Machine Learning Assists the Classification of Reports by Citizens on Disease-Carrying Mosquitoes

Antonio Rodriguez[1], Frederic Bartumeus[2,3,4], and Ricard Gavaldà[1]

[1] Universitat Politècnica de Catalunya, Barcelona (Spain)
[2] Centre for Advanced Studies of Blanes (CEAB-CSIC), 17300 Girona (Spain)
[3] CREAF, Cerdanyola del Vallès, 08193 Barcelona (Spain)
[4] ICREA, Pg Lluís Companys 23, 08010 Barcelona (Spain)

Abstract. *Mosquito Alert* (`www.mosquitoalert.com/en`) is an expert-validated citizen science platform for tracking and controlling disease-carrying mosquitoes. Citizens download a free app and use their phones to send reports of presumed sightings of two world-wide disease vector mosquito species (the Asian Tiger and the Yellow Fever mosquito). These reports are then supervised by a team of entomologists and, once validated, added to a database. As the platform prepares to scale to much larger geographical areas and user bases, the expert validation by entomologists becomes the main bottleneck. In this paper we describe the use of machine learning on the citizen reports to automatically validate a fraction of them, therefore allowing the entomologists either to deal with larger report streams or to concentrate on those that are more strategic, such as reports from new areas (so that early warning protocols are activated) or from areas with high epidemiological risks (so that control actions to reduce mosquito populations are activated). The current prototype flags a third of the reports as "almost certainly positive" with high confidence. It is currently being integrated into the main workflow of the Mosquito Alert platform.

1 Introduction

One of the unintended consequences of globalization is the expansion of invasive species outside of their original habitats. These may have harmful or even devastating effects on the invaded ecosystem. In other cases, these species can carry serious diseases (e.g. Dengue, Chickungunya, Zika, Yellow Fever) affecting humans or other animals. This is the case of the mosquito species that are the focus of this paper.

Mosquito Alert (`www.mosquitoalert.com/en`) [1] is a citizen science, expert-validated, platform developed at Center for Research and Ecological Applications (CREAF) and the Center for Advanced Studies of Blanes (CEAB-CSIC), near Barcelona (Spain). It started in 2014 under the name *AtrapaelTigre* ("catch the tiger") because it initially focused on determining the Asian Tiger mosquito (*Aedes albopictus*) distribution and spreading process in Spain. In the past two years, Mosquito Alert has built up a community, bringing together citizens, scientists (modelers and medical entomologists), and stakeholders (public health administrations and mosquito control services) to help minimize mosquito-borne disease risks in Spain. Based on a solid multidisciplinary team, Mosquito Alert is now starting to scale up to offer a global tool that can aid in the fight against Zika, Chikungunya, Dengue, and Yellow Fever worldwide; this has implied adding the Yellow Fever mosquito (*Aedes aegypti*) in the platform, recently notorious as the main vector of the Zika epidemics in South America [2], extending the geographical area, and moving the system from data-rich to Big Data scenarios. Currently, Mosquito Alert has attracted interest in Latin America, United States and China, where new communities are expected to grow and generate new data.

The "frontend" of the platform is a freely downloadable app for Android and iOS phones that everybody can use to send reports of mosquito sightings. Up to now there have been 20,000 downloads of the app in Spain, and there are several thousands of continuous participants in average throughout the year. The number of participants is increasing rapidly in light of the global attention to the current Zika epidemics in Latin America. Reports from the app users are stored in the "backend" of the platform and help scientists to detect adult mosquitoes and their breeding grounds. This information is used to build distribution and future expansion models, as well as directly helping to control their expansion by activating early warning alerts and control actions to decrease mosquito populations and epidemic risks. It is thus a clear example of *citizen science* [3] in which the general public co-participates in the scientific research and development of the platform, via knowledge sharing, intellectual abilities, resources, or tools. Citizens are actors and at the same time end users experiencing the benefits of such a research process.

More precisely, citizens' actions involve sending reports of spotted tiger mosquitoes or their breeding sites. The latter are small water containers that can proliferate after rain periods either in public spaces (e.g. fountains, sewers, or water drainers) or in private areas (e.g. flower pots in terraces and gardens, rain collecting devices in urban growing gardens, etc). When sending the report to the platform servers, citizens are asked to fill-in a short questionnaire, attach a photography if possible, and give permission to attach the geolocalization of that report. For confidentiality purposes, no information about the citizens is collected other than the random anonymous identifier assigned when they register the app. This unique identifier is used to trace the overall activity and performance of each anonymous user in order to improve engagement and communication with the community. These reports are inspected and validated (or rejected) by a team of entomologists and are included in a database and an interactive webmap. The project, or authorities, can then derive actions from that information, such as dispatching verification and control teams to reported locations.

