A Machine Learning Approach to Water Velocity Estimation for Better Navigation of Marine Gliders

Meghan Leone Stuart   Juan Jose Garcia   Sahin Olut   Taksir Hasan
Ahsan Mahmood   Kipp Williams   Parasara Sridhar Duggirala

Department of Computer Science
University of North Carolina at Chapel Hill
Corresponding author: mstuart@cs.unc.edu

***Abstract*—Marine gliders are autonomous underwater vehicles used for data collection by ocean and climate scientists. To conserve energy on months-long trips these gliders use a propulsion system driven mostly by gravity. This variable buoyancy propulsion system, however, is particularly susceptible to error introduced by strong or chaotic ocean currents. For this reason, most navigation systems rely on prediction of local water velocity to plan their paths. Current physics-based simulation models are limited by on-board computation resources. Consequently, we introduce several variations of recurrent neural network based prediction algorithms and compare their performance to a commonly used statistical forecasting method.**

I. Introduction

Marine gliders are low-energy underwater vehicles propelled by a variable buoyancy system, through which the vehicle inflates a bladder to float, and then deflates the bladder to sink while using steerable wings to direct the glider's motion. However, the low-energy propulsion mechanism leaves gliders vulnerable to significant drift due to ambient water velocity, which makes data acquisition less precise, and has led to the loss of at least one glider, a loss of nearly half a million dollars.

Navigation systems for these gliders currently use predictions of the average ambient water velocity, computed en route, along a glider's next path segment to correctly steer the glider. By default, the gliders we encountered used a constant prediction, simply that the water velocity along the next trip interval is predicted to equal the water velocity measured during the previous trip interval. However, where ocean currents are chaotic, like at the edge of the Gulf Stream off the Southeastern US coast, this prediction is insufficient. Chang et al. [2] address this by forecasting with a physics-based fluid model in a series of glider missions off the Southeastern US coastline in 2011 and 2012. At each timestep, they fit the model to water velocity measurements made previously along the glider path, however, this process is computationally intense. The amount of energy consumed by fitting a physics-based fluid model makes it impossible to do on-board the glider, thus prediction can only occur when the glider surfaces to communicate with a separate server via satellite.

To improve upon the constant prediction while using less energy than a physics based simulation, we test a classic statistical regression timeseries prediction model as well as several recurrent network based models, and compare them across several metrics. The selection of models is informed by related work presenting solutions to prediction tasks.

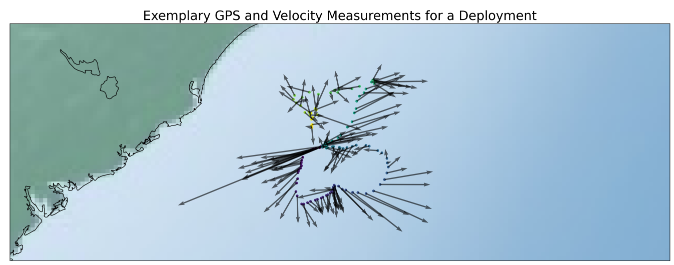


Fig. 1: The path taken by a glider during one deployment. Vectors drawn at each point represent the water velocity reported at each location. Vectors' lengths give relative scale. The series is subsampled every three measurements for ease of visualization.

II. Data

The data used to train and test our models comes from two sources. We drew from an open database hosted by the Southeast Coastal Ocean Observing Regional Association [7] (SECOORA), as well as from data collected in [2] for a total of twelve missions made by seven different gliders off of the Carolina and Georgia coastline. Each of the missions lasted several weeks, during which each glider recorded a series of measurements including depth, water velocity, temperature, CO_2 saturation, and others. The location and water velocity measurements of an example deployment are shown in Figure 1.

The data was cleaned to remove duplicate data, invalid numbers, and glider component malfunctions. After cleaning, the missions contained between 44-450 measurements. Time between measurements varied, both between gliders and within individual missions, but was typically on the order of one hour.

Of these twelve deployments, four were used for preliminary model selection, and the other eight were used for training, validation, and testing. Details of model cross-validation and training are described in the Results section.

