Research on Biomedical Engineering manuscript No.
(will be inserted by the editor)

Cite this article

da Silva, C.C., de Lima, C.L., da Silva, A.C.G. *et al.* Spatiotemporal forecasting for dengue, chikungunya fever and Zika using machine learning and artificial expert committees based on meta-heuristics. *Res. Biomed. Eng.* (2022). https://doi.org/10.1007/s42600-022-00202-6

Spatio-temporal forecasting for dengue, chikungunya fever and zika using machine learning and artificial expert committees based on meta-heuristics

Cecilia Cordeiro da Silva · Clarisse Lins de Lima ·

Ana Clara Gomes da Silva ·

Giselle Machado Magalhães Moreno ·

Anwar Musah · Aisha Aldosery ·

Livia Dutra · Tercio Ambrizzi · Iuri V. G. Borges · Merve Tunali · Selma Basibuyuk · Orhan Yenigün · Tiago Lima Massoni · Kate Jones · Luiza Campos · Patty Kostkova ·

Abel Guilhermino da Silva Filho ·

Wellington Pinheiro dos Santos*

Received: DD Month YEAR / Accepted: DD Month YEAR

Corresponding author: Wellington Pinheiro dos Santos, E-mail: wellington.santos@ufpe.br, ORCID 00000003-2558-6602.

Cecilia Cordeiro da Silva and Abel Guilhermino da Silva Filho
Center for Informatics, CIn-UFPE, Federal University of Pernambuco, Recife, Brazil E-mail: {ccs2,agsf}@cin.ufpe.br

Ana Clara Gomes da Silva and Wellington Pinheiro dos Santos
Department of Biomedical Engineering, Federal University of Pernambuco, Recife, Brazil E-mail: {clara.gomes,wellington.santos}@ufpe.br

Clarisse Lins de Lima

Polytechnique School of the University of Pernambuco, Poli-UPE, Recife, Brazil E-mail:
cll@ecomp.poli.br

Giselle Machado Magalhães Moreno, Livia Dutra, Tercio Ambrizzi, and Iuri Valério Graciano Borges
Department of Atmospheric Sciences, IAG-USP, University of São Paulo, São Paulo, Brazil E-mail:
{gisellemoreno,iurivalerio}@usp.br,{livia.dutra,tercio.ambrizzi}@iag.usp.br

Anwar Musah, Aisha Aldosery, and Patty Kostkova
Centre for Digital Public Health and Emergencies, Institute for Risk and Disaster Reduction, University College London, United Kingdom
E-mail: {a.musah,a.aldosery,p.kostkova}@ucl.ac.uk

Merve Tunali, Selma Basibuyuk, and Orhan Yenigün
Boaziçi University, Institute of Environmental Sciences, Istanbul, Turkey
E-mail: {merve.tunali,yeniguno}@boun.edu.tr, selmabasibuyuk@gmail.com

Tiago Lima Massoni
Department Systems & Computing, Federal University of Campina Grande, Campina Grande, Brazil
E-mail: massoni@dsc.ufcg.edu.br

Kate Jones
Centre for Biodiversity and Environment Research, Department of Genetics, Evolution and Environment, University College London, United Kingdom
E-mail: kate.e.jones@ucl.ac.uk

Luiza Campos
Department of Civil Environmental & Geomatic Engineering, University College London, United Kingdom
E-mail: l.campos@ucl.ac.uk

Abstract

Purpose

Dengue is considered one of the biggest public health problems in recent decades. Climate and demographic changes, the disorderly growth of cities, and international trade have brought new arboviruses such as chikungunya and zika. Control of arboviruses depends on control of the vector: the Aedes aegypti mosquito.

Objective

In this work, we propose a methodology for building disease predictors capable of predicting infected cases and locations based on machine learning. We also propose an Artificial Experts Committee based on meta-heuristic methods to detect the most relevant risk factors.

Method

As a case study, we applied the methodology to forecast dengue, chikungunya and Zika, with data from the City of Recife, Brazil, from 2013 to 2016. We used arboviruses cases data and climatic and environmental information: wind speeds, temperatures and precipitation. Results

The best prediction results were obtained with 10-tree Random Forest regression, with Pearsons correlation above 0.99 and RMSE (%) below 6%. Additionally, the Artificial Experts Committee was able to present the most relevant factors for predicting cases in each twomonth period.

Conclusion

The spatio-temporal prediction results showed the evolution of arboviruses, pointing out as major focuses on both regions richer in urban green areas and low-income neighborhood with irregular water supply. Determining the most relevant factors for prediction, as well as the spatial distribution of cases, can be useful for the planning and execution of public policies aimed at improving the health infrastructure and planning and controlling the vector.

Keywords: Spatio-temporal forecasting · Arboviruses · Dengue fever · Chikungunya fever · Zika · Machine learning forecasting · Artificial expert committees based on meta-heuristics

1 **Introduction**

1.1 Motivation and problem characterization

Dengue has long been considered the most important viral disease transmitted by mosquitoes (arboviruses), being also the most widespread arbovirus in the world [10]. Dengue is clinically manifested in two main forms: the classic dengue, simply called dengue fever, and the hemorrhagic form, sometimes known as dengue hemorrhagic fever (DHF) or dengue shock syndrome (DHF/SCD). Since the early 1970s, the World Health Organization (WHO) has been actively involved in the development and promotion of disease control and treatment strategies [10]. Dengue is transmitted by mosquitoes of the Aedes genus, where the Aedes aegypti mosquito is its main vector. Aedes aegypti is found mainly in urban areas, in water storage deposits [10].

The re-emergence of classic dengue epidemics and the emergence of dengue hemorrhagic fever are considered to be part of the greatest Public Health problems of the second half of the 20th century and the beginning of the 21st century [10]. Demographic changes and the intense migratory flow from rural areas to urban regions have generated disorderly growth in cities, which, added to the lack of good basic sanitation conditions. These factors result in the proliferation of the vector, essentially in tropical and subtropical countries, where periodic outbreaks of the disease [10, 11] have been common.

Due to climate change favorable to the dispersion of vectors and their diseases and the growing number of international flights, favorable to the movement of sick or infected people in an incubation period, Brazil is experiencing the introduction and a rapid process of dispersion towards becoming two new arboviruses endemic to the Americas: the Chikungunya virus, introduced between July and August 2014 after entering the Caribbean in December 2013 and previously causing major epidemics on the African continent and Asia since 2004; and the Zika virus, possibly introduced in the same period during the 2014 World Cup in Brazil [68]. Preventing and combating the proliferation of Aedes aegypti is essential to contain any outbreaks and ensure an improvement in the health conditions of rural and urban populations.

Zika virus is a flavivirus (family Flaviviridae) transmitted by Aedes aegypti. Zika virus causes fever and other general symptoms such as headache, rash, malaise, swelling, and severe joint pain. More severe conditions, including involvement of the central nervous system (Guillain-Barré syndrome, transverse myelitis and meningitis), associated with Zika have been commonly reported [68]. There is evidence of a relationship between Zika virus and microcephaly, as well as other effects on fetuses, but these issues are not yet fully clarified [12, 50].

The prevention of arboviruses depends on prioritizing the elimination of the vector, that is, the Aedes aegypti outbreaks, and on prediction strategies that can provide health managers with adequate information to prevent arbovirus outbreaks [44, 54]. This requires active cooperation between government and health agencies in the development and execution of disease control strategies with the general population and the use of tools that can efficiently and effectively extract information from the various databases that municipalities already have. These data contain information on infrastructure, socio-economic and environmental aspects, distribution of health services and case mapping.

These databases can be linked to others, such as those with climate and environmental information from water agencies and state health departments, in addition to the mapping of the LIRAa index, to obtain temporal and spatio-temporal information of epidemiological interest [18, 54].

Several studies show that there is a strong correlation between the distribution of arboviruses and Aedes aegypti outbreaks with climatic factors, such as historical series of humidity, rainfall and temperatures [31, 41, 44, 60, 67]. Urbanization and changes in the cultivation of certain crops also strongly influence the distribution of the vector [41, 44]. It is also possible to integrate environmental information collected from historical series obtained from satellite images [2, 26, 57] and from various sensors using IoT (Internet of Things) [66], combining a geoprocessing approach with the use of machine learning tools, such as statistical methods, evolutionary and artificial neural networks, to predict the [57] vector distribution.

The combination of several databases and images for prediction can contribute to having large volumes of data, perhaps difficult to generalize [60]. In this sense, it is also necessary to investigate other machine learning approaches beyond the classical methods and the more usual artificial neural networks. This research will also investigate the effectiveness of using deep neural networks (deep nets), given that the deep learning approach has been shown to be effective in solving several problems [21, 46, 64], in addition to other architectures such as the extreme learning machines (ELMs) [36–38] and the support vector machines (SVMs) [15, 65, 73].

In this work, we propose a methodology for building disease predictors capable of predicting infected cases and locations and detecting the most relevant risk factors based on an architecture that supports multiple databases. As a case study, the prediction of arbovirus cases, infected sites and risk factors was investigated, based on historical series of spatial distributions of climate and environmental information (the distribution of wind speed, temperatures and rainfall) from cases of arboviruses and infected sites, for dynamic prediction through machine learning techniques and risk factor analysis using the Artificial Expert Committee based on bioinspired meta-heuristics (examples: artificial ant colonies, artificial bee colonies, genetic algorithms etc.) for attribute selection. More specifically, to validate the proposal, data provided by the Health Department of the City of Recife, from cases of arboviruses from 2013 to 2016, and climate and environmental information, from the Pernambuco Agency for Water and Climate (APAC) and the National Institute of Meteorology (INMET). The prediction uses information from georeferenced historical series of six cycles (bimesters) to predict the consecutive (bimester) cycle. Cases are grouped into bimonthly periods because this is the way adopted by the Unified Health System to observe arboviruses.

1.2 Related works

According to Siriyasatien et al. [60], new data on dengue is constantly being generated and must be incorporated into existing data to ensure that predictive models have a complete set of new data from which to learn, making predictive models current and relevant. This is also true for Zika and Chikungunya. However, ensuring that new observations are incorporated into the existing body of data is essentially a manual task, which is time-consuming and inconvenient, and may not be comprehensive, resulting in ineffective forecasting models. This is a fundamental issue, to develop automatic data update mechanisms on an ongoing basis. This would optimize the effectiveness and efficiency of the forecasting model. This problem is aggravated by the need to update the forecasting models frequently, which would impose very high overheads (manual import of data from the databases). Infrequent updating can decrease the effectiveness and efficiency of the forecasting model and make the planning and management of vector control policies ineffective. Since manual data collection is a laborious task, it is desirable to automatically collect information using mobile applications and the Internet of Things (IoT).

Cortes et al. [19] used data from 2001 to 2014 to forecast 2015 for the cities of Recife and Goiânia. The data were collected monthly and correspond to the number of dengue cases reported by the National Notification System (SINAN), of the Unified Health System. The prediction problem is approached as the adjustment of two time series using the regression methods ARIMA and SARIMA (Season Auto-Regressive Integrated Moving Average). The results were not considered adequate for the prediction of the cases in Recife. The research lacked consideration of other factors important to the prediction than the case history, such as climatic and environmental variables and other information of interest. In addition, there was no type of refinement that would allow prediction at the level of neighborhoods and districts, since information on the geographic positioning of the cases was not collected.

Buczak et al. [13] built two models to predict dengue outbreaks in the cities of Iquitos, Peru, and San Juán, Puerto Rico, based on sets of ARIMA and SARIMA regressors. Data on dengue cases were collected weekly from the respective public health systems of the two countries, from 2009 to 2010 and from 2012 to 2013. Data on rainfall, temperature and vegetation were collected daily by satellite. Geographic positioning information was not considered. Therefore, the methods proposed by Buczak et al. [13] do not return an approximate spatial distribution, which makes it difficult to use the result in local planning, in this case, at the level of neighborhoods in each city. Buczak et al. [13] also do not analyze the weight of each factor in the prediction.

For Albrieu-Llinás et al. [2], remote sensing systems and geographic information offer valuable tools for mapping the distribution of species in a given area. However, the prediction of species occurrences by means of probability distribution maps based on entomological research[1] cross-cutting has limited utility for local authorities. Albrieu-Llinás et al. [2] aimed to examine the temporal evolution of the number of houses infested with immature stages of Aedes aegypti in each individual neighborhood and to investigate the environmental clusters generated with information provided by variables of remote sensing to explain the behavior observed over time. Entomological surveys were carried out between 2011 and 2013 in the city of Clorinda, Argentina, recording the number of homes with breeding sites with Aedes aegypti larvae. 10,981 houses were visited, chosen at random. Data were organized by neighborhood and collected monthly. Clorinda has 32 neighborhoods. A SPOT 5 satellite image was used to obtain seven land cover variables: bare soil, surface water, wetlands, low vegetation (grass), tall vegetation (shrubs and trees), urban buildings and pastures or crops. These variables were subjected to partitioning using the k-means algorithm for grouping neighborhoods into four environmental clusters. The problem of prediction of sites infected by Aedes aegypti was modeled as a regression problem. A regressor based on a generalized linear model was used. The results were also presented in the form of geospatial distributions, using heat maps, that is, pseudo-color maps, assembled after interpolation of the results. As a tool for visualizing spatial distribution and qualitative analysis, the Quantum GIS geographic information system, or QGIS, was used.

[1] Entomological research is understood as research involving insects and their relationship with humans, with other living beings and with the environment.

The method proposed by Albrieu-Llinás et al. [2] showed great potential for local planning. However, other regression methods could have been tested.

Similar to Albrieu-Llinás et al. [2], Scavuzzo et al. [58] proposed an approach to predict sites infected by eggs and larvae of Aedes aegypti using satellite images and position data from infected breeding sites. The study area was the city of Tartagal, in the province of Salta, northwest of Argentina, close to the border with Bolivia. Breeding data were collected weekly, from August 2012 to July 2016, always covering 50 properties chosen at random. To collect the mosquito's eggs, ovitraps were used, traps made of plastic that attract the females to deposit the eggs and leave them trapped [18]. The climatic and environmental variables were obtained through remote sensing: vegetation index (Normalized Difference Vegetation Index, NDVI) and water and humidity index (Normalized Difference Water Index, NDWI), from the MODIS satellite MOD13Q1; temperature distribution, obtained from JAXA's TRRM satellite (Tropical Rainfall Measuring Mission - NASA / Japan Aerospace Exploration Agency). The prediction problem was modeled as a prediction of time series using regression. Linear regression methods were tested, SVM with RBF kernel, MLP with three hidden layers of three neurons each, k closest neighbors (kNN) and decision trees. As quality indexes, the correlation index and the MSE were used. The best results were obtained with kNN, MLP and SVM, with a correlation of 0.888, 0.875 and 0.837, in this order, against linear regression and decision tree, with 0.774 and 0.679, respectively. The results were not presented in the form of geospatial distributions, since the approach did not use the positioning information of each breeding site. The approach of Scavuzzo et al. [58] is similar to that of Scavuzzo et al. [57], which focus on artificial neural networks.

For Beltrán et al. [7], the devastating consequences of newborns infected with the Zika virus make it necessary to combat and stop the spread of this virus and its vectors: the mosquitoes Aedes aegypti. An essential part of the fight against mosquitoes is the use of mobile technology to support routine surveillance and risk assessment by Endemic Control Agents (ACEs). In addition, to improve early warning systems, public health officials need to more accurately predict where an outbreak of the virus and its vector is likely to occur. The ZIKA system, proposed by Beltrán et al. [7], aims to develop a comprehensive framework that combines e-learning to empower ACEs, and provide community-based participatory surveillance and prediction of occurrences and distribution of the zika virus and its vectors In real time. Currently, this system is being implemented in Brazil, in the cities of Campina Grande, Recife, Jaboatão dos Guararapes and Olinda, in the State of Pernambuco and Paraíba, with the highest prevalence of Zika virus disease. The ZIKA system also aims to help ACEs to learn new techniques and good practices to improve virus surveillance and offer a real-time forecast of the virus and vector. The proposed forecasting model can be recalibrated in real time with information from ACEs, government institutions and weather stations to predict the areas most at risk of an outbreak of Zika virus and other arboviruses transmitted by Aedes aegypti in an interactive map. This mapping and alerting system has the potential to help government institutions make quick decisions and use their resources more efficiently to prevent the spread of the Zika virus. Although they propose the use of Random Forest regressors to make predictions, Beltrán et al. [7] did not carry out experiments and focused on the proposal of the mobile application to support participatory surveillance, the proposal being only a theoretical model.

Zhao et al. [77] developed a national pooled model to predict counts of dengue cases across different departments of Colombia. The authors used the assumption that precipitation, air temperature, and land cover type have been shown to be three important determinants of Aedes mosquito abundance and are often used as predictors in dengue forecasting [6, 22, 29, 58]. Precipitation data was obtained from the CMORPH (Climate

Prediction Center morphing method) daily estimated precipitation dataset [42]. The land surface temperatures were extracted from the MODIS Terra Land Surface Temperature 8-day image products (MOD11C2.006). Enhanced vegetation index (EVI) estimates were obtained from the MODIS Terra Vegetation Indices 16-Day image products (MOD13C1.006). Considering the role of social injustice in epidemics, the authors also included population, education coverage, and the Gini Index (a measure of income inequity) as potential predictors, which were retrieved from the Colombian National Administrative Department of Statistics. The dengue case surveillance data were extracted from an electronic platform, SIVIGILA, created by the Colombia national surveillance program and was available at the department level. The national surveillance program receives weekly reports from all public health facilities that provide services to cases of dengue. the dengue cases reported were a mixture of probable and laboratory confirmed cases without distinguishing between the two different case definitions.

Zhao et al. [77] found that for the majority of Colombia departments, the national model more accurately forecasted future dengue cases at the department level compared to the local model. This indicates the similarity in importance of dengue vectors across different administrative regions of Colombia. Pooling data from individual departments creates a training dataset with larger ranges of variables, increasing the extrapolating capacity of Random Forest models. Results with Random Forests were superior than the ones obtained with MLPs and Deep Convolutional neural Networks. The national pooled model trained by a larger dataset had higher prediction accuracy compared to the local models. The authors also discovered that the meteorological and environmental variables were more important for prediction accuracy at smaller forecasting horizons compared to the socio-demographic variables, with socio-demographics being more important at larger forecasting horizons. Poor quality housing and sanitation management with high population density are key risk factors for dengue transmission, closely related to education and poverty. These results demonstrate the complementary nature of these different groups of predictor variables and the importance of their inclusion in dengue forecasting models.

According to Chakraborty et al. [16], Dengue data sets are neither purely linear nor nonlinear. They usually contain both linear and nonlinear patterns. If this is the case, then the individual ARIMA or Artificial Neural Network (ANN) is not adequate to model this situation. Consequently, the combination of linear and nonlinear models can be well suited for accurately modeling such complex autocorrelation structures. Hybrid ARIMA-ANN models have become more popular due to its capacity to forecast complex time series accurately [76]. Neural Network Auto-Regression (NNAR) corresponds to a feed-forward neural network model with only one hidden layer with a time series with lagged values of the series as inputs. Differently from pure ANNs, SVMs, and LSTMs, NNAR is a nonlinear autoregressive model. Popular hybrid models are hybrid ARIMA-ANN [43, 75], hybrid ARIMA-SVM model [53] and hybrid ARIMA-LSTM [17]. These models try to fit both linear and nonlinear patterns of the time series data.

Chakraborty et al. [16] proposes a hybrid ARIMA-NNAR model to capture complex data structures and linear plus nonlinear behavior of dengue data sets. In the first phase, ARIMA catches the linear patterns of the data set. Then the NNAR model is employed to capture the nonlinear patterns in the data using residual values obtained from the base ARIMA model. Three popular open-access dengue data sets, namely San Juan, Iquitos and the Philippines data are used to determine the effectiveness of the proposed model. Different linear and nonlinear models have been studied on these data sets that shows highly nonlinear patterns in these regions. Mean absolute error (MAE); root mean square error (RMSE) and symmetric Mean Absolute Percent Error (SMAPE) are used as evaluation metrics. For the endemic regions San Juan and Iquitos, weekly laboratory-confirmed cases for the time periods from May 1990 through October 2011 and from July 2000 through December 2011, respectively are considered in this study. The Philippines data set contains the monthly recorded cases of dengue per 100,000 population in the Philippines. Monthly incidence of dengue are available for the time period January 2008 through December 2016. The Philippines monthly data set contains a total of 108 monthly observations and we use the total cases reported from all regions in the Philippines in this study. San Juan weekly data set contains a total of 1144 observations whereas Iquitos data set contains only 520 observations. The authors organized the three dengue datasets into training and test sets. They studied ARIMA, ANN, SVM, LSTM, NNAR model for these data. The data set is divided into two samples of training and testing to assess the forecasting performance of the proposed model. The proposed hybrid ARIMA-NNAR model was able to predict nonlinear tendencies in comparison with the other models, i.e. ARIMA, ANN, SVM, NNAR, LSTM, ARIMA-SVM, ARIMA-ANN, and ARIMA-LSTM.

Stolerman et al. [63] developed machine-learning algorithms to analyze climate time series and their connection to the occurrence of dengue epidemic years for seven Brazilian state capitals. The authors focused on the impact of two key variables: frequency of precipitation and average temperature during a determined range of time windows in the annual cycle. The authors used the publicly available datasets of the Brazilian Notifiable Diseases Information System (SINAN), especially the total number of dengue cases per year, from 2002 to 2017, for all Brazilian state capitals. The authors assumed that the numbers reported were sufficient to identify dengue epidemic years. A year is considered epidemic if, for a given city, the incidence of dengue is above 100 cases per 100,000 inhabitants in the period January-December. To find critical climate signatures, the research was restricted to seven state capitals with at least 3 epidemic years and 3 non-epidemic years in the period 20022012. Climate data used in this work was acquired from the National Institute of Meteorology (INMET), including average temperature time series and precipitation for the following cities: Aracajú, Belo Horizonte, Manaus, Recife, Salvador, and São Luís (from 1/1/2001 to 12/31/2012) and for Rio de Janeiro (from 1/1/2002 to 12/31/2013). Instead of regression, the authors reduced the dengue forecasting problem to temporal classification. They used RBF and linear kernel SVMs. To evaluate their results, they employed accuracy, a metric consistent with a classification approach. The authors used the model trained with data from earlier years 2002-2012 to forecast dengue outcomes from 2013-2017. The state capital of São Luís exhibited the higher accuracy (100% corresponding to 3 correct predictions from a total of 3 test years), followed by Manaus and Salvador (80% accuracy corresponding to 4 correct predictions from a total of 5 test years). For Rio de Janeiro, Aracajú, Belo Horizonte and Recife, the authors reached accuracies below 70%. The authors obtained an overall accuracy of 74% considering all 7 capital cities. Therefore, the proposed method correctly predicted the outcome of 23 out of 31 experiments. Their results indicate that each Brazilian state capital considered has its own climate signatures that correlate with the overall number of human dengue-cases. The immediate winter before an epidemic year is a strong factor in epidemic year predictions. However, the authors recognize that their approach to reduce dengue forecasting to classification could be considered somewhat arbitrary, since it is not the canonical way to forecast arboviruses according to the Brazilian Ministry of Health.

2 Materials and methods

2.1 Proposal

In this work, we propose a methodology for predicting cases of arboviruses, infected sites and risk factors, based on historical series of spatial distributions of climatic and environmental information (the distribution of wind speeds, temperatures and rainfall) of cases of arboviruses and infected sites, for dynamic prediction through machine learning techniques and risk factor analysis using bioinspired meta-heuristic algorithms for attribute selection. The methodology for predicting arboviruses proposed in this work is illustrated in the diagram in Figure 1.

The proposed model can be used for real-time dynamic prediction. Ideally, georeferenced information on wind speed, temperature and rainfall would be collected from sensor networks distributed throughout the city, from the perspective of smart city. These sensors could be connected through ad hoc networks (networks for specific applications) or connected directly to the Internet, in an Internet of Things (IoT) approach, feeding specific databases with the measured quantities (wind speed, temperature or rainfall) and the latitude and longitude positions of each sensor collected by GPS (Global Positioning System). As this is not yet the reality of the City of Recife, as a case study, to validate the proposed model, climate and environmental data from 2009 to 2017 were used, from the Pernambuco Agency for Water and Climate (APAC) and the National Institute of Meteorology (INMET). Data on arboviruses cases could also be collected directly from the database of the eSUS system (Electronic Access to the Unified Health System), where data on the occurrence of arboviruses collected from both public and private health units are registered. However, currently, access to this real-time data is restricted and is provided through open data platforms from time to time for research and development purposes. As a case study, to validate the proposed model, we used data provided by the Health Department of the City of Recife, from 2013 to 2016, through the Open Data Portal of the City of Recife. These data are organized by neighborhood and, through geocoding, receive the latitude and longitude coordinates of the center of the neighborhood. However, geocoding of the simplified patient address could also be used, as there is incomplete address information and sufficiently anonymized, that is, no information that identifies the patients. However, to validate the proposed model and considering that the other databases are not as complete as to the spatial distribution, it was decided to keep the representation by neighborhood.

