows in \( W_R \) through random masks \( d_i \) and \( d_f \) acting on the input and hidden state, respectively. Note that, following the dropout variant for RNNs [36], in our LSTM setting with an input sequence, each row of the weight matrix is randomly masked once, and the same dropout masks \( d_i \) and \( d_f \) are used at each time step to drop the same network units over time. In the following sections, we will see how to apply the above variational LSTM (VarLSTM) model to our trajectory prediction task under uncertainty using an encoder–decoder RNN architecture.

### B. Encoder Network

The encoder network is designed as a bidirectional RNN [39] to capture and analyze the temporal patterns in vessel trajectory both in the positive and negative time directions simultaneously. Following the encoder architecture used in [25], we use two VarLSTM to encode the input sequence using one model for each time direction.

In our VarLSTM encoder implementation, the dropout mask \( d_o \) is applied only to the recurrent connections, therefore it does not perturb the input vessel trajectory. In the end, the trainable parameters are \( \theta_E = \{ W_R, \hat{W}_R \} \), one weight matrix for each VarLSTM.

The bidirectional VarLSTM maps the input sequence into two different temporal representations: 1) the forward hidden sequence by iterating the forward layer from \( t = 1 \) to \( \ell \), and 2) the backward hidden sequence by iterating the backward layer from \( t = \ell \) to 1. In this way, the encoder network is able to learn long-term patterns in both temporal directions. The output layer of the encoder is then updated into a compact hidden state representation obtained by concatenating the forward and backward hidden states, i.e., \( h_t = \hat{h}_t + \bar{h}_t \), computing the output vectors of the encoder layer \( \hat{h}_t, \ldots, \hat{h}_\ell \) for an input sequence of length \( \ell \). Each element \( \hat{h}_t \) encodes bidirectional spatio-temporal information of the input sequence extracted from the states of the vessel preceding and following the \( t \)-th component of the sequence.
C. Attention-Based Decoder Network With Uncertainty

In the proposed architecture, the decoder RNN is trained to learn the following conditional probability:

\[
p(y_1, \ldots, y_h|x_1, \ldots, x_t) = \prod_{t=1}^{h} p(y_t|y_1, \ldots, y_{t-1}, z_t, \psi),
\]

(14)

where each factor in (14) can be modeled by an RNN \(D\), with trainable parameters \(\theta_D\), of the form

\[
p(y_t|y_1, \ldots, y_{t-1}, z_t) = D(y_{t-1}, s_{t-1}, z_t; \theta_D).
\]

(15)

The decoder (15) generates the next kinematic state \(\hat{y}_t\) given the context vector \(z_t\), the possibly available information on the high-level intention \(\psi\) of the vessel, the decoder’s hidden state \(s_{t-1}\), and the previous predictions \(\hat{y}_{t-1}\), which are fed back into the model in a recursive fashion as additional inputs to predict further into the future. In addition, (15) is initialized by setting

\[
s_0 = g(h_{t-1}, \psi)
\]

(16)

to map the last encoder (forward) hidden state and the intention into the initial decoder hidden state \(s_0\). Note that, similar to [25], this work partially addresses the multimodal nature of the prediction task with the use of the vessel’s intention (i.e., destination) \(\psi\). However, different from [25] and as an additional measure to avoid overfitting, here the intention information is used only to initialize the decoder through (16), which in this case takes the form

\[
s_0 = \tanh(W_\psi \eta)
\]

(17)

with \(\eta = h_{t-1} \oplus \psi, W_\psi\) being the trainable parameters\(^2\) of the neural network (16).

The attention mechanism [40] is adopted as an intermediate layer between the encoder and the decoder to learn the relationship between the observed and the predicted kinematic states while preserving the spatio-temporal structure of the input. We extend the attention module of [40] implemented in [25] by applying a random dropout mask repeated at all time steps \(d_t\) to the input hidden features, previously computed by the encoder network. This is achieved by allowing the context representation to be a set of fixed-size vectors, or context set \(z = [z_t]_{t=1}^{T}\), where each context vector \(z_t\) in (15) can be computed as a weighted sum of the encoded input states, i.e., \(z_t = \sum_{j=1}^{T} \alpha_{tj} h_j \odot d_t\), where \(\alpha_{tj} = \exp(e_{tj})/\sum_{k=1}^{T} \exp(e_{tk})\) represents the attention weight, and \(e_{tj} = a(s_{t-1}, h_j, \odot d_t; W_a)\) is a variational neural network with parameters \(W_a\) and dropout mask \(d_t\). The variational attention network is trained jointly with the prediction model to quantify the level of matching between the inputs around position \(j\) and the output at position \(t\) based on the \(j\)th encoded input state \(h_j\) and the decoder’s state \(s_{t-1}\) (generating the \(t\)th output).

