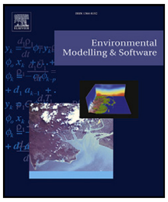
Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft



Position Paper

Cyclone trajectory and intensity prediction with uncertainty quantification using variational recurrent neural networks

Arpit Kapoor [a,b,*], Anshul Negi [c], Lucy Marshall [b,d,e], Rohitash Chandra [a,b]

[a] Transitional Artificial Intelligence Research Group, School of Mathematics and Statistics, University of New South Wales, Sydney, NSW 2052, Australia
[b] Data Analytics for Resources and Environments, Australian Research Council-Industrial Transformation Training Centre (ARC-ITTC), NSW 2052, Australia
[c] Department of Electrical and Electronics Communications, Indian Institute of Technology, Ropar, India
[d] Faculty of Science and Engineering, Macquarie University, Sydney, NSW 2109, Australia
[e] School of Civil and Environmental Engineering, University of New South Wales, Sydney, NSW 2052, Australia

ARTICLE INFO

Keywords:
Cyclone track prediction
Recurrent neural networks
Bayesian neural networks
Long short-term memory
Variational inference

ABSTRACT

Cyclone track forecasting is a critical climate science problem involving time-series prediction of cyclone location and intensity. Machine learning methods have shown much promise in this domain, especially deep learning methods such as recurrent neural networks (RNNs) However, these methods generally make single-point predictions with little focus on uncertainty quantification. Although Markov Chain Monte Carlo (MCMC) methods have often been used for quantifying uncertainty in neural network predictions, these methods are computationally expensive. Variational Inference (VI) is an alternative to MCMC sampling that approximates the posterior distribution of parameters by minimizing a KL-divergence loss between the estimate and the true posterior. In this paper, we present variational RNNs for cyclone track and intensity prediction in four different regions across the globe. We utilise simple RNNs and long short-term memory (LSTM) RNNs and use the energy score (ES) to evaluate multivariate probabilistic predictions. The results show that variational RNNs provide a good approximation with uncertainty quantification when compared to conventional RNNs while maintaining prediction accuracy.

1. Introduction

In the last few decades, the devastating impact of climate change became more obvious with rise in adverse meteorological events such as droughts, heat waves, hurricanes and tropical cyclones (Mendelsohn et al., 2012), and bushfires. The devastating effects of tropical cyclones and hurricanes include flooding, landslides, detrimental winds, and storm surges which makes them a dominant meteorological hazard (Needham et al., 2015; Schrum et al., 2020). Apart from these, tropical cyclones cause a significant amount of economic and environmental damage (Pielke, 2007; Fengjin and Ziniu, 2010; Lionello et al., 2006; Harmelin-Vivien, 1994). Forecasting cyclone tracks (trajectory) and intensities are critical for lowering injuries and saving lives apart from prevention of damage to infrastructure and mitigation of economic losses (McBride and Holland, 1987). Cyclones, typhoons and hurricanes are all tropical storms, with the distinguishing characteristic being the geographical location where they form. Tropical storms that form over North Atlantic Ocean and North-east Pacific are known as *hurricanes*, and those developing in the North-west Pacific are known as *typhoons*, while *cyclones* are the storms that form over the South Pacific and the Indian Ocean (Emanuel, 2003).

Cyclone track forecasting is a multi-dimensional time-series prediction problem with cyclone location (represented by latitude and longitude) and its wind intensity recorded at each time step. Roy et al. presented a comprehensive review of tropical cyclone track forecasting methods (Roy and Kovordányi, 2012a). More recently, Chen et al. presented a machine learning focused review of tropical cyclone forecast modelling (Chen et al., 2020). Traditionally, numerical (Yablonsky et al., 2015) and statistical methods/techniques (McAdie and Lawrence, 2000; Roy and Kovordányi, 2012b) have been utilized for forecasting cyclone trajectories. Regression models have commonly been used for cyclone trajectory (Neumann, 1972) as well as intensity prediction (Atkinson and Holliday, 1977). Atkinson and Holliday (1977) derive a high degree polynomial equation to model the wind speed/pressure relationship for tropical cyclones in the western-North

[*] Corresponding author at: Transitional Artificial Intelligence Research Group, School of Mathematics and Statistics, University of New South Wales, Sydney, NSW 2052, Australia.

E-mail addresses: arpit.kapoor@unsw.edu.au (A. Kapoor), 2019eeb1145@iitrpr.ac.in (A. Negi), lucy.marshall@unsw.edu.au (L. Marshall), rohitash.chandra@unsw.edu.au (R. Chandra).

URLs: http://arpit-kapoor.com (A. Kapoor), http://rohitash-chandra.github.io (R. Chandra).

https://doi.org/10.1016/j.envsoft.2023.105654
Received 13 November 2022; Received in revised form 4 February 2023; Accepted 7 February 2023
Available online 9 February 2023
1364-8152/

pacific ocean. Due to the chaotic nature of cyclones, estimating the risk of cyclones reaching landfall along with their wind intensities has been challenging in previous studies. Hall and Jewson (2007) used a non-parametric model and utilized the complete set of cyclones instead of using only those that reached the landfall for training and outperformed previous approaches that used multi-regression models (Vickery et al., 2000). As an alternative to stochastic methods, deterministic models have also been used for trajectory forecasting of tropical cyclones (Fiorino and Elsberry, 1989; DeMaria, 1987; Mohanty, 1994). The literature also includes methods that combined forecasts from multiple models to generate a consensus prediction. Weber (2003) presented a statistical ensemble of numerical track prediction models assuming dependence on the storm structure, location, and movement. Goerss (2000) and Weber (2003) showed that predictions from an ensemble approach may be more accurate than predictions from individual models.

Machine learning models have more recently been established as a reliable alternative to other statistical techniques for cyclone trajectory and wind intensity prediction. Zhang et al. (2013) used decision trees to predict the diversion of cyclone trajectories from landing in the West Pacific ocean. Lee and Liu (2000) initially predicted cyclone tracks with radial basis function (RBF) neural networks. They also proposed the integration of dynamic link architecture (DLA) for neural dynamics for pattern classification of cyclone tracks. Ali et al. (2007) used simple neural networks (multilayer perceptron) to predict the position of cyclones in the Indian Ocean 24 h in advance, given the past 12 h of observations. The predictions were more accurate when compared to traditional models such as Climatology and Persistence (CLIPER) (Neumann, 1972), Limited Area Model (LAM) and the quasi lagrangian model (QLM) (Mathur, 1991). Ali et al. (2012) used simple neural networks to predict the tropical cyclone heat potential which is a crucial factor influencing the cyclone intensity.

Although machine learning methods show promising results for cyclone track and intensity forecasting problems, they generate single-point predictions that lack uncertainty quantification. Bayesian inference provides a mechanism for estimating unknown model parameters and quantifying uncertainty in predictions (MacKay, 1996). In comparison to single-point estimates such as gradient descent methods, Bayesian inference represents the unknown model parameters using probability (posterior) distributions and applies computational methods such as variational inference (Jordan et al., 1999; Blei et al., 2017; Rezende et al., 2014) and Markov Chain Monte-Carlo (MCMC) methods (Metropolis et al., 1953; Hastings, 1970; Andrieu et al., 2003) to sample (estimate) them. Bayesian inference methods have commonly been used for uncertainty quantification in geoscientific modelling (Pall et al., 2020; Chandra et al., 2019a). Bayesian neural networks (BNNs) (Specht, 1990; Richard and Lippmann, 1991; Wan, 1990; MacKay, 1995) refer to the use of Bayesian inference for inference (training) neural network weights and biases. Zhu et al. (2016) proposed a bagging approach with multiple two-layered Bayesian neural networks as members for forecasting cyclone tracks in the South China Sea. A scaled conjugate gradients algorithm is used to obtain the maximum a posteriori approximation (MAP) of the neural network weights while Laplace approximation is used to approximate a Gaussian posterior distribution. Deo and Chandra (2019) used BNNs via Langevin-gradient MCMC sampling for multi-step ahead prediction of cyclone intensities. However, there is no work for uncertainty quantification using BNNs for cyclone trajectory prediction.

A significant amount of work has been done in the area of Bayesian neural networks and Bayesian deep learning using variational inference (VI) techniques, as they can be more easily integrated (when compared to MCMC) with gradient-based optimization methods (backpropagation). Methods such as Bayes-by-backprop (Blundell et al., 2015) and variational autoencoders (Kingma and Welling, 2013) are prominent implementations of variational inference. However, variational inference provides an approximate inference since it treats the marginalization needed while performing Bayesian inference as an optimization problem (Jordan et al., 1999; Wainwright and Jordan, 2008; Blei et al., 2017), and does not directly sample from the posterior distribution as in the case of MCMC sampling.

Although uncertainty quantification in model predictions is vital due to the chaotic nature of cyclones and related extreme events (Fraedrich and Leslie, 1989), there is limited work using BNNs and variational inference in modelling and prediction. We note that deep learning methods such as long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) recurrent neural networks (RNNs) have been prominent in modelling temporal sequences and hence have been popular for storms and cyclones (Gao et al., 2018; Pan et al., 2019; Alemany et al., 2019; Moradi Kordmahalleh et al., 2016). The progress in the last decade with variational inference and deep learning methods motivates their application for uncertainty quantification in modelling cyclones.

In this paper, we implement variational inference via two selected RNN models to jointly predict tropical cyclone and hurricane trajectory (path) and intensity. We formulate this as a three-dimensional time-series prediction problem where the first two dimensions represent the latitude and the longitude (trajectory), and the third dimension represents the wind intensity of the cyclone. We refer to the variational inference framework as Bayes-RNN and Bayes-LSTM, which implement simple RNNs and LSTM models, respectively. We evaluate the performance of the framework on cyclones and hurricanes that appeared in the last four decades in four selected regions that cover India and the Pacific ocean. We also investigate key model parameters and report prediction accuracy, uncertainty projections, and computational efficiency.

The rest of the paper is organized as follows. Section 2 provides a background on related methods and Section 3 presents the data pre-processing and the proposed method. Section 4 presents experiments and the results, Section 5 discusses the results and Section 6 concludes with future research directions.

2. Related work

2.1. Machine learning for cyclones

Computational intelligence and machine learning methods have been very promising in predicting cyclone trajectories (Chaudhuri et al., 2015; Carr III et al., 2001; Kovordányi and Roy, 2009). Evolutionary algorithms are optimization methods (Hruschka et al., 2009; Zhou et al., 2011; Črepinšek et al., 2013) that provide gradient-free optimization which have been used for training neural network models (Floreano et al., 2008; Stanley et al., 2019; Galván and Mooney, 2021). Neuroevolution has been a suitable alternative to backpropagation; however, they face the limitation of requiring excessive computational time since they heuristically approximate the gradients using evolutionary operators such as crossover and mutation (Back and Schwefel, 1996; Munk et al., 2015). Apart from this, neuroevolution has the ability to be easily applied for training diverse neural network architectures and can be used to address problems that have changing dynamics such as reinforcement learning (Such et al., 2017; Stanley et al., 2019). Neuroevolution has shown promising results in time-series prediction problems studies (Chandra, 2015; Du et al., 2014; Chandra and Zhang, 2012), and has the ability to model dynamic time series where the model can handle dynamic length of the time series, i.e the model can be used for cyclones to make predictions with minimal data (Chandra et al., 2018b).