III. Model selection

To compare a variety of predictive models for this task, we began with a broad literature review of time series prediction models and dynamical systems. Our survey found that Recurrent Neural Networks (RNNs) often outperform statistical methods like Vector Autoregression (VAR), Autoregressive Integrated Moving Average, (ARIMA) and Error Trend Seasonality (ETS) analysis. However, empirical evidence suggests the best architecture is determined by the dataset [3]. Hence, we present results from a classical linear VAR model for time series prediction and compare this with several variations of deep network based models.

A. Classic vector autoregression

Autoregressive models are widely used for time series in many different contexts. In a nutshell, autoregressive models predict by conditioning on previous values. To do that, the signal needs to be stationary, meaning that the mean, variance and autocorrelation are uniform across the signal [1]. A linear autoregressive model of order p, is shown below:

$$V_t = \beta_0 + \beta_1 V_{t-1} + \ldots + \beta_p V_{t-p} + \epsilon_t$$

where $\beta = (\beta_0, \ldots \beta_p)$ are parameters and ϵ_t is residual error. To estimate the parameters, β, we use LASSO regression, equivalent to least squares regression with an regularization term penalizing the ℓ_1 norm of β.

Additionally, we apply several network based models.

B. Recurrent neural network based autoregression

Like DeepAR [5] we approximate the distribution of the next velocity measurement using an RNN framework. Our general framework can be formalized as follows:

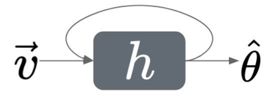
Let $V = \{v_{i,:}\}_{i=1}^{N}$ be a set of i.i.d. sequences. Let $v_{i,t}$ represent the t^{th} velocity measurement of the i^{th} sequence.

$$\begin{aligned} p(V) &= \prod_{i=1}^{N}\prod_{t=1}^{T} p(v_{i,t}|v_{i,<t}) \\ &= \prod_{i=1}^{N}\prod_{t=1}^{T} f(v_{i,t}; h(v_{i,<t})) \end{aligned}$$

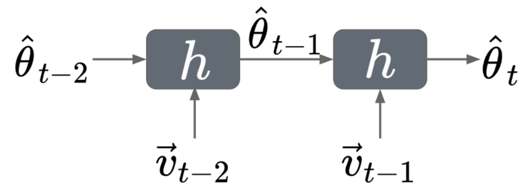
where f is the probability density of the predicted velocity conditioned on h, the recurrent model trained to predict parameters of distribution f. T is the total length of each sequence, and N is total number of sequences. We can choose distribution f as well as the architecture of model h in order to tailor the model to our data.

In this work we show results for two selections of distribution f: a multivariate Gaussian distribution, and a mixture of Gaussians with four components. These distributions were parameterized by two different selections of h conditioned on previous velocity measurements: the standard recurrent unit $h(v_t) = Mv_t + W\hat{\theta}_t + b$, where $\hat{\theta}_t = h(v_{<t})$ is the parameter estimate at the previous time-step, and a long short term memory (LSTM) gate as proposed by Hochreiter et al [4]. The distributions were then sampled to produce velocity predictions v_t.

We additionally report results using the same RNN and LSTM architectures to directly predict velocity vector v_t rather than parameters for its probability distribution.



(a) Standard RNN framework.



(b) Unfolded framework showing temporal relationships.

Fig. 2: The standard RNN framework. $\vec{v}$ is the measured ground truth velocity, and $\hat{\theta}$ is the predicted parameters of the velocity density. We use two different recurrent unit functions, h, trained to fit the data.