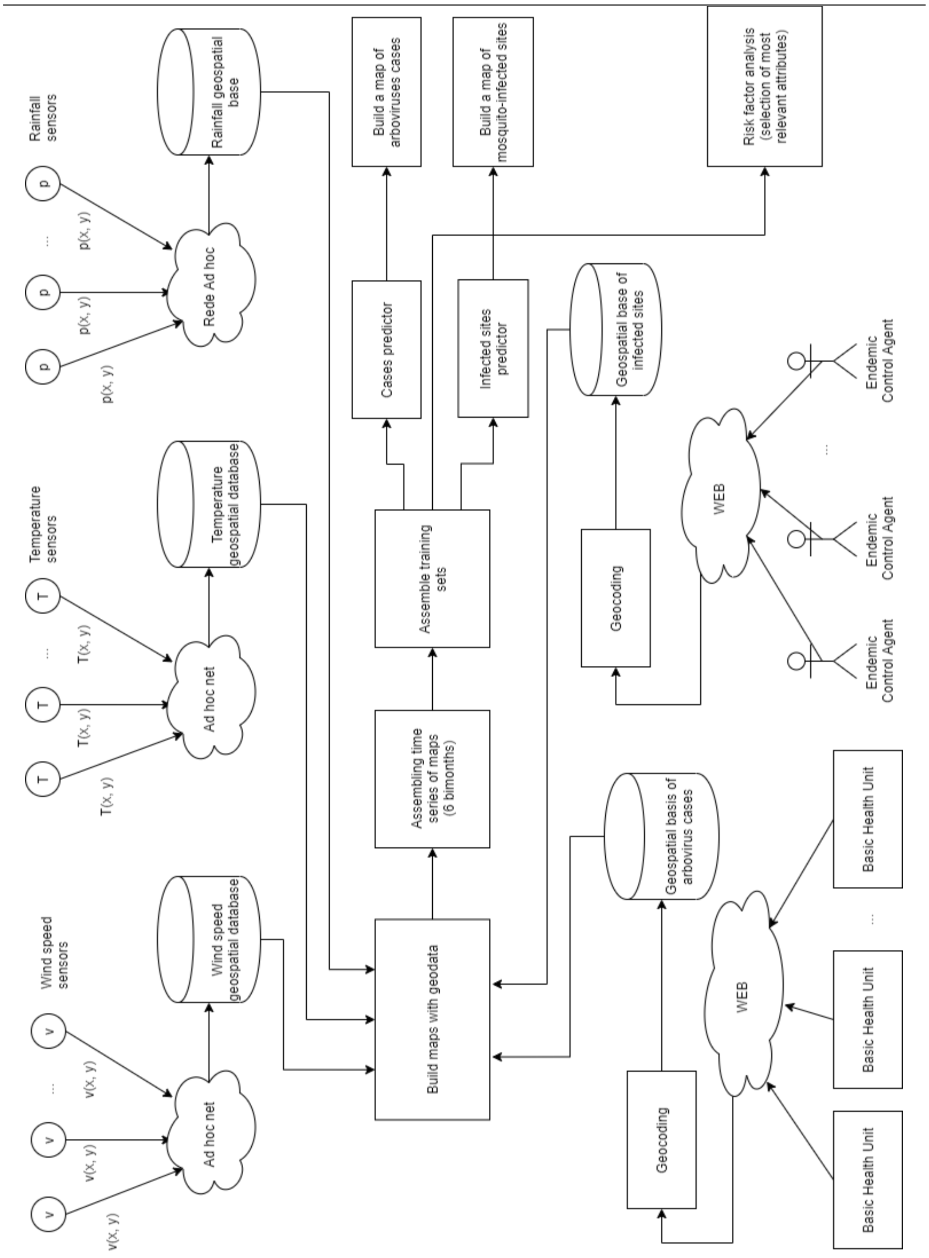


Fig. 1 Proposed methodology for predicting arboviruses: prediction of the spatial distribution of arboviruses cases and infected sites from historical series of climate and environmental information distributions, environmental and clinical surveillance databases

To make a prediction, the monthly accumulated amounts of each of the climatic and environmental variables are collected, that is, temperature, wind speed and rainfall, corresponding to 12 months. For each type of variable, for each month, a map is assembled with the spatial distribution of that variable, that is, an information plan for that month [8]. For data from infected locations, a map is assembled with the spatial distribution of this variable in the two-month period, not the month. Within the scope of SUS, the planning of actions to combat arbovirus outbreaks is carried out considering bimonthly cycles. A similar process is carried out for arbovirus case data. In this approach, we do not separate disease prediction models (dengue, chikungunya and zika). The focus of prevention is on the vector, the mosquito Aedes aegypti [14, 51, 56, 70, 74]. Spatial distributions of arbovirus cases and infected sites for the next two months are predicted. The generated maps must be registered, that is, the *pixels* of each map must correspond exactly to the same spatial position. To assemble the maps by interpolation, we used the free and open geographic information system QGIS[2]. The data is passed to QGIS through spreadsheets, where each geographic position occupies a line and, in the columns, the information of latitude, longitude, and the variables of interest are arranged. The maps are then concatenated.

Training sets are assembled by traversing the pixels. Each pixel corresponds to an instance. The attribute vectors are assembled by scanning the concatenated maps simultaneously, concatenating latitude, longitude, and the following information for each of the six bimesters, in this order: number of cases or number of breeding sites, depending on the type of prediction, if the distribution of cases of arboviruses or infected sites, in that order; temperatures, rainfall and wind speeds, for the first and second month of the bimester, each. The desired output is the number of cases or number of breeding sites, depending on the type of prediction, at the corresponding coordinate.

With the training sets, the best regressor architectures for predicting arbovirus cases and infected sites are investigated. The following architecture families are candidates:

– Support Vector Machine (SVM);
– Random Forest (RF);
– Multilayer Perceptron (MLP);
– Extreme Learning Machine (ELM);
– Echo State Machine (ESM); – Deep Echo State Network (Deep-ESN); – Linear Regression (LR).

Each regressor was evaluated in 30 rounds using 10-fold cross validation. For quantitative evaluation, the following prediction quality metrics were calculated: correlation coefficient (R), Kendall's τ (KE), Spearman's ρ (SP), mean absolute error (MAE), o root mean square error (RMSE), relative absolute error (percent MAE), and relative squared error (percent RMSE). The results were also qualitatively evaluated through the extrapolation of the trained models for a given test set, and the results were interpolated for visualization in the form of a map and compared to the real distribution. For quantitative validation, we used the Weka library, the QGIS geographic information system and specific programs implemented in Python and Octave. For qualitative validation, to generate maps with the results, we use Quantum GIS.

The most relevant factors for each prediction were evaluated using as a strategy hybridized attribute selection methods with bioinspired optimization methods. Selecting attributes returns the relevance of each attribute to the prediction and thus highlights the most relevant variables. The simple ranking method was hybridized, which in turn is based on a decision tree as a reference classifier/regressor. The following methods were investigated, considering a maximum of 500 generations and initial populations of size 20:

[2] QGIS, a Free and Open Geographic Information System, available at https://www.qgis.org/pt_BR/site/, accessed on June 25, 2021.

- Genetic Algorithm (GA);
- Evolutionary Search (Modified Genetic Algorithm, ES);
- Particle Swarm Optimization (PSO); – Artificial Bee Colony (ABC); – Artificial Ant Colony (AAC).

These methods are part of an Artificial Experts Committee to select the most relevant factors for prediction. The proposed architecture is shown in the diagram in Figure 2.

2.2 Area under study

The City of Recife is the capital of the State of Pernambuco, located in the Northeast Region of Brazil, latitude and longitude -8.053889 and -34.880833, respectively (see Figure 3). With a land area of approximately 218 km^2, it occupies a mostly flat territory, formed by hills, islands, peninsulas and mangroves. According to the Brazilian Institute of Geography and Statistics - IBGE[3], Recife is the northeastern city with the best Human Development Index (HDI-M): 0.772, which is considered high. It is the fourth most important Brazilian capital, after Brasília, Rio de Janeiro and São Paulo, and has the fourth most populous urban agglomeration in Brazil, with 4 million inhabitants in 2017, surpassed only by the metropolitan regions of São Paulo, Rio de Janeiro and Belo Horizonte. Recife is home to the richest urban agglomeration in the North-Northeast and the eighth richest in Brazil, in addition to having the fourteenth highest GDP in the country and the highest GDP per capita among the northeastern capitals. The city is the ninth most populous in the country, with 1,637,834 inhabitants, and its metropolitan region is the seventh in Brazil in population.

The city presents 69.2% of households with adequate sanitation, 60.5% of urban households on public streets with trees, and 49.6% of urban households on public streets with adequate urbanization (presence of manhole, sidewalk, paving and curb). When compared to other municipalities in the state, it is in position 20 out of 185, 107 out of 185 and 1 out of 185, respectively. When compared to other cities in Brazil, its position is 1415 out of 5570, 3654 out of 5570 and 444 out of 5570, in that order.

The City of Recife was chosen for the case study of this work not only because of the availability of data, but mainly because of the various arbovirus outbreaks that have already happened: in a more endemic, although less serious form, dengue, in its various forms, including hemorrhagic; and the outbreaks of chikungunya fever and zika virus in 2015, the latter responsible for the occurrence of thousands of cases of microcephaly in newborns in 2015 and 2016 [9, 19, 27, 49]. This attests to the social impact of the dynamic prediction of arboviruses.

[3] Brazilian Institute of Geography and Statistics, IBGE, Recife City Indicators, available at https://cidades.ibge.gov.br/brasil/pe/recife/panorama, accessed on June 28, 2021.

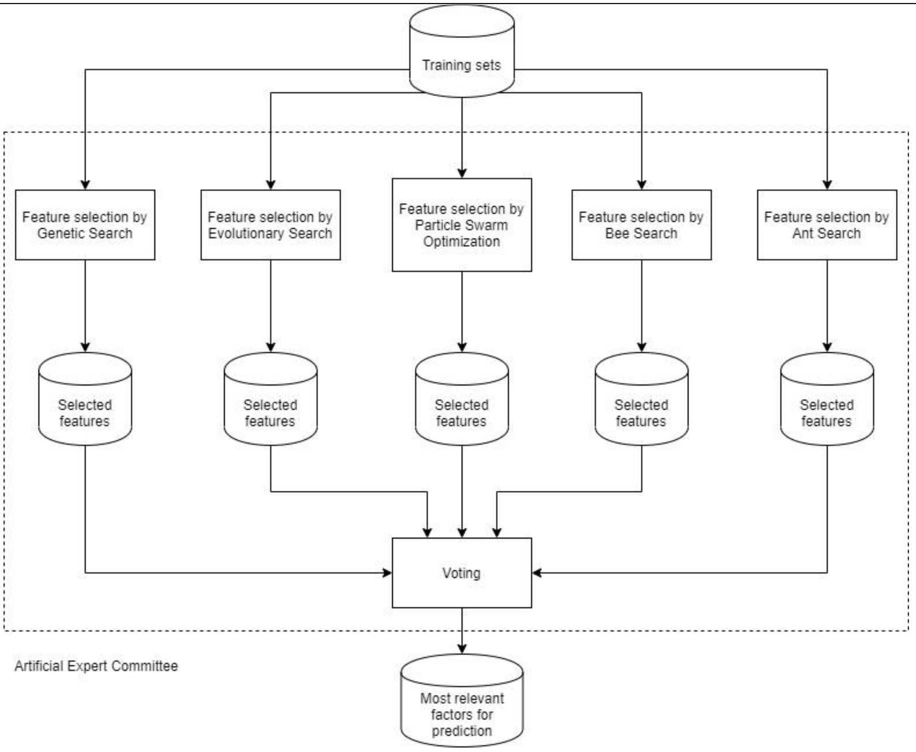

Training sets
Feature selection by Genetic Search
Feature selection by Evolutionary Search
Feature selection by Particle Swarm Optimization
Feature selection by Bee Search
Feature selection by Ant Search
Selected features
Selected features
Selected features
Selected features
Selected features
Voting
Artificial Expert Committee
Most relevant factors for prediction

Fig. 2 Artificial Expert Committee for attribute selection, where each expert is based on a bio-inspired heuristic search algorithm: genetic algorithm (genetic search), modified genetic algorithm (evolutionary search), particle swarm optimization, artificial bee colony (search for bees), and colony of artificial ants (search for ants), for initial populations of 20 individuals and a maximum of 500 generations or iterations.

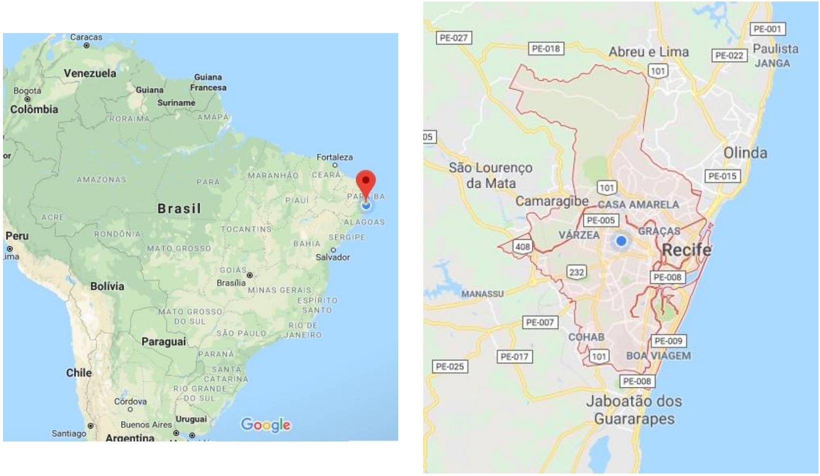


Fig. 3 Profile and geographic location of the City of Recife, capital of the State of Pernambuco, Brazil, latitude -8.053889 and longitude -34.880833.

2.3 Datasets

2.3.1 Geographic Information System SIGH-PE

The Pernambuco Water and Climate Agency, APAC, is a public agency in the State of Pernambuco, created by State Law 14,028 of March 26, 2010 to implement the State Water Resources Policy, to complement the Integrated Water Resources Management System in Pernambuco, SIGRH-PE, and strengthen the planning and regulation of multiple uses of water resources in the State.

APAC performs real-time hydrometeorological monitoring of fluviometric and pluviometric stations through PCDs (Data Collection Platforms) distributed in the region of the State of Pernambuco. Rainfall stations monitor rainfall in millimeters (mm). Fluviometric stations monitor the state of the dams, the quality of the water in the reservoirs, and the hydrographic basins of the State. The information is stored in the Geographic Information System SIGH-PE[4], whose interface can be seen in Figure 4, with the distribution of Data Collection Platforms, PCDs, throughout the State of Pernambuco. Rainfall and fluviometric data are monitored daily. Data can be exported to spreadsheets, manually selecting the PCD of interest and the time interval, with start and end date of data collection. Data is distributed by day or by monthly accrual.

From this base, we chose to use the monthly accumulated rainfall from 2009 to 2017, as the prediction is made using cycles of six bimonths. The coordinates used are the latitude and longitude of the monitoring stations: Codecipe / Santo Amaro, Várzea and Alto da Brasileira.

2.3.2 Meteorological Database for Teaching and Research - BDMEP

The National Institute of Meteorology (INMET) represents Brazil at the World Meteorological Organization (WMO). It is responsible for the traffic of messages collected by the South American meteorological observation network and the other meteorological centers that make up the World Meteorological Surveillance System. INMET is home to a Geographic Information System Center (GISC), part of the main core of the new World Meteorological Organization Information System (WIS), the result of the evolution of the Global Telecommunication System (GTS).

The Institute’s Meteorological Data Collection and Distribution System (temperature, relative humidity, wind direction and speed, atmospheric pressure, precipitation, among other variables) is equipped with upper air sounding stations (radiosonde); surface weather stations, manually operated; and the largest network of automatic stations in South America.

The Meteorological Database for Teaching and Research[5], BDMEP, is a database to support teaching and research activities and other applications in meteorology, hydrology, water resources, public health and the environment. The database houses daily meteorological data in digital form, from historical series of the various conventional meteorological stations in the network of INMET stations with millions of information,

[4] Geographic Information System of the Pernambuco Water and Climate Agency - SIGH-PE, available at http://www.apac.pe.gov.br/sighpe

[5] National Institute of Meteorology - INMET, available at http://www.inmet.gov.br/portal/index.php?r=bdmep/bdmep, accessed on June 25, 2021.

referring to daily measurements, in accordance with the international technical standards of the World Meteorological Organization.

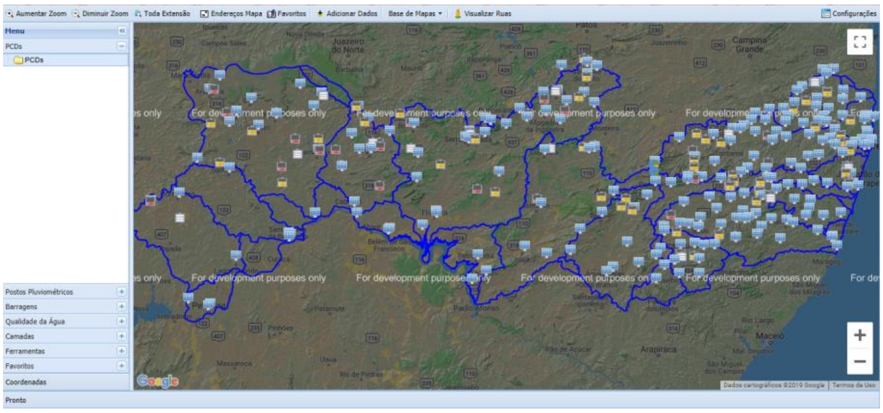


Fig. 4 APAC Geographic Information System Interface, SIGH-PE, with the distribution of Data Collection Platforms, PCDs, throughout the State of Pernambuco. Rainfall data (precipitation in mm) and fluviometric data are monitored daily.

The atmospheric variables available for consultation in the BDMEP are: precipitation occurred in the last 24 hours; dry bulb temperature; wet bulb temperature; maximum temperature; minimum temperature; relative humidity; atmospheric pressure at station level; insolation; wind direction and speed. From this base, it was decided to use in this work the average temperatures and wind speeds per month. The coordinates used are the latitude and longitude of the monitoring stations: Codecipe / Santo Amaro, Várzea and Alto da Brasileira.

2.3.3 Open Data Portal of the City of Recife

The Open Data Portal of the City of Recife[6] was developed by EMPREL, Municipal Informatics Company. It aims to make publicly available access and search for government data generated by departments and municipal management bodies. The publication of data in an open format allows applications or visualizations to be developed, seeking to facilitate data analysis, promote the improvement of services through innovation and creativity, and contribute to a greater participation of society with the municipal government. Data is available in CSV and PDF formats. It is also possible to do direct searches in databases using SQL queries and the JSON protocol.

The Municipal Health Department makes available 11 (eleven) data sets, including data on the Health Districts, Health Surveillance, Mobile Emergency Care Service - SAMU, Health Units, City Academies, and records of dengue cases, zika and chikungunya registered in public and private health units. Case record data are daily and range from 2013 to 2016.

The data contains incomplete patient address information for anonymization purposes. This information can be geocoded to obtain latitude and longitude. Geocoding was partially

[6] Open Data Portal of the City of Recife, available at http://dados.recife.pe.gov.br/dataset/casos-dedengue-zika-e-chikungunya, accessed June 25, 2021.

done using a Google Maps plugin. Some points could not be geocoded, due to errors and incomplete information in the registers. However, as the resolution of climate and environmental variables has low resolution, due to the low density of points (only three points of temperature, rainfall and wind speed per month), we chose to group this case information by neighborhood and use the latitude and longitude coordinates of the central point of the neighborhood. Since the prediction of arboviruses is made through bimonthly collections, the total number of cases in each bimonthly period was used, starting in January.

2.4 Regression models

2.4.1 Linear Regression

The linear regression is the simplest method to predict numeric values. In this method, it is assumed that the data has a linear behavior, and that the prediction variable can be represented as a linear combination of the attributes with their pre-determined weights [71]. Thus, the general model of linear regression is represented by the Equation 1.

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n, \tag{1}$$

where y is the prediction variable; $x_1, x_2,..., x_n$, represent the values of the attributes and w_0, $w_1, w_2,..., w_n$ represent the weights of each attribute. The idea of the linear regression algorithm is, then, to find the optimal weights that best represent the problem. One of the ways to find the optimal weights is to minimize the sum of the squared difference between the predicted value and the actual value [71]. The sum of the squared difference is calculated by Equation 2:

$$S = \sum_{i=1}^{k} \left[y^{(i)} - \sum_{j=0}^{n} w_j x_j^{(i)} \right]^2 . \tag{2}$$

2.4.2 Artificial Neural Networks

Artificial neural networks (ANN), consists in a machine learning technique based on the behavior of the human brain [59]. The neural networks consist of smaller units, artificial neurons, which are fundamental to their functioning. The artificial neurons contains the following elements: (1.) a set of *synapses* or *connectors* - where a signal x_i at the entrance to the synapse j connected to the k neuron is multiplied by the synaptic weight $w_{k,j}$ (2.) an *adder* to add the input signals, weighted by the respective neuron synapses; (3.) an activation function to limit the output of a neuron [32]. Mathematically, an artificial neuron is represented by the Equation 3 and by the Equation 4:

$$u_k = \sum_{j=1}^{n} w_{k,j} x_i, \tag{3}$$

$$y_k = \varphi(u_k + b_k), \tag{4}$$

wherein x_1, x_2, ..., x_n represent the input signals; $w_{k,1}$, $w_{k,2}$, ..., $w_{k,n}$ represent the synaptic weights of the input signals x_i for the k-th neuron; b_k, is the term bias and φ is a neuron activation function. In regression applications, the inputs x_1, x_2, ..., x_n of the input layer correspond to the forecasting window. For instance, in case of temporal forecasting, the inputs are observed time window of the time-series.

The network architecture used in this work was the Multilayer Perceptron (MLP). In this configuration, the neural network has an input layer, two or more hidden layers and an output layer [32]. ANNs have also been widely used to predict disease cases. For example, in the prediction of dengue cases in the city of São Paulo, Brazil [3]. They were also used to predict dengue outbreaks in the northeastern coast of Yucatán, Mexico, and in San Juan, Puerto Rico [45]. Moreover, the ANNs were applied to model cases of infection by Salmonella in the state of Mississippi, USA [1].

2.4.3 Support Vector Regression

The support vector regression is a supervised machine learning technique for data analysis and pattern recognition. The idea of the SVR algorithm is to find the best hyperplane defined by Vapnik's ε-insensitivity loss function. When this hyperplane is found, a linear regression is applied to the corresponding hyperplane. In situations where the problem is linearly separable, the best hyperplane is given by the equation:

$$y = \mathbf{w}^T\mathbf{x} + b, \tag{5}$$

where $\mathbf{w} = (w_1, w_2, ..., w_n)^T$ is the vector of weights, $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ is the feature vector, and b is the bias. For problems that are not linearly separable, the data is mapped to a hyperplane in a larger dimension. Thereupon, the algorithm seeks to solve the problem by applying the linear regression of the equation 5 in the corresponding hyperplane. For nonlinearly separable problems, SVR machines use kernel functions, $K : \mathrm{R} \times \mathrm{R} \rightarrow \mathrm{R}$. Then, the SVR output assumes the following expression:

$$y = K(\mathbf{w}, \mathbf{x}), \tag{6}$$

where the kernel function can be polynomial, sigmoidal, Gaussian, or even assume other mathematical expressions [23, 61, 71].

2.4.4 Extreme Learning Machines

Extreme Learning Machines (ELMs) emerged as an alternative learning method for Single Layer Feed-forward Networks, (SLFNs), that usually relies on the Backpropagation algorithm for classification or regression tasks. Created by [36], ELMs consist of SLFNs, in which the weights and biases between the input and hidden layer are randomly set, while only the output weights are analytically determined, as a linear system, using the Moore-Penrose generalized inverse [4]. The reliability of ELMs as an alternative for SLFNs is found on the fact that the weights and biases between the input and hidden layer of an SLFN do not need to be tuned, if we guarantee that its activation functions are infinitely differentiable [37], and that an SLFN with a hidden layer with sufficient neurons N and with almost any activation function will be able to learn N distinct observations [35].

ELMs also provide other desired characteristics for real applications, including a learning speed that can be greater than conventional feed-forward networks, due to the one-shot training phase; the tendency of achieving a better generalization performance, small training errors and the smallest norm of weights [37] (according to Bartlett [5], the smaller the norm of weights, the better generalization performance the network tends to achieve).

