Deep Learning Architectures Applied to Mosquito Count Regressions in US Datasets

Chapter · October 2021

DOI: 10.1007/978-3-030-89817-5_15

Deep Learning Architectures Applied to Mosquito Count Regressions in US Datasets

Cuauhtemoc Daniel Suarez-Ramirez[1], Mario Alberto Duran-Vega[1], Hector M. Sanchez C.[2], Miguel Gonzalez-Mendoza[1](✉), Leonardo Chang[1], and John M. Marshall[2]

[1] School of Engineering, Computer Science Department, Tecnologico de Monterrey, Nuevo Leon, Mexico
{a01206503,a00755076}@exatec.tec.mx, {mgonza,lchang}@tec.mx
[2] School of Public Health, Epidemiology and Biostatistics Department, University of California, Berkeley, Berkeley, USA
{sanchez.hmsc,john.marshall}@berkeley.edu

Abstract. Deep Learning has achieved great successes in various complex tasks such as image classification, detection and natural language processing. This work describes the process of designing and implementing seven deep learning approaches to perform regressions on mosquito populations from a specific region, given co-variables such as humidity, uv-index and precipitation intensity. The implemented approaches were: Recurrent Neural Networks (LSTM), an hybrid deep learning model, and a Variational Autoencoder (VAE) combined with a Multi-Layer Perceptron (MLP) which instead of using normal RGB images, uses satellite images of twelve channels from Copernicus Sentinel-2 mission. The experiments were executed on the Washington Mosquito Dataset, augmented with weather information. For this dataset, an MLP proved to achieve the best results.

Keywords: Deep Learning · Machine Learning · Computer Vision · Mosquito · LSTM · VAE · MLP

1 Introduction

Mosquito-borne diseases have been one of the main causes of mortality in humans for centuries [2,5,13]. These encompass the malaria parasite, and the Dengue (DENV), Yellow Fever (YFV), Chikungunya (CHIKV) and Zika (ZIKV) arboviruses [1,5,13]. According to the World Mosquito Program (WMP), more than 700 million infections occur each year, with at least 1 million of them resulting in death [14]. This is why WHO has declared some of these diseases as some of the main threats to mankind.

For the Arboviruses, the *Aedes* genus is the main cause of human-to-human transmission [13]; particularly, female individuals of *Ae. aegypti* species [1] are the main vectors of the disease.

I. Batyrshin et al. (Eds.): MICAI 2021, LNAI 13067, pp. 199–212, 2021.
https://doi.org/10.1007/978-3-030-89817-5_15

Quantifying the presence of the population of these mosquitoes is relevant to predict which zones will become foci of infection to take preemptive actions such as the deployment of insecticides or other population control techniques to prevent a fast paced increase in cases.

One of the efforts regarding monitoring mosquito populations, is the dataset created by Washington D.C. Health authorities [3]. This department has been trapping and testing mosquitoes for almost a decade, in response to the Zika outbreak in Latin America and the Caribbean. The Washington Mosquito Dataset (WMD) contains 2024 records from 28 sites and 36 traps collected across 8 wards from the District of Columbia. This data is collected and reported from April through October, which is usually the mosquito season in Washington D.C.

The task of predicting the mosquito populations sizes (regression) using Machine Learning (ML) or Deep Learning (DL) techniques, has been tackled with meager efforts. Sotomayor *et al.* [11] - inspired by Inception Net [12] - used a Multi-Layered Perceptron (MLP) to process tabular data (weather and trap related information), and a Convolutional Neural Network (CNN) to process Satellite information (images) in the CNN. The Hybrid Model (HM) combine later both outputs. The results were produced by using the Washington Mosquito Dataset (WMD) and they are shown on Table 1.

This work aims towards tackling the regression task of predicting the population size of the *Ae. aegypti* in Washington D.C., by combining diverse DL techniques which encompass the use of spatial and temporal information to create more robust architectures.