In order to deal with uncertainty, the decoder (15) is implemented as a unidirectional VarLSTM which generates the sequence of future predicted distributions by iterating \(\forall t = 1, \ldots, h\)

\[
c_t = \hat{y}_{t-1} \oplus z_t
\]

(18)

\[
s_t = \text{VarLSTM}(c_t, s_{t-1}; W_R)
\]

(19)

\[
\hat{y}_t = W_\mu s_t
\]

(20)

\[
[\log b_{11}, \log b_{22}, b_{21}]^T = W_\Sigma s_t,
\]

(21)

where \(\hat{y}_t = x_t, z_t\) is the context vector computed through the attention mechanism, and \(W_\mu\) and \(W_\Sigma\) are trainable parameters\(^2\) of a single neural network mapping the VarLSTM output \(s_t\) to the parameters in (20)–(21) used to estimate the predictive uncertainty at time step \(t\), here modeled as a multivariate Gaussian distribution \(N(\hat{y}_t, \Sigma_t)\). Note that, while the predictive mean \(\hat{y}_t\) can be directly obtained through (20), to stabilize training and enforce positive-definite predictive covariance matrices, the network (21) is trained to predict the elements of a lower triangular matrix with real and positive diagonal entries \(B = \begin{bmatrix} b_{11} & 0 \\ b_{21} & b_{22} \end{bmatrix}\) such that

\[
\Sigma_t = BB^T = \begin{bmatrix} \sigma_{l,x}^2 & \sigma_{l,x} \sigma_{l,y} \rho_l \\ \sigma_{l,x} \sigma_{l,y} \rho_l & \sigma_{l,y}^2 \end{bmatrix}
\]

(22)

is guaranteed to be positive-definite by virtue of the Cholesky decomposition (22). Note that \(\sigma_{l,x}\) and \(\sigma_{l,y}\) in (22) denote the components of the predictive variance \(\sigma_t\) along the \(x\) and, respectively, \(y\) direction. Notice also that in the proposed VarLSTM decoder implementation, the \(d_t\), dropout mask is applied only on the recurrent connections.

This end-to-end solution is trained by the stochastic gradient descent algorithm in order to learn an optimal function approximation (7). A decoder (15) with such a recursive structure offers the advantage of being able to handle sequences of arbitrary length. The model consists of a set of trainable parameters \(\theta_D = [W_\psi, W_R, W_a, W_\mu, W_\Sigma]\), respectively, used to model the decoder initialization (17), the unidirectional VarLSTM (19), the attention mechanism, and the output distribution given by (20)–(21).

D. Intention Regularization

Despite the proposed decoder initialization (17), the encoder–decoder architecture may overfit the intention information, and predict erroneous future maneuvers of the vessel. This is due to the fact that the model tends to generate predictions that are mainly based on the high level intention rather than on the information encoded in the past trajectory. To avoid this behavior, and inspired by the dropout mechanism [32], we apply a regularization technique based on dropout noise [41] by randomly masking the intention information at train time in order to prevent complex coadaptations between the encoded past trajectory and the high-level information.

In particular, we apply a random dropout noise to the intention at each iteration of the training procedure by feeding the intention information \(\psi\) into (17) with some

---

\(^2\)Again, all biases are omitted for simplicity.
probability $q$ (a predefined hyperparameter), or setting it to zero otherwise. Thus, the decoder initialization (17) takes the following form:

$$ s_0 = \tanh(W_\psi(\eta \odot d_\eta)) $$

(23)

where $d_\eta = \frac{p+v}{p}v$, $v \in \{0, 1\}^{p+v}$ is a random binary mask on the input information such that $v_i = 1, i = 1, \ldots, p$, and $v_i = v \sim \text{Bern}(1 - \gamma)$, $i = p + 1, \ldots, p + v$. The random variable $v$ is Bernoulli-distributed with parameter $1 - \gamma$, i.e., it takes the value 0 with probability $\gamma$ for each training sample of the intention components of $\eta$, otherwise it is 1. In other words, this random mask is applied to $\eta$ in order to regularize via dropout noise only the intention information $\psi \in \{0, 1\}^p$, while leaving the encoded feature $\overrightarrow{h}_i \in \mathbb{R}^p$ untouched. Then, a scaling factor $\frac{p+v}{p}$ is applied to the random mask at train time, while leaving the forward pass at test time unchanged. Note that this prescaling performed at train time, commonly referred to as the inverted dropout implementation, does not require any changes to the network to compensate for the absence of information during test time, as usually done in traditional dropout [32].