The proposed ELM algorithm by [37], is as follows:

Suppose a training set:

$$\psi_T = \{(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in \mathrm{R}^n, \mathbf{t}_i \in \mathrm{R}^m\}_{i=1}^{N_T},$$

an activation function $g : \mathrm{R} \rightarrow \mathrm{R}$ and a number N of hidden nodes, do

1. Randomly, assign input weights and bias, $\mathbf{w}_i$ and b_i, respectively, to the input neurons $i = 1, ..., N$;
2. Calculate the hidden layer output matrix $\mathbf{H}$, with entries $H_{ij} = g(w_j x_i + b_j)$;
3. Calculate the ouput weight β, from $\beta = \mathbf{H}^\dagger\mathbf{O}$, where $\mathbf{O} = [o_1, ..., o_N]^T$ is the desired output of the training dataset.

2.4.5 Echo State Networks

Jaeger [40] and Maass et al. [48] marked the birth of Reservoir Computing. The proposed techniques consist of a novel approach for training and using recurrent neural networks, and are named Echo State Networks (ESN) and Liquid State Machines (LSM). Later, a new learning rule named Back-Propagation De-Correlation emerged from a completely different background, but presenting very similar concepts if compared with both previous approaches [69].

The ease of use and excellent performance of those techniques grew the interest of academy for using them, and research on this field gained momentum. In this way, ESN and LSM were applied in several applications, as explained in the following paragraphs.

Jaeger [40] applied ESN to dynamic pattern classification, testing the network for periodic sequence learning, using a target signal prepared from the melody “The Housing of The Rising Sun” and switchable point attractor learning. The network, when properly tuned, showed sufficient capability for generating stable periodic sequences of sounds robust to noise.

Ghani et al. [28] also showed the ESN performance when applied to speech recognition. The network performance was compared to normal feed-forward neural networks, and the results suggested that the reservoir, allied to feed-forward neural networks showed better classification rates than only Neural Networks. Very good results were also achieved when a pre-processing step was added to the classification task.

The proposed approach of ESN consists of using Recurrent Neural Network (RNN) with topology and weights randomly set, where the network is driven by its external input and then the obtained response is used to train a linear regression or classification function [69], where only the readout layer is trained.

ESNs are, however, not fully understood and still have some parameters to be tuned or to be found. Despite this, they often present good performance, managed by the Echo State property (ESP), which states that the networks should asymptotically forget its initial state when driven by an external signal [69]. Furthermore, given the ESP, ESNs should slowly forget the initial inputs when a new one comes, causing the *fading memory*. This property also allows the reminiscence of echoes, enriching the set of nonlinear transformations and mixings of the current and previous signals, inside its reservoir.

Some desired performance characteristics, achieved by RC, are the nonlinear extension of the input data to other domains, where the classification task can be done more accurately. This enables the use of a relatively simple structure and computationally undemanding linear classification or regression algorithms. Also, because reservoirs are created randomly, they tolerate some internal variation in their parameters, without significantly compromising their performance.

The proposed ESN, initially designed by [40], and also detailed in [47], can be summarized as follows:

1. Generate a random reservoir of Recurrent Neural Networks, with random input weights (W^{in}), random recurrent matrices weights (W), and a random leaking rate α;
2. Run the topology using the training input, u(n), and collecting the correspondent results x(n);
3. Compute the linear readout weights, W^{out}, from the reservoir, using linear regression and minimizing the error between the obtained output, $y(n)$ and the desired result, $y^{desired}(n)$;
4. Use the trained network on new input data, computing the obtained output $y(n)$ by employing the output weights W^{out}.

2.4.6 Random Forest

Random Forests are based on decision tree committees organized in bagging [33]. Decision trees separate data iteratively, testing one property at a time. The resulting sheets represent the most specific category. The root represents the raw data. The random forest is built with many of these trees, all with their own class prediction for any input provided. The most voted class is the departure of Random Forest. Random Forests have been used to solve a plethora of biomedical problems, specially to develop intelligent systems to support diagnosis [20, 24, 25].

As the most relevant characteristics to determine the decision boundary between classes of virus DNA sequences are unknown, Random Forests can be powerful methods for classification, as they are able to verify many relevant properties through their different trees. In the bagging process, each tree receives a version of the training set with a reduced number of attributes. Thus, it is possible to build decision criteria that take into account only a few attributes and these criteria can be winners in the vote, determining the final decision of the classifier.

2.5 Metrics

The main metrics we adopted to evaluate the models are the following: the correlation coefficient and the Relative Quadratic Error (RMSE percentage). The correlation coefficient is a statistical measure between expected and forecasted values. This value varies from -1 to 1. When it approaches 1, it indicates a strong positive correlation. Conversely, when the correlation coefficient is close to -1, it indicates that the variables have a strong negative correlation. When the correlation coefficient is close to zero, it indicates that there is no correlation between the variables [71]. The value of the correlation coefficient serves as the global evaluator for the model. Therefore, it is possible to obtain a high correlation coefficient as well as at the same time obtain high values for local errors. For this reason, it cannot be the only metric for assessing model performance. In order to avoid a superficial evaluation of the regressors, we therefore chose the RMSE (%) as an evaluation metric. The Equation 7 shows the expression of the calculation of the relative quadratic error, where p_i is the predicted value and a_i is the actual value, for $i = 1, 2, ..., n$.

$$\mathrm{RMSE}(\%) = \sqrt{\frac{\sum_{i=1}^{n}(p_i - a_i)^2}{\sum_{i=1}^{n} a_i^2}} \times 100\%. \tag{7}$$

In addition to the RMSE (%), we also calculated the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), the Mean Absolute Percentage Error (MAPE) and the Mean Percentage Error (MPE) (Equations 8-11):

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2}, \tag{8}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |e_i|, \tag{9}$$

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n} \left|\frac{e_i}{a_i}\right|, \tag{10}$$

$$\mathrm{MPE} = \frac{100\%}{n}\sum_{i=1}^{n} \left(\frac{p_i - a_i}{p_i}\right), \tag{11}$$

where, p_i is the forecasted value, a_i is the actual value and $e_i = a_i - p_i$ is the difference between the actual value and the forecasted value.

The Pearson's Correlation Coefficient R is defined as follows:

$$R = \frac{\sum_{i=1}^{n}(p_i - \bar{p})(a_i - \bar{a})}{\sqrt{\sum_{i=1}^{n}(p_i - \bar{p})^2 \cdot \sum_{i=1}^{n}(a_i - \bar{a})^2}}, \tag{12}$$

where $\bar{p}$ and $\bar{a}$ are the sample average values for the sets of predicted and actual values, respectively.

Similarly, the Spearman's Rank Correlation Coefficient ρ is defined as following:

$$\rho = \frac{\sum_{i=1}^{n}(R(p_i) - \bar{R}(p))(R(a_i) - \bar{R}(a))}{\sqrt{\sum_{i=1}^{n}(R(p_i) - \bar{R}(p))^2 \cdot \sum_{i=1}^{n}(R(a_i) - \bar{R}(a))^2}}, \tag{13}$$

where $R(p_i)$ and $R(a_i)$ are the ranks of p_i and a_i, whilst $\bar{R}(p)$ and $\bar{R}(a)$ are the sample averages of the ranks of p_i and a_i, respectively.

The Kendall's Rank Correlation τ is given as follows:

$$\tau = \frac{2}{n(n-1)}\sum_{j=1}^{n}\sum_{i=1}^{j-1} \mathrm{sign}(p_i - p_j) \cdot \mathrm{sign}(a_i - a_j), \tag{14}$$

where n is the number of observations and $1 \le i,j \le n$. The signal function, sign, is defined as following:

$$\operatorname{sign}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0, \\ -1, & x < 0 \end{cases}$$

for $x \in$ R.

2.6 Experimental arrangement

The experiments were carried out on a personal computer, Intel I7 processor, 16GB of RAM, Windows 10 operating system. The fusion of databases and construction of knowledge bases was carried out using a program built in Python specifically for this function [52]. The knowledge bases were saved in a text file, in ARFF format. The regression experiments were mostly carried out in the Weka machine learning and data mining environment, except for the ELM, ESM and Deep-ESM methods. The Weka environment was built using the Java language, and can be extended by adding other libraries and plugins [30, 34, 72]. The ELM, ESM and Deep-ESM methods were tested using programs built in Matlab/Octave language, in the GNU Octave environment. Raw results were stored in MS Excel spreadsheets. The statistical graphs in boxplots were built using the SciDavis program, aimed at building scientific graphs [62]. The maps were built using the Quantum GIS geographic information system [39, 55].

3 **Results**

3.1 Forecasting model

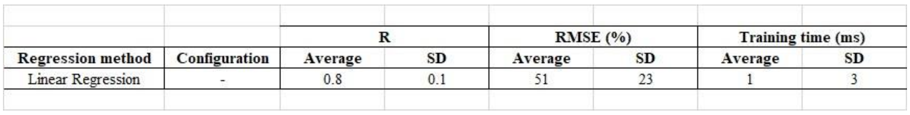
The predictor model of arboviruses cases proposed in this work is based on the spatiotemporal analysis, combining information on climatic and environmental variables from the monthly accumulations, specifically temperature, rainfall and wind speeds, with the bimonthly data on the number of cases. We consider a cycle of six bimonths, that is, one year, the cycle of the Aedes aegypti mosquito. All this information is related to latitude and longitude coordinates, and the prediction result is also associated with geographic positioning. Thus, the attribute vectors have a total of 44 attributes. Data were collected from the respective databases considering a simplified distribution by neighborhood, considering all 94 neighborhoods of the City of Recife. Each neighborhood is associated with a central geographic position, that is, a pair of latitude and longitude. In order to investigate the best regressor architectures, approximate distributions were generated for each variable of interest (attribute) using an irregular grid, generating bases of 1786 vector instances of 44 attributes. Using data from the years 2013, 2014, 2015 and 2016 and considering that the cycle considered for prediction is six bimonths, that is, one year, each of the six bimonths of 2014, 2015 and 2016 are predicted considering the geospatial information of the respective previous years. This process resulted in 18 knowledge bases.

Each regressor was evaluated in 30 rounds using 10-fold cross validation. Thus, for the general evaluation of the prediction, considering that there are a total of 18 bases, each regressor was evaluated in $30 \times 10 \times 18 = 5400$ training and tests. For quantitative evaluation, the following prediction quality metrics were calculated: Pearson’s (R), Kendall’s τ (KE), and Spearman's ρ (SP) correlation coefficients, the mean absolute error (MAE), the mean square error (RMSE), relative absolute error (percent MAE) and relative squared error (percent RMSE). However, in the data analysis, only R was considered as a global quality metric, and RMSE (%) as a local quality metric. The following regressor configurations were evaluated, where Linear Regression (LR) is considered the standard regressor in environmental applications of spatio-temporal analysis:

- Support Vector Machine (SVM): $C = 0.1$, linear kernels (or degree 1), degree 2 and 3 polynomial, and RBF;
- Random Forest (RF): from 10 to 100 trees, increasing the number of trees from 10 to 10;
- Multilayer Perceptron (MLP) with a single hidden layer with 2, 5, 10 and 20 neurons;
- Extreme Learning Machine (ELM): single hidden layer, 50 and 100 neurons, testing linear *kernels*, degree 2 and 3 polynomial, and RBF;
- Echo State Machine (ESM): with a single hidden layer, and deep version with 2, 5 and 10 layers, 10 neurons per layer;
- Linear Regression (LR).

The Correlation Coefficient R, a quantity ranging from 0 (total decorrelation) to 1 (total correlation), was considered to analyze the global behavior of each prediction, that is, how a given regressor method behaves in the most general plane. The R metric is considered an optimistic metric: the result can look very good, something greater than 0.8, for example, and local, punctual errors can be high. Therefore, R cannot be considered alone. To compensate for this, we also chose to look at the RMSE (%) metric, which serves as a measure of local error. The closer to 0% the better. Thus, a regressor is considered good if it has an R considered high and an RMSE (%) considered low. This work considers a high R to be above 0.9, and a low RMSE (%) to be below 5%. Training time was measured in milliseconds (thousandths of seconds). Training times less than 1 millisecond are considered 0. Even though training time is not critical for this proposal, this large one was still measured, to help decide which regressor configurations are better within the same family of regression methods (for example: MLP neural networks with different numbers of neurons in the hidden layer).

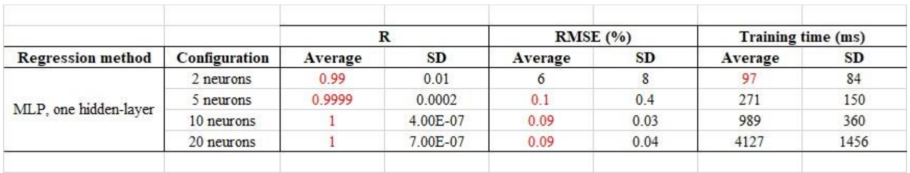
Detailed results of R, RMSE (%) and training time (ms) are displayed in Tables 1, 2, 3, 4, 5 and 6. In Table 1 the results for linear regression are presented. Table 2 presents the results for MLP neural network, a single hidden layer, for 2, 5, 10, and 20 neurons in the hidden layer. In Table 3 are shown the results for SVM, *kernel* polynomial degree 1 (linear), 2, 3, and RBF. Table 4 shows the results for ELM neural network, a single hidden layer with automatically determined number of neurons, kernel polynomial degree 1 (linear), 2, 3, and RBF. In Table 5 the results for neural networks ESM (a single hidden layer with 10 neurons) and Deep-ESM (5 and 10 hidden layers, 10 neurons per layer) are displayed. Finally, in Table 6 are presented the results for Random Forest, from 10 to 100 trees, varying every 10. The best results on average are highlighted in red.



		R		RMSE (%)		Training time (ms)	
Regression method	Configuration	Average	SD	Average	SD	Average	SD
Linear Regression	-	0.8	0.1	51	23	1	3

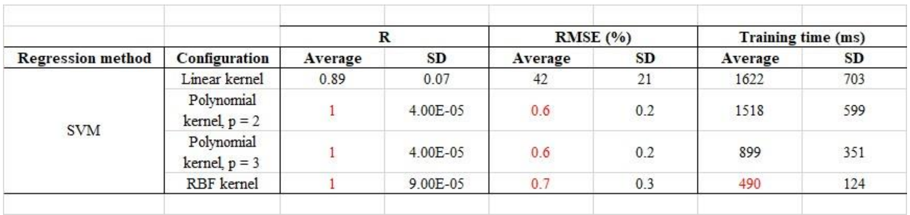
Table 1 Results of R, RMSE (%) and training time (ms), sample mean and standard deviation (SD), for linear regression.

Figures 5, 6 and 7 show the statistical distributions in boxplots of the Correlation Coefficient (R), RMSE (%), and time of model training in milliseconds, respectively, for linear regression, multilayer perceptron artificial neural networks (MLP), extreme learning machines (ELM), single-layer (ESM) and deep echo state machines (Deep ESM), echo-state machines. support vector machines (SVM), and Random Forests.



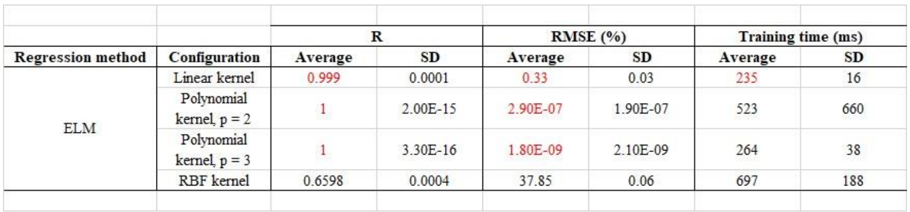
		R		RMSE (%)		Training time (ms)	
Regression method	Configuration	Average	SD	Average	SD	Average	SD
MLP, one hidden-layer	2 neurons	0.99	0.01	6	8	97	84
	5 neurons	0.9999	0.0002	0.1	0.4	271	150
	10 neurons	1	4.00E-07	0.09	0.03	989	360
	20 neurons	1	7.00E-07	0.09	0.04	4127	1456

Table 2 Results of R, RMSE (%) and training time (ms), sample mean and standard deviation (SD), for MLP neural network, a single hidden layer, for 2, 5, 10 and 20 neurons in the hidden layer. The best results on average are highlighted in red.



		R		RMSE (%)		Training time (ms)	
Regression method	Configuration	Average	SD	Average	SD	Average	SD
SVM	Linear kernel	0.89	0.07	42	21	1622	703
	Polynomial kernel, p = 2	1	4.00E-05	0.6	0.2	1518	599
	Polynomial kernel, p = 3	1	4.00E-05	0.6	0.2	899	351
	RBF kernel	1	9.00E-05	0.7	0.3	490	124

Table 3 Results of R, RMSE (%) and training time (ms), sample mean and standard deviation (SD), for SVM, polynomial kernel degree 1 (linear), 2, 3 and RBF. The best results on average are highlighted in red.



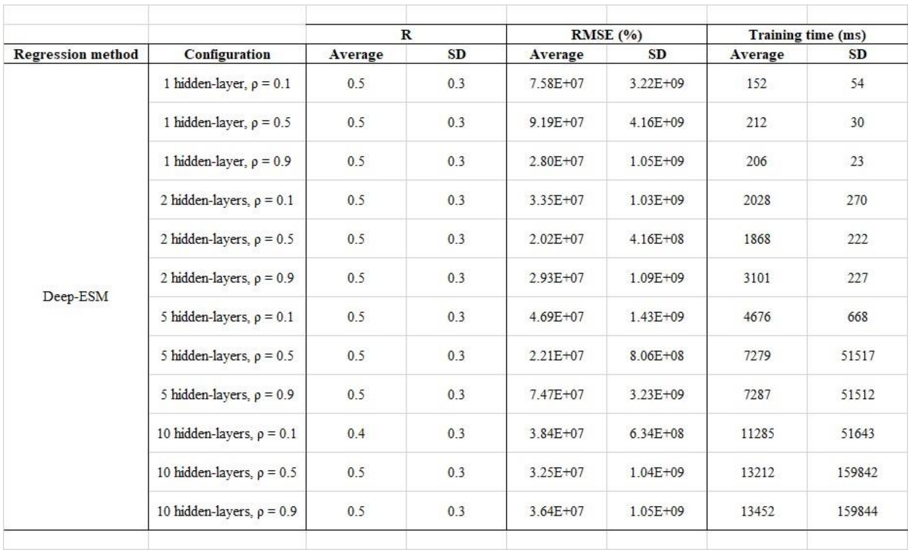
		R		RMSE (%)		Training time (ms)	
Regression method	Configuration	Average	SD	Average	SD	Average	SD
ELM	Linear kernel	0.999	0.0001	0.33	0.03	235	16
	Polynomial kernel, p = 2	1	2.00E-15	2.90E-07	1.90E-07	523	660
	Polynomial kernel, p = 3	1	3.30E-16	1.80E-09	2.10E-09	264	38
	RBF kernel	0.6598	0.0004	37.85	0.06	697	188

Table 4 Results of R, RMSE (%) and training time (ms), sample mean and standard deviation (SD), for ELM neural network, single hidden layer, automatically determined number of neurons in hidden layer, polynomial kernel degree 1 (linear), 2, 3 and RBF. The best results on average are highlighted in red.

3.2 Most relevant features selected by the Artificial Expert Committee

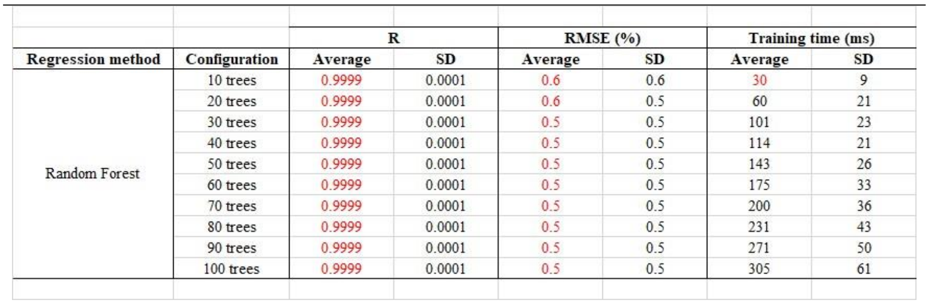
To analyze the most relevant factors for prediction, five bioinspired heuristic search methods were used to select attributes. These methods used as objective function a decision tree as a regressor, evaluated using cross-validation with 10 folds. Each method returned the most relevant attributes and their degree of relevance in percentage, from 0% to 100%. An attribute was considered relevant when its degree of relevance was different from 0%, typically at least 10% in the experiments performed in this work. These five methods formed the Artificial Expert Committee, with the final outcome of relevance decided by voting. Thus, an attribute is considered relevant if it was considered relevant by the simple majority of the artificial expert committee members. All meta-heuristic algorithms were run considering initial populations of candidates for the solution of 20 individuals and a total of 500 iterations or generations. In these experiments, we use the Weka library. The settings used were as follows:

- Genetic Algorithm (or Genetic Search, GS): crossover probability 0.6, mutation probability 0.033, communication frequency 20, 20 individuals, 500 generations;
- Modified Genetic Algorithm (Evolutionary Search, ES): crossover probability 0.6, mutation probability 0.1, bit-flip mutation, generational substitution (children replace parents in the next generation), selection of individuals per tournament, communication frequency of 20, 20 individuals, 500 generations;
- Optimization by Particle Swarm (PSO Search): individual weight 0.34, inertia factor 0.33, social weight 0.33, mutation probability 0.01, communication frequency 20, 20 particles, 500 iterations;
- Artificial Bee Colony (or Bee Search, BS): chaotic coefficient 4.0, chaotic type logistic map, mutation probability 0.01, bit-flip mutation, humidity radius 0.98, mutation radius 0.8, communication frequency 20, 20 bees, 500 iterations;
- Artificial Ant Colony (or Ant Search, AS): chaotic coefficient 4.0, chaotic type logistic map, evaporation 0.9, heuristic 0.7, mutation probability 0.01, mutation type bit-flip, target merits, 2.0 pheromone, 20 communication frequency, 20 ants, 500 iterations.



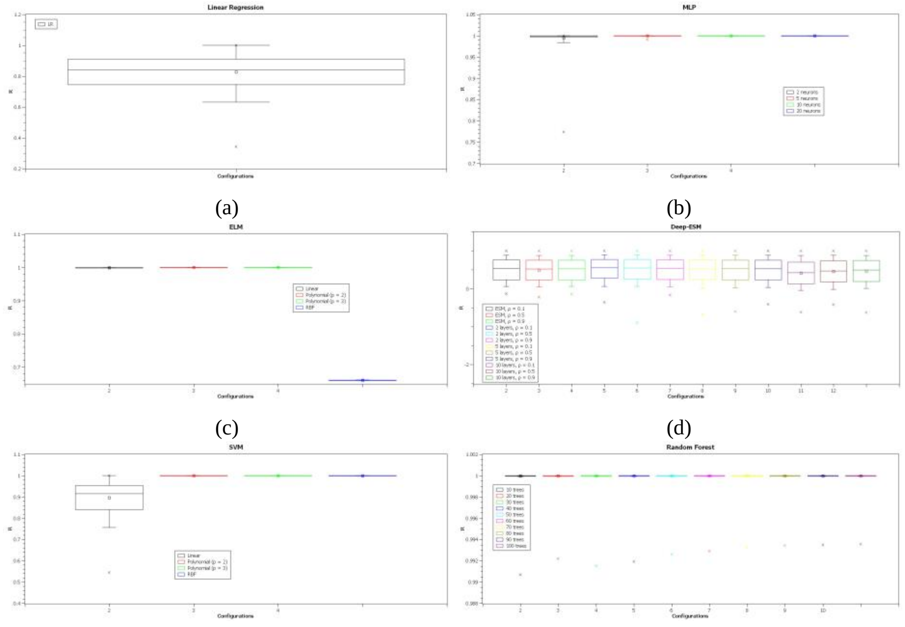
		R		RMSE (%)		Training time (ms)	
Regression method	**Configuration**	**Average**	**SD**	**Average**	**SD**	**Average**	**SD**
Deep-ESM	1 hidden-layer, ρ = 0.1	0.5	0.3	7.58E+07	3.22E+09	152	54
	1 hidden-layer, ρ = 0.5	0.5	0.3	9.19E+07	4.16E+09	212	30
	1 hidden-layer, ρ = 0.9	0.5	0.3	2.80E+07	1.05E+09	206	23
	2 hidden-layers, ρ = 0.1	0.5	0.3	3.35E+07	1.03E+09	2028	270
	2 hidden-layers, ρ = 0.5	0.5	0.3	2.02E+07	4.16E+08	1868	222
	2 hidden-layers, ρ = 0.9	0.5	0.3	2.93E+07	1.09E+09	3101	227
	5 hidden-layers, ρ = 0.1	0.5	0.3	4.69E+07	1.43E+09	4676	668
	5 hidden-layers, ρ = 0.5	0.5	0.3	2.21E+07	8.06E+08	7279	51517
	5 hidden-layers, ρ = 0.9	0.5	0.3	7.47E+07	3.23E+09	7287	51512
	10 hidden-layers, ρ = 0.1	0.4	0.3	3.84E+07	6.34E+08	11285	51643
	10 hidden-layers, ρ = 0.5	0.5	0.3	3.25E+07	1.04E+09	13212	159842
	10 hidden-layers, ρ = 0.9	0.5	0.3	3.64E+07	1.05E+09	13452	159844

Table 5 Results of R, RMSE (%) and training time (ms), sample mean and standard deviation (SD), for ESM and Deep-ESM neural networks. The best results on average are highlighted in red.



