Prediction of the Survival Status for Multispecies Competition System

Youwen Wang,[a] Maria Vasilyeva,[b] and Alexey Sadovski[c]

Department of Mathematics and Statistics, Texas A&M University - Corpus Christi, Corpus Christi, Texas, USA

[a]*Electronic mail: ywang23@islander.tamucc.edu*
[b]*Corresponding author: maria.vasilyeva@tamucc.edu*
[c]*Electronic mail: alexey.sadovski@tamucc.edu*

Abstract. We consider a multispecies competition model in a one- and two-dimensional formulation. To solve the problem numerically, we construct a discrete system using finite volume approximation by space with semi-implicit time approximation. The solution of the multispecies competition model converges to the final equilibrium state that does not depend on the initial condition of the system. The final equilibrium state characterizes the survival status of the multispecies system (one or more species survive or no one survives). In real-world problems values of the parameters are unknown and vary in some range. For such problems, the series of Monte Carlo simulations can be used to estimate the system, where a large number of simulations are needed to be performed with random values of the parameters. A numerical solution is expensive, especially for high-dimensional problems, and requires a large amount of time to perform. In this work, to reduce the cost of simulations, we use a deep neural network to fast predict the survival status. Numerical results are presented for different neural network configurations. The comparison with convenient classifiers is presented.

INTRODUCTION

An ecosystem is a complex interconnected network with a highly non-linear interaction. Modeling approaches used in understanding the ecosystems are based on the availability of data and knowledge [1, 2]. We construct and solve coupled systems of ordinary and/or partial differential equations that describe an underlying physical process in a mathematical model-based approach. Statistical and machine learning approaches are based on data availability. In real life, many factors affect species' population dynamic, and the factor analysis method has long been applied to the analysis of population dynamics as well as ecosystem topic [3, 4]. Many efforts have been put into applying deep learning to unstructured data in the ecosystem, such as identifying species, classifying animal behavior, or estimating biodiversity using data from camera traps, sound recorders, and video recorders [5]. However, deep learning also performs well when applied to structured datasets. After the species' population is estimated via images/ sound/ videos, deep learning can be applied to the population dynamics of the ecosystem, and this is an objective of this paper.

Machine learning is suitable for prediction of non-linear systems [6, 7, 8]. The neural network has been applied to classify structured and unstructured data such as images, signals, and text. Deep Neural Network (DNN) has been widely used in field of health care, bioinformatics, remote sensing, etc. [9, 10] To decipher structured data, Feedforward Neural Network (FNN), Recurrent Neural Network (RNN), and Gated Recurrent Unit based (GRU-based) models have been tested to compare with the MultiLayer Perceptron (MLP), Support Vector Machine (SVM), and k-Nearest Neighbors baselines [11]. Different DNN architectures can be examined by varying the number of hidden layers, the number of neurons in each hidden layer, and the activation function [12]. Noise can be simulated and added to the data set to increase the robustness of the DNN model [13]. In order to assess the performance of DNN, various metrics, such as classification accuracy, precision, recall, f-score, AUC, and log-loss error, are used [14].

Different mathematical models are used to approximate an ecosystem with competing species. Such as Malthusian growth model [15], Lotka–Volterra model [16], as well as Arditi–Ginzburg model [17]. In this paper, we approximate an ecosystem of two competing species with a spatial-temporal multispecies Lotka–Volterra competition model in one-and-two-dimensional formulations. To solve the problem numerically, we construct a discrete system using finite volume approximation by space with semi-implicit time approximation. The solution of the multispecies competition model converges to the final equilibrium state that does not depend on the system's initial condition [18]. The final equilibrium state characterizes the survival status of the system. A numerical solution is expensive, especially for high-dimensional problems, and requires significant time to perform. To reduce the cost of simulations, we present a prediction algorithm based on the different classifiers and machine learning techniques. We generate two datasets related to the problem dimension to train classifiers and neural networks with different architectures. The input data

contains a growth rate, competition term, and diffusion rate of both competing species randomly generated in 10,000 simulations. The output data is the survival status of both species according to their equilibrium final population density.

The paper is organized as follows. In the section "Preliminaries", we present the mathematical model with approximation. Section "Datasets" introduced input and output data. Section "Prediction Algorithms" is to introduce details of classifier algorithms and neural network construction. In the section "Numerical Results", we present the predictive result for different types of classifiers and deep neural network architectures. The paper ends with a conclusion.

PRELIMINARIES

Let $\Omega \subset R^d$ be the computational domain, where d is the dimension ($d = 1$ or 2). Here, we consider domain $\Omega = [0, L]^d$ for simplicity. The following coupled system of equations describes the mathematical model [19, 20]:

$$\begin{aligned} \frac{\partial u^{(1)}}{\partial t} - \nabla \cdot (\varepsilon_1 \nabla u^{(1)}) &= r_1 u^{(1)}(1 - u^{(1)}) - \alpha_{12} u^{(1)} u^{(2)}, \quad x \in \Omega, \quad 0 < t < T \\ \frac{\partial u^{(2)}}{\partial t} - \nabla \cdot (\varepsilon_2 \nabla u^{(2)}) &= r_2 u^{(2)}(1 - u^{(2)}) - \alpha_{21} u^{(1)} u^{(2)}, \quad x \in \Omega, \quad 0 < t < T, \end{aligned} \tag{1}$$

where $u^{(k)}$ is the population of the k-th species, ε_k is the diffusion coefficient, r_k is the k-th population reproductive growth rate, α_{kl} is the interaction coefficient due to competition and $k = 1, 2$.

The system of equation is considered with given initial conditions for both species

$$u^{(1)} = u_{01}, \quad u^{(2)} = u_{02}, \quad x \in \Omega, \quad t = 0, \tag{2}$$

and fixed boundary conditions on $\partial\Omega$ (zero Dirichlet boundary conditions)

$$u^{(1)} = 0, \quad u^{(2)} = 0, \quad x \in \partial\Omega, \quad 0 < t < T, \tag{3}$$

where u_{01} and u_{02} are some given constants.

We solve a system of equations (1) with initial conditions (2) and boundary conditions (3) using a finite volume method with a semi-implicit time approximation. For computational domain Ω, we construct a uniform structured grid with mesh size $h = L/(N-1)$, where N is the number of nodes in each direction. Let $u_i^{(k)}$ be the value of the function $u^{(k)}$ on cell K_i. For the finite volume approximation by space using two-point flux approximation, we obtain the following discrete form

$$\begin{aligned} \frac{\partial u_i^{(1)}}{\partial t}|K_i| + \sum_j T_{1,ij}(u_i^{(1)} - u_j^{(1)}) &= r_1 u_i^{(1)}(1 - u_i^{(1)})|K_i| - \alpha_{12} u_i^{(1)} u_i^{(2)}|K_i|, \\ \frac{\partial u_i^{(2)}}{\partial t}|K_i| + \sum_j T_{2,ij}(u_i^{(2)} - u_j^{(2)}) &= r_2 u_i^{(2)}(1 - u_i^{(2)})|K_i| - \alpha_{21} u_i^{(2)} u_i^{(1)}|K_i|, \end{aligned} \tag{4}$$

for $\forall i = \overline{1, N_c}$, where N_c is the number of cells. Here $T_{k,ij} = \varepsilon_k \, |e_{ij}|/d_{ij}$ ($k = 1, 2$), where d_{ij} is the distance between to cell center points x_i and x_j, $|e_{ij}|$ is the length of the interface between to cells K_i and K_j.