RNNs are well suited for modelling temporal relationships in data due to their structural properties (Elman, 1990); this property makes RNNs quite popular for cyclone track and intensity prediction problems (Alemany et al., 2019; Pan et al., 2019; Moradi Kordmahalleh et al., 2016; Igarashi and Tajima, 2021). Evolutionary algorithms have also been utilized to train RNN models to predict hurricane trajectories (Kordmahalleh et al., 2015). Chandra et al. (Chandra et al., 2015; Chandra and Dayal, 2015) employed neuro-evolution of RNNs for cyclone path prediction for the South Pacific ocean. Deo and Chandra (2016) present a study on identifying the minimal timespan required for robust predictions of cyclone wind intensity using RNNs. Later, Deo and Chandra utilized stacked ensemble learning for cyclone intensity prediction (Deo et al., 2017)

Convolutional neural networks (CNNs) have the ability to preserve spatial correlations and have been prominent in computer vision tasks (Gu et al., 2018). CNN's have also been used to incorporate the spatial dependence caused by different starting locations of various cyclones while training (LeCun et al., 1990). Hong et al. (2017) analysed the hyper-tensional satellite images via a CNN model to track cyclones in North Pacific. Zhang et al. (2018) used matrix neural networks that preserve the spatial correlation by representing inputs as matrices which yielded better accuracy when compared to RNNs such as gated recurrent units (GRUs) and LSTMs.

2.2. Bayesian inference for deep learning

In the case of Bayesian inference, MCMC methods face issues with convergence, scalability, and multi-modal posterior; making them unsuitable for big data problems with high-dimensional parameter space. Sampling from complex posterior distribution is computationally expensive where large number of samples need to be drawn to efficiently sample the posterior distribution. In order to address these issues, a number of strategies have been developed, including combining MCMC with gradient-based (Langevin and Hamiltonian MCMC) (Girolami and Calderhead, 2011; Roberts and Rosenthal, 1998; Neal et al., 2011; Welling and Teh, 2011; Chandra et al., 2018a), and meta-heuristic approaches (Drugan and Thierens, 2003; Strens, 2003; Ter Braak, 2006; Ter Braak and Vrugt, 2008; Strens, 2003) to form efficient proposal distribution in MCMC sampling. Other approaches implement structural changes to MCMC sampling that balance exploration with exploitation such as nested sampling (Skilling et al., 2006) and parallel tempering MCMC (Swendsen and Wang, 1986; Hukushima and Nemoto, 1996). Although MCMC sampling struggles computationally with a large number of parameters, recent progress with parallel tempering and Langevin-gradients has shown to address these limitations for Graph CNNs (Chandra et al. (2021a)) and deep autoencoders (Chandra et al. (2021b)). Furthermore, Chandra et al. (2019b) presented a multiprocessing framework for Langevin-gradient proposal distribution with parallel tempering MCMC for Bayesian neural networks for chaotic time-series prediction and pattern classification problems.

Apart from MCMC, we have variational inference for implementing Bayesian inference. Variational inference assumes the shape of the posterior distribution and designs an optimization problem to find the assumed variational densities that are closest to the true posterior densities by minimizing Kullback–Leibler (KL) divergence. It introduces tractability to the Bayesian inference problem and solves the problem of slow convergence for high-dimensional parameter spaces. Initial variational inference approaches for BNNs used mean field variational Bayes (MFVB) for analytical approximations for regression problems with a single hidden layer neural network (Barber and Bishop, 1998; Hinton and Van Camp, 1993). Variational inference gained popularity in the more recent deep learning revolution due to the need for robust uncertainty quantification in predictions. Graves (2011) presented the computation of derivatives of expectations in the variational objective function, also known as the *evidence lower bound* (ELBO). Bundell et al. proposed "Bayes-by-backprop" that provided a simple implementation of variational inference for neural networks (Blundell et al., 2015). As the sampling operation is not deterministic (non-differentiable), in order to perform gradient optimization over the variational loss the re-parameterization trick is employed (Kingma and Welling, 2013; Rezende et al., 2014). Kingma and Welling (2013) show that re-parameterization of variational lower bound enables the representation of random variables (trainable parameters) as deterministic functions with noise and yields an estimator that can be conveniently optimized via stochastic gradient descent. Bayes-by-backprop also utilizes the re-parameterization trick to estimate the derivatives of the expectations. The weight uncertainty is later utilized to push the trade-off between exploration and exploitation in reinforcement learning. The results in Blundell et al. (2015) show that Bayes-by-backprop outperforms dropout regularization with a rather simple formulation of the loss, also known as the variational free energy.

3. Data and methodology

3.1. Data pre-processing

We consider four cyclone datasets for evaluating the proposed variational inference framework. These datasets include cyclone tracks in the North Indian Ocean, South Pacific Ocean, North-West Pacific ocean, and South Indian ocean which are taken from the Joint Typhoon Warning Center (JTWC) (Anon, 2015). The JTWC has the responsibility of issuing tropical cyclone warnings in these regions for the United States Department of Defense and other government agencies. They primarily protect military ships and aircraft jointly operated with the United States and other countries around the world. The website features yearly data as an archive consisting of individual cyclone tracks as text files.

Each track in these datasets features cyclone latitude, longitude, and wind intensity recorded at regular intervals for the duration of the cyclone. The location and the wind-intensity of ith cyclone in the dataset at time-step t is represented as a vector: $(x_1^i[t], x_2^i[t], x_3^i[t])$. Each cyclone in the dataset has an initial location represented by latitude and longitude pair $(x_1^i[0], x_2^i[0])$ and duration given by the total time (time-steps T_i). The unique initial positions of the cyclones, thus, introduce a spatial artefact in the dataset. As a consequence of the spatial artefact, two cyclones with similar trajectory patterns and differing initial positions are interpreted as different cyclones by the model.

We adopt a phase-shift reconstruction of the trajectories to remove any spatial dependence introduced by differences in starting positions. Since we are only concerned with the trajectory of the cyclone and not the starting position/location, we recenter the cyclone tracks such that all the cyclones start at the same location *i.e.* $(0, 0)$. This is done with the help of the following transformation to the original tracks:

$$(x_1^i[t], x_2^i[t]) = (x_1^i[t] - x_1^i[0], x_2^i[t] - x_2^i[0]) \tag{1}$$

where $i \geq 0$ and the pair $(x_1^i[t], x_2^i[t])$ represents the latitude and the longitude of the cyclone i at the time-step t. A similar transformation can be applied to the wind intensity values as well:

$$x_3^i[t] = x_3^i[t] - x_3^i[0] \tag{2}$$

Similar to the previous notations, $x_3^i[t]$ denotes the wind intensity of cyclone i at time-step t, while $x_3^i[0]$ is the initial wind intensity of the cyclone. Note that (x_1^i, x_2^i, x_3^i) are all column vectors of span $\mathbb{R}^{T_i}$.

We apply Taken's embedding theorem (Takens, 1981) for state–space reconstruction (windowing) of time-series data. The theorem uses a sliding window of size α at regular time interval β. Since the cyclone track data is a multivariate time series, each cyclone track for our respective Bayes-RNN model is represented as a three-dimensional tensor of shape $((T_i - \alpha) \times \alpha \times 3)$. Using this three-dimensional notation, the cyclone track is represented as $(T_i - \alpha)$ windows of length α containing the location of cyclone (x_1, x_2, x_3). We only consider single step-ahead prediction, where each training example in track i at time-step t is represented using an input matrix $\mathbf{X}_t^i$ of shape $(\alpha \times 3)$, and an output vector $\mathbf{Y}_t^i$ of size 3, for a single step ahead (time step):

$$\mathbf{X}_t^i = \begin{bmatrix} x_1^i[t+1] & x_2^i[t+1] & x_3^i[t+1] \\ x_1^i[t+2] & x_2^i[t+2] & x_3^i[t+2] \\ \vdots & \vdots & \vdots \\ x_1^i[t+\alpha] & x_2^i[t+\alpha] & x_3^i[t+\alpha] \end{bmatrix} \varsigma \; \mathbf{Y}_t^i = \begin{bmatrix} x_1^i[t+\alpha+1] \\ x_2^i[t+\alpha+1] \\ x_3^i[t+\alpha+1] \end{bmatrix} \tag{3}$$

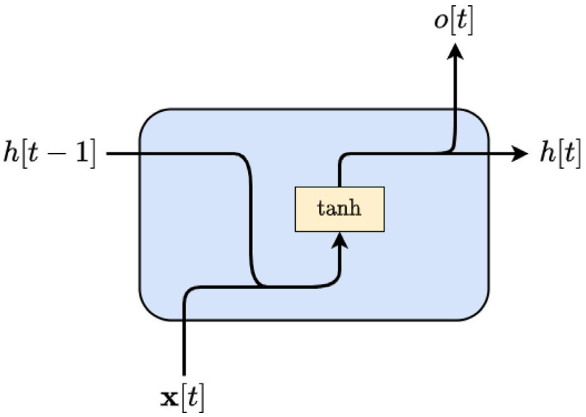


Fig. 1. Architecture of an RNN layer.

where $t \in [0, 1, 2, \dots, T_i - \alpha - 1]$. In the case of multi-step (m) ahead cyclone track prediction (not implemented in this work), the target is a matrix of shape $(m \times 3)$.

3.2. Recurrent neural networks

RNNs have the capability to model dynamic temporal dependencies in sequential data using internal representation of previous states and current inputs (Elman, 1990). This behaviour allows RNNs to predict future states of a time-series by modelling them as a function of previous states. The output of an RNN at time-step t is computed as:

$$h[t] = \tanh\Big(W_{ii}\mathbf{x}[t] + b_{ii} + W_{hi}h[t-1] + b_{hi}\Big) \tag{4}$$

$$o[t] = \tanh\Big(W_{oh}h[t] + b_{oh}\Big) \tag{5}$$

where $h[t]$ and $\mathbf{x}[t]$ are the current hidden state and input vectors, and $h[t-1]$ is the hidden state from previous time-step. As shown in Eq. (4), the hidden state at time-step t is computed as a function of the weighted sum of previous hidden state $h[t-1]$ and input vector $\mathbf{x}[t]$. The weights for this computation are given by W_i, and bias b_h. Finally, the output $o[t]$ is computed as a weighted sum of the hidden states at time-step t, as shown in Eq. (5). The weights for this computation are given by W_o and the bias b_o (see Fig. 1).