IV. Model Evaluation

In the context of predicting water velocity for navigation purposes, the ideal error metric is not clear, as ambient water velocity orthogonal to the intended direction of motion of the glider is more likely to push the glider off path than current roughly parallel to the direction of motion [6]. However, water velocity direction estimation is not enough to perform path correction. With this in mind, we report three metrics. Mean cosine similarity (MCS) is used to measure error in the direction of the predicted velocity.

$$MCS(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N} 1 - \frac{y_i . \hat{y}_i}{||y_i||.||\hat{y}_i||},$$

where y_i is the true measured velocity and $\hat{y}_i$ is our predicted velocity at time-step i. N is the number of forecasts. We also report a relative error metric, mean absolute percentage error (MAPE) to

$$MAPE(y, \hat{y}) = \frac{100}{N}\sum_{i=1}^{N} \frac{||y_i - \hat{y}_i||}{||y_i||},$$

where N is the number of forecasts. MAPE gives the length of the vector difference between the forcasted and true velocities as a percernt of the length of the true velocity vector. Finally, we report the well-known mean squared error (MSE) in $(m/s)^2$.

$$MSE(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N} ||y_i - \hat{y}_i||$$

V. Results

The models were trained and tested using four-fold cross-validation. For each phase of the cross validation, we withheld one entire mission, split the other missions into windows of 20 timesteps each, shuffled them, then split them 90% for training and 10% for validation. Selection of model hyper-parameters, like number of hidden units, window size, optimizer, and loss function, was done by brute force search.

Each model was trained on this data, then evaluated on the withheld mission. This process was repeated four times with four different missions and the metrics reported below are averaged over the four test missions. In this paper, all results reported are from models with predictions conditioned only on previous water velocity measurements. Additional covariates measured by the gliders are omitted during model development for simplicity.

A. *Vector Autoregression using LASSO*

Vector autoregression achieved the lowest MSE across the test suite, however qualitative analysis suggests that it is doing little more than predicting constant current (i.e. predicting $v_{t+1} = v_t$). This can be seen in figure 3. The velocity predictions in orange appear to approximately lag behind the true values in blue by one timestep.

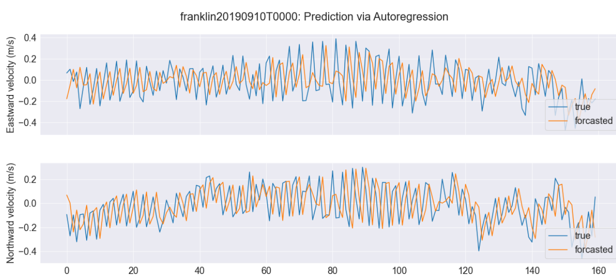


Fig. 3: Vector Auto-regression prediction performed on data collected by Franklin glider in September 2019. This figure shows traces of the eastward and northward velocity components separately. The upper plot is eastward velocity and the lower is northward. The blue trace represents the ground truth measure velocity values and the orange are model forecasts.

B. *RNN Models*

To maintain comparability, we keep the network based model architectures as similar as possible. All RNN/LSTM models use a hidden unit size of 8. The ADAM optimizer was used to minimize the Mean Absolute Error (MAE). For models involving distributions, the loss was derived by sampling from the output distribution.

From table I, we can see that a Gaussian Mixture Model with parameters predicted by an RNN outperforms other models in MAPE and MCS, but is outperformed by VAR in MSE. However, all of the recurrent network based models are very sensitive to hyper parameter selection as well as weight initialization. It is also important to note here, that none of the models perform particularly well in general. It seems that the RNN + GMM architecture outperforms the others overall by conservatively predicting velocities near 0 in series exhibiting

Model	MAPE	MCS	MSE
VAR	627.8945	-0.3465	**0.04525**
RNN + Gaussian	658.78125	-0.05325	0.16275
LSTM + Gaussian	803.52225	-0.11375	0.1265
RNN + GMM	**250.52325**	**-0.00325**	0.11625
LSTM + GMM	1514.53675	-0.17175	2.36725
RNN	623.3465	0.0615	0.776
LSTM	368.7065	-0.19125	0.09225

TABLE I: Metrics for all models averaged over all cross-validation phases

calm water, but is unable to predict unusual events as shown in figure 4. Interestingly, some architectures were able to predict the onset of out-of-distribution events, but were very bad at predicting velocity behavior during said event. This behavior can be seen in figure 5.