This additional regularization of the intention information can be viewed as a multimodal dropout method [42], in which the input features belonging to the same group (or modality) are either all dropped out or all preserved to avoid false coadaptations between different groups of input information, and to handle missing data in one of the groups at test time. In our case, this has been shown to improve the prediction performance by preventing coadaptations between the encoded past observations and the possibly available intention information. The improvements in performance with respect to a previous version of the labeled architecture [34] are provided in Appendix A for a varying probability $\gamma$.

IV. EXPERIMENTS

In this section we describe the experiments carried out to evaluate the proposed encoder–decoder architecture for trajectory prediction with uncertainty.

A. Dataset Benchmark

We used an AIS dataset extracted from the historical data made freely available by the Danish Maritime Authority (DMA) [1], comprising 394 trajectories of tanker vessels belonging to two specific motion patterns of interest in the period ranging from January to February 2020. The complete dataset used in this article is illustrated in Fig. 3. A detailed description of the dataset preparation method can be found in [25, Sec. VI-A], where a flowchart representing all the data preprocessing steps is also available as [25, Fig. 5]. Paths falling into the same maritime pattern are described as sequences of positions in planar coordinates assigned through the Universal Transverse Mercator (UTM) projection (zone 32 V), and correlated by voyage-related attributes including departure and destination areas.

In order to provide regular input sequences to train the model in a supervised fashion, we applied a temporal interpolation to each trajectory using a fixed sampling time of $\Delta = 15$ min to resample the data, and data segmentation by using the sliding window approach to produce fixed-length input and output sequences of length $\ell = h = 12$ vessel positions (i.e., 3 h).

B. Models

We propose an encoder–decoder architecture with attention mechanism composed of a BiLSTM encoder layer with 64 hidden units, and an LSTM decoder layer with 64 hidden units. For the Bayesian modeling of the epistemic uncertainty, we used MC dropout [27] with $M = 100$ samples, and dropout rate applied to recurrent connection $p = 0.05$ in both encoder and decoder layers. The model was trained by applying AdamW optimizer [43] with a learning rate of 0.0001 and weight decay of 0.0001 to minimize the mean absolute error loss function, an early-stopping rule with 3000-epoch patience, and a mini-batch size of 200 samples.

As shown in [7] and [25], the high-level pattern information $\psi$ representing the vessel’s intended destination can be valuable to train a better predictive model. Inspired by [7] and [25], we compare two fundamentally different approaches to account for the vessel behavior based on labeled or unlabeled trajectories, where in the labeled case the neural model is trained to exploit also $\psi$ to perform the prediction task. The encoder–decoder architecture for the labeled case is shown in Fig. 2, where the decoder hidden state is initialized using (16). In particular, we define the following two approaches:
Labeled (L): We train the predictive model and perform prediction using labeled data, i.e., using low-level context representation encoded from a sequence of past observations, as well as additional inputs about high-level intention behavior of the vessel. This model includes the intention regularization mechanism with dropout probability $\gamma = 0.3$, shown to be the highest-performing model in Appendix A.

Unlabeled (U): We train the predictive model and perform prediction using unlabeled data, i.e. using only low-level context representation encoded from a sequence of past observations, without any high-level information about the motion pattern. We adapt the model proposed in Section III by removing the pattern descriptor $\psi$ from the inputs of the decoding network. This results in a simplified version of (16), where only the last encoder (forward) hidden state (or context) is mapped into the initial decoder hidden state, i.e. $s_0 = g(h_l)$. 