3.3. Long short-term memory

Despite their extraordinary effectiveness, RNNs tend to suffer from the problem of *vanishing gradients*, wherein the gradient used for backpropagation decreases exponentially with the number of hidden layers. This causes RNNs to become less useful when it comes to learning relatively longer temporal dependencies. Long short-term memory (LSTM) networks deal with this problem by dividing each unit into multiple gates. A conventional LSTM unit includes an input gate, a cell state, a forget gate, and an output gate. The cell state is responsible for storing information over arbitrary time intervals whereas the other gates regulate the flow of this information. The input to the LSTM at time-step t is processed through these gates by:

$$i[t] = \sigma\Big(W_{ii}\mathbf{x}[t] + b_{ii} + W_{ih}h[t-1] + b_{ih}\Big) \tag{6}$$

$$f[t] = \sigma\Big(W_{fi}\mathbf{x}[t] + b_{fi} + W_{fh}h[t-1] + b_{fh}\Big) \tag{7}$$

$$o[t] = \sigma\Big(W_{oi}\mathbf{x}[t] + b_{oi} + W_{oh}h[t-1] + b_{oh}\Big) \tag{8}$$

$$\hat{c}[t] = \tanh\Big(W_{oi}\mathbf{x}[t] + b_{oi} + W_{oh}h[t-1] + b_{oh}\Big) \tag{9}$$

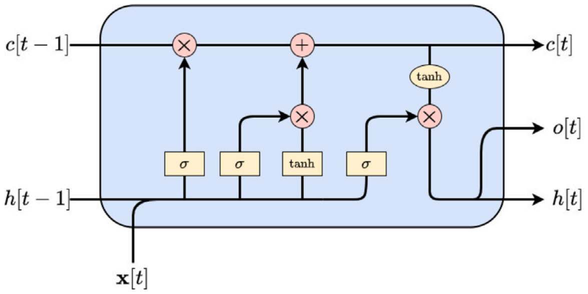


Fig. 2. Architecture of an LSTM layer.

where $x[t]$ is the current input state and $h[t]$ is the current hidden state which also acts as the output for the LSTM state. The forget, input and output gate are represented by $f[t]$, $i[t]$ and $o[t]$, respectively. The output from these gates is used to update the hidden and the memory states at time-step t as:

$$c[t] = f[t] \odot c[t-1] + i[t] \odot \hat{c}[t] \tag{10}$$

$$h[t] = o[t] \odot \tanh(c[t]) \tag{11}$$

where $\hat{c}[t]$ is referred to as the candidate or memory state and $c[t]$ refers to the cell state.

As shown in the Eqs. (6), the input gate uses separate weights and bias $\{W_{ii}, b_{ii}\}$ and $\{W_{ih}, b_{ih}\}$ for the inputs $\mathbf{x}_t$ and previous hidden states h_{t-1}, respectively. The forget gate (Eq. (7)) and the output gate (Eq. (8)) similarly compute a linear combination of hidden and input states along with a *sigmoid* activation function. Although the candidate state (Eq. (9)) is also computed as a function of hidden and input states, the activation function used is a *hyperbolic tangent function* (*tanh*) instead of a *sigmoid*. The cell state is then computed by adding the matrix dot products of $f[t]$ and $c[t-1]$ to the matrix dot product of $i[t]$ and $\hat{c}[t]$. Finally, the hidden state is computed by taking the matrix dot product of $o[t]$ and hyperbolic tan of $c[t]$ (see Fig. 2).

3.4. Variational inference for recurrent neural networks

Given the input vector $\mathbf{x} \in \chi$, the target labels $y \in \mathcal{Y}$ and the neural network parameters $\boldsymbol{\theta}$; the output of neural network model is given by the objective function $f(\mathbf{x}, \boldsymbol{\theta})$. Here, χ is the feature space and $\mathcal{Y}$ is the label space. In this formulation, the model tries to learn the objective function $f(.)$ with parameters θ for the pair $(\mathbf{x}, y)$ during the training process. In the case of a single layer RNN model, the model parameters are given as $\boldsymbol{\theta} = \{\mathbf{u}, \mathbf{v}, \mathbf{w}, b_h, b_o\}$.

This RNN model can be viewed as a probabilistic model with output probability $P(y|\mathbf{x}, \boldsymbol{\theta})$, given the input $\mathbf{x}$ and parameters $\boldsymbol{\theta}$. We note that in the case of classification problems, $P(y|\mathbf{x}, \boldsymbol{\theta})$ represents a categorical distribution; whereas, in the case of regression problems, the target is a real value, *i.e.* $y \in \mathbb{R}$ and $P(y|\mathbf{x}, \boldsymbol{\theta})$ follows a Gaussian distribution.

We can learn the RNN model parameters (weights and biases) by finding the *maximum likelihood estimate* (MLE) of the parameters $\hat{\boldsymbol{\theta}}$, given as,

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} \prod_i P(y_i|\mathbf{x_i}, \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \sum_i \log P(y_i|\mathbf{x_i}, \boldsymbol{\theta})\end{aligned} \tag{12}$$

This can be achieved with the help of a gradient-based optimization which is also known as backpropagation, provided $P(y|\mathbf{x})$ is differentiable with respect to $\boldsymbol{\theta}$. In order to introduce regularization, we can assume a prior distribution for the parameters $\boldsymbol{\theta}$, given by $P(\boldsymbol{\theta})$. $\hat{\boldsymbol{\theta}}$ then represents the *maximum a posteriori* (MAP) estimate of the parameters, given as:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_i \Big(\log P(y_i|\mathbf{x_i}, \boldsymbol{\theta}) + \log P(\boldsymbol{\theta})\Big) \tag{13}$$

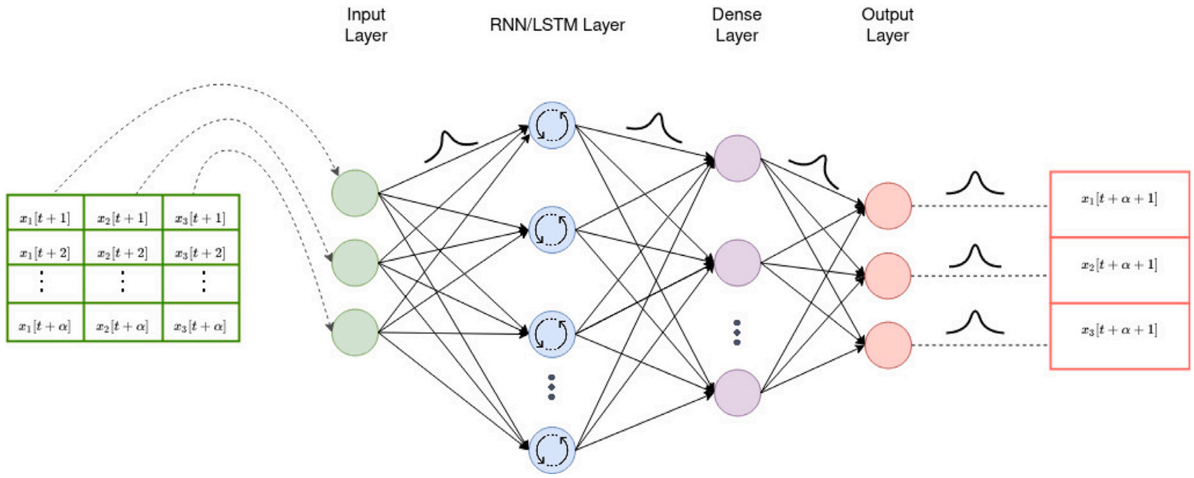


Fig. 3. Bayesian recurrent neural network for cyclone track and wind intensity prediction.

We use a Gaussian prior with a zero mean and a standard deviation τ to introduce L2 regularization which is also known as weight-decay (Krogh and Hertz, 1992) in the backpropagation literature. The prior density is then given as:

$$P(\theta) \propto \frac{1}{(2\pi\tau^2)^{L/2}} \times \exp\left\{ -\frac{1}{2\tau^2}\left(\sum_{l=1}^{L}\theta_l\right)\right\} \tag{14}$$

where L is the total number of trainable parameters in the neural network. The above formulation results in the single-point estimate of θ that maximizes the posterior density. However, to infer the posterior distribution of the model parameters, we can use variational inference to approximate the posterior distribution.

We begin by assuming a variational posterior on the neural network weights given by $q(\theta|\delta)$, parameterized by δ. Variational inference (learning) is then used to find the value of δ that minimize the KL-divergence between the variational posterior and the true posterior:

$$\hat{\delta} = \arg\min_{\delta} \mathrm{KL}[q(\theta|\delta) \ \| \ P(\theta|\mathcal{D})] \tag{15}$$

$$= \arg\min_{\delta} \mathrm{KL}[q(\theta|\delta) \ \| \ P(\theta)] - \mathbb{E}_{q(\theta|\delta)}[\log P(\mathcal{D}|\theta)] \tag{16}$$

where $\mathcal{D}$ represent the data containing the input and output pairs $(\mathbf{x}, y)$, $P(\theta|\mathcal{D})$ is the true posterior and $P(\mathcal{D}|\theta)$ is the likelihood. The loss function then is given as:

$$\mathcal{L} = \mathrm{KL}[q(\theta|\delta) \ \| \ P(\theta)] - \mathbb{E}_{q(\theta|\delta)}[\log P(\mathcal{D}|\theta)] \tag{17}$$

We can use the formulation in Blundell et al. (2015) to approximate this loss function using Monte Carlo sampling given by:

$$\mathcal{L} \approx \sum_{i=1}^{m} \log q(\theta^{(i)}|\delta) - \log P(\theta^{(i)})P(\mathcal{D}|\theta^{(i)}) \tag{18}$$

where $\theta^{(i)}$ represents the ith sample drawn from the variational posterior $q(\theta|\delta)$. The loss value depends directly on the particular weights drawn from the variation posterior.

3.5. *Variational inference framework for cyclone track and intensity prediction*

The raw cyclone trajectories (given by latitude, longitude, and wind intensity) taken from the JTWC are re-centred and reconstructed by windowing the sequence using Taken's theorem, as discussed in the data pre-processing section. Fig. 4 shows the framework for cyclone trajectory prediction using Bayes-RNN and Bayes-LSTM. We note that the cyclones are concatenated and divided into a training and test set defined by year. We present the input sequences (represented by matrix $\mathbf{X}_t$), into two Bayes-RNN models to generate one-step-ahead predictions. One of the models generates the predictions for latitude and longitude via two output neurons. The other generates the predictions for wind intensity. Note that the trailing data points that do not make it into a window are discarded from training data. We then compute the log of *likelihood, prior* and the *variational density* using the Bayes-RNN predictions and the corresponding one-step ahead target (actual value). The *variational density* is calculated by adding the log probability over the Gaussian distributions parameterized by the mean and variance values given in the weight matrix. The *likelihood, prior* and the *variational density* are then used to calculate the loss as described in Eq. (18).

Algorithm 1 provides the details for training the Bayes-RNN model for cyclone trajectory and wind-intensity prediction tasks. We note that the specific RNN model can be either simple RNN or an LSTM-based RNN. The algorithm begins by defining the RNN model ($f(\mathbf{x}, \theta)$), initializing the variational parameters (δ), and setting the hyper-parameters, i.e. number of training epochs (N_{epochs}), number of Monte Carlo samples for computing the variational loss (N_{sample}), and the prior variance (τ^2). We consider the variational posterior to be a diagonal Gaussian distribution. Therefore, the RNN parameters are given as: $\theta \sim \mathcal{N}(\mu, \sigma^2)$; where $\sigma = \log(1 + \exp(\rho))$. The function $f(x) = \log(1 + \exp(x))$ is known as the softplus, which is used to ensure that the value of σ is always positive. In order to sample the parameters θ, we first sample from a standard Gaussian and shift it by mean (μ) and scale by the standard deviation (σ), as shown in steps 1 and 2 of the algorithm. Steps 1 and 2 are repeated to generate $N_{samples}$ (model predictions). In step 3, the model prediction is used to compute the variational loss (Eq. (18)). The loss function is optimized using gradient-based optimization, also known as backpropagation.

Fig. 3 shows the Bayes-RNN framework for cyclone trajectory and intensity prediction. The neural network architecture consists of four layers: an input layer, an RNN or LSTM layer, a dense layer, and an output layer. The input layer as well as the output layer contain three neurons while the number of recurrent and dense layer neurons are decided set empirically for each dataset. We note that we use backpropagation to learn the variational parameters ($\delta = (\mu, \rho)$) that parameterize the variational posterior distribution for the RNN model parameters (θ). Thus, by learning the parameters such as μ and ρ, we learn the distribution over the RNN model parameters (θ).