For approximation by time, we use a semi-implicit scheme and evaluate a reaction term using the previous time layer solution [21, 22, 23]

$$\begin{aligned} \frac{u_i^{(1)} - \check{u}_i^{(1)}}{\tau}|K_i| + \sum_j T_{1,ij}(u_i^{(1)} - u_j^{(1)}) &= r_1 \check{u}_i^{(1)}(1 - \check{u}_i^{(1)})|K_i| - \alpha_{12} \check{u}_i^{(1)} \check{u}_i^{(2)}|K_i|, \\ \frac{u_i^{(2)} - \check{u}_i^{(2)}}{\tau}|K_i| + \sum_j T_{2,ij}(u_i^{(2)} - u_j^{(2)}) &= r_2 \check{u}_i^{(2)}(1 - \check{u}_i^{(2)})|K_i| - \alpha_{21} \check{u}_i^{(2)} \check{u}_i^{(1)}|K_i|, \end{aligned} \tag{5}$$

where τ is the time step size and $\check{u}_i^{(1)}$ and $\check{u}_i^{(2)}$ are the solution from the previous time layer.

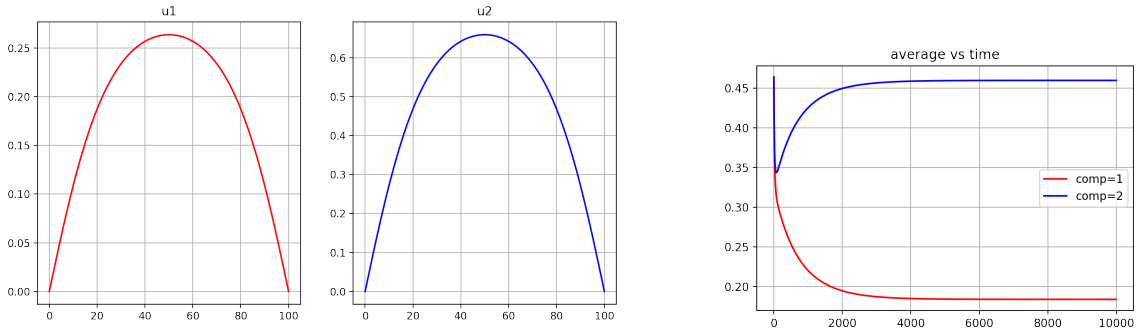


FIGURE 1: Solution at final time for 1D case. Left: $u^{(1)}$. Middle: $u^{(2)}$. Right: dynamic of the average population, $\bar{u}^{(1)}$ (red color) and $\bar{u}^{(2)}$ (blue color)

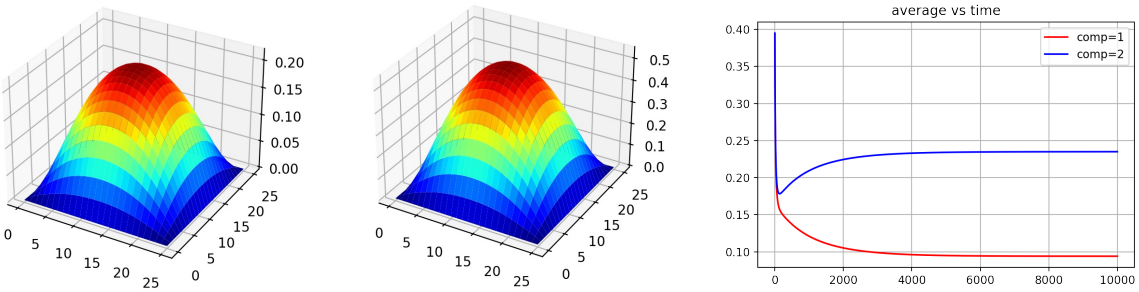


FIGURE 2: Solution at final time for 2D case. Left: $u^{(1)}$. Middle: $u^{(2)}$. Right: dynamic of the average population, $\bar{u}^{(1)}$ (red color) and $\bar{u}^{(2)}$ (blue color)

To illustrate the presented model with a numerical algorithm, we consider a two-species competition model with the following parameters

$$\varepsilon_1 = \varepsilon_2 = 0.01, \quad r_1 = r_2 = 0.05, \quad \alpha_{12} = 0.048, \quad \alpha_{21} = 0.045.$$

We consider model in one-dimensional domain $\Omega = [0,\pi]$ and two-dimensional domain $\Omega = [0,\pi]^2$. We set zero population for both species as boundary conditions on $\partial\Omega$. In simulations, we use the grid with 100 cells for a one-dimensional problem and 25×25 grid for a two-dimensional case. We simulate with $\tau = 1$ and set initial conditions $u_{01} = u_{02} = 0.5$. We perform 10 000 time iterations to represent a dynamic of the solution.

To represent the result and compare the final equilibrium state, we calculate the average solution over the computational domain Ω for each species

$$\bar{u}^{(1)}(t) = \frac{1}{|\Omega|}\int_\Omega u^{(1)}(x,t)\,dx, \quad \bar{u}^{(2)}(t) = \frac{1}{|\Omega|}\int_\Omega u^{(2)}(x,t)\,dx,$$

where $|\Omega|$ is the volume of the domain Ω.

In Figures 1 and 2, we plot the solution for one and two-dimensional spatial boundary conditions at the final time, as well as the dynamic to the average solution over the domain for this two-species system. We observed that after around 3000 time iterations, both one-dimensional and two-dimensional cases would converge to a steady state. We also observed the same boundary effect for both 1D and 2D cases. The final population density is highest in the middle and closer to zero when it gets closer to the domain's boundary. However, 1D and 2D cases differ in the values of both species' final equilibrium population density. In the 1D case, the "more successful species" (i.e., the species with higher final equilibrium population density) converges to 0.46 of population density in the 1D case. In contrast, the "more successful species" in the 2D case only converge to 0.23 of population density. In both 1D and 2D cases, "the less successful species" converge to a final population density of around 0.1.

DATASETS

Numerical simulations using the algorithm presented above can be computationally expensive, especially for multidimensional cases. To reduce the cost, we present a prediction algorithm based on the different classifiers and machine learning techniques. We generate two datasets related to the problem dimension to train classifiers and neural networks with different architectures. The process of dataset generation is illustrated in Figure3.