Table 1. Previous obtained results with Washington Mosquito Dataset

Metric	MLP	CNN	Hybrid
Mean error	14.58	75.05	6.86
SD	17.94	61.21	11.08
R^2	0.9816	0.091	0.9956

The remainder of this work is organized as follows Subsect. 2.1 presents a detailed analysis of multiple MLP implementations. Thereafter, the process of designing our custom dataset is described in Subsect. 2.2. Subsection 2.3 provides the implementation of a CNN using Google Maps imagery. Following, the testing of the Hybrid Model is detailed in Subsect. 2.4. The designing and implementation of the Sentinel Hybrid Model is described in Sect. 2.5. Subsequently, An Hybrid VAE Model description and Implementation is shown in Sect. 2.6. Then, a RNN approach is tested in Subsect. 2.7. Finally, analysis and conclusions are shown in Sect. 3.

Additionally, the datasets and the code for implementing the architectures here presented is located in https://github.com/CuauSuarez/Mosquito-Count-DL.

2 Methods

Sotomayor *et al*. implemented three DL approaches for the task of predicting mosquito populations over WMD [11]: a four-layer MLP; a pre-trained VGG19 network; and an HM which concatenates the output from the first and second model, to create an input for a final MLP. To improve upon these architectures, we implemented a total of seven architectures, four previously tested by Sotomayor *et al*. but slightly improved, and three new approaches. These methods are:

1. An MLP using just the categorical and numerical data from the weather reports (cleaned and pre-processed).
2. An MLP, with WMD augmented with weather information acquired from DarkSky API [7] (https://darksky.net) and World Weather Online (WWO) [9] (https://www.worldweatheronline.com), to improve our previous results with additional information related to the mosquito life-cycle.
3. A CNN using exclusively satellite imagery from Google Maps. This architecture was based on DarkNet-53 using pre-trained weights from YoloV3 [10].
4. An HM based on the work of Sotomayor *et al*. It concatenates the output of an MLP and a CNN, to create an input for the final MLP.
5. The previously presented HM, but instead of using images of three channels provided by Google Maps, it uses twelve-channel images from Copernicus Sentinel-2 mission. Which contains more information like humidity and vegetation.
6. A custom HM which concatenates a latent vector from the satellite image (obtained with a VAE), with the rest of the tabular data.
7. A LSTM, approach which fits naturally due to the temporal nature of the WMD plus weather. Since it is augmented with weather information of two weeks from the trap collection.

The process of developing these seven architectures was that of gradually increasing the complexity of it and including more data regarding time and space.

In the following sections, each of the architectures are further described.

2.1 Multi-Layer Perceptron (MLP)

The first approach we implemented was an MLP with WMD. To make data patterns easier for the model to learn. We pre-processed the data as follows:

1. We removed the columns *Lifestage*, *EggsCollected*, *LarvaeCollected*, *PupaCollected*, *Town*, *State*, and *County*. Since those columns contained the same data across the 2023 rows.
2. We removed duplicated data, such as *X*, *Y* and *Address*, which represent the location of the trap, while keeping *Latitude* and *Longitude* for that purpose.
3. We decomposed the dates into two numerical columns, *week* and *Year*; and we normalized all numerical columns.

We selected a K-Fold Cross-validation with a K of 5 for validating our model. To produce the best possible results by each model, we tried all the combinations of the next hyper-parameters: Learning rate (0.01, 0.001, 0.0001), Dropout (0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.60), Hidden Units (10, 20, 30, 40, 50, 60, 70, 80, 90, 100), Hidden Layers (2, 3, 4).

Table 2. Results of each MLP with the corresponding number of layers in Washington D.C dataset.