4. Experiments and results

In this section, we present the details of the dataset, the experiments, and the results from Bayes-RNN. We compare the performance of the Bayes-RNN that utilize simple RNN and LSTM models with their non-Bayesian counterparts.

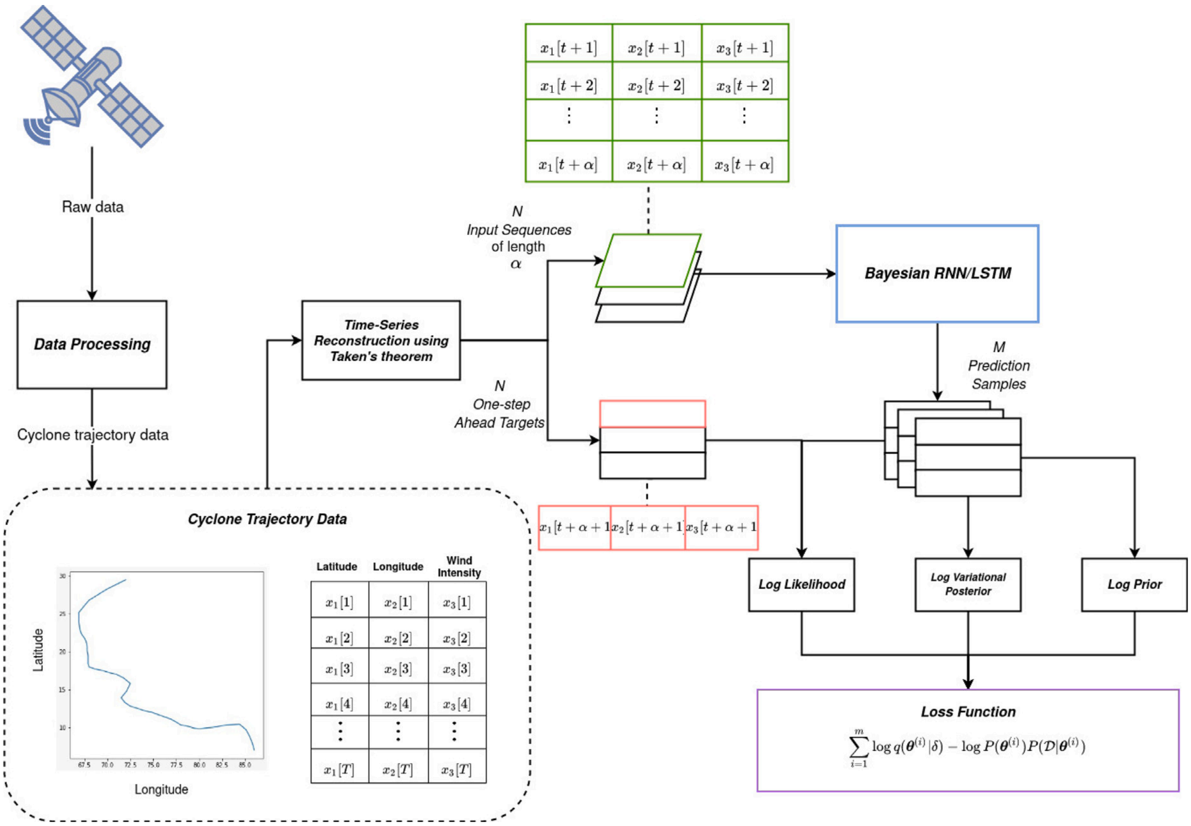

Raw data
Data Processing
Cyclone trajectory data
Time-Series Reconstruction using Taken's theorem
N Input Sequences of length α
$x_1[t+1]$ $x_2[t+1]$ $x_3[t+1]$
$x_1[t+2]$ $x_2[t+2]$ $x_3[t+2]$
$x_1[t+\alpha]$ $x_2[t+\alpha]$ $x_3[t+\alpha]$
Bayesian RNN/LSTM
M Prediction Samples
N One-step Ahead Targets
$x_1[t+\alpha+1$ $x_2[t+\alpha+1$ $x_3[t+\alpha+1$
Log Likelihood
Log Variational Posterior
Log Prior
Loss Function
$\sum_{i=1}^{m} \log q(\theta^{(i)}|\delta) - \log P(\theta^{(i)})P(\mathcal{D}|\theta^{(i)})$
Cyclone Trajectory Data
Latitude
Longitude
Latitude
Longitude
Wind Intensity
$x_1[1]$ $x_2[1]$ $x_3[1]$
$x_1[2]$ $x_2[2]$ $x_3[2]$
$x_1[3]$ $x_2[3]$ $x_3[3]$
$x_1[4]$ $x_2[4]$ $x_3[4]$
$x_1[T]$ $x_2[T]$ $x_3[T]$

Fig. 4. Framework for cyclone track and wind intensity prediction using Bayes-RNN and Bayes-LSTM.

```
Data: Cyclone track dataset
Result: Posterior distribution over neural network parameters $P(\theta|\mathbf{y})$
i. Define recurrent neural network model, $y = f(\mathbf{x}, \theta)$
ii. Initialize the variational parameters, $\delta = (\mu, \rho)$
iii. Set hyper-parameters: $N_{epoch}, N_{sample}$, and $\tau$
iv. Pre-preprocess data into input–output matrices: $(\mathbf{X}, \mathbf{Y})$
for $n = 1, \dots, N_{epoch}$ do
    for $i = 1, \dots, N_{sample}$ do
        1. Sample noise:
            $\epsilon \sim \mathcal{N}(0, I)$
        2. Compute the parameters
            $\theta^{(i)} = \mu + \log(1 + \exp(\rho)) \cdot \epsilon$
    end
    3. Compute variational loss:
        $\mathcal{L} = \sum_{i=1}^{N_{sample}} \log(q(\theta^{(i)}|\delta)) - \log(P(\theta^{(i)})P(\mathcal{D}|\theta^{(i)}))$
    5. Compute gradients: $\Delta\mu$ and $\Delta\rho$
    6. Update the variational parameters:
        $\mu = \mu - \gamma\Delta\mu$
        $\rho = \rho - \gamma\Delta\rho$
end
```

Algorithm 1: Variational inference for uncertainty quantification in Bayes-RNN and Bayes-LSTM with application to cyclone track and intensity prediction.

4.1. Dataset

Table 1 shows the four selected regions from the JTWC with the number of unique cyclone trajectories for each region. Our processed version of the JTWC dataset used for this work is available on github.[1] We note that the terms “hurricane” and “cyclone” are merely distinguished by their names in the dataset and there are no major differences between these terms. We refer to the hurricanes used in the datasets as cyclones, hereafter. The raw dataset consists of cyclone parameters (cyclone no., date & time, latitude, longitude and the wind speed in knots) for each cyclone recorded at 6-hour time intervals. Fig. 5 shows the movement of 20 randomly sampled cyclones in each region. Aside from the data pre-processing mentioned in Section 3.1, we also normalize the coordinates of each cyclone by subtracting the mean value over the region and dividing it by the standard deviation. The dataset includes cyclones from 1985 to 2019, out of which 70% of the data points are used for training while the remaining as test samples.

Table 1
Number of cyclones in each dataset (region).

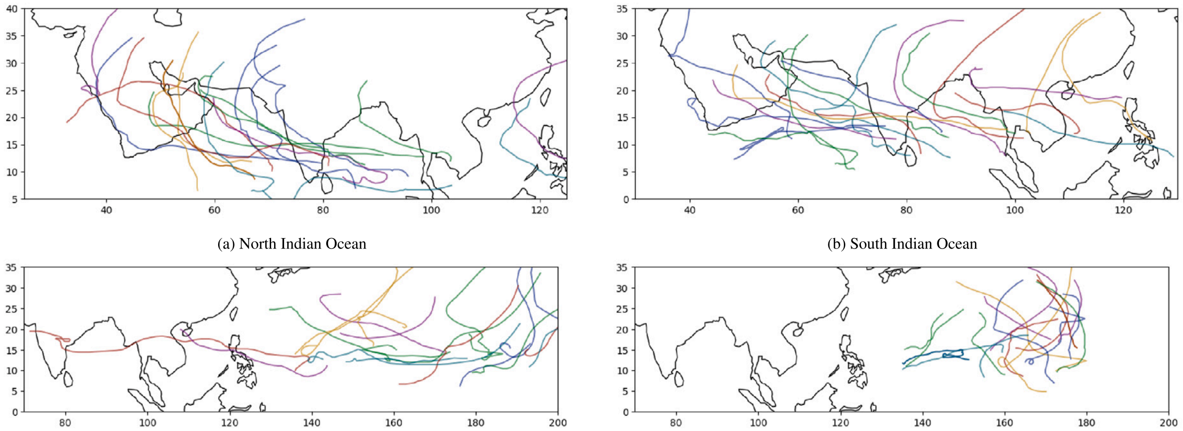
Dataset	Number of cyclone tracks
North Indian Ocean	563
North-west Pacific Ocean	490
South Indian Ocean	650
South Pacific Ocean	380

Fig. 6 presents the histogram of the lengths of different tracks. We observe that more than half of cyclones last for up to 40 to 80 h in duration. While the number of cyclones in the North-west Pacific Ocean and the South Pacific Ocean differs, the cyclone duration in these two regions shows a similar pattern. A similar observation can be made for North Indian and South Indian Oceans as well.

4.2. Experiment setting

We divide the experiments into two major tasks with two different models, i.e cyclone track prediction and intensity prediction. Hence, a model is used for track prediction and a separate one is used for wind intensity. We compare the performance of 4 different types of models (RNN, LSTM, Bayes-RNN, Bayes-LSTM) for the track and intensity

[1] https://github.com/sydney-machine-learning/cyclonedatasets



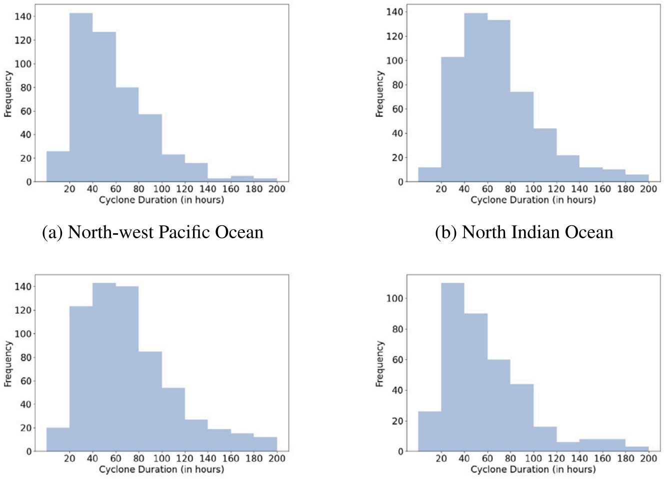
(a) North Indian Ocean

(b) South Indian Ocean

(c) North-west Pacific Ocean

(d) South Pacific Ocean

Fig. 5. Randomly sampled 20 cyclones observed in each of the four datasets (regions).