Classification is strictly done by visual examination of the pictures, if included with the report. Each report is validated by three experts, with a super-expert making the final decision in case of disagreement.

Reports need supervised validation because a certain fraction of the reports are erroneous: non-experienced citizens may report regular mosquitoes as tiger mosquitoes despite the tutorials in the web and the information and guided questionnaire provided in the app. The work of the validating entomologists is one of the bottlenecks for the scalability of the system. This paper describes an application of machine learning to spare entomologists the verification of part of the reports; more specifically, with the datasets gathered so far, the application can flag over 30% of the received reports as true tiger mosquito sightings, with high confidence. This will allow the entomologists to focus on other more valuable tasks, such as verifying new reports from areas where the specimens have not yet been established, or organizing control teams in high-risk epidemiological areas, as well as being able to handle larger geographical areas and larger user bases. Other possibilities to the same effect beyond analyzing the reports could be considered in the future, for example, analyzing the pictures themselves with image processing techniques and cross-validating expert and non-expert supervision to allow citizens to improve over time.

In fact, feedback on pictures is partially available through the public map in the Mosquito Alert webpage. In the pop-ups of the map, participants can see their own pictures with the comments by experts. By checking the expert comments and the pictures, an untrained citizen can quickly learn which are pictures are considered high-quality, i.e. which ones are useful to identify the targeted species. In a future app release, feedback on scores and overall quality of their pictures will be sent directly to the users' cellphones.

The manuscript is structured as follows. In Sections 2 and 3 we describe the datasets used for the study and the preprocessing process. In Section 4 we describe the experiments with different classifiers, their results, and the current choice of a classifier to be implemented. In Section 5 we sketch the architecture of the system as it is currently being implemented in *MosquitoAlert*. Finally in Section 6 we recap the outcomes of the experience and highlight a few possibilities for future work.

2 The Datasets

The project provided two main files for the study. The first lists the users that have downloaded the app including user ID, download time and other related information (app version downloaded). The interesting part is the file with the reports received during 2014 and 2015, whose fields will be discussed shortly.

The dataset contains 16967 users and 10618 reports; about a third of registered users sent no reports, another third sent a single report, and following a Zipfian-looking distribution, the maximum number of reports sent by a single user is 38. Note that in that period the app only allowed to report the tiger mosquito *Aedes albopictus* species.

The key value of this dataset is that it contains the label or classification provided by the entomologists for each report, allowing us to treat the problem as a supervised learning one. Five labels are possible, encoded as integers in the range $-2 \ldots 2$.

- 2: this is for sure a tiger mosquito spotting
- 1: this is probably a tiger mosquito spotting
- 0: there is not enough information to classify
- -1: this is probably not a tiger mosquito spotting
- -2: this is for sure not a tiger mosquito spotting

Reports with label 0 were unfortunately the majority, typically those without a picture, for which entomologists cannot assess the validity for sure. We removed them since it was difficult to use them for either training and testing. This left a total of 2094 usable reports, distributed in classes as follows:

class	2	1	-1	-2
frequency	47%	46%	2%	5%

Therefore, this is a moderately class-imbalanced problem, with positive instances over 7 times as frequent as negative ones.

2.1 Contents of the reports

The most relevant fields for each report are:

- userId,
- app version number
- phone operating system,
- report date and time
- report georeference (latitude and longitude), if available,
- report type (adult mosquito or breeding site),
- the answers to three taxonomic questions present in the questionnaire:
  - Q1: Is it small, black and has white stripes?
  - Q2: Does it have a white stripe in both head and thorax?
  - Q3: Does it have white stripes in both abdomen and legs?

  For each question the user can select one of three options (No/I don't know/Yes), represented by the numbers -1, 0 and 1 respectively;
- an optional comment in free-text format, and
- finally, the label or class assigned by the entomologist, taking values in $\{-2, -1, 1, 2\}$ as explained before.

3 Instance Construction

As usual in machine learning, features had to be transformed and new features built so that classifier building algorithms can use them. Most of the new features are created by aggregating across all reports from the same user, or across all reports from a geographical zone.

In particular, the Mosquito Alert platform locates reports in a grid formed by square cells of 4km × 4km, used as reference for the project. Knowing that a mosquito normally travels about 700 meters by itself during its life (if not transported e.g. by entering a car), we decided to additionally consider circular areas or 1km around each report. It is expected that positive reports tend to appear more often in invaded areas, e.g. around previous positive reports. A report from an area from which no previous positive reports exist is, on the one hand, more likely to be a user error, but on the other hand very important as it may be an early alert for a new invaded area.