C. Results

In this section we demonstrate the effectiveness of the proposed encoder–decoder VarLSTM in learning trajectory predictions with quantified uncertainty for the maritime domain using labeled and unlabeled data. From the complete dataset in Fig. 3, we isolated a set of trajectories to be used as test set. The selected test trajectories are illustrated in Fig. 4, with a color that changes with the distance traveled from a fixed origin point, located in the upper left corner of the figure. Fig. 5(a) shows the average prediction error (APE) computed as the average Euclidean distance between the predicted position and the ground truth at the $i$th prediction sample (i.e., prediction horizon $3$ h) as a function of the vessel’s traveled distance from the fixed origin point. This representation allows mapping all prediction errors on a common axis and identifying regions where the prediction error is high.

Fig. 5(a) shows that the two models achieve comparable prediction errors, apart from a specific waypoint area $WP$ (distance between 80 and 100 nmi), highlighted in Fig. 4, where the APE obtained with the unlabeled model is much higher than that of the labeled model. This is a crossroad area where vessels can take three different paths toward the same destination, which makes it challenging for the prediction system to anticipate which direction the vessel will follow after the crossroad. The plot proves how high-level information is key to improve prediction performance corresponding to deviation areas (e.g., area $WP$ in Fig. 4). The major difference between unlabeled and labeled predictions is that unlabeled use past observed positions to generate future states, while labeled use additional high-level pattern information, i.e., the ship’s intended destination. In the area $WP$ (Fig. 4), the unavailability of any prior information puts the unlabeled model at a disadvantage for deciding which future path to be followed by the vessel after a crossroad. Instead, the labeled model has more information available, and can generate more realistic future trajectories by exploiting the information related to the pattern descriptor.

In Fig. 5(b) we show the performance in terms of predictive uncertainty estimation by using the notion of generalized variance, defined as the determinant of the covariance matrix $\Sigma_k$ in (9) produced by the unlabeled and labeled models; more precisely, in Fig. 5(b), the square root of the determinant of the prediction covariance matrix (averaged over all the trajectories) is plotted, which is proportional to the area of the uncertainty ellipse. Fig. 6 shows the results obtained by the proposed model on a specific trajectory, which intersects the area $WP$ (Fig. 4) at three time steps $k_i$, $i = 1, 2, 3$. We perform the predictions considering the sliding window approach, so the input sequence is fed in each neural model to predict the next output sequence. We can see that the predictions using the unlabeled model shown in Fig. 6(a)–(c) detect the correct direction to be followed only after the critical point of a crossroad at time step $k_1$, while, conversely, the ones using the labeled model shown in Fig. 6(d)–(f) are able to anticipate the direction to be followed by the vessel after the crossroad many steps ahead. Another important point to be made regarding the predictive uncertainty modeling is how the level of uncertainty quantified by the unlabeled model is larger and includes many possible directions when the path...
Fig. 5. (a) APE and (b) predicted total variance. Both are computed at the $i$th prediction sample (i.e., prediction horizon 3 h) and are expressed as a function of the vessel distance from a fixed point, located in the upper left corner of Fig. 4. Panel (b) shows the square root of the determinant of the covariance matrix produced by the network, which is also known as generalized variance and is proportional to the area of the prediction uncertainty ellipse.

Fig. 6. Predictions computed with (a)–(c) unlabeled models compared to (d)–(f) labeled models. Ground truth data is provided by the Danish Maritime Authority and is used pursuant to conditions for use of Danish public data [1].

to be taken is still undetermined [e.g., in correspondence to a crossroad, such as in Fig. 6(b)]. The dataset used to train the network (represented in Fig. 3) contains multiple paths that can be taken by a vessel after the waypoint; for this reason, the unlabeled model computes a predicted trajectory that is the average mode among multiple possible paths, even if the unimodal prediction is not necessarily a valid future behavior, and this is especially apparent in Fig. 6(b). In contrast, the predictive uncertainty diminishes when the decision of the predictor based on the training data is less complicated due to the direct path of the vessel, as shown in Fig. 6(c).
In the labeled case, due to the additional high-level information available, the predictions are shown to be more accurate, especially when the vessel is before the crossroad, as shown in Fig. 6(d) and (e). The labeled model can compute more accurate predictions, as some of the possible future paths (specifically, after the waypoint) can be excluded, precisely on the basis of the additional high-level information. This can be easily noted with a comparative inspection of Fig. 6(b) and (e), where the uncertainty of the unlabeled model, contrary to the labeled one, has to account also for the additional South–East path. It should be noted that the intention information is most useful when it can reduce the multimodal nature of the task to a unimodal prediction task. For instance, in Fig. 6(e), the high-level information allows excluding the South–East path, but cannot help further in deciding exactly which one of the two South–West paths the vessel is going to follow. Still, the labeled model achieves better prediction performance, as the additional information allows going from a higher modality prediction task to a lower modality one.