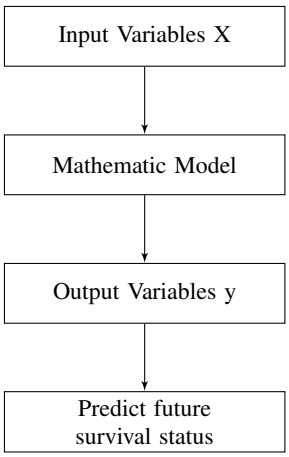


FIGURE 3: The Data Simulation Diagram

We consider one and two-dimensional test problems and simulate the system with random input parameters X_l

$$X_l = (r_1, r_2, \alpha_{12}, \alpha_{21}, \varepsilon_1, \varepsilon_2), \tag{6}$$

where $l = 1, 2, .., L$ and L is the number of simulations. Here r_k is the k-th population growth rate, α_{kl} is the interaction coefficient due to competition ($u^{(l)}$ compete with $u^{(k)}$), and ε_k is the population diffusion rate of the k-th species ($k = 1, 2$).

We perform $L = 10,000$ simulations with random values in the following interval

$$0.01 < r_1, r_2, \alpha_{12}, \alpha_{21}, \varepsilon_1, \varepsilon_2, < 0.1,$$

and

$$\delta < u_{01}, u_{02} < 1.0 - \delta$$

with $\delta = 0.01$.

To generate dataset, we perform numerical simulations for each input parameter X_l to calculate final average population, $\bar{u}^{(1)}$ and $\bar{u}^{(2)}$. Simulations are perform till both population reach equilibrium, $|\bar{u}^{(k)} - \check{\bar{u}}^{(k)}| < tol$ with $tol = 10^{-5}$ for each $k = 1, 2$. Next, we define groups based on the final average population. Groups are represented by which species survive. In the two-species competition model, we have:

- 0: 00 - no one survived and
- 1: 10 - first species survived,
- 2: 01 - second species survived,
- 3: 11 - both species survived.

We note that we use a $\delta = 0.01(1\%)$ as a threshold in the groups definition or

$$g_i = \begin{cases} 00, & \bar{u}^{(1)} < \delta, \text{ and } \bar{u}^{(2)} < \delta, \\ 01, & \bar{u}^{(1)} < \delta, \text{ and } \bar{u}^{(2)} \geq \delta, \\ 10, & \bar{u}^{(1)} \geq \delta, \text{ and } \bar{u}^{(2)} < \delta, \\ 11, & \bar{u}^{(1)} \geq \delta, \text{ and } \bar{u}^{(2)} \geq \delta. \end{cases}$$

Therefore we have the following output data Y_l

$$Y_l = \{g_i\}, \tag{7}$$

where g_i is the group.

Finally, we obtain the following dataset

$$\text{Dataset: } \{(X_l, Y_l), l = 1, ..., L\}.$$

where L is the size of dataset, X_l and Y_l are the input data and output data.

	00	01	10	11
1d	2796	3059	3064	1081
	27.96%	30.59%	30.64%	10.81%
2d	6888	1487	1489	136
	68.88%	14.87%	14.89%	1.36%

TABLE I: Survival Status for one and two-dimensional problems

In Table I, we present results for $L = 10,000$ simulations for both 1D and 2D formulations. For one dimensional case (*1D*), we observe that around 30 percent of simulations result in either 00(no species survives), 01 (species two survives), or 10 (species one survives). In comparison, only around 10 percent of simulations result in both species surviving together. For the two-dimensional case (*2D*), we observe that 68.88 percent of simulations result in 00 (both species go extinct), while around 14.87 percent and 14.89 percent result in 01 (species two survives) or 10 (species one survives), and very rare, about 1.36 percent of the simulation result in 11 (both species survive).

In later work, we intend to apply the classification methods to predict the survival status of species under equilibrium before a catastrophic event. After randomly simulating large amounts of cases with random parameters as input and equilibrium population density, we will only take simulations that lead to the cases in which at least one species survive. After that, we will apply catastrophic events with random lengths over a geographic domain. To predict and determine a group where at least one species survive, we define two groups:

- 0 - no one survived and
- 1 - at least one species survived.

Here 0-group is equivalent to the 00-group in four group representation, and 1-group contains 01, 10, and 11-group. Two and four groups' predictions will be considered in the next section.

PREDICTION ALGORITHMS

We consider different classification algorithms and compare them with machine learning techniques to predict survival groups. The data flow process is illustrated in Figure 4.

First, we construct the classifier

$$Y_l = \mathscr{C}(X_l),$$

where $\mathscr{C}$ is the classifier, X is the input, and Y is the corresponding output.

We consider several convenient classification techniques (types):

- linear classifiers (Logistic Regression, Ridge Classifier, SGD Classifier),
- nearest neighbor (k-nearest neighbor classifier),
- decision trees (decision tree classifier and extremely randomized tree classifier),
- naive Bayes (Gaussian naive Bayes classifier),
- support vector machines (linear support and c-support vector classifiers),

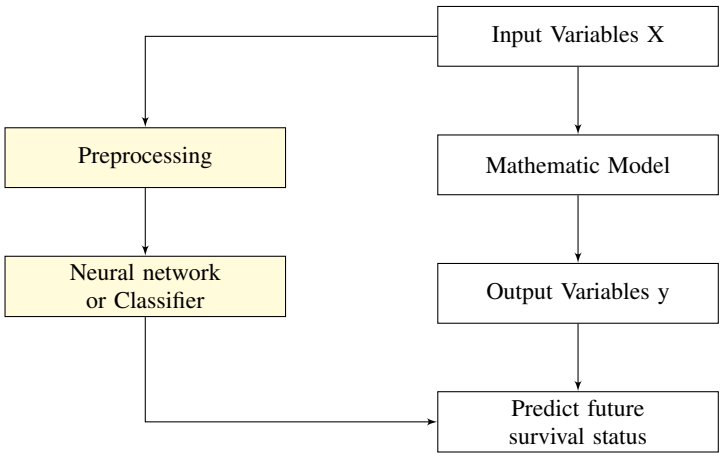
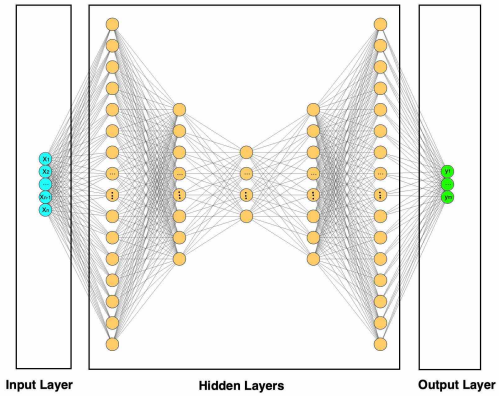
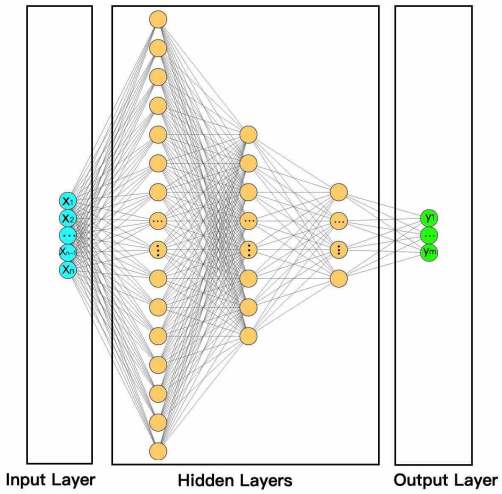