(a) North-west Pacific Ocean

(b) North Indian Ocean

(c) South Indian Ocean

(d) South Pacific Ocean

Fig. 6. Histogram of cyclone duration for the respective ocean regions.

prediction for each of the ocean regions as described in Section 3.1. In order to maintain uniformity amongst the models, and for a fair comparison, we take the most basic form of each model — with just a single hidden layer and a minimal number of hidden neurons to represent the problem, obtained from trial experiments.

The training data consists of time series from each cyclone taken with a window of size 4 and the corresponding target to be predicted. Since each data point is recorded at a 6-hour interval, a window size of 4 represents a 24-hour timeframe. In the case of the track prediction task, each element of the sequence consists of a pair of values (the corresponding latitude and longitude); whereas, for the intensity prediction task, each element in the sequence consists of only the intensity value.

In practice, we have noticed that this form of optimization is highly sensitive to the hyper-parameters, likelihood function, and prior density function. Blundell et al. (2015) use a scaled mixture of two Gaussians as the prior distribution. However, in our case, we empirically found that a high variance Gaussian prior yielded better accuracy performance compared to the sub-optimal results shown by the mixture prior. Thus, we define the Gaussian prior with zero mean and variance equal to 36 for model training.

Finally, we implement the respective models using the Pytorch machine learning library[2] and train the models using *Intel Core i5-1145G7* processor with 8 cores along with 16 GigaByte random access memory (RAM).

4.3. Evaluation of predictions

Continuous ranked probability score (CRPS) (Matheson and Winkler, 1976) is often used to measure the performance of probabilistic predictions that are scalar continuous values. CRPS is a measure of quadratic distance between the forecast cumulative distribution function (CDF) and the empirical CDF of the observation. For a random variable X with CDF F, the CRPS is computed as follows:

$$CRPS(F,x) = \int_{-\infty}^{\infty} \Big(F(x) - \mathbb{1}(y-x)\Big)^2 \, dy \tag{19}$$

where, $\mathbb{1}$ is the Heaviside step function along real line whose value is given by:

$$\mathbb{1}(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

Gneiting and Raftery (2007) show that the CRPS can be equivalently written as follows,

$$CRPS(F,x) = \mathbf{E}\Big[|X-x|\Big] - \frac{1}{2}\Big[|X-X^*|\Big] \tag{21}$$

where, X and X^* are independent copies of the random variable associated with F. CRPS value close to zero is generally desirable. Although CRPS is common with univariate forecasts, it cannot be used for multivariate predictions as in our case. Therefore, we use energy score (ES) (Gneiting et al., 2008) which can be viewed as a generalization of CRPS suitable for multivariate predictions and is computed as,

$$ES(F,x) = \mathbf{E}\Big[\|X-x\|\Big] - \frac{1}{2}\Big[\| X - X^* \|\Big] \tag{22}$$

where $\| \cdot \|$ is the euclidean norm.

In the case of ensemble predictions of size m, ES is evaluated as,

$$ES(F,x) = \frac{1}{m}\sum_{j=1}^{m} \|x_j - x\| - \frac{1}{2m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} \| x_i - x_j \| \tag{23}$$

The energy score provides a direct method of comparison between deterministic, ensemble and density forecasts.

[2] pytorch.org

Table 2
Intensity prediction performance on train and test set of selected cyclone datasets.

Dataset	Method	Train RMSE	Train ES	Test RMSE	Test ES
North Indian Ocean	RNN	0.036 ± 0.021	0.045	0.037 ± 0.040	0.009
	Bayes-RNN	0.069 ± 0.002	0.047	0.072 ± 0.001	0.028
	LSTM	0.026 ± 0.006	**0.020**	0.020 ± 0.006	**0.005**
	Bayes-LSTM	0.047 ± 0.001	0.057	0.073 ± 0.001	0.049
North-west Pacific Ocean	RNN	0.128 ± 0.227	0.016	0.118 ± 0.241	0.006
	Bayes-RNN	0.059 ± 0.003	0.041	0.052 ± 0.005	0.022
	LSTM	0.047 ± 0.002	0.021	0.035 ± 0.001	0.006
	Bayes-LSTM	0.045 ± 0.001	**0.014**	0.034 ± 0.001	**0.004**
South Indian Ocean	RNN	0.038 ± 0.005	0.030	0.039 ± 0.005	0.016
	Bayes-RNN	0.042 ± 0.003	0.019	0.042 ± 0.004	0.028
	LSTM	0.033 ± 0.001	0.019	0.033 ± 0.001	0.013
	Bayes-LSTM	0.037 ± 0.001	**0.033**	0.037 ± 0.001	**0.013**
South Pacific Ocean	RNN	0.095 ± 0.252	0.036	0.100 ± 0.252	0.021
	Bayes-RNN	0.042 ± 0.002	0.025	0.044 ± 0.004	0.021
	LSTM	0.037 ± 0.002	0.018	0.041 ± 0.003	0.021
	Bayes-LSTM	0.038 ± 0.001	**0.011**	0.040 ± 0.001	**0.019**

Table 3
Track prediction performance on train and test set of selected cyclone datasets.

Dataset	Method	Train RMSE	Train ES	Test RMSE	Test ES
North Indian Ocean	RNN	0.058 ± 0.091	0.019	0.062 ± 0.089	0.017
	Bayes-RNN	0.030 ± 0.003	0.003	0.031 ± 0.004	0.010
	LSTM	0.019 ± 0.004	0.010	0.020 ± 0.004	0.012
	Bayes-LSTM	0.017 ± 0.001	**0.003**	0.017 ± 0.001	**0.006**
North-west Pacific Ocean	RNN	0.080 ± 0.120	0.017	0.076 ± 0.116	0.021
	Bayes-RNN	0.036 ± 0.004	0.011	0.035 ± 0.006	0.017
	LSTM	0.022 ± 0.004	0.012	0.021 ± 0.005	**0.010**
	Bayes-LSTM	0.020 ± 0.001	**0.006**	0.020 ± 0.001	0.011
South Indian Ocean	RNN	0.069 ± 0.152	0.031	0.069 ± 0.151	0.011
	Bayes-RNN	0.034 ± 0.003	0.002	0.034 ± 0.004	0.016
	LSTM	0.016 ± 0.004	0.004	0.016 ± 0.004	0.009
	Bayes-LSTM	0.020 ± 0.001	**0.003**	0.020 ± 0.000	**0.007**
South Pacific Ocean	RNN	0.062 ± 0.076	0.021	0.077 ± 0.081	0.023
	Bayes-RNN	0.036 ± 0.003	0.007	0.036 ± 0.005	0.022
	LSTM	0.026 ± 0.006	0.012	0.028 ± 0.007	0.015
	Bayes-LSTM	0.023 ± 0.000	**0.007**	0.022 ± 0.000	**0.014**

In addition to the Energy score, we also report the Root Mean Squared Error (RMSE) of the predictions which are computed as,

$$RMSE = \sqrt{\sum_{i=0}^{N} \frac{(\hat{y}_i - y_i)^2}{N}} \tag{24}$$

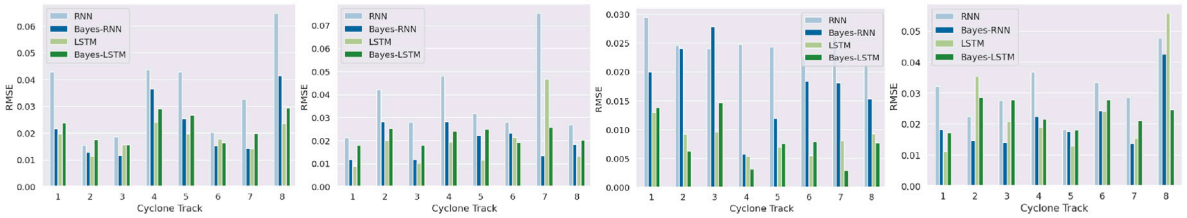
where, $\hat{y}_i$ refers to the predicted value, y_i refers to the target value, and N is the total number of training samples (windows). We chose the RMSE as one of the performance metrics because it is commonly used to benchmark continuous forecasting models in a Spatio-temporal setting.

4.4. Results

Tables 2 and 3 present the results (RMSE and ES) for train and test sets of the track and wind-intensity prediction tasks, respectively, for cyclones in the four different ocean regions. They feature the mean and confidence interval of RMSE (denoted by ±) along with the energy scores (ES) across 100 model predictions using RNN, Bayes-RNN, LSTM and Bayes-LSTM. The model weights for Bayes-RNN and Bayes-LSTM were drawn from their variational posterior distribution 100 times and then used for prediction on the test set. In the case of RNN and LSTM models, 100 models were trained independently. We observe that in the case of intensity prediction for the North Indian ocean dataset, LSTM yields the best ensemble ES and RMSE values on the train and test sets. Bayes-RNN and Bayes-LSTM perform slightly worse than their single-point estimate counterparts. However, in all the other problem sets (both track and intensity prediction) the variational counterparts (Bayes-LSTM and Bayes-RNN) show a lower value of ES and RMSE values on both train and test sets. It is clear that the variational models provide a good approximation of the empirically observed distribution of the single-point RNN and LSTM models.

We also notice that in many cases the Bayes-RNN and Bayes-LSTM provide a lower mean and variance of RMSE values compared to the RNN and LSTM models. This is primarily because RNN and LSTM models have a tendency to get stuck in the local minima which affects the overall distribution of the RMSE values, leading to a higher mean and variance. The variational models do not suffer from this issue and provide a good approximation of the true posterior distribution.

Fig. 7 presents a barplot of RMSE values reported by randomly selected models of each type on 10 random cyclone tracks of each ocean for the cyclone trajectory prediction task. In the case of RNN and LSTM models, we randomly select a model from 100 independently trained models while in Bayes-RNN/Bayes-LSTM models, we randomly sample weights from the learned variational distribution of parameters. We can observe that the LSTM models give consistently better performance (lower RMSE) values as compared to the RNN models. The Bayesian models also give better or similar performance as compared to their vanilla counterparts for most tracks. Similarly, Fig. 8 is a barplot of RMSE values reported by the randomly selected models of each type on 10 random cyclone tracks of each ocean for the cyclone intensity prediction task. We can observe similar trends in model performance as in the trajectory prediction task. However, we also observe that for some of the trajectories the vanilla RNN provides the lowest RMSE value for intensity predictions (North-west Pacific Ocean). This differs from the trend observed in the reported mean of RMSE in Table 2. Vanilla RNN often suffers from the problem of local optima and due



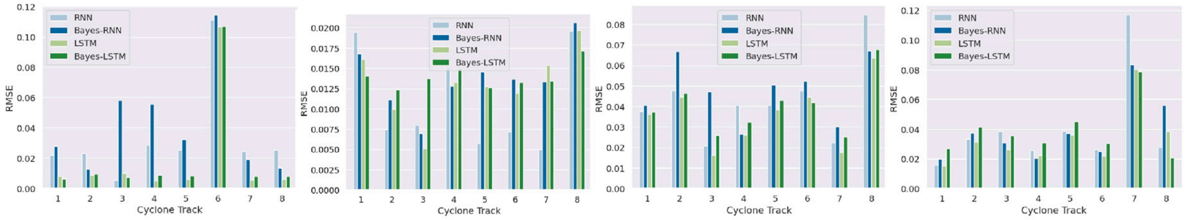
(a) North-west Pacific Ocean

(b) North Indian Ocean

(c) South Indian Ocean

(d) South Pacific Ocean

Fig. 7. RMSE for the performance of each model on 10 random cyclones from each oceanic region for the Track prediction task.