The main features included in the dataset for classifier training are thus:

- Discretized time-of-day of the report (0-6am, 6am-noon, noon-6pm, 6pm-0). This information is relevant as some mosquito species may have different daily activity patterns. Other discretization ranges are of course possible.
- Month of the year. Mosquitoes are visible mostly during summer months, although the distribution curve is affected and shifted to some extent by weather conditions.
- Number of previous reports by the same user; note that some of the following features do not make sense if this is the first report by a user.
- The answers to questions Q1, Q2, Q3.
- The operating system used (Android, iPhone), just in case it happens to be relevant.
- User accuracy: Fraction of previous reports in agreement with entomologists.
- Time between user sign-in and this report.
- Time between last report by the user and this one.
- Average time between reports by this user.
- User Action Areas: Number of cells from which the user has sent report.
- User Mobility Index: This variable tries to express the user mobility between cells and its activity in each of them. A user that always sends reports from an specific cell but has sent only one report from a different cell is definitively different from one who sends the same amount of reports from two different cells. Intending to express this user movement activity, the variable has been computed as the standard deviation of the number of reports sent from every different cell the user has been active on.
- Reports around 1km in the last hour, last day, last week, and last month: Four variables indicating the number of reports in a circle of radius 1km around the location of this report, in the four time periods indicated.
- Valid reports around 1km in the last hour, last day, last week, and last month: Same as before but considering only the proportion of reports validated as positive.
- A boolean indicating whether the report has a comment or not. The user taking the time to enter a comment may indicate more careful work on his/her part.
- The class, in $\{-2, -1, 1, 2\}$.

4 Classifiers tested

Four classifiers were tested in order to predict the report classification, in particular their implementations in R and the Rweka packages:

- Naïve Bayes - 'e1071' R library
- k-nearest neighbors - 'e1071' R library
- Decision trees (C4.5) - Rweka R library
- Random forests - RandomForest R library.

A 10-fold cross-validation procedure was used to assess classifier performance. We omit the discussion of the parameter choice and model validation in each algorithm and other resampling methods.

In all cases, the initial results were rather poor. Very often, the classifier returned one of the majority classes (either 1 or 2) on all instances, never returning -1 or -2. Two variations of the training process were carried out:

1. The four classes -2, -1, 1, 2 were reduced to two by merging "probably negative" (-1) into "negative" (-2) and "probably positive" (1) into "positive" (2). This was a reasonable option since the Mosquito Alert team considered it was too challenging for the classifier to tell these categories apart , e.g. total or partial certainty about a positive or a negative report was somewhat subtle.
2. Oversampling the infrequent class (-2) in the training set by factors 5x to 10x, since classifiers tended to simply return "2" on all instances. Crucially, no replication was added to the testing dataset for each fold, therefore the error rates should be those of the trained classifier on the original distribution, not on the oversampled one.

Results were far better after these modifications except for k-nearest neighbors, which kept performing poorly. Confusion matrices for each of the other three classifiers are given in Tables 1, 2, 3.

		Predicted class	
		negative	positive
True class	negative	5.8%	0.6%
	positive	61.4%	32.2%

Table 1. Confusion matrix, Naive Bayes

Two of the classifiers stood out for their performance: Random Forests and Naïve Bayes. Random Forest achieves the highest accuracy (87.7%). However, it does not have particularly high recall or particularly high precision on any of the two classes, so it is unclear what use it can be given. On the other hand, Naïve Bayes has a substantially smaller accuracy (38%) but has an extremely

		Predicted class	
		negative	positive
True class	negative	3.9%	2.4%
	positive	16.4%	77.3%

Table 2. Confusion matrix, C4.5 Decision Trees. M=10.

		Predicted class	
		negative	positive
True class	negative	3.7%	2.6%
	positive	9.7%	84.0%

Table 3. Confusion matrix, Random Forests. 500 trees.

good precision on the positive class (32.2 / 32.8 = 98.2%). That is, whenever it classifies a report as positive, that report is with very high probability indeed positive. Furthermore, it does classify about a third (32.8%) of the reports as positive and only about 10% of the truly negative reports are labeled as possible. That is what is required for our intended purpose, that is, identifying with high confidence a significant fraction of reports that can be classified without taking entomologists' time. It has also the advantage that it can by interpreted relatively well for the experts; interpretation is harder for Random Forests.

