Conservation Biology: Report — Group II

Modelling animal invasive species distribution

Palash Sethi
f2014260@pilani.bits-pilani.ac.in

Mihir Deshmukh
f2014472@pilani.bits-pilani.ac.in

Dhruval Rana
f2014222@pilani.bits-pilani.ac.in

Divyansh Saini
f2014261@pilani.bits-pilani.ac.in

Satwik Bhattamishra
f2014319@pilani.bits-pilani.ac.in

I. INTRODUCTION

In the past few decades, a number of methods have been proposed to model species distributions yet there is no goto approach for applied researchers to model a given species. Given the publicly available datasets of detailed climatic and topological data along with the effort and collaboration of researchers to generate reliable location data of different species, predictive modelling of species distribution has become substantially feasible and reliable. Species distribution models have been applicable in many fields such as climate change biology, conservation biology and landscape-ecology. Although the type of data available for different species varies, there are usually two categories of data for a given species : detailed presence/absence data or presence only data. Although presence/absence data is certainly more preferable as compared to presence only data, however, in a more practical scenario absence data is relatively more difficult to collect and for most species only the presence only data can be found. In this article we attempt to provide some useful insight into the performance of standard predictive models such as simple models (decision trees, naive bayes), complex models (gradient boosting machines, neural networks) and popularly used models (Maxent). Additionally we analyze the influence of using pseudo-negative samples to assess a model and propose an alternate approach based on generative adversarial networks to model species distribution using presence only data. Our empirical study, based on two invasive species namely european starling and zebra mussel support the validity and effectiveness of our proposed approach.

The remainder of the article is organized as follows. In section II we discuss some background and prior work related to the field. The type of data and its details are discussed in section III. In section IV we compare the performance of standard machine learning models. In the next section we introduce feature embeddings using autoencoders (AE) and generative adversarial networks(GANs) to improve the performance of the models and later use a variant of GANs for the purpose of one-class classification. Finally section VI ends with conclusion.

II. RELATED PRIOR WORK

Most of the species distribution models (SDMs) rely on environmental factors (abiotic) such as climate based features and topological features to model a given species. The idea is to

relate the observed presence of a species to values of the environmental variables. Essentially they rely on the statistical correlation between the observed species distribution and its surrounding environmental variables to predict how likely it is for a species to occur in the conditions of a given geographical location. Simple but widely used models such as maximum entropy model (Maxent)[6] and genetic algorithm for rule-set prediction (GARP)[8] use presence-only data to predict species distributions. Maxent is a robust general-purpose machine learning method which initially originated for a modelling a natural language processing model[2]. Phillips et al. 2006 proposed an approach to use Maxent for modelling species geographic distribution which received considerable attention due to its ability to make robust predictions without much effort in parameter tuning and its easy to use user interface for non-technical users. The essential idea behind Maxent is to estimate the target distribution by find the probability distribution with maximum entropy, subject to a set of constraints based on the given presence only data. GARP is a relatively older SDM which uses genetic algorithm (GA) to determine a rule-set for the given environmental variables. It starts with a random set of mathematical rules and optimizes them using GA to specify ideal range of each variable to determine the potential of the occurrence of a species. GARP is considered to be robust when dealing with smaller samples of presence only data.

Although a lot of work has been done and analyzed[3] to model species distributions using presence only data, most of the models overestimate the extent of the suitable environmental conditions of the given species[3] [6]. Maxent uses pseudo-negative samples to assess its performance and claims to be robust against false positives as compared to GARP. Thus we intend to assess the performance of a given model to handle actual negative cases and analyze the shortcomings of using pseudo-negative samples.

III. DATA

To model the distribution of a particular species we need two sets of data. Firstly we need the locations where the species was observed which is to say that we need an observed distribution map which contains the coordinates of the sightings of that particular species. Secondly we map those coordinates to a GIS database to obtain geographical and environmental features which can help us predict the suitable conditions of the presence of that species.

We chose two invasive species for our experiments namely european starling and zebra mussel. European starling are chunky and blackbird-sized birds with short tails and long, slender beaks and are native to europe. Zebra mussel is a mollusc which originated in the Black Sea and Caspian Sea in Eurasia. Both the species are invasive in north america. Zebra mussel was chosen since a lot of true negative points were availabe for the species. The sightings map is usually a csv file containing longitude and latitude of the place where that organism was sighted. The data for these species are publicly available online[1].