		R		RMSE (%)		Training time (ms)	
Regression method	**Configuration**	**Average**	**SD**	**Average**	**SD**	**Average**	**SD**
Random Forest	10 trees	0.9999	0.0001	0.6	0.6	30	9
	20 trees	0.9999	0.0001	0.6	0.5	60	21
	30 trees	0.9999	0.0001	0.5	0.5	101	23
	40 trees	0.9999	0.0001	0.5	0.5	114	21
	50 trees	0.9999	0.0001	0.5	0.5	143	26
	60 trees	0.9999	0.0001	0.5	0.5	175	33
	70 trees	0.9999	0.0001	0.5	0.5	200	36
	80 trees	0.9999	0.0001	0.5	0.5	231	43
	90 trees	0.9999	0.0001	0.5	0.5	271	50
	100 trees	0.9999	0.0001	0.5	0.5	305	61

Table 6 Results of R, RMSE (%) and training time (ms), sample mean and standard deviation (SD), for Random Forest, from 10 to 100 trees, step of 10 trees. The best results on average are highlighted in red.



(a) (b)

(c) (d)

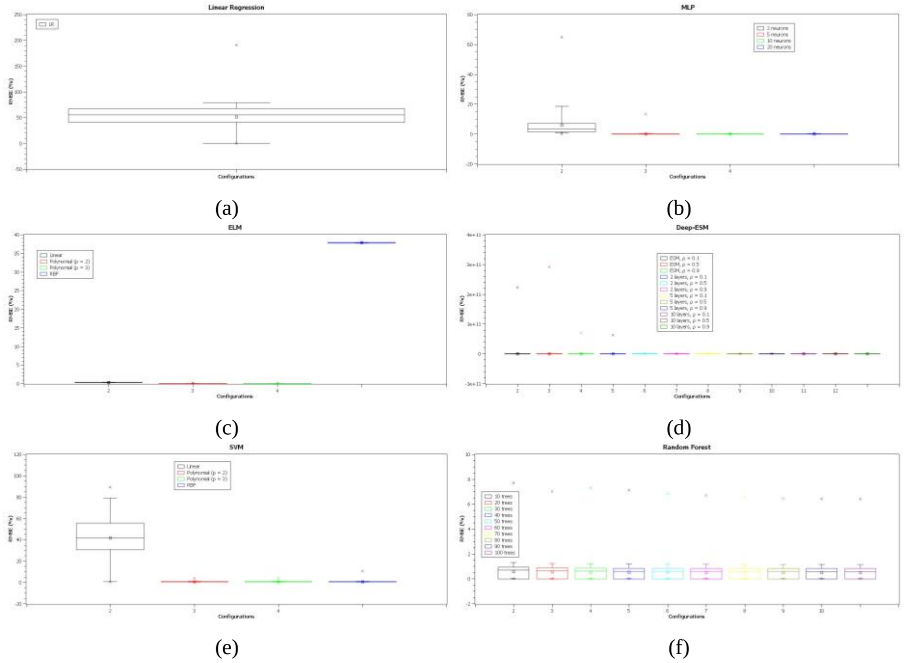
(e) (f)

Fig. 5 Correlation coefficient R for: (a) linear regression; (b) MLP neural networks with a single hidden layer, with 2, 5, 10, and 20 hidden neurons; (c) ELM neural networks with linear kernel, degree 2 and 3 polynomial, and RBF; (d) ESM and DeepESM neural networks, with 2, 5 and 10 hidden layers, with 10 neurons per hidden layer; (e) SVM with linear kernel, polynomial of degrees 2 and 3, and RBF; (f) Random Forests with 10, 20, 30, ..., and 100 trees.

The results were generated considering the initial databases distributed by neighborhood, without applying interpolation, a base for each of the six bimesters, for the years 2014, 2015 and 2016, using data from 2013 to 2016. Each base has, therefore, 94 instances, associated with 94 neighborhoods and districts of the City of Recife, with 44 attributes, which include latitude and longitude of the central point of the neighborhood, number of cases per two months and temperatures, rainfall and wind speeds per month, for each one. of the six quarters of the year. Tables 7, 8 and 9 present the results of the automatic analysis of the most relevant factors for the two months of 2014, 2015 and 2016, in that order.

3.3 Spatio-temporal analysis

Table 10 shows the prediction results for linear regression and Random Forest with 10 trees, considering training sets of 4665 instances and test sets of 1555 instances, obtained from interpolation of real data. The best results are highlighted in red. As quality metrics, the correlation index R, the RMSE (%) error and the correlation indexes of Kendall and Spearman were considered.



(a) (b)

(c) (d)

(e) (f)

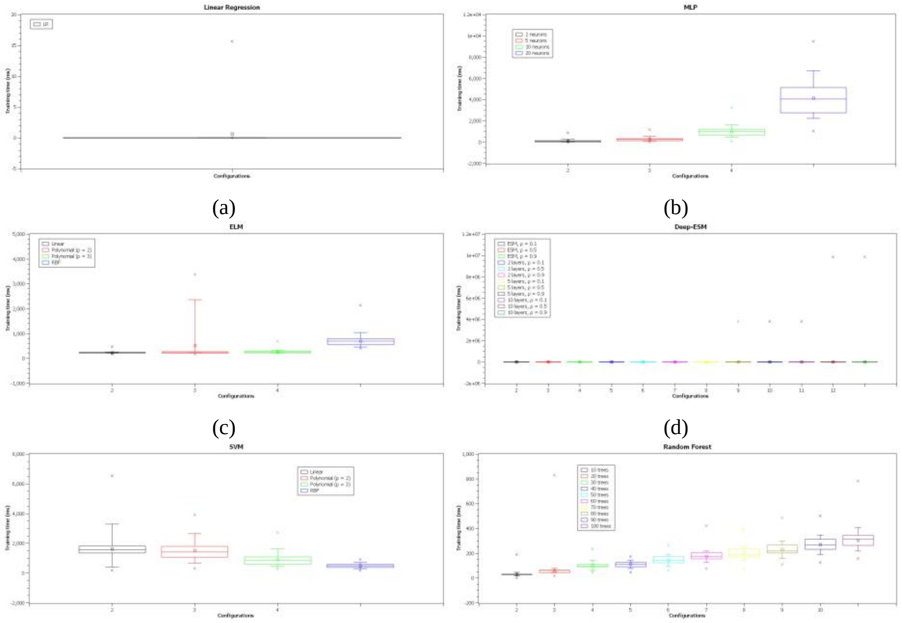
Fig. 6 RMSE (%) for: (a) linear regression; (b) MLP neural networks with a single hidden layer, with 2, 5, 10, and 20 hidden neurons; (c) ELM neural networks with linear kernel, degree 2 and 3 polynomial, and RBF; (d) ESM and DeepESM neural networks, with 2, 5 and 10 hidden layers, with 10 neurons per hidden layer; (e) SVM with linear kernel, polynomial of degrees 2 and 3, and RBF; (f) Random Forests with 10, 20, 30,..., and 100 trees.

Regarding the qualitative results, all forecasting maps were generated with a resolution of 120 dpi, scale 1:182666. Figures 8, 9, and 10 present the forecasting results for the Random Forest with 10 trees, considering training sets of 4665 instances and test sets of 1555 instances, obtained from interpolation of real data, for bimonths 1 (January to February), 2 (March to April), 3 (May to June), 4 (July to August), 5 (September to October), and 6 (November to December), for the years 2014, 2015 and 2016, respectively.

4 Discussion

4.1 Forecasting model

Table 1 shows that the results with linear regression are reasonable in relation to the correlation index R, with mean 0.8 and standard deviation 0.1, and that learning is very fast, with training time average of 1ms and standard deviation of 3ms. However, the RMSE (%) is high: 51 on average with a standard deviation of 23. The boxplot plot in Figure 5(a) shows the distribution of the correlation index R, with half of the results (a box) ranging from 0.75 to 0.9. Considering the distribution of the mustache in the graph, the variation is greater: from 0.65 to 1.00. However, there is a point off that shows that at least one result of R was only 0.35. Thus, the correlation index R varies considerably. Figure 6(a) shows the boxplot of the RMSE distribution (%), with a concentration around 50%, which is a considerably large error. There is still a point off close to 200, which shows that at least one result has an error of almost 200%. Figure 7(a) shows the boxplot of the distribution of training time, showing that linear regression is a very fast regression method, with the absolute majority of results concentrated in less than 1ms, despite a point out close to 16ms, which is still pretty low.



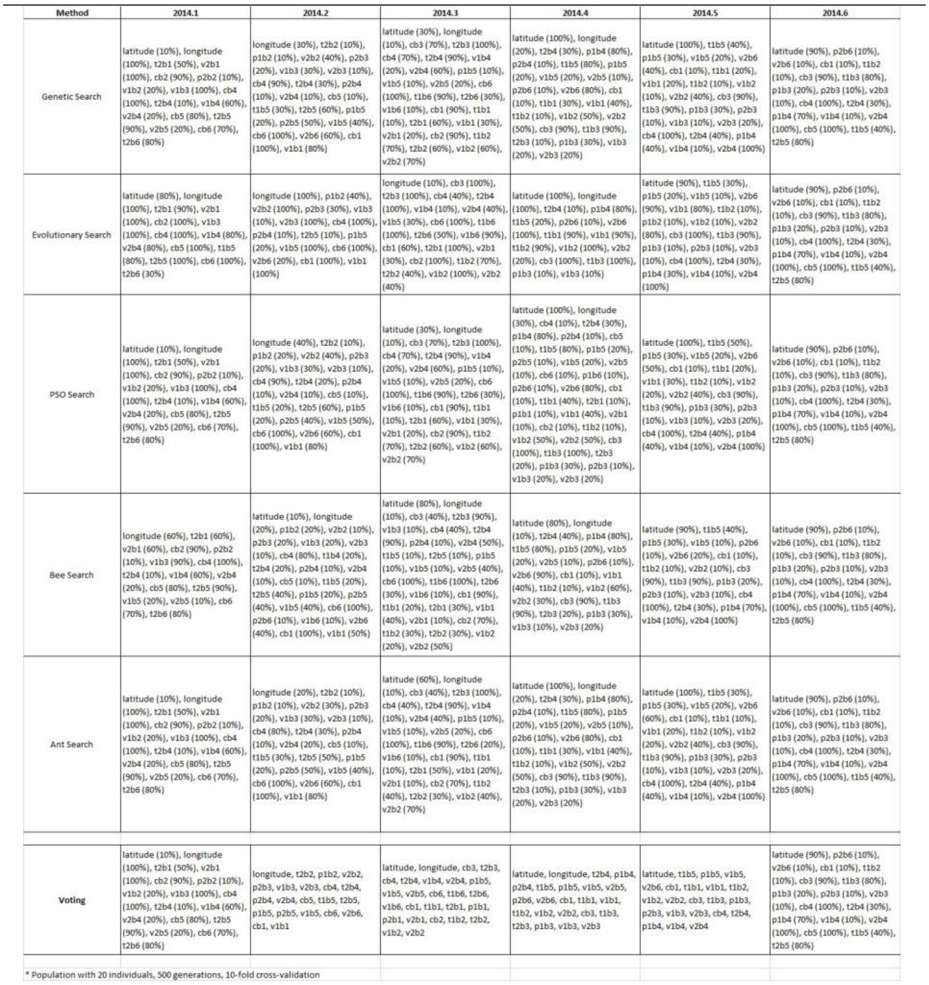
(a) (b)

(c) (d)

(e) (f)

Fig. 7 Training time (ms) for: (a) linear regression; (b) MLP neural networks with a single hidden layer, with 2, 5, 10, and 20 hidden neurons; (c) ELM neural networks with linear kernel, degree 2 and 3 polynomial, and RBF; (d) ESM and DeepESM neural networks, with 2, 5 and 10 hidden layers, with 10 neurons per hidden layer; (e) SVM with linear kernel, polynomial of degrees 2 and 3, and RBF; (f) Random Forests with 10, 20, 30, ..., and 100 trees.

Table 2 presents the results for multilayer perceptron neural networks (MLP), with 2, 5, 10 and 20 neurons in the hidden layer. The results were already interesting with 2 neurons in the hidden layer: the correlation index R proved to be quite high, with a mean of 0.99, and stable, as the standard deviation was only 0.01. The RMSE (%) was also reasonably low: an average of 6% with a standard deviation of 8, but still a little above the limit that we consider great in this work, of 5%. The training time can be considered low: 97ms with a standard deviation of 84ms. Figure 5(b) shows the boxplot of the distribution of the correlation index R, showing that the results of R for the four studied configurations are very similar and considerably high: all distributions are very concentrated in 1, 00. This behavior is a little different only for the configuration with 2 neurons in the hidden layer, with a point out of 0.77, but in general this configuration also resembles the other three. The behavior for the RMSE (%) is somewhat different, according to the boxplot in Figure 6(b): although the configuration with 2 neurons in the hidden layer is concentrated around 5%, the mustache of the distribution reaches 20%. There is also a point off at 65%. The configurations with 5, 10 and 20 neurons in the hidden layer are very similar to each other, except that the configuration with 5 neurons has an off point at approximately 15%. Configurations with 10 and 20 neurons can be considered equivalent. Figure 7(b) shows the distribution of training time in ms. Configurations with 2 and 5 neurons in the hidden layer are significantly faster, while the configuration with 20 neurons is not only slower training in median, but also this training time spreads more, ranging from 1000ms to almost 10000ms, or that is, from 1 to 10 seconds. Although this time is not critical, it can be considered as the best configuration the one with 10 neurons in the hidden layer, as it has a high correlation index R, a low RMSE (%), and consumes less memory (fewer neurons) and have faster training.