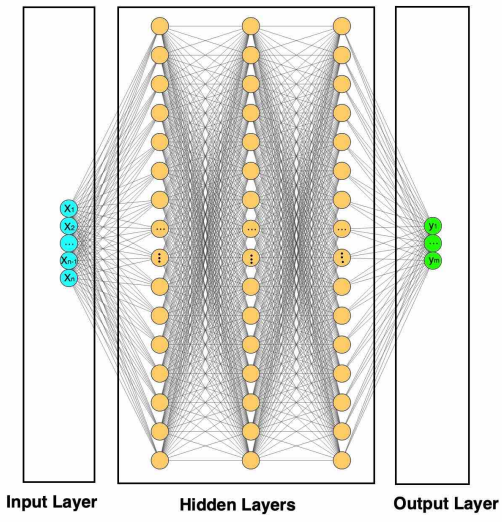
FIGURE 4: The Dataflow Diagram



(a) bottleneck



(b) triangular



(c) flat

FIGURE 5: Neural network types

- ensemble methods (random forest, AdaBoost, bagging, gradient boosting, and histogram-based gradient boosting regression tree classifiers),

For classification, we use a sklearn library [24]

Next, we consider deep neural network

$$Y_l = \mathcal{N}(X_l),$$

where $\mathcal{N}$ is the multilayer neural network (see Figure 5 for illustration). The multilayer neural network is a network of l layers, with X is the input and Y is the corresponding output

$$\mathcal{N}(X) = \sigma(W_l\sigma(...\sigma(W_2\sigma(W_1X + b_1) + b_2)...) + b_l)$$

where W_i are the weight matrices, b_i's are the bias vectors, and σ is the activation function.

The deep neural network we apply here can vary in many ways, such as the number of hidden layers, neurons in each hidden layer, activation function on each layer, training method, etc. There are countless combinations of these architectures of the neural network, so multiple efforts have been put into finding the neural network that works the best [25, 26, 27].

We investigate three ways of hidden layers design in neural networks architecture (types):

- bottleneck with 5 layers (64-32-16-32-64, 32-16-8-16-32), 3 layers (64-32-64, 32-16-32, 16-8-16)
- triangular with 3 layers (64-32-16, 32-16-8),
- flat with 3 layers (64-64-64, 32-32-32, 16-16-16),
- single (64, 32, 16),

with different numbers of neurons and layers (see Figure 5 for illustration). We use a fixed training method (Adam). The activation function in the hidden layers is ReLU (Rectified Linear Unit), and the softmax activation function is for the output layer. Implementation is performed using keras library [28, 29].

NUMERICAL RESULTS

In this work, we compared different combinations of deep neural network architectures. We use 1D and 2D datasets for the construction of the classification algorithm or training of the neural network. Dataset is divided into the train, and test sets, where we take 25 % of data for testing and 75 % for training. We first consider Classifier and Neural Network Prediction Model for 4 Survival Status (00, 01, 10, and 11) in 1D and 2D. We then presented Classifier and Neural Network Prediction Model for 2 Survival Status ("No species survive" and "At least one species survives") in 1D and 2D.

Four groups prediction

First, we present the performance of prediction models for 4 Survival Status Groups (00, 01, 10, and 11). We examined both 1D and 2D simulation datasets.

For the 1D dataset, Table II presents the performance of 4 survival groups using the Classifier prediction model, and Table III presents that of Neural Network (NN) prediction model. Overall, we observed that NN models generally performed better than Classifier models, indicating that using NN will help us improve our prediction accuracy. Within Classifier models, we observed that the Ensemble-Method-based and Support-Vector-Machine-based models give the best test accuracy. In contrast, the Linear-Models-based models give the worst test accuracy. Within NN models, the Bottleneck-Layered models give the best test accuracy, while the Single-Layered models gives the worst test accuracy. For NN models, besides the difference in test accuracy among different design types (bottleneck, flat, triangular, or single), we also observed that as the number of layers and number of neurons in each layers increase, the test accuracy also increases, this "pattern" is agreed by the fact that the best performing model is the model that has bottleneck layer design, the largest number of layers, and the largest number of neurons at the same time.

Model	Type	Accuracy (Train)	Accuracy (Test)
Hist Gradient Boosting Classifier	Ensemble Methods	100.00%	95.36%
RBF SVC	Support Vector Machines	97.01%	95.28%
Random Forest Classifier	Ensemble Methods	100.00%	94.08%
Gradient Boosting Classifier	Ensemble Methods	97.75%	93.80%
Bagging Classifier	Ensemble Methods	99.59%	92.32%
Gaussian NB	Naive Bayes	90.07%	89.16%
K Neighbors Classifier	Nearest Neighbors	93.43%	88.84%
Decision Tree Classifier	Decision Trees	100.00%	88.84%
Linear SVC	Support Vector Machines	88.21%	87.60%
AdaBoost Classifier	Ensemble Methods	89.61%	87.24%
Logistic Regression	Linear Models	86.56%	86.04%
Ridge Classifier	Linear Models	86.21%	85.96%
SGD Classifier	Linear Models	86.43%	85.76%
Extra Trees Classifier	Ensemble Methods	100.00%	78.64%

TABLE II: Classifier Prediction Models for 4 Survival Groups (00, 01, 10, and 11) for 1D