MLP results				
Metric	Previous MLP	MLP-2	MLP-3	MLP-4
Mean error	14.58	8.82	8.79	8.78
R^2	0.9816	0.61	0.64	0.65
(Only Aedes) MLP results				
Metric	Previous MLP	MLP-2	MLP-3	MLP-4
Mean error	–	3.98	3.99	4.00
R^2	–	0.49	0.48	0.48

The best combination of hyper-parameters for all mosquito species were: RMSProp with learning rate of 0.001, 4 layers, with a hidden size of 70, and a dropout of 0.1. This resulted in a mean absolute error of **8.78**.

As shown in Table 2, our MLP architecture of 4 layers obtained a better median average than the MLP implemented by Sotomayor *et al*. However, we are not sure how R^2 was measured for the MLP presented by Sotomayor *et al*. Since a mean average of 14.58 is unlikely to lead to the $R^2 = 0.98$ they reported.

The WMD contain information of several species of mosquitoes, like *Culex pipens* or *Psorophora columbiae*. But *Aedes aegypti* is the main cause of human-to-human transmission of Arboviruses. We prepared a smaller version of the dataset, containing only samples of this species. We trained and tested our model with this dataset. The best mix of hyper-parameters were: RMSProp with learning rate of 0.01, 4 layers, with a hidden size of 60, and a dropout of 0.1. This led in a mean absolute error of **3.98** (Table 2).

The results obtained by our MLP, improved the results presented by the MLP of Sotomayor *et al*. However, one further improvement is to augment the WMD with weather information. Since weather information is related to the mosquito life-cycle and their population number.

2.2 Approach 2 (MLP + Weather Data)

Aedes aegypti has a life cycle that lasts around 1.5 weeks (with intense sunlight) or 3 weeks (in cold periods). It involves fours stages: egg, larva, pupa, and adult. Variables like temperature or food availability, affects the life cycle of the mosquito. In good environmental conditions, mosquitoes reach their adult form in around 10 days (Fig. 1) [11].

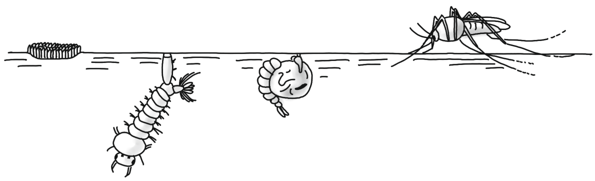

Fig. 1. Egg (2–7 days) larva (4 or more) pupa (2 days) emerging adult.

To include information about the mosquito life-cycle in our model, we built a WMD dataset augmented with weather information. To create this dataset, we requested information from the past 14 days since the trap collection date, and the GPS coordinates.

To build the aforementioned dataset, we used the DarkSky API [7]. This framework provides weather information according to a given latitude, longitude and date. Which is information available in WMD.

A previous approach proposed by Sotomayor *et al.* involved augmenting the WMD by requesting weather information from the day the trap was collected. However, since the life-cycle of *Aedes aegypti* depends of the weather conditions from the previous 10 to 14 days, we augmented WMD with weather information from the previous 14 days before the trap was collected. We trained our MLP with the WMD augmented with weather information (Table 3). The best combination of hyper-parameters for all mosquito species were: RMSProp with learning rate of 0.01, 3 layers, with a hidden size of 140, and a dropout of 0.3. This resulted in a mean absolute error of **9.06**.

Moreover, we repeated this experiment with a smaller dataset containing only *Aedes aegypti* (Table 3). The best mix of hyper-parameters were: RMSProp with learning rate of 0.01, 2 layers, with a hidden size of 120, and a dropout of 0.2. This led to a mean absolute error of **4.10**.

Results obtained by using WMD augmented with weather information were inferior when compared to the original WMD. This could be caused by the noise generated by the large amount of variables used to train our MLP. Therefore, we performed multiple feature selection approaches to improve our results (Pearson, Kendall, Spearman and PCA).

Table 3. Results of each MLP by number of layers, using WMD + Weather.