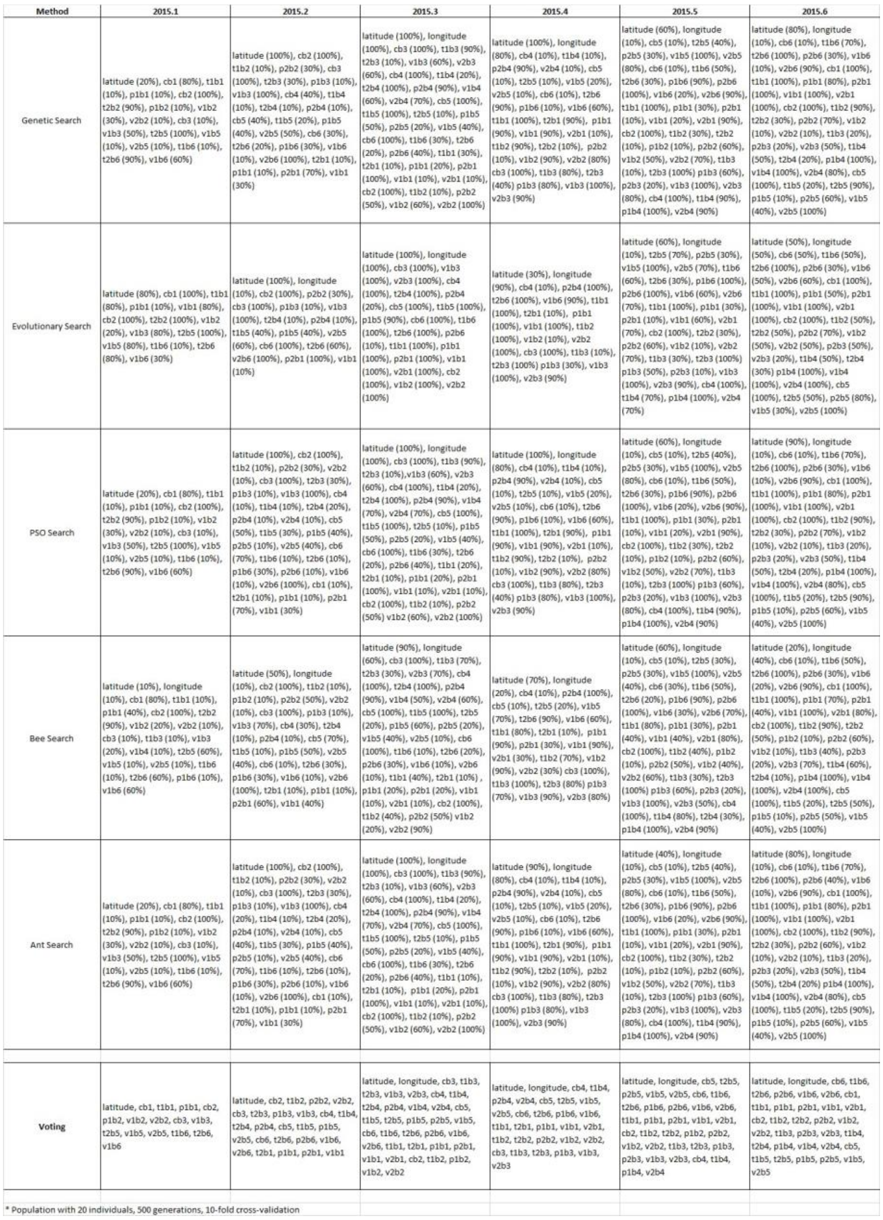


Method	2014.1	2014.2	2014.3	2014.4	2014.5	2014.6
Genetic Search	latitude (10%), longitude (100%), t2b1 (50%), v2b1 (100%), cb2 (90%), p2b2 (10%), v1b2 (20%), v1b3 (100%), cb4 (100%), t2b4 (10%), v1b4 (60%), v2b4 (20%), cb5 (80%), t2b5 (90%), v2b5 (20%), cb6 (70%), t2b6 (80%)	longitude (30%), t2b2 (10%), p1b2 (10%), v2b2 (40%), p2b3 (20%), v1b3 (30%), v2b3 (10%), cb4 (90%), t2b4 (30%), p2b4 (10%), v2b4 (10%), cb5 (10%), t1b5 (30%), t2b5 (60%), p1b5 (20%), p2b5 (50%), v1b5 (40%), cb6 (100%), v2b6 (60%), cb1 (100%), v1b1 (80%)	latitude (30%), longitude (10%), cb3 (70%), t2b3 (100%), cb4 (70%), t2b4 (90%), v1b4 (20%), v2b4 (60%), p1b5 (10%), v1b5 (10%), v2b5 (20%), cb6 (100%), t1b6 (90%), t2b6 (30%), v1b6 (10%), cb1 (90%), t1b1 (10%), t2b1 (60%), v1b1 (30%), v2b1 (20%), cb2 (90%), t1b2 (70%), t2b2 (60%), v1b2 (60%), v2b2 (70%)	latitude (100%), longitude (20%), t2b4 (30%), p1b4 (80%), p2b4 (10%), t1b5 (80%), p1b5 (20%), v1b5 (20%), v2b5 (10%), p2b6 (10%), v2b6 (80%), cb1 (10%), t1b1 (30%), v1b1 (40%), t1b2 (10%), v1b2 (50%), v2b2 (50%), cb3 (90%), t1b3 (90%), t2b3 (10%), p1b3 (30%), v1b3 (20%), v2b3 (20%)	latitude (100%), t1b5 (40%), p1b5 (30%), v1b5 (20%), v2b6 (40%), cb1 (10%), t1b1 (20%), v1b1 (20%), t1b2 (10%), v1b2 (10%), v2b2 (40%), cb3 (90%), t1b3 (90%), p1b3 (30%), p2b3 (10%), v1b3 (10%), v2b3 (20%), cb4 (100%), t2b4 (40%), p1b4 (40%), v1b4 (10%), v2b4 (100%)	latitude (90%), p2b6 (10%), v2b6 (10%), cb1 (10%), t1b2 (10%), cb3 (90%), t1b3 (80%), p1b3 (20%), p2b3 (10%), v2b3 (10%), cb4 (100%), t2b4 (30%), p1b4 (70%), v1b4 (10%), v2b4 (100%), cb5 (100%), t1b5 (40%), t2b5 (80%)
Evolutionary Search	latitude (80%), longitude (100%), t2b1 (90%), v2b1 (100%), cb2 (100%), v1b3 (100%), cb4 (100%), v1b4 (80%), v2b4 (80%), cb5 (100%), t1b5 (80%), t2b5 (100%), cb6 (100%), t2b6 (30%)	longitude (100%), p1b2 (40%), v2b2 (100%), p2b3 (30%), v1b3 (10%), v2b3 (100%), cb4 (100%), p2b4 (10%), t2b5 (10%), p1b5 (20%), v1b5 (100%), cb6 (100%), v2b6 (20%), cb1 (100%), v1b1 (100%)	longitude (10%), cb3 (100%), t2b3 (100%), cb4 (40%), t2b4 (100%), v1b4 (10%), v2b4 (40%), v1b5 (30%), cb6 (100%), t1b6 (100%), t2b6 (50%), v1b6 (90%), cb1 (60%), t2b1 (100%), v2b1 (30%), cb2 (100%), t1b2 (70%), t2b2 (40%), v1b2 (100%), v2b2 (40%)	latitude (100%), longitude (100%), t2b4 (10%), p1b4 (80%), t1b5 (20%), p2b6 (10%), v2b6 (100%), t1b1 (90%), v1b1 (90%), t1b2 (90%), v1b2 (100%), v2b2 (20%), cb3 (100%), t1b3 (100%), p1b3 (10%), v1b3 (10%)	latitude (90%), t1b5 (30%), p1b5 (20%), v1b5 (10%), v2b6 (90%), v1b1 (80%), t1b2 (10%), p1b2 (10%), v1b2 (10%), v2b2 (80%), cb3 (100%), t1b3 (90%), p1b3 (10%), p2b3 (10%), v2b3 (10%), cb4 (100%), t2b4 (30%), p1b4 (30%), v1b4 (10%), v2b4 (100%)	latitude (90%), p2b6 (10%), v2b6 (10%), cb1 (10%), t1b2 (10%), cb3 (90%), t1b3 (80%), p1b3 (20%), p2b3 (10%), v2b3 (10%), cb4 (100%), t2b4 (30%), p1b4 (70%), v1b4 (10%), v2b4 (100%), cb5 (100%), t1b5 (40%), t2b5 (80%)
PSO Search	latitude (10%), longitude (100%), t2b1 (50%), v2b1 (100%), cb2 (90%), p2b2 (10%), v1b2 (20%), v1b3 (100%), cb4 (100%), t2b4 (10%), v1b4 (60%), v2b4 (20%), cb5 (80%), t2b5 (90%), v2b5 (20%), cb6 (70%), t2b6 (80%)	longitude (40%), t2b2 (10%), p1b2 (20%), v2b2 (40%), p2b3 (20%), v1b3 (30%), v2b3 (10%), cb4 (90%), t2b4 (20%), p2b4 (10%), v2b4 (10%), cb5 (10%), t1b5 (20%), t2b5 (60%), p1b5 (20%), p2b5 (40%), v1b5 (50%), cb6 (100%), v2b6 (60%), cb1 (100%), v1b1 (80%)	latitude (30%), longitude (10%), cb3 (70%), t2b3 (100%), cb4 (70%), t2b4 (90%), v1b4 (20%), v2b4 (60%), p1b5 (10%), v1b5 (10%), v2b5 (20%), cb6 (100%), t1b6 (90%), t2b6 (30%), v1b6 (10%), cb1 (90%), t1b1 (10%), t2b1 (60%), v1b1 (30%), v2b1 (20%), cb2 (90%), t1b2 (70%), t2b2 (60%), v1b2 (60%), v2b2 (70%)	latitude (100%), longitude (30%), cb4 (10%), t2b4 (30%), p1b4 (80%), p2b4 (10%), cb5 (10%), t1b5 (80%), p1b5 (20%), p2b5 (10%), v1b5 (20%), v2b5 (10%), cb6 (10%), p1b6 (10%), p2b6 (10%), v2b6 (80%), cb1 (10%), t1b1 (40%), t2b1 (10%), p1b1 (10%), v1b1 (40%), v2b1 (10%), cb2 (10%), t1b2 (10%), v1b2 (50%), v2b2 (50%), cb3 (100%), t1b3 (100%), t2b3 (20%), p1b3 (30%), p2b3 (10%), v1b3 (20%), v2b3 (20%)	latitude (100%), t1b5 (50%), p1b5 (30%), v1b5 (20%), v2b6 (50%), cb1 (10%), t1b1 (20%), v1b1 (30%), t1b2 (10%), v1b2 (20%), v2b2 (40%), cb3 (90%), t1b3 (90%), p1b3 (30%), p2b3 (10%), v1b3 (10%), v2b3 (20%), cb4 (100%), t2b4 (40%), p1b4 (40%), v1b4 (10%), v2b4 (100%)	latitude (90%), p2b6 (10%), v2b6 (10%), cb1 (10%), t1b2 (10%), cb3 (90%), t1b3 (80%), p1b3 (20%), p2b3 (10%), v2b3 (10%), cb4 (100%), t2b4 (30%), p1b4 (70%), v1b4 (10%), v2b4 (100%), cb5 (100%), t1b5 (40%), t2b5 (80%)
Bee Search	longitude (60%), t2b1 (60%), v2b1 (60%), cb2 (90%), p2b2 (10%), v1b3 (90%), cb4 (100%), t2b4 (10%), v1b4 (60%), v2b4 (20%), cb5 (80%), t2b5 (90%), v1b5 (20%), v2b5 (10%), cb6 (70%), t2b6 (80%)	latitude (10%), longitude (20%), p1b2 (20%), v2b2 (10%), p2b3 (20%), v1b3 (20%), v2b3 (10%), cb4 (80%), t1b4 (20%), t2b4 (20%), p2b4 (10%), v2b4 (10%), cb5 (10%), t1b5 (20%), t2b5 (40%), p1b5 (20%), p2b5 (40%), v1b5 (40%), cb6 (100%), p2b6 (10%), v1b6 (10%), v2b6 (40%), cb1 (100%), v1b1 (50%)	latitude (80%), longitude (10%), cb3 (40%), t2b3 (90%), v1b3 (10%), cb4 (40%), t2b4 (90%), p2b4 (10%), v2b4 (50%), t1b5 (10%), t2b5 (10%), p1b5 (10%), v1b5 (10%), v2b5 (40%), cb6 (100%), t1b6 (100%), t2b6 (30%), v1b6 (10%), cb1 (90%), t1b1 (20%), t2b1 (30%), v1b1 (40%), v2b1 (10%), cb2 (70%), t1b2 (30%), t2b2 (30%), v1b2 (20%), v2b2 (50%)	latitude (80%), longitude (10%), t2b4 (40%), p1b4 (80%), t1b5 (80%), p1b5 (20%), v1b5 (20%), v2b5 (10%), p2b6 (10%), v2b6 (90%), cb1 (10%), v1b1 (40%), t1b2 (10%), v1b2 (60%), v2b2 (30%), cb3 (90%), t1b3 (90%), t2b3 (20%), p1b3 (30%), v1b3 (10%), v2b3 (20%)	latitude (90%), t1b5 (40%), p1b5 (30%), v1b5 (10%), p2b6 (10%), v2b6 (20%), cb1 (10%), t1b2 (10%), v2b2 (10%), cb3 (90%), t1b3 (90%), p1b3 (20%), p2b3 (10%), v2b3 (10%), cb4 (100%), t2b4 (30%), p1b4 (70%), v1b4 (10%), v2b4 (100%)	latitude (90%), p2b6 (10%), v2b6 (10%), cb1 (10%), t1b2 (10%), cb3 (90%), t1b3 (80%), p1b3 (20%), p2b3 (10%), v2b3 (10%), cb4 (100%), t2b4 (30%), p1b4 (70%), v1b4 (10%), v2b4 (100%), cb5 (100%), t1b5 (40%), t2b5 (80%)
Ant Search	latitude (10%), longitude (100%), t2b1 (50%), v2b1 (100%), cb2 (90%), p2b2 (10%), v1b2 (20%), v1b3 (100%), cb4 (100%), t2b4 (10%), v1b4 (60%), v2b4 (20%), cb5 (80%), t2b5 (90%), v2b5 (20%), cb6 (70%), t2b6 (80%)	longitude (20%), t2b2 (10%), p1b2 (10%), v2b2 (30%), p2b3 (20%), v1b3 (30%), v2b3 (10%), cb4 (80%), t2b4 (30%), p2b4 (10%), v2b4 (20%), cb5 (10%), t1b5 (30%), t2b5 (50%), p1b5 (20%), p2b5 (50%), v1b5 (40%), cb6 (100%), v2b6 (60%), cb1 (100%), v1b1 (80%)	latitude (60%), longitude (10%), cb3 (40%), t2b3 (100%), cb4 (40%), t2b4 (90%), v1b4 (10%), v2b4 (40%), p1b5 (10%), v1b5 (10%), v2b5 (20%), cb6 (100%), t1b6 (90%), t2b6 (20%), v1b6 (10%), cb1 (90%), t1b1 (10%), t2b1 (50%), v1b1 (20%), v2b1 (10%), cb2 (70%), t1b2 (40%), t2b2 (30%), v1b2 (40%), v2b2 (70%)	latitude (100%), longitude (20%), t2b4 (30%), p1b4 (80%), p2b4 (10%), t1b5 (80%), p1b5 (20%), v1b5 (20%), v2b5 (10%), p2b6 (10%), v2b6 (80%), cb1 (10%), t1b1 (30%), v1b1 (40%), t1b2 (10%), v1b2 (50%), v2b2 (50%), cb3 (90%), t1b3 (90%), t2b3 (10%), p1b3 (30%), v1b3 (20%), v2b3 (20%)	latitude (100%), t1b5 (30%), p1b5 (30%), v1b5 (20%), v2b6 (60%), cb1 (10%), t1b1 (10%), v1b1 (20%), t1b2 (10%), v1b2 (20%), v2b2 (40%), cb3 (90%), t1b3 (90%), p1b3 (30%), p2b3 (10%), v1b3 (10%), v2b3 (20%), cb4 (100%), t2b4 (40%), p1b4 (40%), v1b4 (10%), v2b4 (100%)	latitude (90%), p2b6 (10%), v2b6 (10%), cb1 (10%), t1b2 (10%), cb3 (90%), t1b3 (80%), p1b3 (20%), p2b3 (10%), v2b3 (10%), cb4 (100%), t2b4 (30%), p1b4 (70%), v1b4 (10%), v2b4 (100%), cb5 (100%), t1b5 (40%), t2b5 (80%)
Voting	latitude (10%), longitude (100%), t2b1 (50%), v2b1 (100%), cb2 (90%), p2b2 (10%), v1b2 (20%), v1b3 (100%), cb4 (100%), t2b4 (10%), v1b4 (60%), v2b4 (20%), cb5 (80%), t2b5 (90%), v2b5 (20%), cb6 (70%), t2b6 (80%)	longitude, t2b2, p1b2, v2b2, p2b3, v1b3, v2b3, cb4, t2b4, p2b4, v2b4, cb5, t1b5, t2b5, p1b5, p2b5, v1b5, cb6, v2b6, cb1, v1b1	latitude, longitude, cb3, t2b3, cb4, t2b4, v1b4, v2b4, p1b5, v1b5, v2b5, cb6, t1b6, t2b6, v1b6, cb1, t1b1, t2b1, p1b1, p2b1, v2b1, cb2, t1b2, t2b2, v1b2, v2b2	latitude, longitude, t2b4, p1b4, p2b4, t1b5, p1b5, v1b5, v2b5, p2b6, v2b6, cb1, t1b1, v1b1, t1b2, v1b2, v2b2, cb3, t1b3, t2b3, p1b3, v1b3, v2b3	latitude, t1b5, p1b5, v1b5, v2b6, cb1, t1b1, v1b1, t1b2, v1b2, v2b2, cb3, t1b3, p1b3, p2b3, v1b3, v2b3, cb4, t2b4, p1b4, v1b4, v2b4	latitude (90%), p2b6 (10%), v2b6 (10%), cb1 (10%), t1b2 (10%), cb3 (90%), t1b3 (80%), p1b3 (20%), p2b3 (10%), v2b3 (10%), cb4 (100%), t2b4 (30%), p1b4 (70%), v1b4 (10%), v2b4 (100%), cb5 (100%), t1b5 (40%), t2b5 (80%)

* Population with 20 individuals, 500 generations, 10-fold cross-validation

Table 7 Result of the analysis of the most relevant factors through the algorithms of Genetic Search, Evolutionary Search, Particle Swarm Optimization, Bee Search and Ant Search, for 20 solution candidates evolving in 500 generations, for the year 2014.

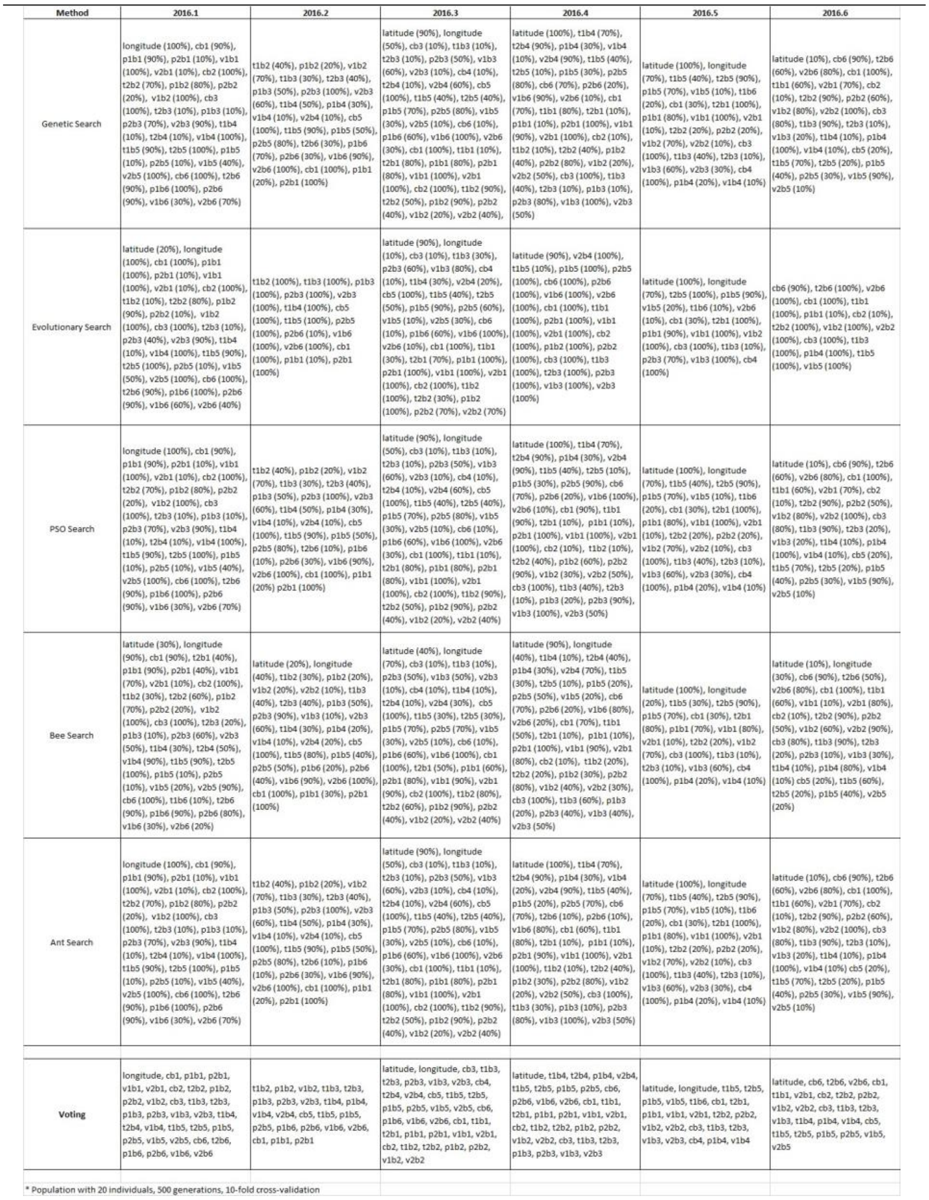
In Table 4 are shown the results for extreme learning machines (ELM), with linear kernel, polynomial degrees 2 and 3, and radial basis function (RBF). Except for the RBF kernel, all configurations showed very high mean R correlation index (from 0.999 to 1.000) and very low standard deviation. Likewise, all polynomial configurations reached RMSE (%) errors well below the 5% considered in this work as a standard. The average training time also ranged, in these polynomial configurations, from 200ms to 500ms. Figure 5(c) shows the boxplot of the distribution of the correlation index R, with practically similar and considerably high results for the linear, polynomial degree 2 and 3 kernels, touching 1.00 and with almost zero dispersion. The RBF kernel was also stable, but with a distribution of R concentrated in 0.66. Figure 6(c) presents a similar scenario: RMSE (%) errors fairly concentrated between 0.00% and 0.20%, except for the RBF kernel, again. In Figure 7(c) the training time distributions of the used architectures are presented. The behaviors of linear and polynomial degree 2 and 3 kernels could be considered similar if the polynomial degree 2 kernel configuration did not have an unexpected scattering, with mustache reaching 2400ms and point outside 3500ms. Again, the results with RBF were not interesting. Thus, linear ELM can be considered the most suitable ELM configuration to solve the problem.



Method	2015.1	2015.2	2015.3	2015.4	2015.5	2015.6
Genetic Search	latitude (20%), cb1 (80%), t1b1 (10%), p1b1 (10%), cb2 (100%), t2b2 (90%), p1b2 (10%), v1b2 (30%), v2b2 (10%), cb3 (10%), v1b3 (50%), t2b5 (100%), v1b5 (10%), v2b5 (10%), t1b6 (10%), t2b6 (90%), v1b6 (60%)	latitude (100%), cb2 (100%), t1b2 (10%), p2b2 (30%), cb3 (100%), t2b3 (30%), p1b3 (10%), v1b3 (100%), cb4 (40%), t1b4 (10%), t2b4 (10%), p2b4 (10%), cb5 (40%), t1b5 (20%), p1b5 (40%), v2b5 (50%), cb6 (30%), t2b6 (20%), p1b6 (30%), v1b6 (10%), v2b6 (100%), t2b1 (10%), p1b1 (10%), p2b1 (70%), v1b1 (30%)	latitude (100%), longitude (100%), cb3 (100%), t1b3 (90%), t2b3 (10%), v1b3 (60%), v2b3 (60%), cb4 (100%), t1b4 (20%), t2b4 (100%), p2b4 (90%), v1b4 (60%), v2b4 (70%), cb5 (100%), t1b5 (100%), t2b5 (10%), p1b5 (50%), p2b5 (20%), v1b5 (40%), cb6 (100%), t1b6 (30%), t2b6 (20%), p2b6 (40%), t1b1 (30%), t2b1 (10%), p1b1 (20%), p2b1 (100%), v1b1 (10%), v2b1 (10%), cb2 (100%), t1b2 (10%), p2b2 (50%), v1b2 (60%), v2b2 (100%)	latitude (100%), longitude (80%), cb4 (10%), t1b4 (10%), p2b4 (90%), v2b4 (10%), cb5 (10%), t2b5 (10%), v1b5 (20%), v2b5 (10%), cb6 (10%), t2b6 (90%), p1b6 (10%), v1b6 (60%), t1b1 (100%), t2b1 (90%), p1b1 (90%), v1b1 (90%), v2b1 (10%), t1b2 (90%), t2b2 (10%), p2b2 (10%), v1b2 (90%), v2b2 (80%) cb3 (100%), t1b3 (80%), t2b3 (40%) p1b3 (80%), v1b3 (100%), v2b3 (90%)	latitude (60%), longitude (10%), cb5 (10%), t2b5 (40%), p2b5 (30%), v1b5 (100%), v2b5 (80%), cb6 (10%), t1b6 (50%), t2b6 (30%), p1b6 (90%), p2b6 (100%), v1b6 (20%), v2b6 (90%), t1b1 (100%), p1b1 (30%), p2b1 (10%), v1b1 (20%), v2b1 (90%), cb2 (100%), t1b2 (30%), t2b2 (10%), p1b2 (10%), p2b2 (60%), v1b2 (50%), v2b2 (70%), t1b3 (10%), t2b3 (100%) p1b3 (60%), p2b3 (20%), v1b3 (100%), v2b3 (80%), cb4 (100%), t1b4 (90%), p1b4 (100%), v2b4 (90%)	latitude (80%), longitude (10%), cb6 (10%), t1b6 (70%), t2b6 (100%), p2b6 (30%), v1b6 (10%), v2b6 (90%), cb1 (100%), t1b1 (100%), p1b1 (80%), p2b1 (100%), v1b1 (100%), v2b1 (100%), cb2 (100%), t1b2 (90%), t2b2 (30%), p2b2 (70%), v1b2 (10%), v2b2 (10%), t1b3 (20%), p2b3 (20%), v2b3 (50%), t1b4 (50%), t2b4 (20%), p1b4 (100%), v1b4 (100%), v2b4 (80%), cb5 (100%), t1b5 (20%), t2b5 (90%), p1b5 (10%), p2b5 (60%), v1b5 (40%), v2b5 (100%)
Evolutionary Search	latitude (80%), cb1 (100%), t1b1 (80%), p1b1 (10%), v1b1 (80%), cb2 (100%), t2b2 (100%), v1b2 (20%), v1b3 (80%), t2b5 (100%), v1b5 (80%), t1b6 (10%), t2b6 (80%), v1b6 (30%)	latitude (100%), longitude (10%), cb2 (100%), p2b2 (30%), cb3 (100%), p1b3 (10%), v1b3 (100%), t2b4 (10%), p2b4 (10%), t1b5 (40%), p1b5 (40%), v2b5 (60%), cb6 (100%), t2b6 (60%), v2b6 (100%), p2b1 (100%), v1b1 (10%)	latitude (100%), longitude (100%), cb3 (100%), v1b3 (100%), v2b3 (100%), cb4 (100%), t2b4 (100%), p2b4 (20%), cb5 (100%), t1b5 (100%), p1b5 (90%), cb6 (100%), t1b6 (100%), t2b6 (100%), p2b6 (10%), t1b1 (100%), p1b1 (100%), p2b1 (100%), v1b1 (100%), v2b1 (100%), cb2 (100%), v1b2 (100%), v2b2 (100%)	latitude (30%), longitude (90%), cb4 (10%), p2b4 (100%), t2b6 (100%), v1b6 (90%), t1b1 (100%), t2b1 (10%), p1b1 (100%), v1b1 (100%), t1b2 (100%), v1b2 (10%), v2b2 (100%), cb3 (100%), t1b3 (10%), t2b3 (100%) p1b3 (30%), v1b3 (100%), v2b3 (90%)	latitude (60%), longitude (10%), t2b5 (70%), p2b5 (30%), v1b5 (100%), v2b5 (70%), t1b6 (60%), t2b6 (30%), p1b6 (100%), p2b6 (100%), v1b6 (60%), v2b6 (70%), t1b1 (100%), p1b1 (30%), p2b1 (10%), v1b1 (60%), v2b1 (70%), cb2 (100%), t2b2 (30%), p2b2 (60%), v1b2 (10%), v2b2 (70%), t1b3 (30%), t2b3 (100%) p1b3 (50%), p2b3 (10%), v1b3 (100%), v2b3 (90%), cb4 (100%), t1b4 (70%), p1b4 (100%), v2b4 (70%)	latitude (50%), longitude (50%), cb6 (50%), t1b6 (50%), t2b6 (100%), p2b6 (30%), v1b6 (50%), v2b6 (60%), cb1 (100%), t1b1 (100%), p1b1 (50%), p2b1 (100%), v1b1 (100%), v2b1 (100%), cb2 (100%), t1b2 (50%), t2b2 (50%), p2b2 (70%), v1b2 (50%), v2b2 (50%), p2b3 (50%), v2b3 (20%), t1b4 (50%), t2b4 (30%) p1b4 (100%), v1b4 (100%), v2b4 (100%), cb5 (100%), t2b5 (50%), p2b5 (80%), v1b5 (30%), v2b5 (100%)
PSO Search	latitude (20%), cb1 (80%), t1b1 (10%), p1b1 (10%), cb2 (100%), t2b2 (90%), p1b2 (10%), v1b2 (30%), v2b2 (10%), cb3 (10%), v1b3 (50%), t2b5 (100%), v1b5 (10%), v2b5 (10%), t1b6 (10%), t2b6 (90%), v1b6 (60%)	latitude (100%), cb2 (100%), t1b2 (10%), p2b2 (30%), v2b2 (10%), cb3 (100%), t2b3 (30%), p1b3 (10%), v1b3 (100%), cb4 (10%), t1b4 (10%), t2b4 (20%), p2b4 (10%), v2b4 (10%), cb5 (50%), t1b5 (30%), p1b5 (40%), p2b5 (10%), v2b5 (40%), cb6 (70%), t1b6 (10%), t2b6 (10%), p1b6 (30%), p2b6 (10%), v1b6 (10%), v2b6 (100%), cb1 (10%), t2b1 (10%), p1b1 (10%), p2b1 (70%), v1b1 (30%)	latitude (100%), longitude (100%), cb3 (100%), t1b3 (90%), t2b3 (10%),v1b3 (60%), v2b3 (60%), cb4 (100%), t1b4 (20%), t2b4 (100%), p2b4 (90%), v1b4 (70%), v2b4 (70%), cb5 (100%), t1b5 (100%), t2b5 (10%), p1b5 (50%), p2b5 (20%), v1b5 (40%), cb6 (100%), t1b6 (30%), t2b6 (20%), p2b6 (40%), t1b1 (20%), t2b1 (10%), p1b1 (20%), p2b1 (100%), v1b1 (10%), v2b1 (10%), cb2 (100%), t1b2 (10%), p2b2 (50%) v1b2 (60%), v2b2 (100%)	latitude (100%), longitude (80%), cb4 (10%), t1b4 (10%), p2b4 (90%), v2b4 (10%), cb5 (10%), t2b5 (10%), v1b5 (20%), v2b5 (10%), cb6 (10%), t2b6 (90%), p1b6 (10%), v1b6 (60%), t1b1 (100%), t2b1 (90%), p1b1 (90%), v1b1 (90%), v2b1 (10%), t1b2 (90%), t2b2 (10%), p2b2 (10%), v1b2 (90%), v2b2 (80%) cb3 (100%), t1b3 (80%), t2b3 (40%) p1b3 (80%), v1b3 (100%), v2b3 (90%)	latitude (60%), longitude (10%), cb5 (10%), t2b5 (40%), p2b5 (30%), v1b5 (100%), v2b5 (80%), cb6 (10%), t1b6 (50%), t2b6 (30%), p1b6 (90%), p2b6 (100%), v1b6 (20%), v2b6 (90%), t1b1 (100%), p1b1 (30%), p2b1 (10%), v1b1 (20%), v2b1 (90%), cb2 (100%), t1b2 (30%), t2b2 (10%), p1b2 (10%), p2b2 (60%), v1b2 (50%), v2b2 (70%), t1b3 (10%), t2b3 (100%) p1b3 (60%), p2b3 (20%), v1b3 (100%), v2b3 (80%), cb4 (100%), t1b4 (90%), p1b4 (100%), v2b4 (90%)	latitude (90%), longitude (10%), cb6 (10%), t1b6 (70%), t2b6 (100%), p2b6 (30%), v1b6 (10%), v2b6 (90%), cb1 (100%), t1b1 (100%), p1b1 (80%), p2b1 (100%), v1b1 (100%), v2b1 (100%), cb2 (100%), t1b2 (90%), t2b2 (30%), p2b2 (70%), v1b2 (10%), v2b2 (10%), t1b3 (20%), p2b3 (20%), v2b3 (50%), t1b4 (50%), t2b4 (20%), p1b4 (100%), v1b4 (100%), v2b4 (80%), cb5 (100%), t1b5 (20%), t2b5 (90%), p1b5 (10%), p2b5 (60%), v1b5 (40%), v2b5 (100%)
Bee Search	latitude (10%), longitude (10%), cb1 (80%), t1b1 (10%), p1b1 (40%), cb2 (100%), t2b2 (90%), v1b2 (20%), v2b2 (10%), cb3 (10%), t1b3 (10%), v1b3 (20%), v1b4 (10%), t2b5 (60%), v1b5 (10%), v2b5 (10%), t1b6 (10%), t2b6 (60%), p1b6 (10%), v1b6 (60%)	latitude (50%), longitude (10%), cb2 (100%), t1b2 (10%), p1b2 (10%), p2b2 (50%), v2b2 (10%), cb3 (100%), p1b3 (10%), v1b3 (70%), cb4 (30%), t2b4 (10%), p2b4 (10%), cb5 (70%), t1b5 (10%), p1b5 (50%), v2b5 (40%), cb6 (10%), t2b6 (30%), p1b6 (30%), v1b6 (10%), v2b6 (100%), t2b1 (10%), p1b1 (10%), p2b1 (60%), v1b1 (40%)	latitude (90%), longitude (60%), cb3 (100%), t1b3 (70%), t2b3 (30%), v2b3 (70%), cb4 (100%), t2b4 (100%), p2b4 (90%), v1b4 (50%), v2b4 (60%), cb5 (100%), t1b5 (100%), t2b5 (20%), p1b5 (60%), p2b5 (20%), v1b5 (40%), v2b5 (10%), cb6 (100%), t1b6 (10%), t2b6 (20%), p2b6 (30%), v1b6 (10%), v2b6 (10%), t1b1 (40%), t2b1 (10%) , p1b1 (20%), p2b1 (20%), v1b1 (10%), v2b1 (10%), cb2 (100%), t1b2 (40%), p2b2 (50%) v1b2 (20%), v2b2 (90%)	latitude (70%), longitude (20%), cb4 (10%), p2b4 (100%), cb5 (10%), t2b5 (20%), v1b5 (70%), t2b6 (90%), v1b6 (60%), t1b1 (80%), t2b1 (10%), p1b1 (90%), p2b1 (30%), v1b1 (90%), v2b1 (30%), t1b2 (70%), v1b2 (90%), v2b2 (30%) cb3 (100%), t1b3 (100%), t2b3 (80%) p1b3 (70%), v1b3 (90%), v2b3 (80%)	latitude (60%), longitude (10%), cb5 (10%), t2b5 (30%), p2b5 (30%), v1b5 (100%), v2b5 (40%), cb6 (30%), t1b6 (50%), t2b6 (20%), p1b6 (90%), p2b6 (100%), v1b6 (30%), v2b6 (70%), t1b1 (80%), p1b1 (30%), p2b1 (40%), v1b1 (40%), v2b1 (80%), cb2 (100%), t1b2 (40%), p1b2 (10%), p2b2 (50%), v1b2 (40%), v2b2 (60%), t1b3 (30%), t2b3 (100%) p1b3 (60%), p2b3 (20%), v1b3 (100%), v2b3 (50%), cb4 (100%), t1b4 (80%), t2b4 (30%), p1b4 (100%), v2b4 (90%)	latitude (20%), longitude (40%), cb6 (10%), t1b6 (50%), t2b6 (100%), p2b6 (30%), v1b6 (20%), v2b6 (90%), cb1 (100%), t1b1 (100%), p1b1 (70%), p2b1 (40%), v1b1 (100%), v2b1 (80%), cb2 (100%), t1b2 (90%), t2b2 (50%), p1b2 (10%), p2b2 (60%), v1b2 (10%), t1b3 (40%), p2b3 (20%), v2b3 (70%), t1b4 (60%), t2b4 (10%), p1b4 (100%), v1b4 (100%), v2b4 (100%), cb5 (100%), t1b5 (20%), t2b5 (50%), p1b5 (10%), p2b5 (50%), v1b5 (40%), v2b5 (100%)
Ant Search	latitude (20%), cb1 (80%), t1b1 (10%), p1b1 (10%), cb2 (100%), t2b2 (90%), p1b2 (10%), v1b2 (30%), v2b2 (10%), cb3 (10%), v1b3 (50%), t2b5 (100%), v1b5 (10%), v2b5 (10%), t1b6 (10%), t2b6 (90%), v1b6 (60%)	latitude (100%), cb2 (100%), t1b2 (10%), p2b2 (30%), v2b2 (10%), cb3 (100%), t2b3 (30%), p1b3 (10%), v1b3 (100%), cb4 (20%), t1b4 (10%), t2b4 (20%), p2b4 (10%), v2b4 (10%), cb5 (40%), t1b5 (30%), p1b5 (40%), p2b5 (10%), v2b5 (40%), cb6 (70%), t1b6 (10%), t2b6 (10%), p1b6 (30%), p2b6 (10%), v1b6 (10%), v2b6 (100%), cb1 (10%), t2b1 (10%), p1b1 (10%), p2b1 (70%), v1b1 (30%)	latitude (100%), longitude (100%), cb3 (100%), t1b3 (90%), t2b3 (10%), v1b3 (60%), v2b3 (60%), cb4 (100%), t1b4 (20%), t2b4 (100%), p2b4 (90%), v1b4 (70%), v2b4 (70%), cb5 (100%), t1b5 (100%), t2b5 (10%), p1b5 (50%), p2b5 (20%), v1b5 (40%), cb6 (100%), t1b6 (30%), t2b6 (20%), p2b6 (40%), t1b1 (10%), t2b1 (10%), p1b1 (20%), p2b1 (100%), v1b1 (10%), v2b1 (10%), cb2 (100%), t1b2 (10%), p2b2 (50%), v1b2 (60%), v2b2 (100%)	latitude (90%), longitude (80%), cb4 (10%), t1b4 (10%), p2b4 (90%), v2b4 (10%), cb5 (10%), t2b5 (10%), v1b5 (20%), v2b5 (10%), cb6 (10%), t2b6 (90%), p1b6 (10%), v1b6 (60%), t1b1 (100%), t2b1 (90%), p1b1 (90%), v1b1 (90%), v2b1 (10%), t1b2 (90%), t2b2 (10%), p2b2 (10%), v1b2 (90%), v2b2 (80%) cb3 (100%), t1b3 (80%), t2b3 (100%) p1b3 (80%), v1b3 (100%), v2b3 (90%)	latitude (40%), longitude (10%), cb5 (10%), t2b5 (40%), p2b5 (30%), v1b5 (100%), v2b5 (80%), cb6 (10%), t1b6 (50%), t2b6 (30%), p1b6 (90%), p2b6 (100%), v1b6 (20%), v2b6 (90%), t1b1 (100%), p1b1 (30%), p2b1 (10%), v1b1 (20%), v2b1 (90%), cb2 (100%), t1b2 (30%), t2b2 (10%), p1b2 (10%), p2b2 (60%), v1b2 (50%), v2b2 (70%), t1b3 (10%), t2b3 (100%) p1b3 (60%), p2b3 (20%), v1b3 (100%), v2b3 (80%), cb4 (100%), t1b4 (90%), p1b4 (100%), v2b4 (90%)	latitude (80%), longitude (10%), cb6 (10%), t1b6 (70%), t2b6 (100%), p2b6 (40%), v1b6 (10%), v2b6 (90%), cb1 (100%), t1b1 (100%), p1b1 (80%), p2b1 (100%), v1b1 (100%), v2b1 (100%), cb2 (100%), t1b2 (90%), t2b2 (30%), p2b2 (60%), v1b2 (10%), v2b2 (10%), t1b3 (20%), p2b3 (20%), v2b3 (50%), t1b4 (50%), t2b4 (20%) p1b4 (100%), v1b4 (100%), v2b4 (80%), cb5 (100%), t1b5 (20%), t2b5 (90%), p1b5 (10%), p2b5 (60%), v1b5 (40%), v2b5 (100%)
Voting	latitude, cb1, t1b1, p1b1, cb2, p1b2, v1b2, v2b2, cb3, v1b3, t2b5, v1b5, v2b5, t1b6, t2b6, v1b6	latitude, cb2, t1b2, p2b2, v2b2, cb3, t2b3, p1b3, v1b3, cb4, t1b4, t2b4, p2b4, cb5, t1b5, p1b5, v2b5, cb6, t2b6, p2b6, v1b6, v2b6, t2b1, p1b1, p2b1, v1b1	latitude, longitude, cb3, t1b3, t2b3, v1b3, v2b3, cb4, t1b4, t2b4, p2b4, v1b4, v2b4, cb5, t1b5, t2b5, p1b5, p2b5, v1b5, cb6, t1b6, t2b6, p2b6, v1b6, v2b6, t1b1, t2b1, p1b1, p2b1, v1b1, v2b1, cb2, t1b2, p1b2, v1b2, v2b2	latitude, longitude, cb4, t1b4, p2b4, v2b4, cb5, t2b5, v1b5, v2b5, cb6, t2b6, p1b6, v1b6, t1b1, t2b1, p1b1, v1b1, v2b1, t1b2, t2b2, p2b2, v1b2, v2b2, cb3, t1b3, t2b3, p1b3, v1b3, v2b3	latitude, longitude, cb5, t2b5, p2b5, v1b5, v2b5, cb6, t1b6, t2b6, p1b6, p2b6, v1b6, v2b6, t1b1, p1b1, p2b1, v1b1, v2b1, cb2, t1b2, t2b2, p1b2, p2b2, v1b2, v2b2, t1b3, t2b3, p1b3, p2b3, v1b3, v2b3, cb4, t1b4, p1b4, v2b4	latitude, longitude, cb6, t1b6, t2b6, p2b6, v1b6, v2b6, cb1, t1b1, p1b1, p2b1, v1b1, v2b1, cb2, t1b2, t2b2, p2b2, v1b2, v2b2, t1b3, p2b3, v2b3, t1b4, t2b4, p1b4, v1b4, v2b4, cb5, t1b5, t2b5, p1b5, p2b5, v1b5, v2b5