V. CONCLUSION

In this article, we proposed an attention-based recurrent encoder–decoder architecture to address the problem of trajectory prediction with uncertainty quantification applied to a maritime domain case study. The predictive uncertainty is estimated through Bayesian learning by combining both aleatoric and epistemic uncertainty, with the latter modeled via MC dropout. Experimental results show that the proposed architecture is able to learn maritime sequential patterns from historical AIS data, and successfully predict future vessel trajectories with a reliable quantification of the predictive uncertainty. Two models are compared and show how prediction performance can be improved by exploiting high-level intention behavior of vessels (e.g., their intended destination) when available. Possible future lines of research on this topic include: the investigation of multimodal prediction techniques in combination with high-level intention modeling to further improve the prediction performance when the intention information alone is not sufficient to fully account for the multimodality of the prediction task; the effects of seasonality patterns and weather conditions in the available data on the performance of the trajectory predictor; the conditioning of trajectory predictions based on the type of vessel; and the integration of uncertainty-aware prediction modules into real-world intelligent situational awareness systems [46].

APPENDIX

INTENTION REGULARIZATION PERFORMANCE

Herein, we investigate how the prediction model can benefit from the intention regularization technique proposed in Section III-D for the novel version of the labeled architecture (Lv2), by comparing the results with those obtained with the previous version of the labeled architecture (Lv1) presented in [34], in which the intention information is injected for each time step during the decoding phase.

For this comparison, we used the complete AIS dataset from [25] shown in Fig. 3. In our experiment, we split the original dataset composed of 394 full trajectories into 284 trajectories for training, 32 for validation, and 78 for testing. The windowing procedure proposed in [25] produces 8574 input/output sequences of length $\ell = h = 12$ for training, 1054 sequences for validation, and 2379 sequences for the testing phase. In the evaluation of our experiment, we use the APE for different horizons (i.e., 1, 2, and 3 h), and the average displacement error (ADE) as the average Euclidean distance between the predicted trajectory and the ground truth (i.e., over all the predicted positions of a trajectory) [44], [45]. The performance evaluation of the proposed intention regularization method is shown in Table I, where only the most informative results are shown for a varying probability of intention dropout mask. As shown in Table I, the proposed architecture Lv2 performs better than Lv1, with the best performance achieved by setting $\gamma = 0.3$.

| Model          | Mask (γ) | 1h  | 2h  | 3h  | ADE   |
|----------------|----------|-----|-----|-----|-------|
| Lv1 [34]       | -        | 0.57| 1.14| 1.80| 0.97  |
| Lv2 (ours)     | 0.3      | 0.51| 1.03| 1.78| 0.88  |
| Lv2 (ours)     | 0.5      | 0.54| 1.06| 1.81| 0.91  |

REFERENCES

[1] “Data from the Danish AIS system,” Accessed: Nov. 02, 2022. [Online]. Available: https://dma.dk/safety-at-sea/navigational-information/ais-data
[2] Z. Xiao, X. Fu, L. Zhang, and R. S. M. Goh, “Traffic pattern mining and forecasting technologies in maritime traffic service networks: A comprehensive survey,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 5, pp. 1796–1825, May 2020.
[3] L. M. Millefiori, P. Braca, K. Bryan, and P. Willett, “Modeling vessel kinematics using a stochastic mean-reverting process for long-term prediction,” IEEE Trans. Aerosp. Electron. Syst., vol. 52, no. 5, pp. 2313–2330, Oct. 2016.
[4] N. Forti, L. M. Millefiori, P. Braca, and P. Willett, “Bayesian filtering for dynamic anomaly detection and tracking,” IEEE Trans. Aerosp. Electron. Syst., vol. 58, no. 3, pp. 1528–1544, Jun. 2022.
[5] N. Forti, E. d’Afflitos, P. Braca, L. M. Millefiori, P. Willett, and S. Carniel, “Maritime anomaly detection in a real-world scenario: Ever Given grounding in the Suez Canal,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 8, pp. 13904–13910, Aug. 2022.
[6] N. Forti, L. M. Millefiori, and P. Braca, “Unsupervised extraction of maritime patterns of life from Automatic Identification System data,” in Proc. MTS/IEEE OCEANS, 2019.
[7] B. Ristic, B. La Scala, M. Morelande, and N. Gordon, “Statistical analysis of motion patterns in AIS data: Anomaly detection and motion prediction,” in Proc. 11th Int. Conf. Inf. Fusion, 2008.
[8] L. P. Perera, P. Oliveira, and C. Guedes Soares, “Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction,” IEEE Trans. Intell. Transp. Syst., vol. 13, no. 3, pp. 1188–1200, Sep. 2012.
[9] F. Mazzarella, V. F. Arguedas, and M. Vespe, “Knowledge-based vessel position prediction using historical AIS data,” in Proc. Sensor Data Fusion: Trends, Solutions, Appl., 2015.
[10] S. Hexenberg, A. L. Flätén, B. H. Eriksen, and E. F. Brekke, “AIS-based vessel trajectory prediction,” in Proc. 20th Int. Conf. Inf. Fusion, 2017.