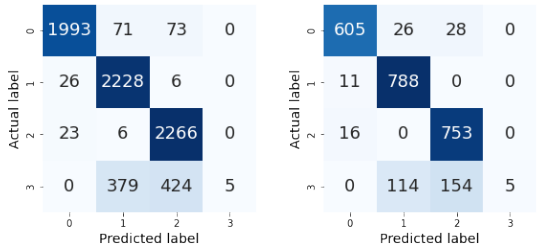
Model	Type	Accuracy (Train)	Accuracy (Test)
5 layers (64-32-16-32-64)	Bottleneck	97.20%	96.88%
3 layers (64-64-64)	Flat	96.60%	95.72%
3 layers (32-16-32)	Bottleneck	95.91%	95.32%
3 layers (64-32-64)	Bottleneck	96.31%	95.04%
5 layers (32-16-8-16-32)	Bottleneck	96.07%	94.92%
3 layers (32-32-32)	Flat	95.61%	94.80%
3 layers (64-32-16)	Triangular	95.64%	94.72%
3 layers (32-16-8)	Triangular	95.41%	94.28%
3 layers (16-16-16)	Flat	94.73%	93.92%
3 layers (16-8-16)	Bottleneck	94.51%	93.08%
1 layer (64)	Single	92.87%	92.24%
1 layer (32)	Single	90.23%	89.56%
1 layer (16)	Single	86.61%	86.48%

TABLE III: Neural Networks Prediction Models for 4 Survival Groups (00, 01, 10, and 11) for 1D

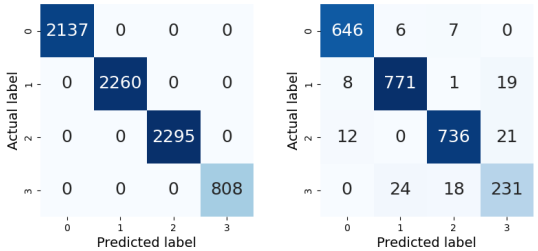
Model	Type	Accuracy (Train)	Accuracy (Test)
Hist Gradient Boosting Classifier	Ensemble Methods	100.00%	97.96%
Gradient Boosting Classifier	Ensemble Methods	99.85%	97.52%
RBF SVC	Support Vector Machines	98.24%	97.48%
Random Forest Classifier	Ensemble Methods	100.00%	97.32%
Bagging Classifier	Ensemble Methods	99.81%	97.20%
Decision Tree Classifier	Decision Trees	100.00%	96.40%
Gaussian NB	Naive Bayes	92.75%	92.96%
K Neighbors Classifier	Nearest Neighbors	95.08%	92.00%
AdaBoost Classifier	Ensemble Methods	89.89%	88.44%
Linear SVC	Support Vector Machines	88.07%	87.56%
Extra Trees Classifier	Ensemble Methods	100.00%	83.56%
Ridge Classifier	Linear Models	81.15%	79.48%
SGD Classifier	Linear Models	80.03%	79.28%
Logistic Regression	Linear Models	75.16%	74.08%

TABLE IV: Classifier Prediction Models for 4 Survival Groups (00, 01, 10, and 11) for 2D

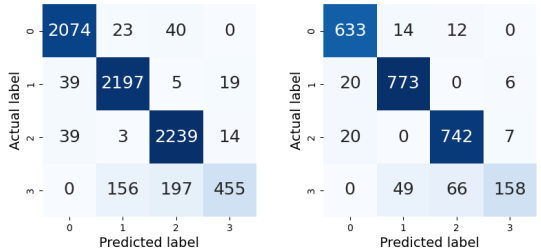
Figure 6 presents the best and worst performing models for 4 survival groups under 1D. From this figure, we observed that the worst Classifier and NN model did not do well in predicting survival group "11" (both species survive). This might be because the proportion of group "11" is relatively small in the total simulation results (10.81 %), making it hard for models such as Logistic Classifier to learn from. However, the best performing Classifier and NN model did a good job even for the prediction of group "11".



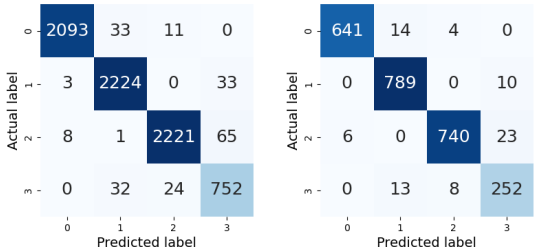
(a) LogisticRegression



(b) HistGradientBoostingClassifier



(c) NN (64, single)



(d) NN (64-32-16-32-64, bottleneck)

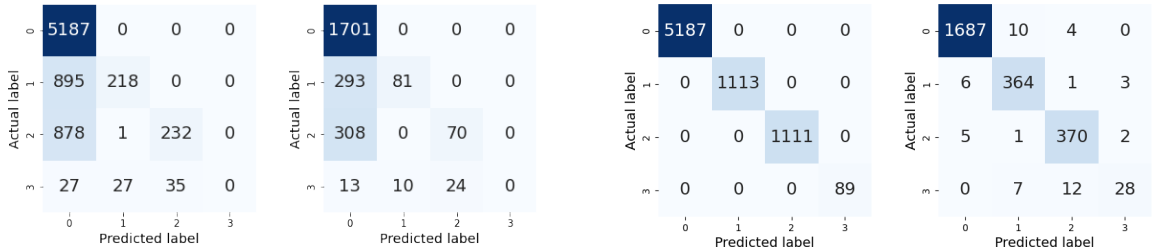
FIGURE 6: Confusion matrix for 4 Survival Groups (00, 01, 10, and 11) for 1D. Train dataset (left) and Test dataset (right)

Model	Type	Accuracy (Train)	Accuracy (Test)
3 layers (64-32-64)	Bottleneck	99.00%	98.76%
5 layers (64-32-16-32-64)	Bottleneck	98.87%	98.68%
5 layers (32-16-8-16-32)	Bottleneck	99.09%	98.68%
3 layers (64-64-64)	Flat	98.59%	98.64%
3 layers (64-32-16)	Triangular	98.88%	98.60%
3 layers (32-16-32)	Bottleneck	98.73%	98.36%
3 layers (32-32-32)	Flat	98.33%	98.20%
3 layers (16-16-16)	Flat	98.57%	98.04%
3 layers (16-8-16)	Bottleneck	98.23%	97.72%
3 layers (32-16-8)	Triangular	97.51%	96.92%
1 layer (64)	Single	91.00%	90.84%
1 layer (32)	Single	83.87%	83.24%
1 layer (16)	Single	70.51%	69.36%

TABLE V: Neural Networks Prediction Models for 4 Survival Groups (00, 01, 10, and 11) for 2D

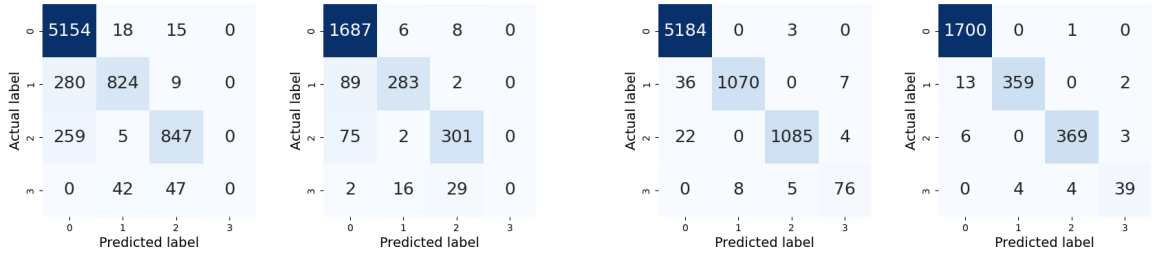
For the 2D dataset, Table IV presents the performance of 4 survival groups using the Classifier prediction model, and Table V presents that of Neural Network (NN) prediction model. We observed similar results as the 1D case, NN models generally outperformed Classifier models. Within Classifier models, the best performing model is an Ensemble-Methods-based Hist Gradient Boosting Classifier model, while the worst performing model is a Linear-Models-based Logistic Regression model. Within NN models, the best performing model is a Bottleneck-Layered 62-32-64 model, while the worst performing model is a Single-Layered 16 neurons model. Moreover, all the NN models with 3 or more layers can gain a test accuracy of more than 96%.