* Population with 20 individuals, 500 generations, 10-fold cross-validation

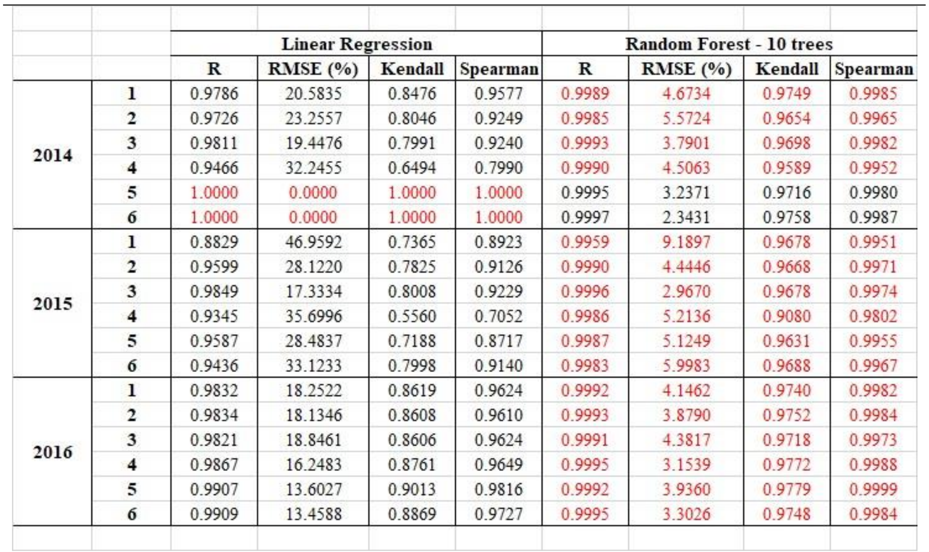
Table 8 Result of the analysis of the most relevant factors through the algorithms of Genetic Search, Evolutionary Search, Particle Swarm Optimization, Bee Search and Ant Search, for 20 solution candidates evolving in 500 generations, for the year 2015.



Method	2016.1	2016.2	2016.3	2016.4	2016.5	2016.6
Genetic Search	longitude (100%), cb1 (90%), p1b1 (90%), p2b1 (10%), v1b1 (100%), v2b1 (10%), cb2 (100%), t2b2 (70%), p1b2 (80%), p2b2 (20%), v1b2 (100%), cb3 (100%), t2b3 (10%), p1b3 (10%), p2b3 (70%), v2b3 (90%), t1b4 (10%), t2b4 (10%), v1b4 (100%), t1b5 (90%), t2b5 (100%), p1b5 (10%), p2b5 (10%), v1b5 (40%), v2b5 (100%), cb6 (100%), t2b6 (90%), p1b6 (100%), p2b6 (90%), v1b6 (30%), v2b6 (70%)	t1b2 (40%), p1b2 (20%), v1b2 (70%), t1b3 (30%), t2b3 (40%), p1b3 (50%), p2b3 (100%), v2b3 (60%), t1b4 (50%), p1b4 (30%), v1b4 (10%), v2b4 (10%), cb5 (100%), t1b5 (90%), p1b5 (50%), p2b5 (80%), t2b6 (30%), p1b6 (70%), p2b6 (30%), v1b6 (90%), v2b6 (100%), cb1 (100%), p1b1 (20%), p2b1 (100%)	latitude (90%), longitude (50%), cb3 (10%), t1b3 (10%), t2b3 (10%), p2b3 (50%), v1b3 (60%), v2b3 (10%), cb4 (10%), t2b4 (10%), v2b4 (60%), cb5 (100%), t1b5 (40%), t2b5 (40%), p1b5 (70%), p2b5 (80%), v1b5 (30%), v2b5 (10%), cb6 (10%), p1b6 (60%), v1b6 (100%), v2b6 (30%), cb1 (100%), t1b1 (10%), t2b1 (80%), p1b1 (80%), p2b1 (80%), v1b1 (100%), v2b1 (100%), cb2 (100%), t1b2 (90%), t2b2 (50%), p1b2 (90%), p2b2 (40%), v1b2 (20%), v2b2 (40%),	latitude (100%), t1b4 (70%), t2b4 (90%), p1b4 (30%), v1b4 (10%), v2b4 (90%), t1b5 (40%), t2b5 (10%), p1b5 (30%), p2b5 (80%), cb6 (70%), p2b6 (20%), v1b6 (90%), v2b6 (10%), cb1 (70%), t1b1 (80%), t2b1 (10%), p1b1 (10%), p2b1 (100%), v1b1 (90%), v2b1 (100%), cb2 (10%), t1b2 (10%), t2b2 (40%), p1b2 (40%), p2b2 (80%), v1b2 (20%), v2b2 (50%), cb3 (100%), t1b3 (40%), t2b3 (10%), p1b3 (10%), p2b3 (80%), v1b3 (100%), v2b3 (50%)	latitude (100%), longitude (70%), t1b5 (40%), t2b5 (90%), p1b5 (70%), v1b5 (10%), t1b6 (20%), cb1 (30%), t2b1 (100%), p1b1 (80%), v1b1 (100%), v2b1 (10%), t2b2 (20%), p2b2 (20%), v1b2 (70%), v2b2 (10%), cb3 (100%), t1b3 (40%), t2b3 (10%), v1b3 (60%), v2b3 (30%), cb4 (100%), p1b4 (20%), v1b4 (10%)	latitude (10%), cb6 (90%), t2b6 (60%), v2b6 (80%), cb1 (100%), t1b1 (60%), v2b1 (70%), cb2 (10%), t2b2 (90%), p2b2 (60%), v1b2 (80%), v2b2 (100%), cb3 (80%), t1b3 (90%), t2b3 (10%), v1b3 (20%), t1b4 (10%), p1b4 (100%), v1b4 (10%), cb5 (20%), t1b5 (70%), t2b5 (20%), p1b5 (40%), p2b5 (30%), v1b5 (90%), v2b5 (10%)
Evolutionary Search	latitude (20%), longitude (100%), cb1 (100%), p1b1 (100%), p2b1 (10%), v1b1 (100%), v2b1 (10%), cb2 (100%), t1b2 (10%), t2b2 (80%), p1b2 (90%), p2b2 (10%), v1b2 (100%), cb3 (100%), t2b3 (10%), p2b3 (40%), v2b3 (90%), t1b4 (10%), v1b4 (100%), t1b5 (90%), t2b5 (100%), p2b5 (10%), v1b5 (50%), v2b5 (100%), cb6 (100%), t2b6 (90%), p1b6 (100%), p2b6 (90%), v1b6 (60%), v2b6 (40%)	t1b2 (100%), t1b3 (100%), p1b3 (100%), p2b3 (100%), v2b3 (100%), t1b4 (100%), cb5 (100%), t1b5 (100%), p2b5 (100%), p2b6 (10%), v1b6 (100%), v2b6 (100%), cb1 (100%), p1b1 (10%), p2b1 (100%)	latitude (90%), longitude (10%), cb3 (10%), t1b3 (30%), p2b3 (60%), v1b3 (80%), cb4 (10%), t1b4 (30%), v2b4 (20%), cb5 (100%), t1b5 (40%), t2b5 (50%), p1b5 (90%), p2b5 (60%), v1b5 (10%), v2b5 (30%), cb6 (10%), p1b6 (60%), v1b6 (100%), v2b6 (10%), cb1 (100%), t1b1 (30%), t2b1 (70%), p1b1 (100%), p2b1 (100%), v1b1 (100%), v2b1 (100%), cb2 (100%), t1b2 (100%), t2b2 (30%), p1b2 (100%), p2b2 (70%), v2b2 (70%)	latitude (90%), v2b4 (100%), t1b5 (10%), p1b5 (100%), p2b5 (100%), cb6 (100%), p2b6 (100%), v1b6 (100%), v2b6 (100%), cb1 (100%), t1b1 (100%), p2b1 (100%), v1b1 (100%), v2b1 (100%), cb2 (100%), p1b2 (100%), p2b2 (100%), cb3 (100%), t1b3 (100%), t2b3 (100%), p2b3 (100%), v1b3 (100%), v2b3 (100%)	latitude (100%), longitude (70%), t2b5 (100%), p1b5 (90%), v1b5 (20%), t1b6 (10%), v2b6 (10%), cb1 (30%), t2b1 (100%), p1b1 (90%), v1b1 (100%), v1b2 (100%), cb3 (100%), t1b3 (10%), p2b3 (70%), v1b3 (100%), cb4 (100%)	cb6 (90%), t2b6 (100%), v2b6 (100%), cb1 (100%), t1b1 (100%), p1b1 (10%), cb2 (10%), t2b2 (100%), v1b2 (100%), v2b2 (100%), cb3 (100%), t1b3 (100%), p1b4 (100%), t1b5 (100%), v1b5 (100%)
PSO Search	longitude (100%), cb1 (90%), p1b1 (90%), p2b1 (10%), v1b1 (100%), v2b1 (10%), cb2 (100%), t2b2 (70%), p1b2 (80%), p2b2 (20%), v1b2 (100%), cb3 (100%), t2b3 (10%), p1b3 (10%), p2b3 (70%), v2b3 (90%), t1b4 (10%), t2b4 (10%), v1b4 (100%), t1b5 (90%), t2b5 (100%), p1b5 (10%), p2b5 (10%), v1b5 (40%), v2b5 (100%), cb6 (100%), t2b6 (90%), p1b6 (100%), p2b6 (90%), v1b6 (30%), v2b6 (70%)	t1b2 (40%), p1b2 (20%), v1b2 (70%), t1b3 (30%), t2b3 (40%), p1b3 (50%), p2b3 (100%), v2b3 (60%), t1b4 (50%), p1b4 (30%), v1b4 (10%), v2b4 (10%), cb5 (100%), t1b5 (90%), p1b5 (50%), p2b5 (80%), t2b6 (10%), p1b6 (10%), p2b6 (30%), v1b6 (90%), v2b6 (100%), cb1 (100%), p1b1 (20%) p2b1 (100%)	latitude (90%), longitude (50%), cb3 (10%), t1b3 (10%), t2b3 (10%), p2b3 (50%), v1b3 (60%), v2b3 (10%), cb4 (10%), t2b4 (10%), v2b4 (60%), cb5 (100%), t1b5 (40%), t2b5 (40%), p1b5 (70%), p2b5 (80%), v1b5 (30%), v2b5 (10%), cb6 (10%), p1b6 (60%), v1b6 (100%), v2b6 (30%), cb1 (100%), t1b1 (10%), t2b1 (80%), p1b1 (80%), p2b1 (80%), v1b1 (100%), v2b1 (100%), cb2 (100%), t1b2 (90%), t2b2 (50%), p1b2 (90%), p2b2 (40%), v1b2 (20%), v2b2 (40%)	latitude (100%), t1b4 (70%), t2b4 (90%), p1b4 (30%), v2b4 (90%), t1b5 (40%), t2b5 (10%), p1b5 (30%), p2b5 (90%), cb6 (70%), p2b6 (20%), v1b6 (100%), v2b6 (10%), cb1 (90%), t1b1 (90%), t2b1 (10%), p1b1 (10%), p2b1 (100%), v1b1 (100%), v2b1 (100%), cb2 (10%), t1b2 (10%), t2b2 (40%), p1b2 (60%), p2b2 (90%), v1b2 (30%), v2b2 (50%), cb3 (100%), t1b3 (40%), t2b3 (10%), p1b3 (20%), p2b3 (90%), v1b3 (100%), v2b3 (50%)	latitude (100%), longitude (70%), t1b5 (40%), t2b5 (90%), p1b5 (70%), v1b5 (10%), t1b6 (20%), cb1 (30%), t2b1 (100%), p1b1 (80%), v1b1 (100%), v2b1 (10%), t2b2 (20%), p2b2 (20%), v1b2 (70%), v2b2 (10%), cb3 (100%), t1b3 (40%), t2b3 (10%), v1b3 (60%), v2b3 (30%), cb4 (100%), p1b4 (20%), v1b4 (10%)	latitude (10%), cb6 (90%), t2b6 (60%), v2b6 (80%), cb1 (100%), t1b1 (60%), v2b1 (70%), cb2 (10%), t2b2 (90%), p2b2 (50%), v1b2 (80%), v2b2 (100%), cb3 (80%), t1b3 (90%), t2b3 (20%), v1b3 (20%), t1b4 (10%), p1b4 (100%), v1b4 (10%), cb5 (20%), t1b5 (70%), t2b5 (20%), p1b5 (40%), p2b5 (30%), v1b5 (90%), v2b5 (10%)
Bee Search	latitude (30%), longitude (90%), cb1 (90%), t2b1 (40%), p1b1 (90%), p2b1 (40%), v1b1 (70%), v2b1 (10%), cb2 (100%), t1b2 (30%), t2b2 (60%), p1b2 (70%), p2b2 (20%), v1b2 (100%), cb3 (100%), t2b3 (20%), p1b3 (10%), p2b3 (60%), v2b3 (50%), t1b4 (30%), t2b4 (50%), v1b4 (90%), t1b5 (90%), t2b5 (100%), p1b5 (10%), p2b5 (10%), v1b5 (20%), v2b5 (90%), cb6 (100%), t1b6 (10%), t2b6 (90%), p1b6 (90%), p2b6 (80%), v1b6 (30%), v2b6 (20%)	latitude (20%), longitude (40%), t1b2 (30%), p1b2 (20%), v1b2 (20%), v2b2 (10%), t1b3 (40%), t2b3 (40%), p1b3 (50%), p2b3 (90%), v1b3 (10%), v2b3 (60%), t1b4 (30%), p1b4 (20%), v1b4 (10%), v2b4 (20%), cb5 (100%), t1b5 (80%), p1b5 (40%), p2b5 (50%), p1b6 (20%), p2b6 (40%), v1b6 (90%), v2b6 (100%), cb1 (100%), p1b1 (30%), p2b1 (100%)	latitude (40%), longitude (70%), cb3 (10%), t1b3 (10%), p2b3 (50%), v1b3 (50%), v2b3 (10%), cb4 (10%), t1b4 (10%), t2b4 (10%), v2b4 (30%), cb5 (100%), t1b5 (30%), t2b5 (30%), p1b5 (70%), p2b5 (70%), v1b5 (30%), v2b5 (10%), cb6 (10%), p1b6 (60%), v1b6 (100%), cb1 (100%), t2b1 (50%), p1b1 (60%), p2b1 (80%), v1b1 (90%), v2b1 (90%), cb2 (100%), t1b2 (80%), t2b2 (60%), p1b2 (90%), p2b2 (40%), v1b2 (20%), v2b2 (40%)	latitude (90%), longitude (40%), t1b4 (10%), t2b4 (40%), p1b4 (30%), v2b4 (70%), t1b5 (30%), t2b5 (10%), p1b5 (20%), p2b5 (50%), v1b5 (20%), cb6 (70%), p2b6 (20%), v1b6 (80%), v2b6 (20%), cb1 (70%), t1b1 (50%), t2b1 (10%), p1b1 (10%), p2b1 (100%), v1b1 (90%), v2b1 (80%), cb2 (10%), t1b2 (20%), t2b2 (20%), p1b2 (30%), p2b2 (80%), v1b2 (40%), v2b2 (30%), cb3 (100%), t1b3 (60%), p1b3 (20%), p2b3 (40%), v1b3 (40%), v2b3 (50%)	latitude (100%), longitude (20%), t1b5 (30%), t2b5 (90%), p1b5 (70%), cb1 (30%), t2b1 (80%), p1b1 (70%), v1b1 (80%), v2b1 (10%), t2b2 (20%), v1b2 (70%), cb3 (100%), t1b3 (10%), t2b3 (10%), v1b3 (60%), cb4 (100%), p1b4 (20%), v1b4 (10%)	latitude (10%), longitude (30%), cb6 (90%), t2b6 (50%), v2b6 (80%), cb1 (100%), t1b1 (60%), v1b1 (10%), v2b1 (80%), cb2 (10%), t2b2 (90%), p2b2 (50%), v1b2 (60%), v2b2 (90%), cb3 (80%), t1b3 (90%), t2b3 (20%), p2b3 (10%), v1b3 (30%), t1b4 (10%), p1b4 (80%), v1b4 (10%) cb5 (20%), t1b5 (60%), t2b5 (20%), p1b5 (40%), v2b5 (20%)
Ant Search	longitude (100%), cb1 (90%), p1b1 (90%), p2b1 (10%), v1b1 (100%), v2b1 (10%), cb2 (100%), t2b2 (70%), p1b2 (80%), p2b2 (20%), v1b2 (100%), cb3 (100%), t2b3 (10%), p1b3 (10%), p2b3 (70%), v2b3 (90%), t1b4 (10%), t2b4 (10%), v1b4 (100%), t1b5 (90%), t2b5 (100%), p1b5 (10%), p2b5 (10%), v1b5 (40%), v2b5 (100%), cb6 (100%), t2b6 (90%), p1b6 (100%), p2b6 (90%), v1b6 (30%), v2b6 (70%)	t1b2 (40%), p1b2 (20%), v1b2 (70%), t1b3 (30%), t2b3 (40%), p1b3 (50%), p2b3 (100%), v2b3 (60%), t1b4 (50%), p1b4 (30%), v1b4 (10%), v2b4 (10%), cb5 (100%), t1b5 (90%), p1b5 (50%), p2b5 (80%), t2b6 (10%), p1b6 (10%), p2b6 (30%), v1b6 (90%), v2b6 (100%), cb1 (100%), p1b1 (20%), p2b1 (100%)	latitude (90%), longitude (50%), cb3 (10%), t1b3 (10%), t2b3 (10%), p2b3 (50%), v1b3 (60%), v2b3 (10%), cb4 (10%), t2b4 (10%), v2b4 (60%), cb5 (100%), t1b5 (40%), t2b5 (40%), p1b5 (70%), p2b5 (80%), v1b5 (30%), v2b5 (10%), cb6 (10%), p1b6 (60%), v1b6 (100%), v2b6 (30%), cb1 (100%), t1b1 (10%), t2b1 (80%), p1b1 (80%), p2b1 (80%), v1b1 (100%), v2b1 (100%), cb2 (100%), t1b2 (90%), t2b2 (50%), p1b2 (90%), p2b2 (40%), v1b2 (20%), v2b2 (40%)	latitude (100%), t1b4 (70%), t2b4 (90%), p1b4 (30%), v1b4 (20%), v2b4 (90%), t1b5 (40%), p1b5 (20%), p2b5 (70%), cb6 (70%), t2b6 (10%), p2b6 (10%), v1b6 (80%), cb1 (60%), t1b1 (80%), t2b1 (10%), p1b1 (10%), p2b1 (90%), v1b1 (100%), v2b1 (100%), t1b2 (10%), t2b2 (40%), p1b2 (30%), p2b2 (80%), v1b2 (20%), v2b2 (50%), cb3 (100%), t1b3 (30%), p1b3 (10%), p2b3 (80%), v1b3 (100%), v2b3 (50%)	latitude (100%), longitude (70%), t1b5 (40%), t2b5 (90%), p1b5 (70%), v1b5 (10%), t1b6 (20%), cb1 (30%), t2b1 (100%), p1b1 (80%), v1b1 (100%), v2b1 (10%), t2b2 (20%), p2b2 (20%), v1b2 (70%), v2b2 (10%), cb3 (100%), t1b3 (40%), t2b3 (10%), v1b3 (60%), v2b3 (30%), cb4 (100%), p1b4 (20%), v1b4 (10%)	latitude (10%), cb6 (90%), t2b6 (60%), v2b6 (80%), cb1 (100%), t1b1 (60%), v2b1 (70%), cb2 (10%), t2b2 (90%), p2b2 (60%), v1b2 (80%), v2b2 (100%), cb3 (80%), t1b3 (90%), t2b3 (10%), v1b3 (20%), t1b4 (10%), p1b4 (100%), v1b4 (10%) cb5 (20%), t1b5 (70%), t2b5 (20%), p1b5 (40%), p2b5 (30%), v1b5 (90%), v2b5 (10%)
Voting	longitude, cb1, p1b1, p2b1, v1b1, v2b1, cb2, t2b2, p1b2, p2b2, v1b2, cb3, t1b3, t2b3, p1b3, p2b3, v1b3, v2b3, t1b4, t2b4, v1b4, t1b5, t2b5, p1b5, p2b5, v1b5, v2b5, cb6, t2b6, p1b6, p2b6, v1b6, v2b6	t1b2, p1b2, v1b2, t1b3, t2b3, p1b3, p2b3, v2b3, t1b4, p1b4, v1b4, v2b4, cb5, t1b5, p1b5, p2b5, p1b6, p2b6, v1b6, v2b6, cb1, p1b1, p2b1	latitude, longitude, cb3, t1b3, t2b3, p2b3, v1b3, v2b3, cb4, t2b4, v2b4, cb5, t1b5, t2b5, p1b5, p2b5, v1b5, v2b5, cb6, p1b6, v1b6, v2b6, cb1, t1b1, t2b1, p1b1, p2b1, v1b1, v2b1, cb2, t1b2, t2b2, p1b2, p2b2, v1b2, v2b2	latitude, t1b4, t2b4, p1b4, v2b4, t1b5, t2b5, p1b5, p2b5, cb6, p2b6, v1b6, v2b6, cb1, t1b1, t2b1, p1b1, p2b1, v1b1, v2b1, cb2, t1b2, t2b2, p1b2, p2b2, v1b2, v2b2, cb3, t1b3, t2b3, p1b3, p2b3, v1b3, v2b3	latitude, longitude, t1b5, t2b5, p1b5, v1b5, t1b6, cb1, t2b1, p1b1, v1b1, v2b1, t2b2, p2b2, v1b2, v2b2, cb3, t1b3, t2b3, v1b3, v2b3, cb4, p1b4, v1b4	latitude, cb6, t2b6, v2b6, cb1, t1b1, v2b1, cb2, t2b2, p2b2, v1b2, v2b2, cb3, t1b3, t2b3, v1b3, t1b4, p1b4, v1b4, cb5, t1b5, t2b5, p1b5, p2b5, v1b5, v2b5