Thus, we chose Naïve Bayes as the classifier to be implemented in the prototype, keeping in mind that Random Forests is a good candidate if some other usage that requires to prime overall accuracy appears in the future. The possibility of combining the two (or more) classifiers via voting is being investigated.

Figure 1 presents the ROC curve of the Naïve Bayes classifiers, and Table 4 lists the features by decreasing order of importance. It can be observed that the most important ones are the questionnaire answers Q2, Q3, and Q1, together with the number of reports within 1km in the last month. However, trying to predict on the bases of these four variables alone created a noticeable decrease in performance, e.g. the other variables do help. The least significant ones are (not very surprisingly) the fact that the user is new and the phone's operating system.

5 Integration in the project

The MosquitoAlert system is a hook-model [4] consisting of three main blocks: the MosquitoAlert app (available for Android and iOS), a corresponding Django-based server-side functionality including a data-base containing the reports send by the app-users, and an online platform `http://www.mosquitoalert.com`, providing three different levels of services. At the first level we have a platform called EntoLab. This is a restricted access service through which a set of experts can make a previous filtering of inappropriate reports and classify the rest as either positive or negative ones. Only classified reports are afterwards made visible

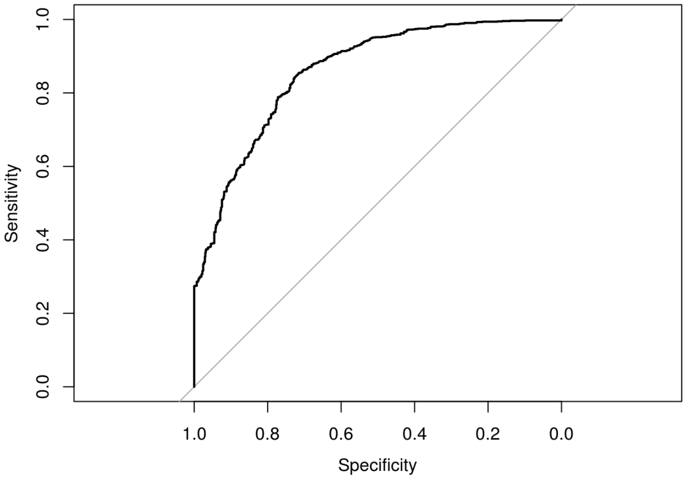

Fig. 1. ROC curve for the NB classifier. The area under curve is 0.8, and the maximum sensitivity (recall) that can be achieved without any false positives is close to 0.3.

to the rest of the services. An intermediate level called ManagersPortal grants on-demand-access to stakeholders (e.g. public health administrations, mosquito control services, private companies) with particular interests on the information about the spread of the mosquito and areas with imminent risk of being invaded. Finally, an open access level allows citizens to visualize all the information gathered, synthesized in the form of interactive maps (e.g. observations, app downloads), where they can find their individual contributions validated by the experts. This top level constitutes the necessary reward that closes the hook-model loop.

Because of the inherent bias of the set of reports towards positive ones (as well as in the classifier itself) the idea is to implement the classifier as a filter that yields an ordered list of the pending reports based on its positive score (i.e. the probability of being positive). Afterwards, and based on the ROC of the classifier (plus any aside considerations like current geographical interest) the experts can decide how many reports from the top of the list can be considered as correctly classified and do not need further expert supervision. Because of the relative low computational cost of the Naïve Bayes classifier, this filtering is implemented as a batch process scheduled at regular intervals (i.e. daily) but can be tuned and triggered as desired by the platform managers. This may be reconsidered in the future if computationally heavier classifiers were used.

The reports data-base contains all received reports, either already classified or pending to be classified. Thus, a batch implementation of the algorithm consists of the following steps:

Variable name	Importance
reportQ2Answ	0.7424
reportQ3Answ	0.7038
reports1kmLastMonth	0.6623
reportQ1Answ	0.6615
userNumReports	0.6405
userNumActionAreas	0.6348
validReports1kmLastMonth	0.6216
userTimeForFirstReport	0.6197
reports1kmLastWeek	0.6158
userAccuracy	0.6085
userTimeSinceLastReport	0.5923
userMeanTimeBetweenReports	0.5912
validReports1kmLastWeek	0.5688
reports1kmLastDay	0.5594
validReports1kmLastDay	0.5452
validReports1kmLastHour	0.5301
reports1kmLastHour	0.5281
userMobilityIndex	0.5260
reportMonth	0.5199
reportNote	0.5081
reportTimeOfDay	0.5057
os	0.4650
newUser	0.3648

Table 4. Variable importance in the NB classifier. Numbers are the values of the model coefficients after standarization.