Figure 7 presents the best and worst performing models for 4 survival groups under 2D. From this figure, we observed that the worst Classifier and NN model did not do well in predicting survival group "11" (both species survive), this might be because the proportion of group "11" is so small in the total simulation results (1.36%), forming an imbalanced dataset that makes it impossible for models such as Logistic Classifier to predict group "11". The best performing Classifier and NN model did a better job for the prediction of group "11", though not as powerful as 1D scenario.



(a) LogisticRegression

(b) HistGradientBoostingClassifier



(c) NN (64, single)

(d) NN (64-32-16-32-64, bottleneck)

FIGURE 7: Confusion matrix for 4 Survival Groups (00, 01, 10, and 11) for 2D. Train dataset (left) and Test dataset (right)

Two groups prediction

Second, we present the performance of prediction models for 2 Survival Status Groups ("No species survive" and "At least one species survives"). We examined both 1D and 2D simulation datasets.

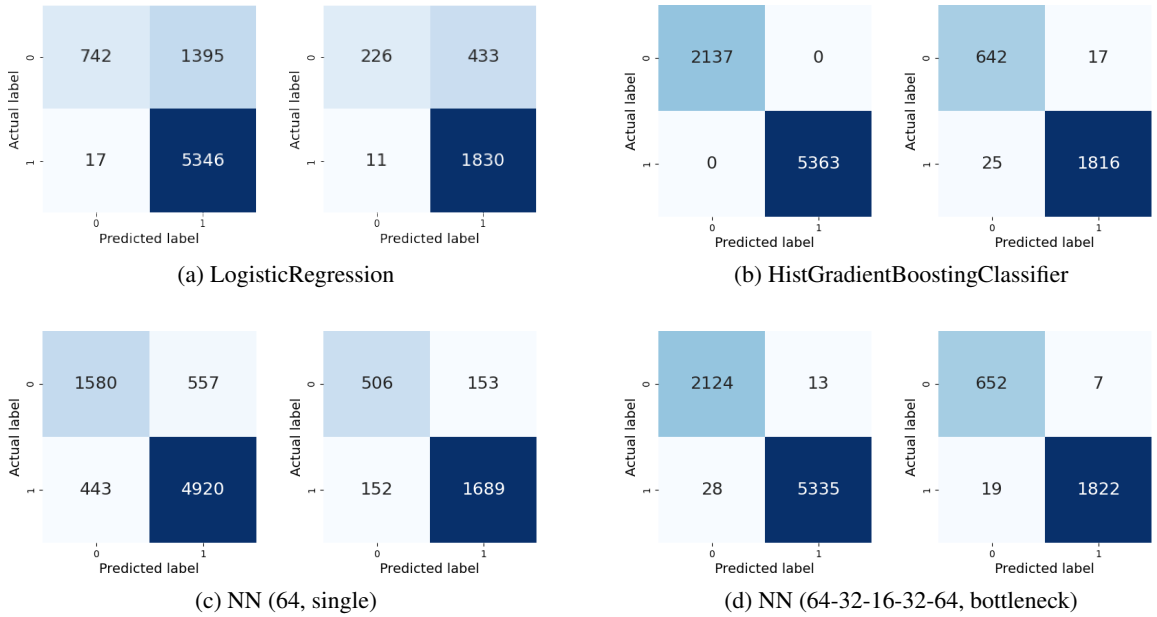
Model	Type	Accuracy (Train)	Accuracy (Test)
Hist Gradient Boosting Classifier	Ensemble Methods	100.00%	98.32%
Random Forest Classifier	Ensemble Methods	100.00%	97.80%
Bagging Classifier	Ensemble Methods	99.88%	97.20%
Gradient Boosting Classifier	Ensemble Methods	98.55%	96.64%
RBF SVC	Support Vector Machines	97.15%	96.56%
Decision Tree Classifier	Decision Trees	100.00%	96.00%
K Neighbors Classifier	Nearest Neighbors	96.55%	93.52%
Extra Trees Classifier	Ensemble Methods	100.00%	89.80%
Gaussian NB	Naive Bayes	87.03%	88.36%
Linear SVC	Support Vector Machines	86.59%	87.68%
AdaBoost Classifier	Ensemble Methods	88.08%	87.56%
Ridge Classifier	Linear Models	86.04%	87.12%
SGD Classifier	Linear Models	82.35%	83.40%
Logistic Regression	Linear Models	81.17%	82.24%

TABLE VI: Classifier Prediction Models for 2 Survival Groups ("No species survive" and "At least one species survives") for 1D

For the 1D dataset, table VI presents the performance of 2 survival groups using the Classifier prediction model, and table VII presents that of the Neural Network (NN) prediction model. Overall, we observed that, similar to the prediction of 4 survival groups, NN models generally performed better than Classifier models. Within Classifier models, we observed that similar to the prediction of 2 survival groups, the Ensemble-Method-based and Support-Vector-Machine-based models give the best test accuracy, while the Linear-Models-based models give the worst test accuracy. Within NN models, however, the Flat-Layered models give the best test accuracy (in contrast, in the prediction of 4 survival groups, the most powerful model is the Bottleneck-Layered model), while the Single-Layered

Model	Type	Accuracy (Train)	Accuracy (Test)
3 layers (64-64-64)	Flat	99.63%	99.20%
3 layers (32-32-32)	Flat	99.52%	99.20%
3 layers (64-32-16)	Triangular	99.20%	99.12%
5 layers (64-32-16-32-64)	Bottleneck	99.45%	98.96%
3 layers (64-32-64)	Bottleneck	99.15%	98.96%
5 layers (32-16-8-16-32)	Bottleneck	98.91%	98.88%
3 layers (32-16-8)	Triangular	99.36%	98.72%
3 layers (16-16-16)	Flat	99.13%	98.64%
3 layers (32-16-32)	Bottleneck	98.52%	97.64%
3 layers (16-8-16)	Bottleneck	93.88%	94.08%
1 layer (32)	Single	86.60%	87.96%
1 layer (64)	Single	86.67%	87.80%
1 layer (16)	Single	86.59%	87.60%