[1]https://www.eddmaps.org/distribution/

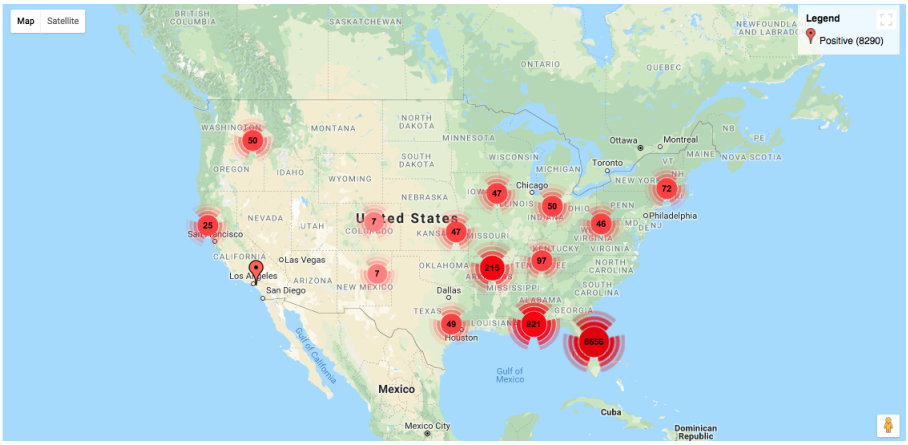

Figure 1: Sightings of European Starling

Figure 2: European Starling

Figure 3: Zebra Mussel

The second type of data we require are the values of different environmental variables that will help our model to characterize the presence of a species in a given location. Using the coordinates obtained from the sightings map we could retrieve different kinds of abiotic features such as the statistics about the temperature, elevation, precipitation...etc in the region. We have used the bioclim data containing 40 features, the first 35 of which includes different aggregates (by week, month, annual) of temperature, radiation, precipitation and moisture. The last 5 features are the first 5 principal components of the original 35 features. The data and the full list of the features are available online[2].

The dataset is available in the standard ESRI format and thus there is a file for each feature containing the values in a grid based manner. The first six lines of the ESRI file is constant in every file and denotes the coordinates of origin and the cell size of the grid. Given the cell size and the coordinates of origin we can estimate the nearest grid point with respect to our required coordinate. Figure 11 shows an instance how a particular value is obtained for a given coordinate from a given environmental variable file.

Thus the second set of data containing the environmental is flexible in terms of selection and we could incorporate a number of additional features to improve our model. Although it is not entirely obvious to decide which feature actually plays a role in providing a good performance and which one could have a negative effect as noise. The importance of the

[2]https://www.climond.org/BioclimData.aspx

features in predicting the species distribution could be evaluated using leave-one-out cross-validation but one should still be cautionary about adding too many noisy features into the data.

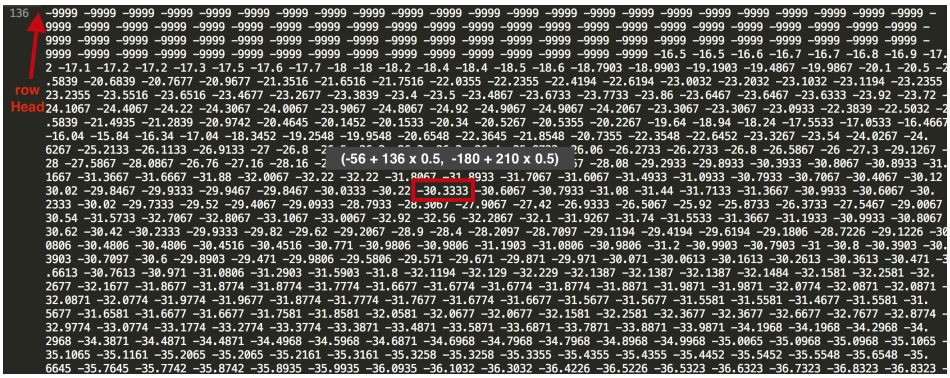

Figure 4: ESRI format instance

The data preprocessing steps could be summarized as follows:

- Collect and clean the distribution map
- Index the environmental variables from the ESRI format files
- Merge with environmental data
- Remove Null values
- Generate Pseudo Random points for absence data

IV. STANDARD MODEL ASSESSMENTS

We test a set of machine learning models which are commonly used for data science problem. We test the models on european starling dataset and evaluate their performance using AUC ROC score and log-loss metric. The set of models that we evaluate include tree based models (Decision Tree, Random forests, Xgboost), feed-forward neural networks, Naive Bayes model, Kernel SVMs (rbf, poly) and Maxent. Although it is widely considered that discriminative models such as RF, Xgboost and NNs will generally outperform generative models like Naive Bayes and Maxent, it has been suggested with some theoretical and empirical evidence[5] that generative can outperform discriminative models in some cases such as smaller sample size and imbalanced dataset. Thus we do not rule out models such as naive bayes from our study.