(a) North-west Pacific Ocean

(b) North Indian Ocean

(c) South Indian Ocean

(d) South Pacific Ocean

Fig. 8. RMSE for the performance of each model on 10 random cyclones from each oceanic region for the intensity prediction task.

to this some of the samples of RNN fail to reach the global optimum solution during training. This leads to a higher overall mean and variance in the RMSE of predictions. Furthermore, the intensity values are highly discretized (in the multiples of 5) as compared to the trajectory coordinates. This leads to a much more uneven dataset and so the disparity in RMSE values of different cyclones is also much more prominent in the intensity prediction task as compared to trajectory prediction. For example, cyclone track no. 6 for the North-West pacific ocean in the sub Fig. 8(a) has a disproportionally large RMSE value when compared with other cyclones.

Fig. 9 contains the subplots of actual trajectories as well as the trajectories predicted by the RNN, LSTM, Bayes-LSTM and Bayes-RNN models for random cyclones from the 4 respective ocean regions. The dates when the recording for each cyclone started have also been provided in the captions. Do note that the model was trained on neither of these cyclones prior to prediction.

Here we can easily observe that for the sampled cyclone in the North-West Pacific Ocean, the Bayes-LSTM and the LSTM models predict the trajectory very accurately for the most part, much more so than the Bayes-RNN and RNN models. We see a somewhat larger discrepancy between the predicted tracks and the actual track for the North Indian ocean, but the general trend for the cyclone has still been captured. For the South Indian and South Pacific Oceans, we observe very similar predictions by the models, and the actual trajectory has to a large extent been accurately predicted.

Fig. 10 allows us to see how the prediction by Bayesian models would probably work in real-time, by showing the true position of the cyclone, and the possible positions as predicted by the Bayes-LSTM model in form of a probability distribution, visualized by a contour.

Fig. 11 further plots full cyclones selected randomly from each ocean along with their true values and the probability distribution of the prediction at each point as given by the Bayes-LSTM and Bayes-RNN models, in the form of a contour.

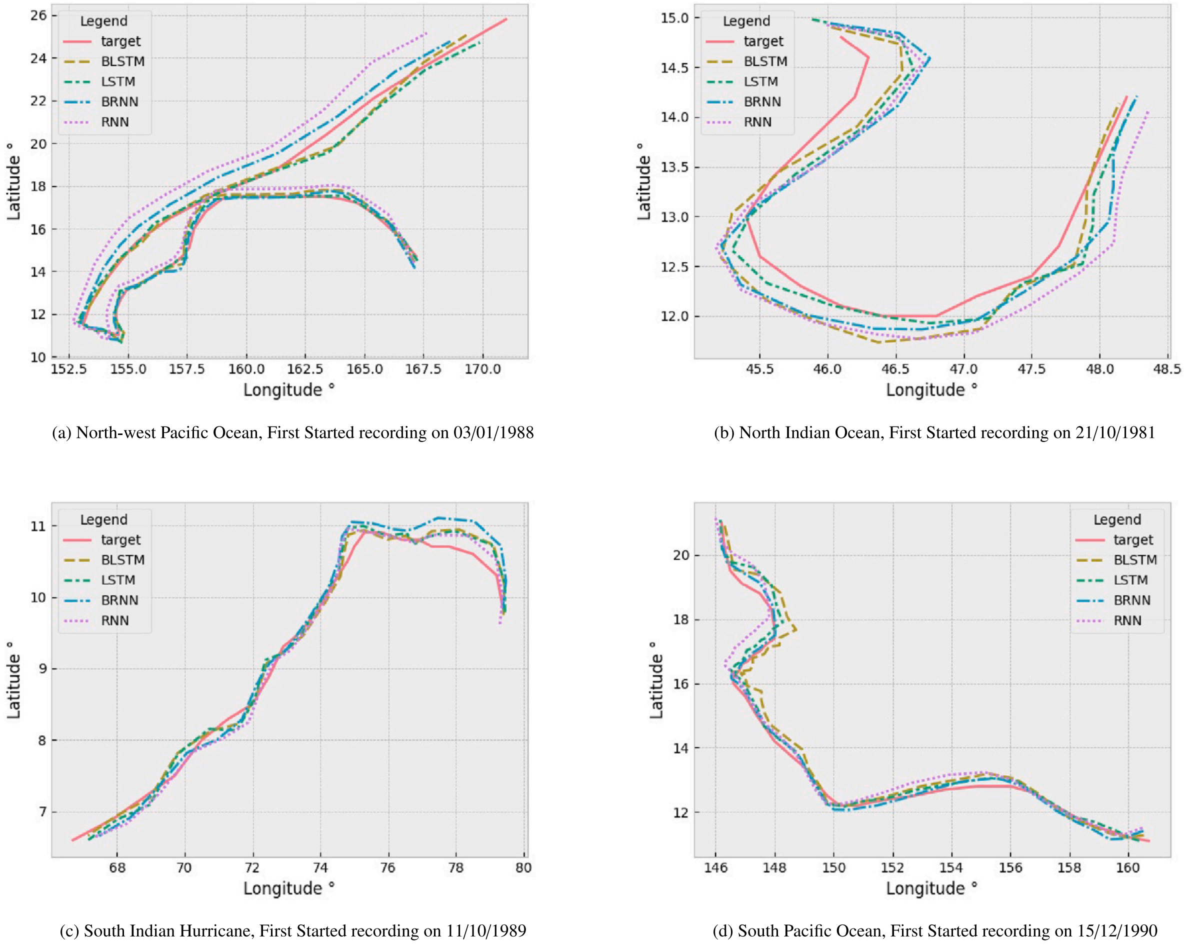
5. Discussion

Given the chaotic and spatio-temporal nature of cyclones (Fraedrich and Leslie, 1989; Zhang and Tao, 2013), recurrent neural networks (including simple RNNs, and LSTM networks) are generally preferred models for prediction of cyclone trajectories. The results of cyclone track and intensity prediction show that Bayes-RNN and Bayes-LSTM models powered via variational inference show either comparable or better prediction accuracy in most of the cases when compared to canonical simple RNN and LSTM models. We were able to approximate the posterior distribution of RNN model weights via variational inference, which has been used to quantify uncertainty in the prediction using the predictive (posterior) distribution.

Bayes-RNN and Bayes-LSTM models via variational inference can be highly sensitive to hyper-parameters (Blundell et al., 2015) such as prior variance τ^2, learning rate, number of Markov samples $N_{samples}$. Thus, the choice of hyper-parameters can lead to drastically different performance results and may affect the convergence. Although, we have used a rather less informative prior, i.e. a zero centred isotropic Gaussian with large variance, a more informative prior may provide a high level of regularization leading to difference in the test performance.

Variational inference methodologies assumes a family of distributions whose parameters are learnt as a result of the optimization process. On the other hand, MCMC methods draw samples directly from the posterior distribution. Hence, a limitation of Bayes-RNN is that we are restricted by the family of distribution we assume as the variational distribution. Although with infinite compute capability MCMC may provide the true posterior via the drawn samples, MCMC methods face scalability issues with high dimensional parameter (such as in the case of neural networks); however, these are continuously being addressed with efficient proposal distributions and sampling schemes (Chandra et al., 2019b; Chandra and Kapoor, 2020; Chandra et al., 2021a,b). Overall, we found that the proposed approach provides a good approximation of the posterior distribution given the family of variational distribution. In our case, we assumed Gaussian distribution over the neural network weights; however, this assumption can be updated later with more insights.

RNN and LSTM models are excellent models for capturing temporal dependencies; however, these models lack the ability to capture spatial correlation in the cyclone data, as shown by Zhang et al. (2018). Recently, it has been shown that temporal convolutional neural networks (TCNN) excel in preserving the spatial correlations while also learning the temporal dependencies in such data (Lea et al., 2017; Yan et al., 2020); however, it has not been specifically tested for cyclone prediction. In future work, the proposed Bayesian framework via variational inference could be extended to TCNN models for improved spatial correlation representation. We model the cyclone track prediction as a multivariate time-series forecasting problem where we only use historical values of latitude and longitude information. The wind-intensity



(a) North-west Pacific Ocean, First Started recording on 03/01/1988

(b) North Indian Ocean, First Started recording on 21/10/1981

(c) South Indian Hurricane, First Started recording on 11/10/1989

(d) South Pacific Ocean, First Started recording on 15/12/1990

Fig. 9. Combined trajectories predicted by different models for cyclones/hurricanes in different oceans.

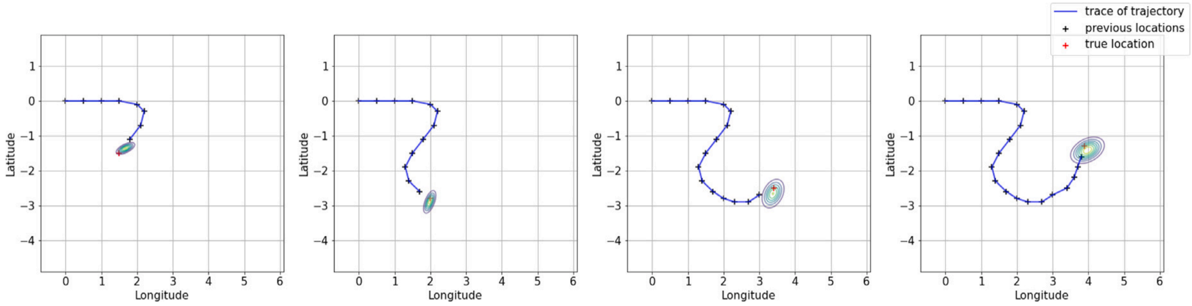
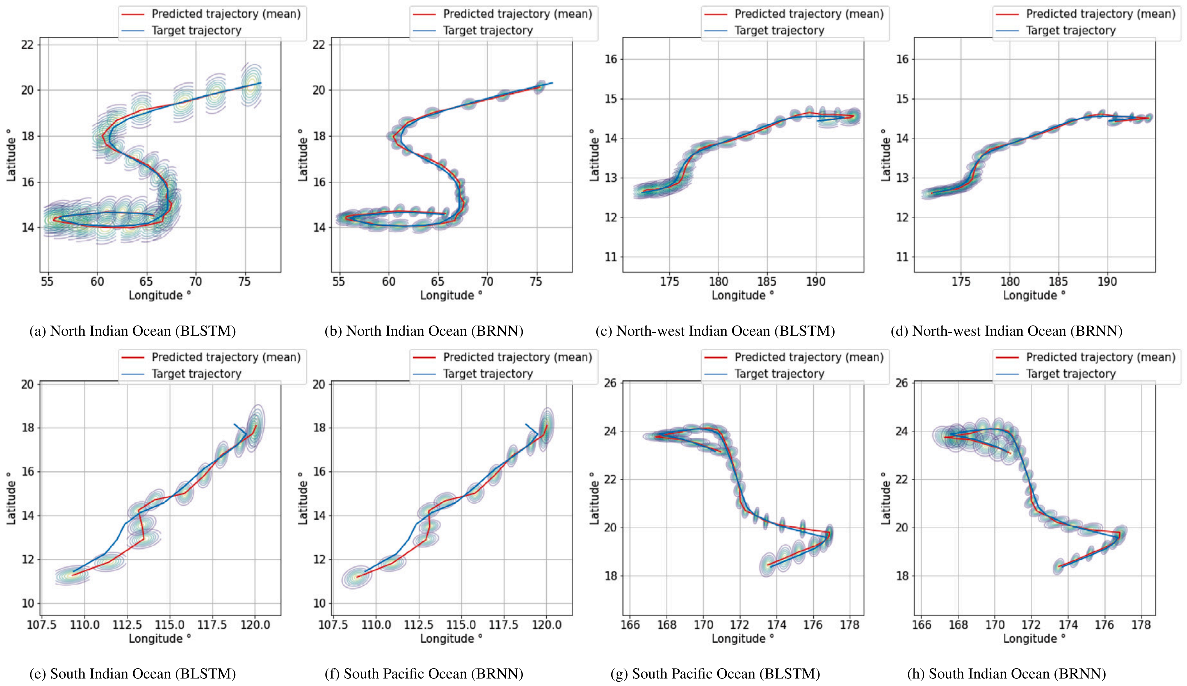


Fig. 10. Progression of a randomly selected cyclone track from the North Indian Ocean dataset plotted at 5th, 9th, 13th and 17th timestep of the trajectory where the contours show the likelihood of the location of the cyclone at the next timestep.

is modelled as a univariate prediction problem. This methodology can be extended to multi-step ahead prediction and addition to this, a multivariate approach could include more features, such as the distance of the cyclone to the landfall, the sea surface temperature, which are not taken into account in this study. These features could possibly lead to better predictions which can be a part of future studies.