Metric	Previous MLP	MLP-2	MLP-3	MLP-4
Mean error	14.58	9.061	9.060	9.11
R^2	0.9816	0.64	0.60	0.67
(Only Aedes)				
Metric	Previous MLP	MLP-2	MLP-3	MLP-4
Mean error	–	4.10	4.15	4.16
R^2	–	0.54	0.55	0.50

We trained multiple MLP models selecting features from the aforementioned coefficients. However, even after performing feature selection, additional weather information from DarkSky did not improve the results. Thus, we used another weather information provider to confirm the validity of the information provided by DarkSky. For this additional test, we used World Weather Online (WWO) API. We downloaded all the weather information from 14 days before the trap was collected, and trained our MLP models. However, the result of our best model was a median error of **9.85**, which was inferior when compared with the results presented at Table 3. DarkSky API and WWO API return similar weather information, since we evaluated two providers of meteorological information. And despite we included attributes directly related to the mosquito life-cycle, our results did not improve. Thus, we think WMD is biased or noisy.

2.3 Approach 3: CNN

Mosquitoes like *Aedes aegypti*, use almost any kind of clean water container to lay eggs [4]. One of the findings of Sotomayor *et al.* [11], is that we can use satellite images of the trap's surroundings to extract spatial features and improve the mosquito count prediction (for example, water containers).

We implemented a CNN regression model, which takes as input the satellite image from the location of the trap (obtained from Google Maps, and scaled to 416×416). To process the aforementioned images, the selected architecture was a pre-trained Darknet-53 [10] followed by 3 dense layers. We performed a cross validation of 5 folds to validate our model and measure the accuracy (Table 4).

The best combination of hyper-parameters for all mosquito species were: RMSProp with learning rate of 0.001, 3 layers, weight decay of 0.004. This results in a mean absolute error of **13.11**. For *Aedes* the best combination were: RMSProp with learning rate of 0.001, 3 layers, with a hidden size of 120, and a weight decay of 0.004. This led to a mean absolute error of **5.22**.

According to the results, our model performed better than the CNN model presented by Sotomayor *et al*.

Table 4. Results of our pre-trained CNN

CNN		
Metric	Prev	Avg
Mean error	75.05	13.11
Aedes CNN		
Metric	Prev	Avg
Mean error	–	5.22

2.4 Approach 4: Hybrid Model (CNN + MLP)

The MLP we proposed in Subsect. 2.1 and the CNN model in Subsect. 2.3 improved the results of the same architectures implemented by Sotomayor *et al*. Following the idea of the hybrid architecture introduced in their work, we combined both architectures into one. By adding an additional NN at the end, which takes as input the output of the MLP and the CNN model. To evaluate this architecture, we used K-Fold Cross-validation of 5 folds, the results are shown in Table 5. The best mix of hyper-parameters for all mosquito species were: RMSProp with learning rate of 0.0001, 3 layers, weight decay of 0.004, and dropout of 0.3. This led in a mean absolute error of **10.85**. For *Aedes* the best combination were: RMSProp with learning rate of 0.0001, 3 layers, weight decay of 0.004, and dropout of 0.3. This resulted in a mean absolute error of **4.8**. For this hybrid architecture, results were inferior as the presented by Sotomayor *et al*.

Table 5. Results of our Hybrid Model compared to the previous Hybrid Model from Sotomayor *et al*. [11]

Hybrid model		
Metric	Prev	Avg
Mean error	6.86	10.85
Aedes hybrid model		
Metric	Prev	Avg
Mean error	–	4.8

2.5 Approach 5: Sentinel CNN + MLP

This approach was inspired by the hybrid model proposed by Sotomayor *et al*. However, our HM implementation showed in Sect. 2.4 was not able to improve or replicate the work of Sotomayor *et al*. Nevertheless, we implemented some improvements to the aforementioned model, en compare the results with the previous approaches.

The hybrid model presented by Sotomayor *et al*. improved the results of their predictor by using satellite images provided by Google Maps. However, despite their good results, we noted some possible improvements.