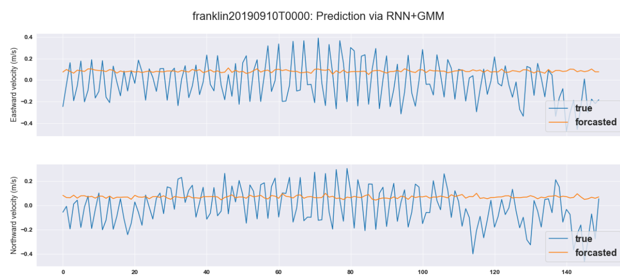


Fig. 4: RNN + GMM prediction performed on the same data collected by Franklin glider in September 2019

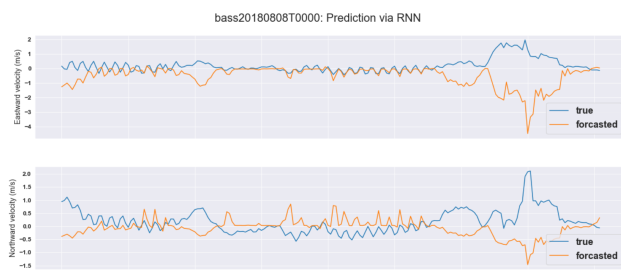


Fig. 5: RNN direct velocity prediction performed on data collected by Bass glider in August 2018.

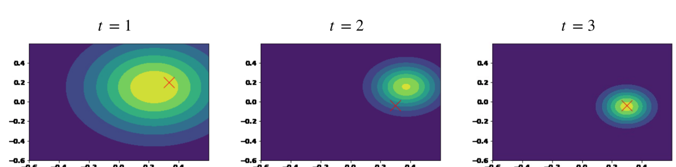


Fig. 6: RNN parameterized predictions of multivariate Gaussian distribution of $v_{t+1}|v_t$ for three subsequent time steps. Red "X" marks the true value.

VI. Conclusion and Future Plans

The exploration done in this paper has led to a few important conclusions for future work on this project. First, it's clear that velocity data alone is not sufficient to make the desired predictions, at least with the methods tested thus far. Second, because of the sensitivity of the recurrent network based algorithms to initialization and parameter tuning, it's not possible to declare that any one model outperformed the others to a significant degree, however all of the recurrent network based architectures showed non-linear behavior and had some regions of good fit. For this reason, we believe it is likely that these models can be improved with more training data and by including as additional covariates glider measurements including GPS location at last surfacing or dead-reckoning location, water temperature, and pressure. These improvements are currently under investigation.

Additionally, the ideal loss function to both train and evaluate these models is not clear. Additional testing in glider simulations and with gliders in controlled environments is necessary to validate any models before deployment.

VII. Acknowledgements

Authors Parasara Sridhar Duggirala, Meghan Leone Stuart, and Kipp Williams acknowledge the support of the Air Force Office of Scientific Research under award number FA9550-19-1-0288 and National Science Foundation (NSF) under grant numbers CNS 1935724 and CNS 2038960.

References

[1] R. Adhikari and R. K. Agrawal. An introductory study on time series modeling and forecasting. arXiv preprint arXiv:1302.6613, 2013.

[2] D. Chang, F. Zhang, and C. R. Edwards. Real-time guidance of underwater gliders assisted by predictive ocean models. Journal of Atmospheric and Oceanic Technology, 32(3):562–578, 2015.

[3] H. Hewamalage, C. Bergmeir, and K. Bandara. Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions. arXiv:1909.00590 [cs, stat], Sept. 2019.

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

[5] D. Salinas, V. Flunkert, and J. Gasthaus. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks.arXiv:1704.04110 [cs, stat], Feb. 2019.URL http://arxiv.org/abs/1704.04110.

[6] R. N. Smith, J. Kelly, Y. Chao, B. H. Jones, and G. S. Sukhatme. Towards the improvement of autonomous glider navigational accuracy through the use of regional ocean models. In ASME2010 29th International Conference on Ocean, Offshore and Arctic Engineering, pages 597–606. American Society of Mechanical Engineers Digital Collection, 2010.

[7] SECOORA Data Portal. URL https://portal.secoora.org/.