TABLE VII: Neural Networks Prediction Models for 2 Survival Groups ("No species survive" and "At least one species survives") for 1D



(a) LogisticRegression

(b) HistGradientBoostingClassifier

(c) NN (64, single)

(d) NN (64-32-16-32-64, bottleneck)

FIGURE 8: Confusion matrix for 2 Survival Groups ("No species survive" and "At least one species survives") for 1D. Train dataset (left) and Test dataset (right)

models still give the worst test accuracy. For NN models, besides the difference in test accuracy among different design types (bottleneck, flat, triangular, or single), we observed that the "pattern" under prediction of 4 survival groups (test accuracy rises as the number of layers and number of neurons increase) no longer holds for prediction of 2 survival groups. In the prediction of 2 survival groups, the 3-Layers Flat and Triangular models outperformed the 5-Layers Bottleneck model, though the difference is not much.

Figure 8 presents the best and worst performing models for two survival groups under 1D. From this figure, we observed that in the worst performing models, the NN-based model gives 305 wrong test predictions, less than that of the Classifier-based model (which gives 444 wrong test predictions); in the best performing models, the NN-based model gives 26 wrong tesst prediction, two times less than Classifier-based model (which gives 42 wrong test prediction).

For the 2D dataset, table VIII presents the performance of 2 survival groups using the Classifier prediction model, and table IX presents that of Neural Network (NN) prediction model. We observed similar results as the 1D case,

Model	Type	Accuracy (Train)	Accuracy (Test)
Hist Gradient Boosting Classifier	Ensemble Methods	100.00%	98.80%
Random Forest Classifier	Ensemble Methods	100.00%	98.68%
Bagging Classifier	Ensemble Methods	99.92%	98.68%
Decision Tree Classifier	Decision Trees	100.00%	98.20%
Gradient Boosting Classifier	Ensemble Methods	99.31%	97.92%
RBF SVC	Support Vector Machines	97.91%	97.76%
K Neighbors Classifier	Nearest Neighbors	96.04%	93.20%
Extra Trees Classifier	Ensemble Methods	100.00%	89.40%
AdaBoost Classifier	Ensemble Methods	88.29%	86.76%
Gaussian NB	Naive Bayes	84.21%	83.12%
Linear SVC	Support Vector Machines	82.87%	82.64%
Ridge Classifier	Linear Models	82.76%	81.72%
SGD Classifier	Linear Models	81.63%	81.08%
Logistic Regression	Linear Models	80.13%	79.40%

TABLE VIII: Classifier Prediction Models for 2 Survival Groups ("No species survive" and "At least one species survives") for 2D

Model	Type	Accuracy (Train)	Accuracy (Test)
3 layers (64-32-64)	Bottleneck	99.55%	99.80%
5 layers (64-32-16-32-64)	Bottleneck	99.53%	99.32%
3 layers (64-32-16)	Triangular	99.47%	99.28%
3 layers (16-16-16)	Flat	99.32%	99.20%
3 layers (32-32-32)	Flat	99.36%	98.96%
3 layers (16-8-16)	Bottleneck	99.29%	98.96%
3 layers (32-16-32)	Bottleneck	99.32%	98.88%
5 layers (32-16-8-16-32)	Bottleneck	99.16%	98.88%
3 layers (32-16-8)	Triangular	99.19%	98.80%
3 layers (64-64-64)	Flat	99.15%	98.76%
1 layer (32)	Single	83.72%	83.44%
1 layer (64)	Single	83.21%	83.04%
1 layer (16)	Single	83.07%	82.68%

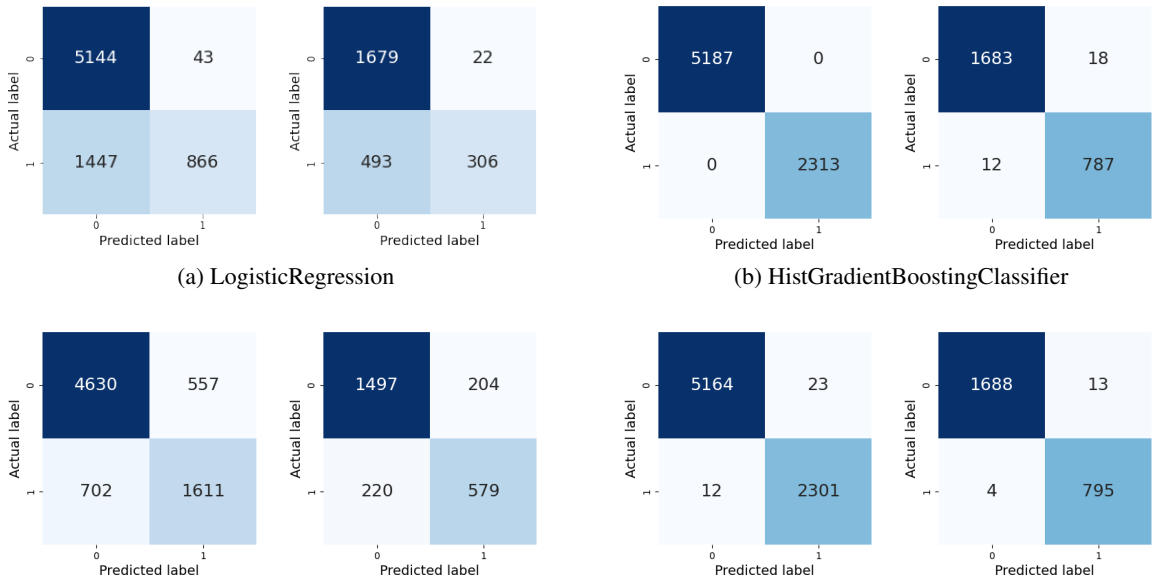
TABLE IX: Neural Networks Prediction Models for 2 Survival Groups ("No species survive" and "At least one species survives") for 2D

NN models generally outperformed Classifier models. Within Classifier models, the best performing model is still Ensemble-Methods-based models, while the worst performing model is still Linear-Models-based models. Within NN models, same as a prediction of 4 survival groups under 2D, the best performing model is a Bottleneck-Layered 62-32-64 model, while the worst performing model is a Single-Layered 16 neurons model. Moreover, all the NN models with 3 or more layers can gain a test accuracy of more than 98% (for prediction of 4 survival groups that test accuracy is above 96%).

Figure 9 presents the best and worst performing models for 2 survival groups under 2D. From this figure, we observed that in the worst performing models, the NN-based model gives 424 wrong test predictions, less than that of the Classifier-based model (which gives 515 wrong test predictions); in the best performing models, the NN-based model only gives 17 wrong test prediction, two times less than Classifier-based model (which gives 30 wrong test prediction). Overall, both Classifier-based and NN-based prediction models are more powerful under the 2D scenario, which means it performs better in the scenario that is closer to the real world.