6. Conclusion and future work

We presented a Bayesian framework via variational inference for approximating the posterior distribution of deep learning model parameters for the cyclone trajectory and wind intensity prediction problem. We first removed the spatial artefact in the original cyclone data by re-centring the cyclone tracks to standardize the starting position. We provided a comparison of the proposed Bayesian framework and the results showed better accuracy in most cases when compared to the canonical methods (RNN and LSTM). We provided a visualization to quantify uncertainty in predictions for the case of track prediction, which has been challenging since it is a spatio-temporal prediction problem and therefore requires the model to capture trends along both, the time and the spatial dimension simultaneously. In addition, the uncertainty in track prediction is difficult to represent on a two-dimensional graph of latitude and longitude axis, when the data is represented chronologically due to the spatial correlation.



(a) North Indian Ocean (BLSTM) (b) North Indian Ocean (BRNN) (c) North-west Indian Ocean (BLSTM) (d) North-west Indian Ocean (BRNN)

(e) South Indian Ocean (BLSTM) (f) South Pacific Ocean (BRNN) (g) South Pacific Ocean (BLSTM) (h) South Indian Ocean (BRNN)

Fig. 11. Trajectories of various cyclones in each of the four ocean areas as predicted by BRNN and BLSTM along with contour plot of confidence intervals.

Software and data availability

We provide open source code and data for our proposed framework via Github repository with following meta information.

- Name: VarRNN-Cyclones
- Developers: Arpit Kapoor and Anshul Negi
- Contact email: kapoor.arpit97@gmail.com
- Compatible Operating System: Mac/Linux/Windows
- Developed and tested: Ubuntu 20.04 (Linux)
- Size of repository: 1.55MB
- Year published: 2022
- Source: Github[3]

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The link to data and code has been shared as part of the submission.

Acknowledgment

We thank the Australian Government for supporting this research through the Australian Research Council's Industrial Transformation Training Centre in Data Analytics for Resources and Environments (DARE) (project IC190100031).

References

Alemany, S., Beltran, J., Perez, A., Ganzfried, S., 2019. Predicting hurricane trajectories using a recurrent neural network. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, no. 01. pp. 468–475.

Ali, M., Jagadeesh, P., Lin, I.-I., Hsu, J.-Y., 2012. A neural network approach to estimate tropical cyclone heat potential in the Indian ocean. IEEE Geosci. Remote Sens. Lett. 9 (6), 1114–1117.

Ali, M., Kishtawal, C., Jain, S., 2007. Predicting cyclone tracks in the north Indian ocean: An artificial neural network approach. Geophys. Res. Lett. 34 (4).

Andrieu, C., De Freitas, N., Doucet, A., Jordan, M.I., 2003. An introduction to MCMC for machine learning. Mach. Learn. 50 (1), 5–43.

Anon, 2015. Joint Typhoon warning center (JWTC) tropical cyclone best track data site. https://www.metoc.navy.mil/jtwc/jtwc.html?best-tracks.

Atkinson, G.D., Holliday, C.R., 1977. Tropical cyclone minimum sea level pressure/maximum sustained wind relationship for the western north Pacific. Mon. Weather Rev. 105 (4), 421–427.

Back, T., Schwefel, H.-P., 1996. Evolutionary computation: An overview. In: Proceedings of IEEE International Conference on Evolutionary Computation. IEEE, pp. 20–29.

Barber, D., Bishop, C.M., 1998. Ensemble learning in Bayesian neural networks. Nato ASI Ser. F Comput. Syst. Sci. 168, 215–238.

Blei, D.M., Kucukelbir, A., McAuliffe, J.D., 2017. Variational inference: A review for statisticians. J. Amer. Statist. Assoc. 112 (518), 859–877.

Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D., 2015. Weight uncertainty in neural network. In: Proceedings of the 32nd International Conference on Machine Learning. pp. 1613–1622.

Carr III, L.E., Elsberry, R.L., Peak, J.E., 2001. Beta test of the systematic approach expert system prototype as a tropical cyclone track forecasting aid. Weather Forecast. 16 (3), 355–368.

Chandra, R., 2015. Multi-objective cooperative neuro-evolution of recurrent neural networks for time series prediction. In: 2015 IEEE Congress on Evolutionary Computation. CEC, IEEE, pp. 101–108.

Chandra, R., Azam, D., Müller, R.D., Salles, T., Cripps, S., 2019a. BayesLands: A Bayesian inference approach for parameter uncertainty quantification in Badlands. Comput. Geosci. 131, 89–101.

Chandra, R., Bhagat, A., Maharana, M., Krivitsky, P.N., 2021a. Bayesian graph convolutional neural networks via tempered MCMC. IEEE Access 9, 130353–130365.

Chandra, R., Dayal, K., 2015. Cooperative neuro-evolution of Elman recurrent networks for tropical cyclone wind-intensity prediction in the south Pacific region. In: Evolutionary Computation (CEC), 2015 IEEE Congress on. IEEE, pp. 1784–1791.

Chandra, R., Dayal, K., Rollings, N., 2015. Application of cooperative neuro-evolution of elman recurrent networks for a two-dimensional cyclone track prediction for the south Pacific region. In: Neural Networks (IJCNN), 2015 International Joint Conference on. IEEE, pp. 1–8.

Chandra, R., Jain, K., Deo, R.V., Cripps, S., 2019b. Langevin-gradient parallel tempering for Bayesian neural learning. Neurocomputing http://dx.doi.org/10.1016/j.neucom.2019.05.082.

[3] https://github.com/DARE-ML/variational-rnn-cyclones

Chandra, R., Jain, M., Maharana, M., Krivitsky, P.N., 2021b. Revisiting Bayesian autoencoders with MCMC. arXiv preprint arXiv:2104.05915.

Chandra, R., Kapoor, A., 2020. Bayesian neural multi-source transfer learning. Neurocomputing 378, 54–64.

Chandra, R., Müller, R.D., Deo, R., Buttersworth, N., Salles, T., Cripps, S., 2018a. Multi-core parallel tempering Bayeslands for basin and landscape evolution. ArXiv Preprint ArXiv:1806.10939.

Chandra, R., Ong, Y.-S., Goh, C.-K., 2018b. Co-evolutionary multi-task learning for dynamic time series prediction. Appl. Soft Comput. 70, 576–589. http://dx.doi.org/10.1016/j.asoc.2018.05.041.

Chandra, R., Zhang, M., 2012. Cooperative coevolution of elman recurrent neural networks for chaotic time series prediction. Neurocomputing 86, 116–123.

Chaudhuri, S., Dutta, D., Goswami, S., Middey, A., 2015. Track and intensity forecast of tropical cyclones over the north Indian ocean with multilayer feed forward neural nets. Meteorol. Appl. 22 (3), 563–575.

Chen, R., Zhang, W., Wang, X., 2020. Machine learning in tropical cyclone forecast modeling: A review. Atmosphere 11 (7), 676.

Črepinšek, M., Liu, S.-H., Mernik, M., 2013. Exploration and exploitation in evolutionary algorithms: A survey. ACM Comput. Surv. 45 (3), 1–33.

DeMaria, M., 1987. Tropical cyclone track prediction with a barotropic spectral model. Mon. Weather Rev. 115 (10), 2346–2357.

Deo, R., Chandra, R., 2016. Identification of minimal timespan problem for recurrent neural networks with application to cyclone wind-intensity prediction. In: 2016 International Joint Conference on Neural Networks. IJCNN, IEEE, pp. 489–496.

Deo, R., Chandra, R., 2019. Multi-step-ahead cyclone intensity prediction with Bayesian neural networks. In: Pacific Rim International Conference on Artificial Intelligence. Springer, pp. 282–295.

Deo, R.V., Chandra, R., Sharma, A., 2017. Stacked transfer learning for tropical cyclone intensity prediction. arXiv preprint arXiv:1708.06539.

Drugan, M.M., Thierens, D., 2003. Evolutionary markov chain monte carlo. In: International Conference on Artificial Evolution (Evolution Artificielle). Springer, pp. 63–76.

Du, W., Leung, S.Y.S., Kwong, C.K., 2014. Time series forecasting by neural networks: A knee point-based multiobjective evolutionary algorithm approach. Expert Syst. Appl. 41 (18), 8049–8061.

Elman, J.L., 1990. Finding structure in time. Cogn. Sci. 14 (2), 179–211.

Emanuel, K., 2003. Tropical cyclones. Ann. Rev. Earth Planet. Sci. 31 (1), 75–104.

Fengjin, X., Ziniu, X., 2010. Characteristics of tropical cyclones in China and their impacts analysis. Nat. Hazards 54 (3), 827–837.

Fiorino, M., Elsberry, R.L., 1989. Some aspects of vortex structure related to tropical cyclone motion. J. Atmos. Sci. 46 (7), 975–990.

Floreano, D., Dürr, P., Mattiussi, C., 2008. Neuroevolution: From architectures to learning. Evolut. Intell. 1 (1), 47–62.

Fraedrich, K., Leslie, L., 1989. Estimates of cyclone track predictability. I: Tropical cyclones in the Australian region. Q. J. R. Meteorol. Soc. 115 (485), 79–92.

Galván, E., Mooney, P., 2021. Neuroevolution in deep neural networks: Current trends and future challenges. IEEE Trans. Artif. Intell. 2 (6), 476–493.

Gao, S., Zhao, P., Pan, B., Li, Y., Zhou, M., Xu, J., Zhong, S., Shi, Z., 2018. A nowcasting model for the prediction of Typhoon tracks based on a long short term memory neural network. Acta Oceanol. Sinica 37, 8–12.

Girolami, M., Calderhead, B., 2011. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. J. R. Stat. Soc. Ser. B Stat. Methodol. 73 (2), 123–214.

Gneiting, T., Raftery, A.E., 2007. Strictly proper scoring rules, prediction, and estimation. J. Amer. Statist. Assoc. 102 (477), 359–378.

Gneiting, T., Stanberry, L.I., Grimit, E.P., Held, L., Johnson, N.A., 2008. Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds. Test 17 (2), 211–235.