A. METRIC DESCRIPTION

Both log-loss and AUC ROC score are classification metrics based on probabilities. The AUC ROC score is a reliable binary classification metric which signifies how separable the predicted probabilities are as compared to the true values. To compute the AUC ROC the

true positive rate and false positive rate for different thresholds are plotted and the area under them is taken as the final value. Log-loss is variant of the likelihood function and the lower the value of the log-loss is the better the performance of the model. Log loss takes into account the uncertainty of the prediction based on how much it varies from the actual label and thus gives a more nuanced view into the performance of the model.

B. RESULTS

Table I denotes the performance of different models on the european starling dataset. The negative samples were generated by sample random coordinates which are not close to the places where the species were observed and then mapping their corresponding environmental data. The gradient boosting machines and the neural networks lead to the highest scores although the models are still subject to hyperparameter tuning.

TABLE I: Model Performances (%) (5-fold Cross Validation)

Method	AUC ROC	Log-loss
Maxent	90.12	0.091
Naive Bayes	86.32	0.165
SVM (RBF)	94.84	0.087
SVM (Poly)	90.81	0.1003
Decision Trees	92.53	0.082
Random Forest	95.67	0.048
Neural Networks	96.56	**0.042**
Xgboost	**96.81**	0.046

V. PROPOSED METHOD

The methods used earlier are subject to the way the pseudo-negative samples are chosen. Recent work[9] has suggested based on some theoretical analysis that having true negative samples significantly changes the behavior of the models and are thus important. But since absence data are not usually available our goal is capitalize on the presence data as much as possible. We first show how features generated from deep learning architectures such as AE [1] and GANs[4] improve the model performance, then we discuss why the use of pseudo-negative samples are not ideal and finally we evaluate a one-class classification variant of GAN to model the data using presence-only data.

A. USING FEATURE EMBEDDINGS

In this approach we add a method to the data preprocessing step where transform the data from its original space to a lower-level representation using autoencoders and GANs. Using stacked autoencoders we aim to learn a stochastic non-linear mapping $\mathbb{R}^n \rightarrow \mathbb{R}^d$ to transform the data to a lower d-dimensional space. A standard autoencoder takes an input vector $v \in [0,1]^n$ and maps it to a latent representation $y \in [0,1]^d$ through a deterministic

mapping $y = f_\theta(v) = g(Av + b)$, parameterized by $\theta = \{A, b\}$. $A \in \mathbb{R}^{n \times d}$ is a weight matrix and b is a bias vector. The latent representation is then mapped back to a reconstructed vector $\hat{v} \in [0,1]^n$ in the input space $\hat{v} = h_{\theta'} = g(A'y + b')$ where $\theta' = \{A', b'\}$. The parameters (θ) of the model are optimized to get the minimum reconstruction error between $\hat{v}$ and v.

Generative adversarial networks is a recently proposed method which uses a game theoretic approach to learn the true distribution of the data. It is composed of two networks namely the generator and the discriminator where the goal of the generator is to generate an input for the discriminator such that the discriminator will predict it as true. The goal of the discriminator is to distinguish between the true input and the input generated by the generator. Thus the problem is formulated as a min-max problem and the generator and discriminator are optimized on the given loss functions:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{1}$$

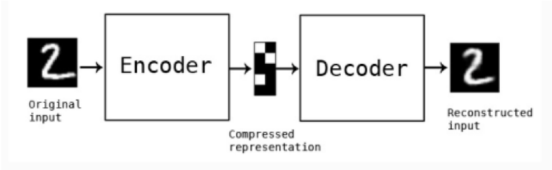

Figure 5: Autoencoder Schematic

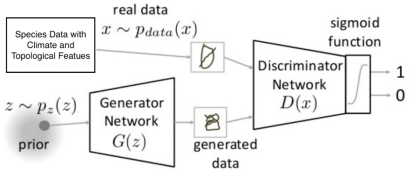

Figure 6: GAN Schematic

Figure 5 and Figure 6 show a schematic representation of GANs and autoencoders. To obtain features from the network we first train the GAN in the standard way and then remove the final layer of the discriminator to get the features learned by the discriminator network. We then apply a machine learning model on this feature space to get the final result. The idea is that the transformation learned by the network could make it easier for the models to distinguish between the two classes of data in the transformed space as compared to the original vector space. Table II shows the results obtained after incorporation the learned representations and we can clearly see that using the transformation helps improve the performance of the random forest model.