* Population with 20 individuals, 500 generations, 10-fold cross-validation

Table 9 Result of the analysis of the most relevant factors through the algorithms of Genetic Search, Evolutionary Search, Particle Swarm Optimization, Bee Search and Ant Search, for 20 solution candidates evolving in 500 generations, for the year 2016.



		Linear Regression				Random Forest - 10 trees			
		R	RMSE (%)	Kendall	Spearman	R	RMSE (%)	Kendall	Spearman
2014	1	0.9786	20.5835	0.8476	0.9577	0.9989	4.6734	0.9749	0.9985
	2	0.9726	23.2557	0.8046	0.9249	0.9985	5.5724	0.9654	0.9965
	3	0.9811	19.4476	0.7991	0.9240	0.9993	3.7901	0.9698	0.9982
	4	0.9466	32.2455	0.6494	0.7990	0.9990	4.5063	0.9589	0.9952
	5	1.0000	0.0000	1.0000	1.0000	0.9995	3.2371	0.9716	0.9980
	6	1.0000	0.0000	1.0000	1.0000	0.9997	2.3431	0.9758	0.9987
2015	1	0.8829	46.9592	0.7365	0.8923	0.9959	9.1897	0.9678	0.9951
	2	0.9599	28.1220	0.7825	0.9126	0.9990	4.4446	0.9668	0.9971
	3	0.9849	17.3334	0.8008	0.9229	0.9996	2.9670	0.9678	0.9974
	4	0.9345	35.6996	0.5560	0.7052	0.9986	5.2136	0.9080	0.9802
	5	0.9587	28.4837	0.7188	0.8717	0.9987	5.1249	0.9631	0.9955
	6	0.9436	33.1233	0.7998	0.9140	0.9983	5.9983	0.9688	0.9967
2016	1	0.9832	18.2522	0.8619	0.9624	0.9992	4.1462	0.9740	0.9982
	2	0.9834	18.1346	0.8608	0.9610	0.9993	3.8790	0.9752	0.9984
	3	0.9821	18.8461	0.8606	0.9624	0.9991	4.3817	0.9718	0.9973
	4	0.9867	16.2483	0.8761	0.9649	0.9995	3.1539	0.9772	0.9988
	5	0.9907	13.6027	0.9013	0.9816	0.9992	3.9360	0.9779	0.9999
	6	0.9909	13.4588	0.8869	0.9727	0.9995	3.3026	0.9748	0.9984

Table 10 Prediction results for linear regression and Random Forest with 10 trees, considering training sets of 4665 instances and test sets of 1555 instances, obtained from interpolation of real data. As quality metrics, the correlation index R, the RMSE (%) error and the correlation indexes of Kendall and Spearman were considered. Results are presented as sample average and standard deviation (SD). The best results are highlighted in red.

Table 3 shows the results for support vector machines (SVM), with linear (or degree 1 polynomial), 2- and 3-degree polynomial and RBF kernels. The results with linear kernel somewhat resemble the results with linear regression, although a little better and more stable (low standard deviation): correlation index R of 0.87, with standard deviation of 0.07, and RMSE (%) 42% with a standard deviation of 21%. The training time for the linear kernel, however, was very variable and relatively long: 1622ms with a standard deviation of 703ms. The linear kernel was the slowest SVM configuration of all evaluated. The results with degree 2 and 3 polynomial kernels and RBF were considered very good: very low R correlation indices, practically 1.00, with very low standard deviations (4.00×10^{-5} for degree polynomial kernels 2 and 3, and 9.00×10^{-5} for the RBF kernel). The RMSE results (%) were also well below 5%: 0.6% with standard deviation 0.2% for grade 2 and 3 polynomial kernels, and 0.7% with standard deviation 0.3% for the RBF kernel. The shortest average training time was obtained with the RBF kernel: 490ms with a standard deviation of 124ms. Figure 5(e) shows the boxplot of the distribution of the correlation index R. It can be noted that there is a concentration between 0.84 and 0.96 for the linear kernel (degree 1 polynomial). It is also evident that the 2- and 3-degree polynomial and RBF kernels achieved excellent results: concentration at 1.00 with practically zero deviation. From a statistical point of view, the results obtained for the correlation index R for the 2- and 3-degree polynomial and RBF kernels are equivalent. The same is true for the RMSE (%) error, as shown in the boxplot in Figure 6(e), with a slight disadvantage for the RBF kernel, which has a point out of 10%. Analyzing the boxplot in Figure 7(e), despite the statistical equivalence between the methods with 2- and 3-degree polynomial and RBF kernels, it is clear that the training time is lower and more stable for the RBF kernel, which is the most adequate SVM configuration to solve the problem.

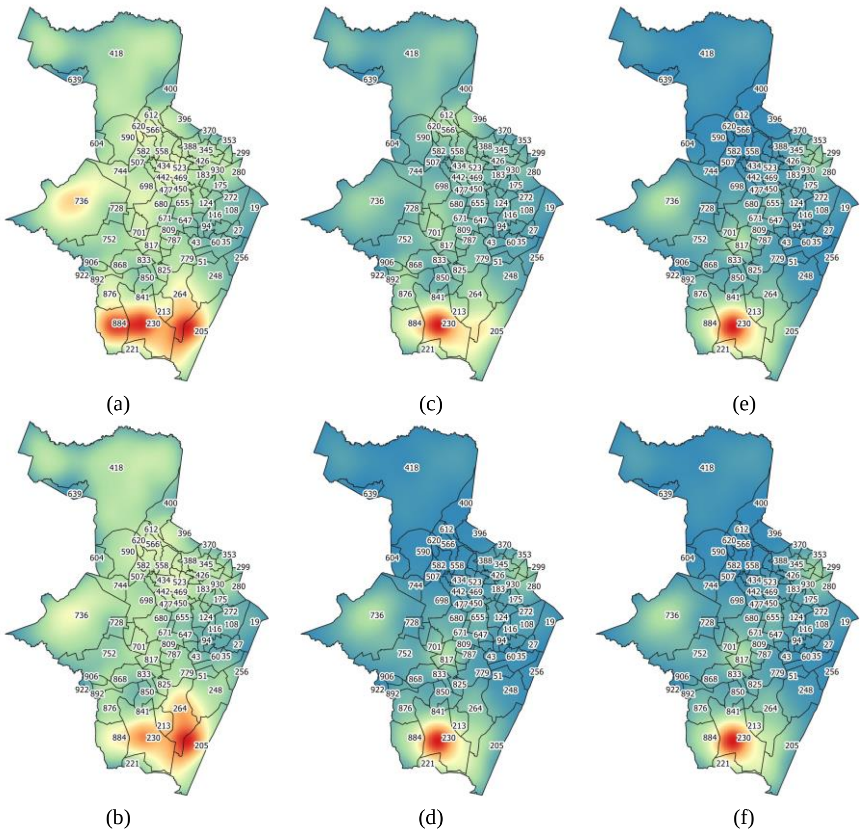
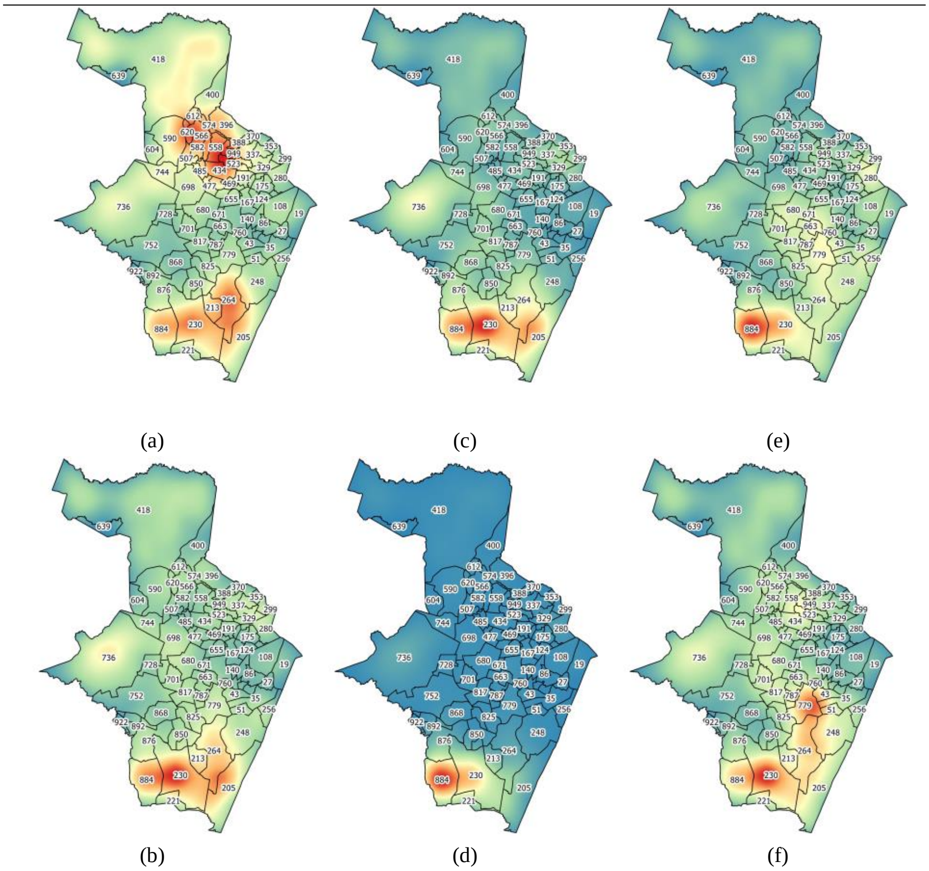


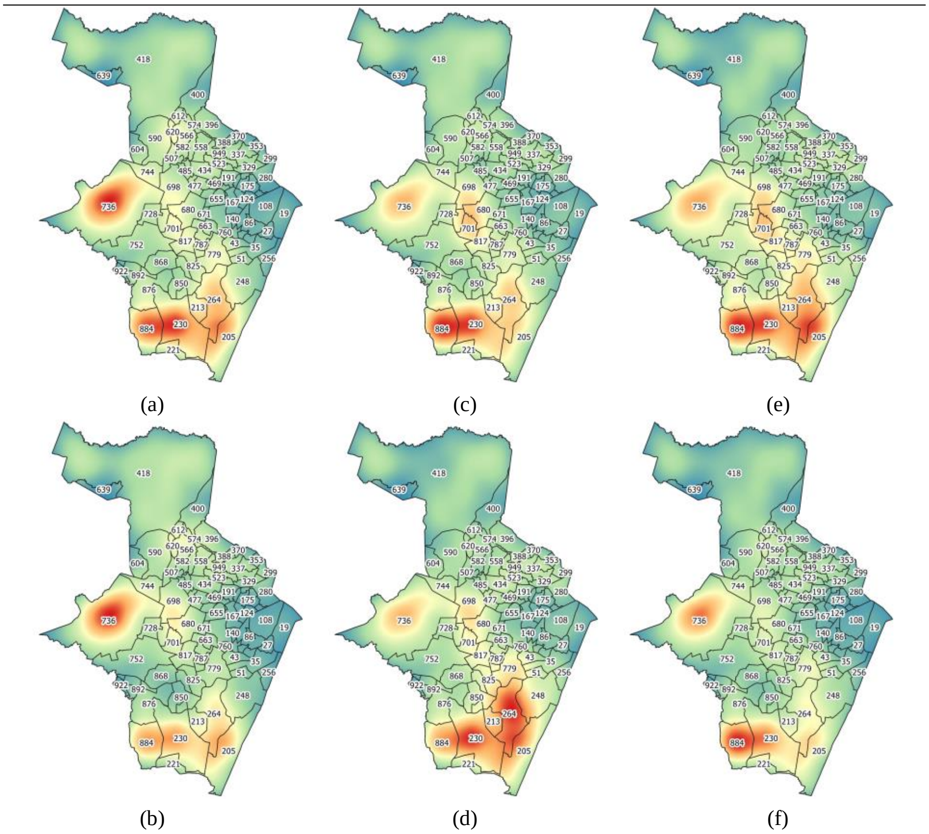
Fig. 8 Prediction results for the Random Forest with 10 trees, considering training sets of 4665 instances and test sets of 1555 instances, obtained from interpolation of real data, for bimonths 1 (January to February, cf. a), 2 (March to April, cf. b), 3 (May to June, cf. c), 4 (July to August, cf. d), 5 (September to October, cf. e), and 6 (November to December, cf. f), for the year 2014. Resolution of 120 dpi, scale 1:182666. The pseudocolor scale is inverse spectral: low numbers of cases are represented close to blue (0-1 cases per neighbourhood); intermediate situations tend to be between green and yellow (2-3 cases per block); high numbers of cases tend to red (greater than 3 cases per block).



(a) (c) (e)

(b) (d) (f)

Fig. 9 Prediction results for the Random Forest with 10 trees, considering training sets of 4665 instances and test sets of 1555 instances, obtained from interpolation of real data, for bimonths 1 (January to February, cf. a), 2 (March to April, cf. b), 3 (May to June, cf. c), 4 (July to August, cf. d), 5 (September to October, cf. e), and 6 (November to December, cf. f), for the year 2015. Resolution of 120 dpi, scale 1:182666. The pseudocolor scale is inverse spectral: low numbers of cases are represented close to blue (0-1 cases per neighbourhood block); intermediate situations tend to be between green and yellow (2-3 cases per block); high numbers of cases tend to red (greater than 3 cases per block).



(a) (c) (e)

(b) (d) (f)

Fig. 10 Prediction results for the Random Forest with 10 trees, considering training sets of 4665 instances and test sets of 1555 instances, obtained from interpolation of real data, for bimonths 1 (January to February, cf. a), 2 (March to April, cf. b), 3 (May to June, cf. c), 4 (July to August, cf. d), 5 (September to October, cf. e), and 6 (November to December, cf. f), for the year 2016. Resolution of 120 dpi, scale 1:182666. The pseudocolor scale is inverse spectral: low numbers of cases are represented close to blue (0-1 cases per neighbourhood block); intermediate situations tend to be between green and yellow (2-3 cases per block); high numbers of cases tend to red (greater than 3 cases per block).

Table 5 displays the results for single-layer and deep echo state machines (ESM and Deep ESM, respectively), with 1, 2, 5 and 10 hidden layers. ESM and Deep ESM are neural networks trained in the same way as ELMs, but their neurons are different: they are called reservoirs and are influenced not only by inputs, but also by values passed from the outputs (feedback). Thus, it was expected that its results would be equal to or better than those obtained with the ELM, but they were much lower, making this approach really inadequate to the problem: the mean values of the correlation index R are practically stagnant at 0.5, with a deviation of 0.3, while the RMSE (%) error is on the order of 10^7 with even larger deviations, on the order of 10^9, while the training time increases a lot with the inclusion of more layers, as also shows the boxplot in Figure 7(d). The boxplots of Figures 5(d) and 6(d) not only confirm the inadequacy of ESM and Deep ESM in solving the problem, but also indicate that these methods, regarding the correlation index R and RMSE (%) error, are statistically equivalent.

Table 6 illustrates the results for Random Forest, for configurations of 10 to 100 trees, step of 10 trees. The results are very good and are repeated for all tested configurations: correlation index average R of 0.9999 with a standard deviation of 0.0001, that is, practically 1.00 in all experiments; RMSE (%) mean 0.6% with standard deviation 0.6% for 10 trees, 0.6% with standard deviation 0.5% for 20 trees, and 0.5% with deviation 0.5% from 30 trees onwards. The boxplot in Figure 5(f) shows that, for the correlation index R, all tested configurations are equivalent and can be very good, with results practically concentrated in 1.00 and with points outside still acceptable, between 0.990 and 0.994. The boxplot in Figure 6(f) shows the distribution of the results of the RMSE (%) error, showing that the results are concentrated between 0 and 1%, therefore below the limit of 5% that was adopted in this work, with points out between 7% and 8%. The results also show that the Random Forest settings are statistically equivalent. Thus, training time can be considered as a tiebreaker. Table 6 shows that the configuration with 10 trees has the fastest training: 30ms average with a standard deviation of 9ms. The boxplot in Figure 7(f) shows that the training time increases linearly with the increase in the number of trees, as well as the times start to spread more. In the 10-tree configuration, training times are concentrated around 25ms with very little scattering, with a point off at 200ms. Thus, the Random Forest configuration with 10 trees is the most suitable for the problem because it is the fastest training, although all the tested configurations are statistically equivalent and equally good according to the correlation index R and the RMSE (%) error.

4.2 Most relevant features selected by the Artificial Expert Committee

Table 7 shows the prediction results for the six bimesters of 2014 considering the year 2013. Analyzing the result of the artificial analysts committee, it can be seen that, for the prediction of the January and February bimester 2014 (2014.1), the following factors were considered most relevant: geographic position (latitude and longitude); temperature (t2b1) and wind speed (v2b1) in February 2013; number of arboviruses cases in March and April 2013 (cb2); rainfall in April 2013 (p2b2); wind speeds in March (v1b2) and May (v1b3), 2013; number of cases in July and August 2013 (cb4); temperature in August 2013 (t2b4); wind speeds in July (v1b4) and August (v2b4) of 2013; number of cases in September and October 2013 (cb5); temperature (t2b5) and wind speed (v2b5) in October 2013; number of cases in November and December 2013 (cb6) and temperature (t2b6) in December 2013. The importance of geographic position and velocities, temperatures and number of cases from March to December 2013 in predicting the number of cases is evident in the months of January and February 2014.

In the prediction of the two-month period from March to April 2014 (2014.2), as shown in Table 7, the importance of the following factors is evident: geographic position (longitude); temperatures in bimonths 2 (March to April 2013), 4 (July to August 2013) and 5 (September and October 2013); pressure in periods 2 to 5 (from March to October 2013); wind speeds in the two months of 2 to 4 (March to July 2013), 6 (November and December 2013) and 1 (January and February 2014); and number of cases from the 4th to the 1st period (July 2013 to February 2014).

Regarding the prediction of the period from May to June 2014 (2014.3), as shown in Table 7, the following factors appeared as the most relevant: geographic position (latitude and longitude); temperature from 3 to 4 months (May to August 2013) and from 6 to 2 (November 2013 to April 2014); wind speeds from the 4th to the 2nd period (July 2013 to April 2014); rainfall in the 5th (September to October, 2013) and 1st (January to February,

2014) periods; and number of cases throughout the period, except for bimester 5 (September to October 2013).

Considering the forecasting of the two-month period from July to August 2014 (2014.4), as shown in Table 7, the following factors stood out: geographic position (latitude and longitude); temperature from 4 to 5 (July to October 2013) and 1 to 3 (January to June 2014); rainfall in the quarters 4 to 6 (July to December 2013) and 3 (May to June 2014); wind speeds from the 5th to the 3rd quarter (September 2013 to June 2014); and number of cases in bimesters 1 (January to February 2014) and 3 (May to June 2014).

Taking into account predicting the two-month period from September to October 2014 (2014.5), as shown in Table 7, the following factors were highlighted: geographic position (latitude); temperature in bimesters 5 (September to October 2013), and from 1 to 4 (January to August 2014); rainfall in the 5th (September to October, 2013) and 3rd to 4th (May to August, 2014) periods; wind speed in all six quarters; and number of cases in bimesters 1 (January to February 2014) and from 3 to 4 (May to August 2014).

In predicting the two-month period from November to December 2014 (2014.6), as shown in Table 7, the following factors stood out: geographic position (latitude); rainfall in the 6th (November to December, 2013) and 3rd to 4th (May to August, 2014) periods; wind speeds in the 6th (November to December, 2013) and 3rd to 4th (May to August, 2014) periods; temperature in bimonths 2 (March to April 2014) and from 4 to 5 (July to October 2014); and number of arboviruses cases in the period 1 (January to February 2014) and from 3 to 5 (May to August 2014).