1. Google Maps API, does not provide images for given dates. To make a good prediction, we need to gather images which are closer to the day the trap collection date. To extract features like soil humidity and vegetation that might affect the mosquito count prediction.
2. *Aedes aegypti* can only flight around 1 km, thus, each image needs to cover that maximum space.
3. Google Maps API only offers images of 3 channels (RGB), but other APIs like Sentinel-02, offer images of 12 channels; which contain information about vegetation, soil humidity, and other features that could help the predictor to achieve better results.

Following these improvements, we compiled a dataset of Sentinel-02 images, one for each of the rows of WMD. Sentinel-02 images have 12 channels instead of 3, thus, we need to implement and train a Darknet-53 architecture capable of handling images of 12 channels. Unfortunately, our dataset is too small to train a Deep Neural Network such as Darknet-53. Therefore, we used a pre-trained Darknet-53 model trained for Yolov3. However, as Sentinel-2 images have 12 channels, we split each image of 12 channels into 3 images of 3 channels each, to make them compatible with the pre-trained Darknet-53. Each image contained information like vegetation and soil humidity.

We pass each of the three images generated by the Sentinel-02 through Darknet-53 to extract spatial features. Then, we concatenate the resulting 3 outputs with the result of the MLP network. Finally, we sent the resulting vector to a NN of 3 Dense Layers. The best combination of hyper-parameters for all mosquito species were: RMSProp with learning rate of 0.00001, 3 layers, weight decay of 0.004, and dropout of 0.3. This results in a mean absolute error of **11.10**. For *Aedes* the best combination were: RMSProp with learning rate of 0.00001, 3 layers, weight decay of 0.004, and dropout of 0.3. This resulted in a mean absolute error of **4.8** (Table 6).

Table 6. Results of our hybrid sentinel model

Hybrid sentinel model		
Metric	Prev	Avg
Mean error	6.86	11.10
Aedes hybrid sentinel model		
Metric	Prev	Avg
Mean error	–	4.8

2.6 Approach 6: VAE + MLP

Following the same line of thought, the next approach we tested was based on combining the spatial information (matrices processed with CNN) from satellite images plus the numerical information (vectors processed with MLP) from weather data. In this case, before directly using the images, we opted to encode them into latent vectors, which are lower dimensional representations of the data, as the most important data that is subtracted from the satellite images (Sentinel) relies on info like the water bodies, number of houses, potential nests, vegetation, etc., which does not require multiple dimensions to be represented (each of them). In other words, we transformed the matrix information into vectors.

Variational Autoencoder (VAE). Grouped inside the Generative Models, Autoencoders (AEs) is an architecture composed of an encoder part and a decoder one. The purpose of this architecture is to reduce the dimensionality of the input data (to save memory) and then being able to recreate it [8]. After the encoder section, data is represented by a low dimensionality vector in a "latent space" which, in concept, encodes the most important features of the input. Thus, it could be used in the same ways as dimensionality reduction architectures.

Traditional AEs present an issue in their latent space, it is not continuous. This has two main problems:

1. Trying to generate new similar images to the data-set from random vectors from the latent space won't always have good results.
2. Images that are similar do not necessarilty are closer in the latent space.

Variational Autoencoders (VAEs) are a modification to AEs to introduce a Divergence Measure (DM) to the latent space to assure that samples are distributed in a continuous matter; Usually, for the DM, the Kleiber Divergence (KD) is used, but this is perceived as restrictive and not preserving the variance across vectors. Thus, the Maximum Mean Discrepancy (MMD) was used as the error measure [6].

Note: Satellite data (from Google) was not used as it only gives images of the current time, not historical.

$$\begin{aligned}\mathrm{MMD}(p(z)\|\|q(z)) = \mathbb{E}_{p(z),p(z')}[k(z,z')] + \mathbb{E}_{q(z),q(z')}[k(z,z')] \\ - 2\mathbb{E}_{p(z),q(z')}[k(z,z')]\end{aligned} \tag{1}$$

where $k(z, z')$ is a generic kernel; in this case, the radial basis kernel was used.