TABLE II: Model Performances (%) (5-fold Cross Validation)

Method	AUC ROC	Log-loss
Maxent	82.41	0.112
Random Forest	85.27	0.076
SVM (RBF)	83.55	-
AE + RF [2 layers]	88.53	0.058
GAN + RF [2 layers]	87.38	0.062

B. GAN AS ONE-CLASS CLASSIFIER

The general problem of modelling the species distribution using presence-only data is essentially a problem of one-class classification. Using pseudo-negative samples tends to make the model biased towards the way the pseudo-negative samples were generated. This could lead to model giving high accuracy in cross-validation but performing poorly on the actual test data.

There are three standard ways to generate pseudo-negative samples. The first and the simplest one is to sample a random number between the maximum and minimum value of each feature. Sampling for each feature gives us an additional point. The second one is to compute the mean and variance (or other parameters) for each feature and sample from a Gaussian distribution (or other distribution). But that method would required prior knowledge of the distribution of each feature. The third method involves sampling a random coordinate where the species has not been observed.

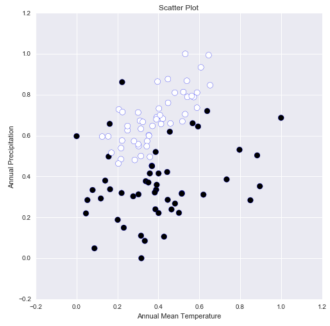

Figure 7: Scatter plot: Annual Precipitation vs Annual Temperature

Figure 7 shows the scatter plot between two of the features for zebra mussel. The light blue points correspond to true positive (presence) points and the dark blue points correspond to true negative (absence) points.

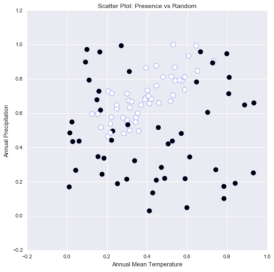

Figure 8: Random

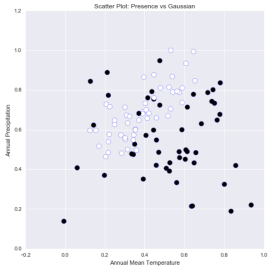

Figure 9: Gaussian

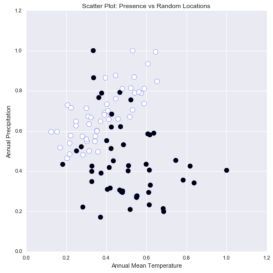

Figure 10: R-Locations

Figure 8, 9 and 10 show the pseudo-negative points generated by the three methods explained earlier and it is trivial to observe that none of the methods is similar to the actual plot of the true negative points in Figure 7. We thus emphasize against the usage of

pseudo-negative sampling as it could create biased models which may perform poorly on test data.

We thus model our problem using a one-class classifier[7] of GAN proposed earlier this year. In case of the standard GAN the discriminator cannot act as a final predictor since when the network is properly trained the generator is able to generate real-enough image to confuse the discriminator such that the discriminator will predict both the true input and generator's output with a probability close to 0.5. The major difference between the standard GAN and the one-class classifier variant is the change in the architecture of the generator. Here, the authors proposed to use a network $\mathcal{R}$ which is essentially an autoencoder instead of the standard generator. The network $\mathcal{R}$ takes an original input and provides the reconstructed vector to the discriminator. The goal of the discriminator is to distinguish between the original input and the value provided by network $\mathcal{R}$. Figure 11 shows the architecture of the network.