1. Training & testing:
   (a) select all labeled (validated) reports with report generation date posterior to a fixed date (in order to control computational cost and capture relative up-to-date information);
   (b) preprocess data and generate the set of training/test instances;
   (c) split the instance set into training and testing subsets (or using cross-validation);
   (d) train the classifier with the training set;
   (e) test the classifier with the test set and compute the ROC.

2. Classifying:
   (a) select all pending reports (also with a report generation date posterior to a given date);
   (b) preprocess data and generate the set of instances; at this point we need the previous set of training/test instances to compute features like Reports around 1Km and Valid reports around 1Km;
   (c) classify the instances;
   (d) order the set of instances by decreasing positive score;

6 Conclusions and Future Work

In summary, a simple machine learning method opens the possibility of saving at least a third of the expert time with small rate of false positives.

An alternative use of classifiers that is being considered is have them exclude both surely positive and surely negative reports, and send the entomologists only the reports that the classifier is uncertain of. To this end one could consider the combination of Random Forests (which, as mentioned, have overall higher accuracy than Naïve Bayes) with ROC curve analysis. At the time of writing we are also starting to explore the option of deleting "uncertain" classes (-1, 1) from the training set, so that training is carried out only on the basis of "certain" labels (-2, 2). Preliminary experiments indicate considerable precision improvement, but further research is needed to assure the classifiers trained by these methods fit the platform needs.

Another aspect that could be exploited for classification is the use of "crowd intelligence" to extract information from the pictures. This is currently done at the platform `www.crowdcrafting.org`, where mosquitoalert.com is one among other projects being crowdcrafted. Mosquito pictures sent by citizen are redirected at this platform for people to validate. The results from this crowd validation is also visualized in the map, together with the expert validation. In future app versions, the possibility will be given to validate pictures from the cellphones itself. So users will not only be asked to make pictures but also to validate other user's pictures. Either from a web platform (crowdcrafting.org) or from the Mosquito Alert app, the system will collect citizen (i.e. non-expert) classification information to be compared with the expert one. In the future, we will be able to add citizen validation scores as input features, analyze the convergence between expert and citizen validations by region or collectively, and exploit all this new information for training the classifiers.

It is clear that one route to go is to develop and test new features and algorithms. A set of new features could involve the direct extraction of information from the pictures themselves, for example, through image processing techniques. The classification system can be fed with new input features, and as the supervised set of reports is constantly growing it makes sense to re-test the chosen classifiers or look for new ones in order to improve the overall performance of the classification system.

Integrating the classifier with expert-mandated rules is another necessary step. As mentioned before, some reports may be more strategic or urgent than others, e.g. those arriving from new areas with no past sightings or particularly vulnerable to disease. The possibility of using the non-expert user base for additional cross-validation while simultaneously speeding-up their learning curve, would require also careful integration with the classifier. Gamification could be an avenue to study.

One of the main future challenges of the platform *Mosquito Alert* is the scalability of the system in order to support the growing flow of reports and an increase on its complexity. It is true that at the moment a considerable flow of reports can be comfortably handed with a single machine. This could certainly change be certainly true if the aforementioned image-processing techniques were incorporated. But even with the current structure, some challenges may appear soon. In particular, creating an instance out of a new report requires locating the previous report within the same 4km × 4km cell and within a radius of 1km. For a large report stream, this may be complex without incorporating the proper data structures. Alternatively, the platform currently uses the PostgresSQL database, which has an extension called PostGIS which adds support for geographic objects allowing location queries to be run in SQL.

References

1. *Mosquito Alert: A citizen platform for studying and controlling mosquitos which transmit global diseases.*, Web content, `http://www.mosquitoalert.com/en/`. Accessed: July 11th, 2016.
2. *Zika virus disease epidemic: Preparedness planning guide for diseases transmitted by Aedes aegypti and Aedes albopictus*, Paula Vasconcelos, Laurence Marrama, Emma Wiltshire, Dragoslav Domanovic and Andrea Würz, Available at: `http://ecdc.europa.eu/en/publications/Publications/zika-preparedness-planning-guide-aedes-mosquitoes.pdf`. Accessed: July 11th, 2016.
3. *White Paper on Citizen Science for Europe*, Fermín Serrano Sanz, Teresa Holocher-Ertl, Barbara Kieslinger, Francisco Sanz García and Cândida G. Silva, Available at: `http://www.socientize.eu/sites/default/files/white-paper_0.pdf`. Accessed: July 11th, 2016.
4. *Hooked: how to build habit-forming products*, Nir Eyal and Ryan Hoover, 2014. ISBN-13: 978-0241184837.