[11] B. R. Dalsnes, S. Hexenberg, A. L. Flätén, B.-O. H. Eriksen, and E. F. Brekke, “The neighbor course distribution method with Gaussian mixture models for AIS-based vessel trajectory prediction,” in Proc. 21st Int. Conf. Inf. Fusion, 2018, pp. 580–587.

[12] A. Valsamis, K. Tserpes, D. Zissis, D. Anagnostopoulos, and T. Varvarigou, “Employing traditional machine learning algorithms for Big Data streams analysis: The case of object trajectory prediction,” J. Syst. Softw., vol. 127, pp. 249–257, 2017.

[13] D. Nguyen, V. Radaide, G. Hajduck, R. Garello, and R. Fablet, “A multi-task deep learning architecture for maritime surveillance using AIS data streams,” in Proc. IEEE Int. Conf. Data Sci. Adv. Analytics, 2018, pp. 331–340.

[14] J. Y. Yu, M. O. Sghaier, and Z. Grabowiecka, “Deep learning approaches for AIS data association in the context of maritime domain awareness,” in Proc. Int. Conf. Inf. Fusion, 2020, pp. 1–8.

[15] H. Tang, Y. Yin, and H. Shen, “A model for vessel trajectory prediction based on long short-term memory neural network,” J. Mar. Eng. Technol., vol. 21, no. 3, pp. 136–145, 2022.

[16] M. Gao, G. Shi, and S. Li, “Online prediction of ship behavior with automatic identification system sensor data using bidirectional long short-term memory recurrent neural network,” Sensors, vol. 18, no. 12, 2018, Art. no. 4211.

[17] C.-H. Yang, C.-H. Wu, J.-C. Shao, Y.-C. Wang, and C.-M. Hsieh, “AIS-based intelligent vessel trajectory prediction using Bi-LSTM,” IEEE Access, vol. 10, pp. 24302–24315, 2022.

[18] C. Wang, H. Ren, and H. Li, “Vessel trajectory prediction based on AIS data and bidirectional GRU,” in Proc. Int. Conf. Comput. Vis., AIS, 2020, pp. 260–264.

[19] B. Murray and L. P. Perera, “A dual linear autoencoder approach for vessel trajectory prediction using historical AIS data,” Ocean Eng., vol. 209, 2020, Art. no. 107478.

[20] S. Wang and Z. He, “A prediction model of vessel trajectory based on generative adversarial network,” J. Navigation, vol. 74, no. 5, pp. 1161–1171, 2021.

[21] R. W. Liu et al., “STMGCN: Mobile edge computing-empowered vessel trajectory prediction using spatio-temporal multi-graph convolutional network,” IEEE Trans. Ind. Inform., vol. 18, no. 11, pp. 7977–7987, Nov. 2022.

[22] X. Zhou, Z. Liu, F. Wang, Y. Xie, and X. Zhang, “Using deep learning to forecast maritime vessel flows,” Sensors, vol. 20, no. 6, 2020, Art. no. 1761.

[23] D. D. Nguyen, C. L. Van, and M. I. Ali, “Vessel trajectory prediction using sequence-to-sequence models over spatial grid,” in Proc. ACM Int. Conf. Distrib. Event-Based Syst., 2018, pp. 258–261.

[24] N. Forti, L. M. Millefori, P. Braca, and P. Willett, “Prediction of vessel trajectories from AIS data via sequence-to-sequence recurrent neural networks,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2020, pp. 8936–8940.