Goerss, J.S., 2000. Tropical cyclone track forecasts using an ensemble of dynamical models. Mon. Weather Rev. 128 (4), 1187–1193.

Graves, A., 2011. Practical variational inference for neural networks. Adv. Neural Inf. Process. Syst. 24.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al., 2018. Recent advances in convolutional neural networks. Pattern Recognit. 77, 354–377.

Hall, T.M., Jewson, S., 2007. Statistical modeling of north Atlantic tropical cyclone tracks. Tellus A 59 (4), 486–498.

Harmelin-Vivien, M.L., 1994. The effects of storms and cyclones on coral reefs: A review. J. Coast. Res. 211–231.

Hastings, W.K., 1970. Monte Carlo sampling methods using Markov chains and their applications. Oxford University Press.

Hinton, G.E., Van Camp, D., 1993. Keeping the neural networks simple by minimizing the description length of the weights. In: Proceedings of the Sixth Annual Conference on Computational Learning Theory. ACM, pp. 5–13.

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

Hong, S., Kim, S., Joh, M., Song, S.-k., 2017. Globenet: Convolutional neural networks for typhoon eye tracking from remote sensing imagery. arXiv preprint arXiv:1708.03417.

Hruschka, E.R., Campello, R.J., Freitas, A.A., et al., 2009. A survey of evolutionary algorithms for clustering. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) 39 (2), 133–155.

Hukushima, K., Nemoto, K., 1996. Exchange Monte Carlo method and application to spin glass simulations. J. Phys. Soc. Japan 65 (6), 1604–1608.

Igarashi, Y., Tajima, Y., 2021. Application of recurrent neural network for prediction of the time-varying storm surge. Coast. Eng. J. 63 (1), 68–82.

Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K., 1999. An introduction to variational methods for graphical models. Mach. Learn. 37 (2), 183–233.

Kingma, D.P., Welling, M., 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Kordmahalleh, M.M., Sefidmazgi, M.G., Homaifar, A., Liess, S., 2015. Hurricane trajectory prediction via a sparse recurrent neural network. In: Proceedings of the 5th International Workshop on Climate Informatics.

Kovordányi, R., Roy, C., 2009. Cyclone track forecasting based on satellite images using artificial neural networks. ISPRS J. Photogramm. Remote Sens. 64 (6), 513–521.

Krogh, A., Hertz, J.A., 1992. A simple weight decay can improve generalization. In: Advances in Neural Information Processing Systems. pp. 950–957.

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

LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E., Jackel, L.D., 1990. Handwritten digit recognition with a back-propagation network. In: Advances in Neural Information Processing Systems. pp. 396–404.

Lee, R.S.T., Liu, J.N.K., 2000. In: Rodrigues, M.A. (Ed.), An approach using Elastic Graph Dynamic Link model for automating the satellite interpretation of tropical cyclone patterns, in Invariants for Pattern Recognition and Classification. World Scientific, pp. 189–207.

Lionello, P., Bhend, J., Buzzi, A., Della-Marta, P., Krichak, S., Jansa, A., Maheras, P., Sanna, A., Trigo, I., Trigo, R., 2006. Cyclones in the mediterranean region: Climatology and effects on the environment. In: Developments in Earth and Environmental Sciences, Vol. 4. Elsevier, pp. 325–372.

MacKay, D.J., 1995. Probable networks and plausible predictions—a review of practical Bayesian methods for supervised neural networks. Network: Comput. Neural Syst. 6 (3), 469–505.

MacKay, D.J., 1996. Hyperparameters: Optimize, or integrate out? In: Maximum Entropy and Bayesian Methods. Springer, pp. 43–59.

Matheson, J.E., Winkler, R.L., 1976. Scoring rules for continuous probability distributions. Manage. Sci. 22 (10), 1087–1096.

Mathur, M.B., 1991. The national meteorological center's Quasi-Lagrangian model for Hurricane prediction. Mon. Weather Rev. 119 (6), 1419–1447.

McAdie, C.J., Lawrence, M.B., 2000. Improvements in tropical cyclone track forecasting in the Atlantic basin, 1970–98. Bull. Am. Meteorol. Soc. 81 (5), 989–997.

McBride, J.L., Holland, G.J., 1987. Tropical-cyclone forecasting: A worldwide summary of techniques and verification statistics. Bull. Am. Meteorol. Soc. 68 (10), 1230–1238.

Mendelsohn, R., Emanuel, K., Chonabayashi, S., Bakkensen, L., 2012. The impact of climate change on global tropical cyclone damage. Nature Clim. Change 2 (3), 205–209.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. J. Chem. Phys. 21 (6), 1087–1092. http://dx.doi.org/10.1063/1.1699114.

Mohanty, U., 1994. Tropical cyclones in the bay of Bengal and deterministic methods for prediction of their trajectories. Sadhana 19 (4), 567–582.

Moradi Kordmahalleh, M., Gorji Sefidmazgi, M., Homaifar, A., 2016. A sparse recurrent neural network for trajectory prediction of Atlantic Hurricanes. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016. pp. 957–964.

Munk, D.J., Vio, G.A., Steven, G.P., 2015. Topology and shape optimization methods using evolutionary algorithms: A review. Struct. Multidiscip. Optim. 52, 613–631.

Neal, R.M., et al., 2011. MCMC using Hamiltonian dynamics. Handb. Markov Chain Monte Carlo 2 (11).

Needham, H.F., Keim, B.D., Sathiaraj, D., 2015. A review of tropical cyclone-generated storm surges: Global data sources, observations, and impacts. Rev. Geophys. 53 (2), 545–591.

Neumann, C.J., 1972. An Alternate to the HURRAN (Hurricane Analog) Tropical Cyclone Forecast System. National Weather Service.

Pall, J., Chandra, R., Azam, D., Salles, T., Webster, J.M., Scalzo, R., Cripps, S., 2020. Bayesreef: A Bayesian inference framework for modelling reef growth in response to environmental change and biological dynamics. Environ. Model. Softw. 125, 104610.

Pan, B., Xu, X., Shi, Z., 2019. Tropical cyclone intensity prediction based on recurrent neural networks. Electron. Lett. 55 (7), 413–415.

Pielke, Jr., R.A., 2007. Future economic damage from tropical cyclones: Sensitivities to societal and climate changes. Phil. Trans. R. Soc. A 365 (1860), 2717–2729.

Rezende, D.J., Mohamed, S., Wierstra, D., 2014. Stochastic backpropagation and approximate inference in deep generative models. In: International Conference on Machine Learning. PMLR, pp. 1278–1286.

Richard, M.D., Lippmann, R.P., 1991. Neural network classifiers estimate Bayesian a posteriori probabilities. Neural Comput. 3 (4), 461–483.

Roberts, G.O., Rosenthal, J.S., 1998. Optimal scaling of discrete approximations to Langevin diffusions. J. R. Stat. Soc. Ser. B Stat. Methodol. 60 (1), 255–268.

Roy, C., Kovordányi, R., 2012a. Tropical cyclone track forecasting techniques– a review. Atmos. Res. 104, 40–69.

Roy, C., Kovordányi, R., 2012b. Tropical cyclone track forecasting techniques– a review. Atmos. Res. 104–105, 40–69. http://dx.doi.org/10.1016/j.atmosres.2011.09.012, URL http://www.sciencedirect.com/science/article/pii/S0169809511002973.

Schrum, P., Scheller, R.M., Duveneck, M.J., Lucash, M.S., 2020. Base-Hurricane: A new extension for the Landis-II forest landscape model. Environ. Model. Softw. 133, 104833. http://dx.doi.org/10.1016/j.envsoft.2020.104833, URL https://www.sciencedirect.com/science/article/pii/S1364815220308902.

Skilling, J., et al., 2006. Nested sampling for general Bayesian computation. Bayesian Anal. 1 (4), 833–859.

Specht, D.F., 1990. Probabilistic neural networks. Neural Netw. 3 (1), 109–118.

Stanley, K.O., Clune, J., Lehman, J., Miikkulainen, R., 2019. Designing neural networks through neuroevolution. Nat. Mach. Intell. 1 (1), 24–35.

Strens, M., 2003. Evolutionary MCMC sampling and optimization in discrete spaces. In: Proceedings of the 20th International Conference on Machine Learning. ICML-03, pp. 736–743.

Such, F.P., Madhavan, V., Conti, E., Lehman, J., Stanley, K.O., Clune, J., 2017. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. arXiv preprint arXiv:1712.06567.

Swendsen, R.H., Wang, J.-S., 1986. Replica Monte Carlo simulation of spin-glasses. Phys. Rev. Lett. 57 (21), 2607.

Takens, F., 1981. Detecting strange attractors in turbulence. In: Dynamical Systems and Turbulence, Warwick 1980. Springer, pp. 366–381.

Ter Braak, C.J., 2006. A Markov chain Monte Carlo version of the genetic algorithm differential evolution: Easy Bayesian computing for real parameter spaces. Stat. Comput. 16 (3), 239–249.

Ter Braak, C.J., Vrugt, J.A., 2008. Differential evolution Markov chain with snooker updater and fewer chains. Stat. Comput. 18 (4), 435–446.

Vickery, P., Skerlj, P., Twisdale, L., 2000. Simulation of hurricane risk in the US using empirical track model. J. Struct. Eng. 126 (10), 1222–1237.

Wainwright, M.J., Jordan, M.I., 2008. Graphical Models, Exponential Families, and Variational Inference. Now Publishers Inc.

Wan, E.A., 1990. Neural network classification: A Bayesian interpretation. IEEE Trans. Neural Netw. 1 (4), 303–305.

Weber, H.C., 2003. Hurricane track prediction using a statistical ensemble of numerical models. Mon. Weather Rev. 131 (5), 749–770.

Welling, M., Teh, Y.W., 2011. Bayesian learning via stochastic gradient Langevin dynamics. In: Proceedings of the 28th International Conference on Machine Learning. ICML-11, pp. 681–688.

Yablonsky, R.M., Ginis, I., Thomas, B., 2015. Ocean modeling with flexible initialization for improved coupled tropical cyclone-ocean model prediction. Environ. Model. Softw. 67, 26–30. http://dx.doi.org/10.1016/j.envsoft.2015.01.003, URL https://www.sciencedirect.com/science/article/pii/S1364815215000213.

Yan, J., Mu, L., Wang, L., Ranjan, R., Zomaya, A.Y., 2020. Temporal convolutional networks for the advance prediction of ENSO. Sci. Rep. 10 (1), 1–15.

Zhang, Y., Chandra, R., Gao, J., 2018. Cyclone track prediction with matrix neural networks. In: 2018 International Joint Conference on Neural Networks. IJCNN, IEEE, pp. 1–8.

Zhang, W., Leung, Y., Chan, J.C., 2013. The analysis of tropical cyclone tracks in the western north Pacific through data mining. Part I: Tropical cyclone recurvature. J. Appl. Meteorol. Climatol. 52 (6), 1394–1416.

Zhang, F., Tao, D., 2013. Effects of vertical wind shear on the predictability of tropical cyclones. J. Atmos. Sci. 70 (3), 975–983.

Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P.N., Zhang, Q., 2011. Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm Evol. Comput. 1 (1), 32–49.

Zhu, L., Jin, J., Cannon, A.J., Hsieh, W.W., 2016. Bayesian neural networks based bootstrap aggregating for tropical cyclone tracks prediction in south China sea. In: International Conference on Neural Information Processing. Springer, pp. 475–482.