Based on this, we used the architecture shown in Fig. 2. Here, c is the number of channels of the original image, Sentinel-1 has 12 different channels and we tested using only the RGB ones (3 channels) or the 11 channels (only 11 channels are provided by the sentinel-02 web service). Moreover, the *n_latent* dimension was chosen to be 50 as it produced low reconstruction error.

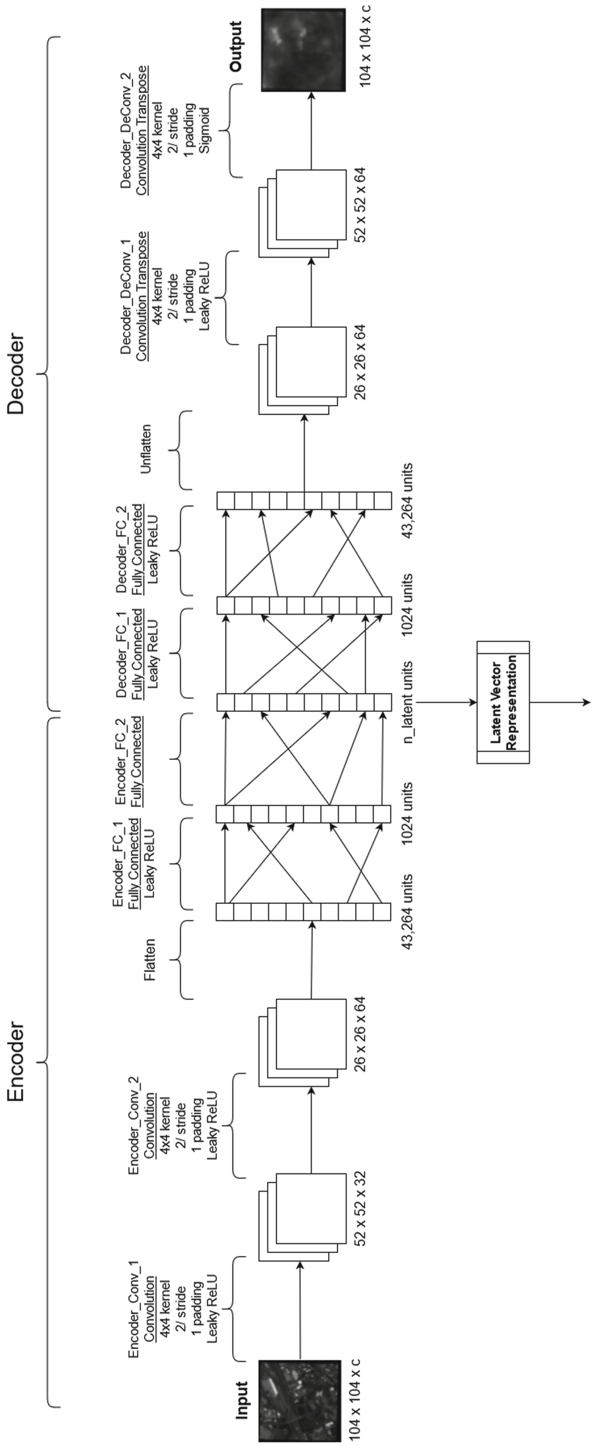

Fig. 2. VAE architecture for sentinel images.

Apart from the MMD, VAEs use the traditional loss function for comparing images (Negative Log Likelihood). The obtained errors where: **0.0001** when using just the RGB channels, and **0.0001** when using all the channels.

Adding the Numerical Data. Now that the images are processed, the rest of data was added following a mixed model approach. This means that the data from the tables is processed through a Neural Network (NN) having a lower dimension vector as an output, this is then concatenated to the Images information, and the resulting vector is then passed through another NN to obtain the final prediction.