[25] S. Capobianco, L. M. Millefori, N. Forti, P. Braca, and P. Willett, “Deep learning methods for vessel trajectory prediction based on recurrent neural networks,” IEEE Trans. Aerosp. Electron. Syst., vol. 57, no. 6, pp. 4329–4346, Dec. 2021.

[26] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 6405–6416.

[27] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in Proc. Int. Conf. Int. Conf. Mach. Learn., 2016, pp. 1050–1059.

[28] A. Kendall and Y. Gal, “What uncertainties do we need in Bayesian deep learning for computer vision?,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5574–5584.

[29] A. Bhattacharyya, M. Fritz, and B. Schiele, “Long-term onboard prediction of people in traffic scenes under uncertainty,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4194–4202.

[30] Y. Xiao and W. Y. Wang, “Quantifying uncertainties in natural language processing tasks,” in Proc. Conf. Artif. Intell., 2019, pp. 7322–7329.

[31] V. Kuleshov, N. Fenner, and S. Ermon, “Accurate uncertainties for deep learning using calibrated regression,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 2801–2809.

[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.

[33] S. Jung, I. Schlangen, and A. Charlish, “A memonic Kalman filter for non-linear systems with extensive temporal dependencies,” IEEE Signal Process. Lett., vol. 27, pp. 1005–1009, 2020.

[34] S. Capobianco, N. Forti, L. M. Millefori, P. Braca, and P. Willett, “Uncertainty-aware recurrent encoder-decoder networks for vessel trajectory prediction,” in Proc. IEEE 24th Int. Conf. Inf. Fusion, 2021.

[35] R. M. Neal, Bayesian Learning for Neural Networks. Berlin, Germany: Springer, 1996.

[36] Y. Gal and Z. Ghahramani, “A theoretically grounded application of dropout in recurrent neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 1027–1035.

[37] C. M. Bishop, Neural Networks for Pattern Recognition. London, U.K.: Oxford Univ. Press, 1996.

[38] A. Graves, A. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2013, pp. 6645–6649.

[39] M. Schuster and K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Trans. Signal Process., vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[40] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in Proc. Int. Conf. Learn. Representations, 2015, arXiv:1409.0473.

[41] S. Wager, S. Wang, and P. S. Liang, “Dropout training as adaptive regularization,” in Proc. Adv. Neural Inf. Process. Syst., 2013, pp. 351–359.

[42] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, “ModDrop: Adaptive multi-modal gesture recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 8, pp. 1692–1706, Aug. 2016.

[43] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in Proc. Int. Conf. Learn. Representations, 2019, arXiv:1711.05101.

[44] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, “You’ll never walk alone: Modeling social behavior for multi-target tracking,” in Proc. IEEE 12th Int. Conf. Comput. Vis., 2009, pp. 261–268.

[45] A. Alahi, K. Goel, V. Ramamoorthy, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social LSTM: Human trajectory prediction in crowded spaces,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 961–971.

[46] N. Forti, E. d’Afflizzo, P. Braca, L. M. Millefori, S. Carniel, and P. Willett, “Next-gen intelligent situational awareness systems for maritime surveillance and autonomous navigation,” Proc. IEEE, vol. 110, no. 10, pp. 1532–1537, Oct. 2022.

Samuele Capobianco received the M.S. degree in computer engineering and the Ph.D. degree in smart computing from the University of Florence, Florence, Italy, in 2016 and 2020, respectively.

In 2018, he was a Visiting Researcher with the Department of Human Centred Computing, Monash University, Melbourne, VIC, Australia. In 2020, he was a Visiting Scientist with the Centre for Maritime Research and Experimentation (CMRE), La Spezia, Italy. His research interests include machine learning and pattern recognition, with emphasis on applications to signal processing, computer vision, and data fusion.
Leonardo Maria Millefiori (Member, IEEE) received the B.Sc. degree in aerospace information engineering and the M.S. degree (summa cum laude) in telecommunication engineering with a focus on radar systems and remote sensing from the Sapienza University of Rome, Italy, in 2010 and 2013, respectively, and the Ph.D. degree (honors) in information engineering from the University of Pisa, Italy, in 2022.

He was a Visiting Researcher with the NATO Science and Technology Organization Center for Maritime Research and Experimentation (CMRE), La Spezia, Italy, where he joined the Research Department as a Research Scientist, in 2014. His research interests include target motion modeling, statistical learning and signal processing, target tracking, data fusion, and machine learning.