CONCLUSION

In this paper, the two-species competition model is considered in a one- and two-dimensional formulation. The finite volume method with semi-implicit time scheme is used to construct a discrete system which used to generate two datasets related to the problem dimension. The input data contains a growth rate "r", competition term "α", and diffusion rate "ε" of both competing species randomly generated in 10,000 simulations. As an output data, we use a



(a) LogisticRegression

(b) HistGradientBoostingClassifier

(c) NN (64, single)

(d) NN (64-32-16-32-64, bottleneck)

FIGURE 9: Confusion matrix for 2 Survival Groups ("No species survive" and "At least one species survives") for 2D. Train dataset (left) and Test dataset (right)

groups related to the survival status that based on their equilibrium final population density. Datasets are used to train several classifiers and neural networks with different design of the hidden layers. We observe that the regular linear regression based classifiers cannot explain complex interaction of two-species model and have a lower test accuracy. Among all considered classifiers, we found that the ensemble methods and support vector machine classifier can provide prediction with better test accuracy. However, the machine learning neural networks with bottleneck design, larger number of layers and neurons give even better results.

REFERENCES

1. N. Schuwirth, F. Borgwardt, S. Domisch, M. Friedrichs, M. Kattwinkel, D. Kneis, M. Kuemmerlen, S. D. Langhans, J. Martínez-López, and P. Vermeiren, "How to make ecological models useful for environmental management," **411**, 108784.
2. J. D. Murray, *Mathematical biology: I. An introduction* (Springer, 2002).
3. G. C. Varley and G. R. Gradwell, "Recent Advances in Insect Population Dynamics," **15**, 1–24.
4. S. D. Albon, T. N. Coulson, D. Brown, F. E. Guinness, J. M. Pemberton, and T. H. Clutton-Brock, "Temporal changes in key factors and key age groups influencing the population dynamics of female red deer," **69**, 1099–1110.
5. S. Christin, E. Hervet, and N. Lecomte, "Applications for deep learning in ecology," **10**, 1632–1644.
6. Y. Wang, S. W. Cheung, E. T. Chung, Y. Efendiev, and M. Wang, "Deep multiscale model learning," **406**, 109071.
7. M. Vasilyeva and A. Tyrylgin, "Machine learning for accelerating macroscopic parameters prediction for poroelasticity problem in stochastic media," Computers & Mathematics with Applications **84**, 185–202 (2021).
8. M. Vasilyeva, W. T. Leung, E. T. Chung, Y. Efendiev, and M. Wheeler, "Learning macroscopic parameters in nonlinear multiscale simulations using nonlocal multicontinua upscaling techniques," Journal of Computational Physics **412**, 109323 (2020).
9. O. Ahmed and A. Brifcani, "Gene Expression Classification Based on Deep Learning," in *2019 4th Scientific International Conference Najaf (SICN)*, pp. 145–149.
10. A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: A review," **16**, 031001.
11. P. Bizopoulos and D. Koutsouris, "Deep Learning in Cardiology," **12**, 168–193.
12. E. Christodoulou, S. Moustakidis, N. Papandrianos, D. Tsaopoulos, and E. Papageorgiou, "Exploring deep learning capabilities in knee osteoarthritis case study for classification," in *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pp. 1–6.
13. S. Zheng, S. Chen, P. Qi, H. Zhou, and X. Yang, "Spectrum sensing based on deep learning classification for cognitive radios," **17**, 138–148.
14. H. S. Basavegowda and G. Dagnew, "Deep learning approach for microarray cancer data classification," **5**, 22–33.
15. T. R. Malthus, *An Essay on the Principle of Population ... The Fourth Edition*.
16. A. J. Lotka, "Fluctuations in the Abundance of a Species considered Mathematically," **119**, 12–12.

17. R. Arditi and L. R. Ginzburg, "Coupling in predator-prey dynamics: Ratio-Dependence," **139**, 311–326.
18. Z. P. Chairez, *Spatial-Temporal Models of Multi-Species Interaction to Study Impacts of Catastrophic Events*, Ph.D. thesis, Texas A&M University-Corpus Christi (2020).
19. M. Vasilyeva, Y. Wang, S. Stepanov, and A. Sadovski, "Numerical investigation and factor analysis of the spatial-temporal multi-species competition problem," arXiv preprint arXiv:2209.02867 (2022).
20. M. Vasilyeva, A. Sadovski, and D. Palaniappan, "Multiscale solver for multi-component reaction-diffusion systems in heterogeneous media," (2022), 10.48550/ARXIV.2209.04495.
21. P. N. Vabishchevich, "Additive operator-difference schemes," in *Additive Operator-Difference Schemes* (de Gruyter, 2013).
22. P. Vabishchevich and M. Vasilyeva, "Explicit-implicit schemes for convection-diffusion-reaction problems," Numerical Analysis and Applications **5**, 297–306 (2012).
23. M. Vasilyeva, "Efficient decoupling schemes for multiscale multicontinuum problems in fractured porous media," arXiv preprint arXiv:2209.01158 (2022).
24. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, and D. Cournapeau, "Scikit-learn: Machine Learning in Python," , 6.
25. J. Pinto, M. Mestre, J. Ramos, R. S. Costa, G. Striedner, and R. Oliveira, "A general deep hybrid model for bioreactor systems: Combining first principles with deep neural networks," **165**, 107952.
26. S. Hochreiter, "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions," **06**, 107–116.
27. K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," **1**, 24–35 ().
28. A. Gulli and S. Pal, *Deep learning with Keras* (Packt Publishing Ltd, 2017).
29. N. Ketkar, "Introduction to keras," in *Deep learning with Python* (Springer, 2017) pp. 97–111.
30. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," the Journal of machine Learning research **12**, 2825–2830 (2011).
31. M. Bekkar and D. H. K. Djemaa, "Evaluation Measures for Models Assessment over Imbalanced Data Sets," , 13.
32. F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, "An Introductory Review of Deep Learning for Prediction Models With Big Data," **3**.
33. A. Lavecchia, "Deep learning in drug discovery: Opportunities, challenges and future prospects," **24**, 2017–2032.
34. K. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," **1** (), 10.1038/s42256-018-0006-z.
35. M. Vasilyeva, W. T. Leung, E. T. Chung, Y. Efendiev, and M. Wheeler, "Learning macroscopic parameters in nonlinear multiscale simulations using nonlocal multicontinua upscaling techniques," **412**, 109323.
36. S. Wang, J. Cao, and P. S. Yu, "Deep Learning for Spatio-Temporal Data Mining: A Survey," **34**, 3681–3700.