The hyper-parameters that we chose to modify are: $e1$ as the number of the dimensions of the embeddings (for the categorical data), nX which are the different number of hidden layers marked on the architecture. Moreover, we also chose different optimizers (Adam and RMSProp), and learning rates (0.01, 0.001, 0.005, 0.0005, 0.0001). The final setup used the 5 most important parameters (according to the correlation coefficients), RMSProp as the optimizer (with learning rate 0f 0.0001), $e1$ was chosen for each particular categorical parameters, $n1 = 20$, $n2 = 10$, $n3 = 100$, $n4 = 50$. The error for this model was of **8.58** when measuring mean absolute error.

As an alternative, another architecture was used applying a NN also to the latent vector. After experimentation, the hyper parameters were chose as: RMSProp with learning rate of 0.0001, $n1 = 30$, $n2 = 10$, $n3 = 30$, $n4 = 50$, $n5 = 30$. This resulted in a mean absolute error of **8.89**.

In general, these both architectures are troublesome for choosing the correct hyper-parameters (and number of layers/units), it is usual to arrive to solutions where the same number is always the same number (highly non-convex topology of the solutions), and the results are plainly surpassed by a simpler NN where only the structured data is used. Thus, the efforts were redirected to a model that only uses this kind of data and considers the temporal nature of the information.

2.7 Approach 7: Recurrent Neural Network (LSTM)

As the weather information is obtained per day, it makes sense to use time-series-analysis techniques in these kind of data, considering the cycle of 14-days for the development of the mosquitoes. Then, we proposed a mixed model where the temporal data is processed through a LSTM+MLP, the numerical data is passed through a MLP, the categorical data is embedded and then passed through a MLP, and finally, either we take a weighted average of the 3 outputs (first approach), or concatenate the resulting vectors and pass it through a final MLP.

Weighted Average. For calculating the dimension of the LSTM, the following formula was used:

$$\mathrm{n1} = \lfloor \frac{\mathrm{batchsize}}{a_h \times \#columns} + 1 \rfloor \tag{2}$$

The "TRAPDAYS" column was created by us to count the number of days that the trap was set. An improvement that can be done to this is to consider also the time of the day when the trap was set and recovered to include decimal values; is not the same to set the trap on the evening of day 1 and retiring on the morning of day 2, when compared to setting the trap on the morning of day 1 and retiring it on the evening of day 2.

Then, after trying several combinations of hyper-parameters, the chosen were: RMSProp with learning rate of 0.001, $a_h = 3$, $l1 = 2$, $n2 = 15$, $n3 = 15$, and a weighted average giving a value of 2 to the numerical data. This resulted in a mean absolute error of **4.80** which is much better than the models using images.

Mixed Model. This architecture was the most explored approach of the temporal ones. The best combination of hyper-parameters gave a mean absolute error of **4.36** which is considerably better than the weighted average approach.

This combination has a mean absolute error of **4.36** which is considerably better than the weighted average approach.

3 Analysis and Conclusions

Mosquitoes are one of the most dangerous creatures for the human beings due to the pathogens they transmit. The development of tools to predict mosquito populations, is of great interest to design control strategies that might help to reduce mosquito-related infections.

We implemented a total of 7 approaches for the prediction of mosquito populations. To present the results of each approach in a more readable manner, we have grouped them in Table 7. Surprisingly, the lowest error was obtained by applying a MLP in the WMD. However, we decided to analyze the dataset to check if the quality of the data was not a factor in getting poor results.

To have a better understating of the quality of WMD, we performed a correlation analysis, calculating Pearson, Spearman and Kendall coefficients. We found that the highest correlation (Spearman) with the dependent variable (Total of mosquitoes) was 0.29 and it corresponded to the "GENUS" variable; the rest of the variables had correlation coefficients lower than this (by 0.05 or more points). We concluded that WMD has poor data quality for this particular application, due to the low correlation coefficients that variables like temperature or precipitation, show for the total number of mosquitoes. Additionally, to corroborate the quality of the extracted weather information from DarkSky, we gathered weather information from WWO to compare output of the models. However, both datasets showed no improvement in the results. Thus, we think that the WMD is a noisy dataset or, at least, it is not fit for getting further insights from it.