Dr. Millefiori was the recipient of the NATO STO Scientific Achievement Award in 2020 as member of the Data Knowledge and Effectiveness (DKOE) research group at CMRE for “Advances in Artificial Intelligence and Information Fusion for Maritime Situational Awareness.”

Nicola Forti received the M.S. degree in automation engineering, and the Ph.D. degree in information engineering from the University of Florence, Florence, Italy, in 2013 and 2016, respectively.

From 2015 to 2016, he was a Visiting Researcher with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA. In 2017, he was a Postdoctoral Researcher with the Systems and Control group, University of Florence. In 2018, he joined the NATO Science & Technology Organization (STO) Centre for Maritime Research and Experimentation (CMRE), La Spezia, Italy, where he is currently a Scientist with the Research Department. His research interests include statistical signal processing, probabilistic modeling, Bayesian filtering, target tracking, data fusion, and machine learning.

Dr. Forti was the recipient of the NATO STO Scientific Achievement Award in 2020 as member of the Data Knowledge and Effectiveness (DKOE) research group at CMRE for “Advances in Artificial Intelligence and Information Fusion for Maritime Situational Awareness.”

Paolo Braca (Senior Member, IEEE) received the laurea degree (summa cum laude) in electronic engineering and the Ph.D. degree (honors) in information engineering from the University of Salerno, Salerno, Italy, in 2006 and 2010, respectively.

In 2009, he was a Visiting Scholar with the ECE Department, University of Connecticut, Stamford, CT, USA. From 2010 to 2011, he was a Postdoctoral Associate with the University of Salerno. In 2011, he joined the NATO Science and Technology Organization (STO) Centre for Maritime Research and Experimentation (CMRE), La Spezia, Italy, where he is currently a Senior Scientist and a Project Manager. Furthermore, he led a number of research projects funded by the European Commission, by the U.S. Office of Naval Research (ONR), and other national and international institutions. He has coauthored more than 200 publications in international scientific journals, conference proceedings, and NATO technical reports. He was awarded with the National Scientific Qualification to function as Associate and Full Professor in Italian Universities, in 2017 and 2018, respectively. In 2019, he was appointed as Adjunct Professor with the University of Cassino, Italy. His research interests include the general area of machine learning and statistical signal processing with emphasis on detection and estimation theory, wireless sensor network, multi-agent algorithms, target tracking and data fusion, adaptation and learning over graphs, radar (sonar) signal processing, and machine learning.

Dr. Braca was the recipient of the Best Student Paper Award (first runner-up) at the FUSION conference in 2009, the NATO Science and Technology Organization (STO) Scientific Achievement Award (SAA) 2017 for his contribution to the “Development and Demonstration of Networked Autonomous ASW,” the Best Paper Award (first runner-up) at the SSPD conference in 2019 for the paper he coauthored, and the IET 2019 Premium Award for Best Paper published on the IET Radar Sonar & Navigation. He was also a recipient of the NATO STO SAA, in 2020, as a Team Leader for the “Advances in Artificial Intelligence and Information Fusion for Maritime Situational Awareness.” He was the recipient of the Young Scientist Contest Award at the Signal Processing Symposium 2021 for the paper he coauthored. He serves as Associate Editor for the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, ISIF Journal of Advances in Information Fusion, and IET Radar, Sonar & Navigation. In 2017, he was a Lead Guest Editor of the Special Issue “Sonar Multi-Sensor Applications and Techniques” in IET Radar, Sonar & Navigation. He served as Associate Editor for the IEEE SIGNAL PROCESSING MAGAZINE, IEEE TRANSACTIONS ON SIGNAL PROCESSING, and EURASIP Journal Advances on Signal Processing. He is in the Technical Committee of the major international conferences in the field of signal processing and data fusion.

Peter Willett (Fellow, IEEE) received the B.Sc. degree in engineering science from the University of Toronto, Toronto, ON, Canada, in 1982 and the Ph.D. degree from Princeton University, Princeton, NJ, USA, in 1986.

He has been a Faculty Member with the Electrical and Computer Engineering Department, University of Connecticut, Stamford, CT, USA, since 1986. Since 1998, he has been a Professor. His research interests include statistical signal processing, detection, machine learning, communications, data fusion, and tracking.