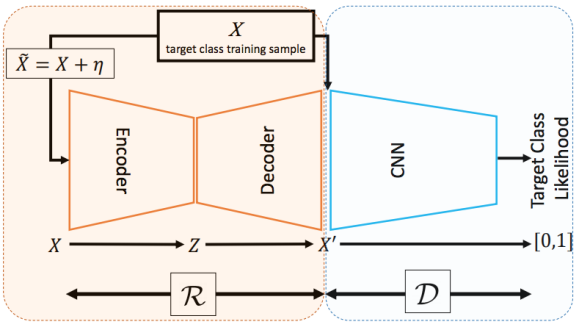

Figure 11: One-class variant: Architecture

TABLE III: Model Performances (%) (5-fold Cross Validation)

Method	Random Forest	One-Class GAN
AUC ROC	75.08	78.22
False Positive Rate	47.15	34.31
Precision	92.29	87.10

Table III shows the comparison between the performance of a standard random forest model with that of a one-class classifier variant of GAN. The GAN was trained using only the presence only data whereas the random forest was trained with properly generated pseudo-negative samples. The models were test against known presence and absence data. Although the AUC ROC score of GAN is clearly higher it is important to observe that the false positive rate of random forest is significantly higher implying that the model overestimates the suitable regions for the species. The confusion matrix of the two models are given in figure 12 and 13 which show that the false positives by random forest is around 260 whereas that of GAN is around 190. This is still subject to further tuning of GAN which could improve its performance.

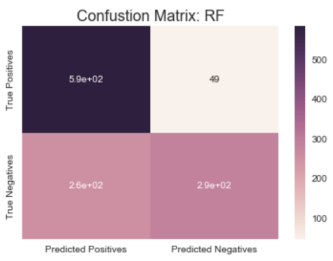

Figure 12: Confusion Matrix : RF

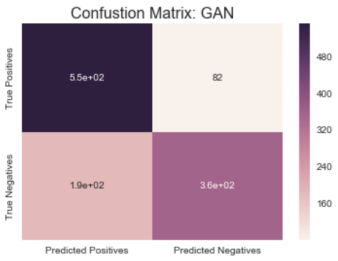

Figure 13: Confusion Matrix : GAN

VI. CONCLUSION

In this article we first assessed the performance of standard machine learning methods on european starling dataset. Additionally we showed how representations generated by deep learning architecture could improve the general methods. We then analyzed the implications of using pseudo-negative samples and why ideally the problem should be formulated as a one-class classification problem rather than forcing it to be binary classification problem by introducing pseudo-negative samples. We then showed how a one-class classifier variant of GAN compares to a properly trained random forest model when they are tested on actual negative samples. One should note that the pseudo-negative samples chosen for random forest are optimized to give the best performance and using a random method for generating pseudo-negative samples could substantially decrease its performance on real data. The one-class classifier network could still be improved by tuning its parameters but a rough version still works well.

REFERENCES

[1] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 37–49, 2012.

[2] Adam L Berger, Vincent J Della Pietra, and Stephen A Della Pietra. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71, 1996.

[3] Jane Elith. Predicting distributions of invasive species, 2015.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[5] Andrew Y Ng and Michael I Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pages 841–848, 2002.

[6] Steven J Phillips, Robert P Anderson, and Robert E Schapire. Maximum entropy modeling of species geographic distributions. *Ecological modelling*, 190(3-4):231–259, 2006.

[7] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. *arXiv preprint arXiv:1802.09088*, 2018.

[8] David Stockwell. The garp modelling system: problems and solutions to automated spatial prediction. *International journal of geographical information science*, 13(2):143–158, 1999.

[9] Tomáš Václavík and Ross K Meentemeyer. Invasive species distribution modeling (isdm): Are absence data and dispersal constraints needed to predict actual distributions? *Ecological modelling*, 220(23):3248–3258, 2009.

VII. CONTRIBUTIONS

The contributions of the team members for different aspects of the project are mentioned below. For each task the contributions are in descending order of the amount of work done by each team member unless mentioned otherwise.

- Data Collection: Satwik, Mihir, Palash, Divyansh, Dhruval
- Data Preprocessing: Satwik, Mihir, Dhruval
- Testing Maxent/Garp : Mihir, Palash
- Implementing Standard Machine Learning Models: Satwik, Divyansh, Mihir
- Evaluating different Machine learning models: Dhruval, Mihir, Palash
- Assessing Pseudo-negative samples effect: Satwik, Divyansh, Dhruval
- Implementing Autoencoder for feature transformation: Satwik, Palash
- Implementing GAN: Satwik