Other possibility to consider is to test more methods to get data from images and different time-windows for the temporal architectures. As using these information increased the error of the model, implementing other techniques to get relevant attributes from this data could increase the value of the metrics.

One possibility could be to implement an image segmentation network to search the bodies of water in the images instead of processing the complete satellite image.

Table 7. Best results obtained for each of the used approaches

Approach	Mean absolute error
VAE + MLP	8.58
Pre-trained CNN	5.22
Sentinel CNN + MLP	4.8
Implemented hybrid model [11]	4.8
LSTM + Weighted average	4.80
LSTM + Mixed model	4.36
Simple MLP-3 + Weather data	4.10
Simple MLP-2	**3.98**

Although adding extra temporal weather information was not helpful for this dataset, using information directly related to the mosquito life-cycle might still work with other datasets since mosquitoes depend of rain, water containers, and temporal conditions like the temperature over time to develop. In the future, it would be worth to test these 7 approaches with other mosquito datasets besides WMD.

References

1. Arredondo-García, J., Méndez-Herrera, A., Medina-Cortina, H.: Arbovirus en latinoamérica. Acta pediátrica de México **37**(2), 111–131 (2016). http://dx.doi.org/10.18233/APM37No2pp111-131
2. Bhatt, S., et al.: The global distribution and burden of dengue. Nature **496**, 504–507 (2013). https://doi.org/10.1038/nature12060
3. D.C. Office of the Chief Technology Officer: Mosquito Washington Dataset (2019). https://opendata.dc.gov/datasets/DCGIS::mosquito-trap-sites/about. Accessed 9 Aug 2020
4. Christophers, S.R.: Aedes aegypti: the yellow fever mosquito. CUP Archive (1960)
5. Ferreira, A.G., Fairlie, S., Moreira, L.A.: Insect vectors endosymbionts as solutions against diseases. Curr. Opin. Insect Sci. **40**, 56–61 (2020). https://doi.org/10.1016/j.cois.2020.05.014. http://www.sciencedirect.com/science/article/pii/S221457452030078X
6. Gretton, A., Borgwardt, K.M., Rasch, M., Schökopf, B., Smola, A.J.: A kernel method for the two-sample problem. In: Advances in Neural Information Processing Systems, pp. 513–520 (2007)
7. Apple Inc: Dark Sky API (2020). https://darksky.net/. Accessed 9 Aug 2020
8. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2014)

9. World Weather Online: World Weather Online (2020). https://www.worldweatheronline.com/. Accessed 11 Nov 2020
10. Redmon, J., Farhadi, A.: YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767 (2018)
11. Sotomayor, L., Vallejo, E., Sánchez Castellanos, H.: Application of a hybrid neural network trained with satellite imagery and weather data to predict mosquito populations based on mosquito trap captures. Ph.D. thesis, Tecnologico de Monterrey, December 2019
12. Szegedy, C., et al.: Going deeper with convolutions. In: Computer Vision and Pattern Recognition (CVPR) (2015). https://doi.org/10.1109/CVPR.2015.7298594. http://arxiv.org/abs/1409.4842
13. Weaver, S.C., Charlier, C., Vasilakis, N., Lecuit, M.: Zika, chikungunya, and other emerging vector-borne viral diseases. Annu. Rev. Med. **69**(1), 395–408 (2018). https://doi.org/10.1146/annurev-med-050715-105122, pMID: 28846489
14. World Mosquito Program (WMP): What are mosquito-borne diseases? (2020). https://doi.org/10.1016/j.cppeds.2009.01.001. https://www.worldmosquitoprogram.org/en/learn/mosquito-borne-diseases. Accessed 9 Aug 2020