Table 8 shows the prediction results for the six quarters of 2015 considering the year 2014. Analyzing the result of the consensus or voting in the artificial analysts committee, we have, for the prediction of the January bimester to February 2015 (2015.1), the following factors considered most relevant: geographic position (latitude); number of arboviruses cases in the period 1 to 3 (January to June 2014); temperature in bimesters 1 (January 2014), and from 5 to 6 (September to December 2014); rainfall in the two months from 1 to 2 (January to April 2014); and wind speeds in the two months from 2 to 3 (March to June 2014) and from 5 to 6 (September to December 2014).

Regarding the prediction of the two-month period from March to April 2015 (2015.2), as shown in Table 8, the following factors stood out: geographic position (latitude); number of arboviruses cases in the two-month period from 2 to 6 (March to December 2014); temperature and rainfall throughout practically the entire period; and wind speeds in the two months from 2 to 3 (March to April 2014) and from 5 to 1 (September 2014 to February 2015).

When predicting the number of cases in the period from May to June 2015 (2015.3), according to Table 8, the following factors were considered more relevant: geographic position (latitude and longitude); number of arboviruses cases in the period 3 to 6 (May to December 2014) and 2 (March to April 2015); temperature and wind speed throughout the period; and rainfall from the 4th to the 2nd period (July 2014 to April 2015).

The prediction of the number of cases from July to August 2015 (2015.4), as shown in Table 8, depended predominantly on the following factors: geographic position (latitude and longitude); number of arboviruses cases from 4 to 6 (July to December 2014) and 3 (May to June 2015); temperature and wind speed throughout the period; and rainfall in the 4th (July to August 2014) and 6th to 3rd (December 2014 to June 2015) periods.

Additionally, the prediction of the number of arbovirus cases from September to October 2015 (2015.5), as shown in Table 8, depended predominantly on the following factors: geographic position (latitude and longitude); number of arboviruses cases in the period from

5 to 6 (September to December 2014), 2 (March to April 2015) and 4 (July to August 2015); and rainfall, temperature and wind speed throughout the period.

The prediction of the number of arbovirus cases from November to December 2015 (2015.6), as shown in Table 8, depended mostly on the following factors: geographic position (latitude and longitude); number of arboviruses cases in the period from 6 to 2 (December 2014 to April 2015) and 5 (September to October 2015); and rainfall, temperature and wind speed throughout the period.

Table 9 shows the prediction results for the six bimesters of 2016 considering the year 2015. Analyzing the result of the consensus or voting in the artificial analysts committee, we have, for the prediction of the January bimester to February 2016 (2016.1), the following factors considered most relevant: geographic position (longitude); number of arboviruses cases in the two months of 1 to 3 (January to June 2015) and 6 (November to December 2015); temperature from 2 to 6 months (March to December 2015); rainfall in the two months from 1 to 3 (January to June 2015) and from 5 to 6 (September to December 2015); and wind speed throughout the period considered for prediction.

In the prediction of the two-month period from March to April 2016 (2016.2), as shown in Table 9, the importance of the following factors is evident: temperature in the two-month period from 2 to 5 (March to October 2015); wind speeds in the two months of 2 to 4 (March to August 2015) and 6 (November to December 2015); and rainfall throughout the period. Interestingly, geographic position was not considered relevant.

Regarding the predicting the May-June 2016 period (2016.3), as shown in Table 9, the following factors were considered most relevant: geographic position (latitude and longitude); temperature in the quarters from 3 to 5 (May to October 2015) and from 1 to 2 (January to April 2016); and rainfall in bimonths 3 (May to June 2015) and from 5 to 2 (September 2015 to April 2016). Wind speed and the number of arbovirus cases were considered important throughout the period considered for prediction.

Considering forecasting the two-month period from July to August 2016 (2016.4), as shown in Table 9, the following factors were considered most relevant: geographic position (latitude); temperature from 4 to 5 (July to October 2015) and from 1 to 3 (January to June 2016); wind speeds in the 4th (July to August 2015) and 6th to 3rd (November 2015 to June 2016) periods; and number of arboviruses cases in the 6th to 3rd period (November 2015 to June 2016). Rainfall was considered relevant for the entire prediction period.

In the prediction of the two-month period from September to October 2016 (i.e. 2016.5), according to Table 9, the following factors were considered most relevant: geographic position (latitude and longitude); temperature in the quarters from 5 to 6 (September to December 2015) and from 2 to 3 (March to June 2016); wind speed in the 5th period (September to October 2015) and from 1st to 4th (January to August 2016); and the number of cases in bimesters 1 (January to February 2016) and from 3 to 4 (May to August 2016). Rainfall was considered important for the entire prediction period.

The prediction of the number of arbovirus cases from November to December 2016 (2016.6), as shown in Table 9, depended mostly on the following factors: geographic position (latitude); number of arboviruses cases in the period from 6 to 3 (November 2015 to June 2016) and 5 (September to October 2016); rainfall in the 2nd period (March to April 2016) and from 4th to 5th (July to October 2016). Temperature and wind speed were considered relevant for the entire prediction period.

4.3 Spatio-temporal analysis

Prediction results using Random Forest tend to qualitatively agree with linear regression results, but show more concentrated areas with smoother boundaries. As the metrics of Table 10, especially the RMSE (%) error, point to the superiority of Random Forest over linear regression for this problem, it is expected that the results of predictions with linear regression will point to some regions a little better and others a little worse than they really are. The prediction of the 2014 first bimester is shown in Figure 8(a). The situation in Várzea neighborhood (region 736) appears clearly, and it is evident that there are a reasonable number of cases in this neighborhood. Forecastings for the remaining bimesters are shown in Figures 8(b), 8(c), 8(d), 8(e), and 8(f).

The prediction made with Random Forest for the first two months of 2015 clearly shows the emergence of cases of arboviruses in the north-central neighborhoods, as shown in Figure 9(a). Furthermore, the result with Random Forest better illustrates the existence of two different peaks in the north-central neighborhoods: one between Vasco da Gama (region 558) and Casa Amarela (region 434), and the other between Dois Irmãos (region 590), Dois Unidos (region 396) and Brejo da Guabiraba (region 612). The prediction for bimester 2 shows greater control of arboviruses cases, with a general decrease in cases, although there is still a concentration of cases in Ibura (region 230), spreading between Cohab (region 884), Jordão (region 221), Ipsep (region 213), Boa Viagem (region 205), and towards Pina (region 248), cf. Figure 9(b). Figure 9(c), in relation to Figure 9(d), shows that the Random Forest predictor managed to show a decrease in the area of cases in Boa Viagem in bimester 3 when compared to 2. Figure 9(d) shows that, for bimester 4, the number of cases in Cohab (region 884) exceeded that of Ibura (region 230). This situation remains constant in bimester 5, as illustrated in Figure 9(e). The prediction of bimester 6 shows that more cases arose, spreading from the south to the center, from Ibura (region 230), which again has the largest number of cases, to Afogados (region 779), Mustardinha (region 787), Bongi (region 809), Brasília Teimosa (region 256), and São José (region 35), passing through Ipsep (region 213), Boa Viagem (region 205), and Pina (region 248), as shown in Figure 9(f). In this case, the situation was also well illustrated by the prediction with Random Forest, as shown in Figure 9(f).

The prediction made with Random Forest for the bimester 1 of 2016 very clearly shows Várzea (region 736) as a peak of arboviruses, along with Ibura (region 230), Cohab (region 884), and, in a weaker way, Boa Viagem (region 205), as shown in Figure 10(a). The prediction of bimester 2, illustrated in Figure 10(b), shows greater control in the number of cases in the southern districts, but Várzea (region 736) persists practically in the same way as in bimester 1, with a few cases scattered in the Midwestern neighborhoods. The prediction for bimester 3 shows again peaks of cases in Cohab (region 884) and Ibura (region 230), and a slight spread of cases in Ibura (region 230) towards Boa Viagem (region 205), Ipsep (region 213) and Pina (region 248), but with greater control of cases in Várzea (region 736) and in the central-western districts, as shown in Figure 10(c). The prediction of bimester 4 illustrates the increase in cases in the direction of Boa Viagem (region 205), Ipsep (region 213) and Pina (region 248), confirming the trend observed in bimester 3, according to Figure 10(d). The prediction for bimester 5 shows a slight improvement compared to bimester 4, as shown in Figure 10(e). The prediction for bimester 6 shows a general decrease in cases, but there is still a concentration of cases in Cohab (region 884) and Ibura (region 230), and the number of cases in Várzea (region 736) increases a little, as shown in Figure 10(f).

The concentration of the highest occurrence of cases in the western region, which is more humid and abundant in green areas, points to the hypothesis that this situation is related to the more favorable development of Aedes aegypti in green areas close to urban regions. In contrast, the concentration of high numbers of cases in the southwestern region of the city, especially in low-income neighborhoods, suggests a strong relationship with infrastructure problems, especially with regular access to water. This leads the inhabitants of this region to maintain irregular water reservoirs, which become potential breeding grounds for mosquitoes.

5 **Conclusion**

In this work, we propose a methodology to build predictive models, based on machine learning, to predict the spatio-temporal distribution of diseases from databases of reported cases and geographic information bases. As a case study, we applied the methodology to the prediction of arboviruses, specifically dengue, chikungunya and Zika, whose vector is the Aedes aegypti mosquito, with data from the City of Recife, Brazil, from 2013 to 2016, which included the 2015 Zika virus outbreak associated with the occurrence of malformations in newborns. We used open and anonymous data, available for search on the Recife City Open Data Portal and obtained from the National Notification System, SINAN, of the Unified Health System (SUS) in Brazil. We also use the following geographic, climatic and environmental information: wind speeds, temperatures and precipitation, obtained from meteorological information systems. Since the density of environmental stations is very small, the distribution of these climatic and environmental variables was estimated in a regular grid using interpolation. The prediction of cases was performed every two months, considering data from the last 12 months.

The best case prediction results were obtained with Random Forest regression. Due to the low computational cost and stability of the results, we chose Random Forest with 10 trees. Pearson's correlation coefficient values were above 0.99, while the RMSE (%) remained below 6%. Kendall and Spearman indexes also remained high: their values were greater than 0.99 for Spearman (close to the Pearson coefficient) and greater than 0.90 for Kendall (a more rigorous index than Pearson and Spearman index). The superior performance of Random Forest when compared to other regression models shows that the regression problem is difficult to generalize, given that Random Forest is based on decision tree committees organized as bagging and the regression is performed by a weighted average of the results of the different decision trees that make up the model.

Qualitative spatio-temporal prediction results also show that it is possible to observe the dynamics of arboviruses transmitted by the Aedes aegypti mosquito with relative precision. The case distributions obtained are smooth, as the numbers of cases obtained were concentrated in the neighborhoods and then interpolated. However, the qualitative results suggest that more accurate distributions can be obtained if an IoT-based approach is adopted for measuring wind speeds, temperatures and rainfall in different neighborhood locations, in order to increase data density. Similarly, approximate information about the region of occurrence of the case could be used. However, this must be done with care to avoid exposure and location of the patient's place of residence.

In this work, we also propose an Artificial Experts Committee to select the most relevant factors for prediction. This committee is composed of meta-heuristic search and optimization methods, namely: genetic algorithms, particle swarm optimization, bee colony optimization, and ant colony optimization. These artificial specialists were trained using a decision tree as an objective function, using populations of the same size evolving in the same number of generations. The most relevant factors for prediction are defined by voting.

In this way, the specificities of each virtual specialist end up being compensated and the most relevant factors for prediction are better defined, which can help public health managers in defining the most important factors for the control of a vector. In the case of dengue, chikungunya and Zika, as the vector is Aedes aegypti, the great influence of factors such as wind speed, temperatures and rainfall point to the strong seasonality of these diseases, which, in turn, indicates their dependence of the vector, ie the mosquito and its life cycle. These and other factors, such as the influence of past numbers of cases, can be very useful for the planning and execution of public policies aimed at improving the health infrastructure and planning and controlling the vector.

As future work, we intend to build a client-server web system to support the spatiotemporal prediction of arboviruses, infectious diseases and other diseases of interest. This system will have the ability to make spatial and temporal predictions from the insertion of multiple georeferenced databases, in addition to indicating the most relevant factors for prediction from the Artificial Experts Committee. It is our intention that this system be made available as free software, to adapt to different realities, in different countries and regions, so that public health authorities can have quick access to information that support decision-making.

Acknowledgements

The authors are grateful to the Brazilian research agencies FACEPE, FAPESP and CNPq, and to UKRI, research grant number NE/T013664/1, for the partial financial support of this research.

Conflict of Interest

All authors declare they have no conflicts of interest.

Compliance with Ethical Standards

This study was funded by the Federal University of Pernambuco, the Brazilian research agencies FACEPE, FAPESP and CNPq, and the University College London held UKRI research grant number NE/T013664/1.

References

1. L. Akil and H. A. Ahmad. Salmonella infections modelling in Mississippi using neural network and geographical information system (GIS). *BMJ Open*, 6(3), 2016. ISSN 2044-6055. doi: 10.1136/bmjopen-2015-009255. URL https://bmjopen.bmj.com/content/6/3/e009255.
2. G. Albrieu-Llinás, M. O. Espinosa, A. Quaglia, M. Abril, and C. M. Scavuzzo. Urban environmental clustering to assess the spatial dynamics of Aedes aegypti breeding sites. *Geospatial Health*, 13(1), 2018.
3. O. S. Baquero, L. M. R. Santana, and F. Chiaravalloti-Neto. Dengue forecasting in São Paulo city with generalized additive models, artificial neural networks and seasonal

autoregressive integrated moving average models. *PLOS ONE*, 13(4):1–12, 04 2018. doi: 10.1371/journal.pone.0195065. URL https://doi.org/10.1371/journal.pone.0195065.

4. J. C. A. Barata and M. S. Hussein. The Moore–Penrose Pseudoinverse: A Tutorial Review of the Theory. *Brazilian Journal of Physics*, 42(1):146–165, 2012. ISSN 16784448. doi: 10.1007/s13538-011-0052-z. URL http://dx.doi.org/10.1007/s13538-0110052-z.
5. P. L. Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE transactions on Information Theory*, 44(2):525–536, 1998.
6. M. A. Beketov, Y. A. Yurchenko, O. E. Belevich, and M. Liess. What environmental factors are important determinants of structure, species richness, and abundance of mosquito assemblages? *Journal of Medical Entomology*, 47(2):129–139, 2014.
7. J. D. Beltrán, A. Boscor, W. P. dos Santos, T. Massoni, and P. Kostkova. ZIKA: A New System to Empower Health Workers and Local Communities to Improve Surveillance Protocols by E-learning and to Forecast Zika Virus in Real Time in Brazil. In *Proceedings of the 2018 International Conference on Digital Health*, pages 90–94. ACM, 2018.
8. G. S. Bhunia and P. K. Shit. *Geospatial Analysis of Public Health*. Springer International Publishing, 2019.
9. C. Braga, C. F. Luna, C. M. Martelli, W. V. d. Souza, M. T. Cordeiro, N. Alexander, M. d. F. P. M. d. Albuquerque, J. C. Silveira-Jr., and E. T. Marques. Seroprevalence and risk factors for dengue infection in socio-economically distinct areas of Recife, Brazil. *Acta Tropica*, 113(3):234–240, 2010.
10. I. A. Braga and D. Valle. Aedes aegypti: histórico do controle no Brasil. *Epidemiologia e Serviços de Saúde*, 16(2):113–118, 2007.
11. I. A. Braga, J. B. P. Lima, S. d. S. Soares, and D. Valle. Aedes aegypti resistance to temephos during 2001 in several municipalities in the states of Rio de Janeiro, Sergipe, and Alagoas, Brazil. *Memórias do Instituto Oswaldo Cruz*, 99(2):199–203, 2004.
12. P. Brasil, J. P. Pereira-Jr, C. Raja-Gabaglia, L. Damasceno, M. Wakimoto, R. M. Ribeiro-Nogueira, P. C. d. Sequeira, A. Machado-Siqueira, L. M. A. d. Carvalho, and D. C. d. Cunha. Zika virus infection in pregnant women in Rio de Janeiro: preliminary report. *New England Journal of Medicine*, 2016.
13. A. L. Buczak, B. Baugher, L. J. Moniz, T. Bagley, S. M. Babin, and E. Guven. Ensemble method for dengue prediction. *PloS One*, 13(1):e0189988, 2018.
14. C. W. Cardoso, I. A. Paploski, M. Kikuti, M. S. Rodrigues, M. M. Silva, G. S. Campos, S. I. Sardi, U. Kitron, M. G. Reis, and G. S. Ribeiro. Outbreak of exanthematous illness associated with Zika, chikungunya, and dengue viruses, Salvador, Brazil. *Emerging Infectious Diseases*, 21(12):2274, 2015.
15. S. Ch, S. Sohani, D. Kumar, A. Malik, B. Chahar, A. Nema, B. K. Panigrahi, and R. Dhiman. A support vector machine-firefly algorithm based forecasting model to determine malaria transmission. *Neurocomputing*, 129:279–288, 2014.
16. T. Chakraborty, S. Chattopadhyay, and I. Ghosh. Forecasting dengue epidemics using a hybrid methodology. *Physica A: Statistical Mechanics and its Applications*, 527: 121266, 2019.
17. H. K. Choi. Stock price correlation coefficient prediction with ARIMA-LSTM hybrid model. *arXiv preprint arXiv:1808.01560*, 2018.

18. G. Coelho, P. C. Silva, and R. L. Frutuoso, editors. *Levantamento rápido de índices para Aedes aegypti LIRAa para vigilância entomológica do Aedes aegypti no Brasil: Metodologia para avaliação dos índices de Breateau e predial e tipos de recipientes*. BRASIL, Ministério da Saúde, Brasília, 1 edition, 2012.
19. F. Cortes, C. M. T. Martelli, R. A. de Alencar Ximenes, U. R. Montarroyos, J. B. S. Junior, O. G. Cruz, N. Alexander, and W. V. de Souza. Time series analysis of dengue surveillance data in two Brazilian cities. *Acta Tropica*, 182:190–197, 2018.
20. M. A. de Santana, J. M. S. Pereira, F. L. da Silva, N. M. de Lima, F. N. de Sousa, G. M. S. de Arruda, R. d. C. F. de Lima, W. W. A. da Silva, and W. P. dos Santos. Breast cancer diagnosis based on mammary thermography and extreme learning machines. *Research on Biomedical Engineering*, 34(1):45–53, 2018.
21. M. Denil, B. Shakibi, L. Dinh, N. De Freitas, et al. Predicting parameters in deep learning. In *Advances in neural information processing systems*, pages 2148–2156, 2013.
22. N. C. Dom, A. A. Hassan, Z. Abd Latif, and R. Ismail. Generating temporal model using climate variables for the prediction of dengue cases in Subang Jaya, Malaysia. *Asian Pacific Journal of Tropical Disease*, 3(5):352–361, 2013.
23. H. Drucker, C. J. Burges, L. Kaufman, A. Smola, V. Vapnik, et al. Support vector regression machines. *Advances in Neural Information Processing Systems*, 9:155–161, 1997.
24. C. W. Espinola, J. C. Gomes, J. M. S. Pereira, and W. P. dos Santos. Detection of major depressive disorder using vocal acoustic analysis and machine learningan exploratory study. *Research on Biomedical Engineering*, 37(1):53–64, 2021.
25. C. W. Espinola, J. C. Gomes, J. M. S. Pereira, and W. P. dos Santos. Vocal acoustic analysis and machine learning for the identification of schizophrenia. *Research on Biomedical Engineering*, 37(1):33–46, 2021.
26. R. J. Fajardo-Herrera, J.-C. Valdelamar-Villegas, and D. Arrieta-Pérez. Prediction of the potential establishment of the mosquito Aedes aegypti in non-residential urban spaces in Colombia using eco-urban and landscape variables. *Gestión y Ambiente*, 20(1):95, 2017.
27. G. Falbo and J. E. Cabral Filho. Facing a severe epidemic outbreak: a fight against arboviruses. *Revista Brasileira de Saúde Materno Infantil*, 16:S3 – S4, 11 2016. ISSN 1519-3829. URL http://www.scielo.br/scielo.php?script=sci_arttext&pid= S1519-38292016000800001&nrm=iso.
28. A. Ghani, T. M. McGinnity, L. P. Maguire, and J. Harkin. Neuro-inspired speech recognition with recurrent spiking neurons. In *International Conference on Artificial Neural Networks*, pages 513–522. Springer, 2008.
29. M. Gharbi, P. Quenel, J. Gustave, S. Cassadou, G. La Ruche, L. Girdary, and L. Marrama. Time series analysis of dengue incidence in Guadeloupe, French West Indies: forecasting models using climate variables as predictors. *BMC Infectious Diseases*, 11 (1):1–13, 2011.
30. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11 (1):10–18, 2009.
31. A. Hamlet, K. Jean, W. Perea, S. Yactayo, J. Biey, M. Van Kerkhove, N. Ferguson, and T. Garske. The seasonal influence of climate and environment on yellow fever transmission across Africa. *PLoS Neglected Tropical Diseases*, 12(3):e0006284, 2018.
32. S. Haykin. *Redes Neurais: Princípios e Prática*. Bookman, Porto Alegre, 2001.

33. T. K. Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282, 1995.

34. G. Holmes, A. Donkin, and I. H. Witten. Weka: A machine learning workbench. In *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*, pages 357–361. IEEE, 1994.

35. G.-B. Huang and H. A. Babri. Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. *IEEE Transactions on Neural Networks*, 9(1):224–229, 1998.

36. G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 985–990. IEEE, 2004.

37. G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

38. G.-B. Huang, H. Zhou, X. Ding, and R. Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529, 2012.

39. M. Hugentobler. Quantum GIS. In *Encyclopedia of GIS*, pages 935–939. Springer, 2008.

40. H. Jaeger. The echo state approach to analysing and training recurrent neural networkswith an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.

41. C. C. Jansen and N. W. Beebe. The dengue vector Aedes aegypti: what comes next. *Microbes and Infection*, 12(4):272–279, 2010.

42. R. J. Joyce, J. E. Janowiak, P. A. Arkin, and P. Xie. CMORPH: A method that produces global precipitation estimates from passive microwave and infrared data at high spatial and temporal resolution. *Journal of Hydrometeorology*, 5(3):487–503, 2004.

43. M. Khashei and M. Bijari. A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. *Applied Soft Computing*, 11(2):2664–2675, 2011.

44. M. U. Kraemer, M. E. Sinka, K. A. Duda, A. Q. Mylne, F. M. Shearer, C. M. Barker, C. G. Moore, R. G. Carvalho, G. E. Coelho, W. Van Bortel, et al. The global distribution of the arbovirus vectors Aedes aegypti and Ae. albopictus. *eLife*, 4:e08347, 2015.

45. A. E. Laureano-Rosario, A. P. Duncan, P. A. Mendez-Lazaro, J. E. Garcia-Rejon, S. Gomez-Carro, J. Farfan-Ale, D. A. Savic, and F. E. Muller-Karger. Application of Artificial Neural Networks for Dengue Fever Outbreak Predictions in the Northwest Coast of Yucatan, Mexico and San Juan, Puerto Rico. *Tropical Medicine and Infectious Disease*, 3(1):5, 2018.

46. Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

47. M. Lukoševicius. A practical guide to applying echo state networks. Inˇ *Neural networks: tricks of the trade*, pages 659–686. Springer, 2012.

48. W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.

49. T. Magalhaes, C. Braga, M. T. Cordeiro, A. L. S. Oliveira, P. M. S. Castanha, A. P. R. Maciel, N. M. L. Amancio, P. N. Gouveia, V. J. P. da Silva-Jr., T. F. L. Peixoto, H. Britto, P. V. Lima, A. R. S. Lima, K. D. Rosenberger, T. Jaenisch, and E. T. A. Marques. Zika virus displacement by a chikungunya outbreak in Recife, Brazil. *PLoS Neglected Tropical Diseases*, 11(11):e0006055, 2017.

50. R. B. Martines. Notes from the field: evidence of Zika virus infection in brain and placental tissues from two congenitally infected newborns and two fetal lossesBrazil, 2015. *MMWR. Morbidity and Mortality Weekly Report*, 65, 2016.
51. S. V. Mayer, R. B. Tesh, and N. Vasilakis. The emergence of arthropod-borne viral diseases: A global prospective on dengue, chikungunya and zika fevers. *Acta Tropica*, 166:155–163, 2017.
52. T. E. Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9(3), 2007.
53. P.-F. Pai and C.-S. Lin. A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega*, 33(6):497–505, 2005.
54. K. K. Paul, P. Dhar-Chowdhury, C. E. Haque, H. M. Al-Amin, D. R. Goswami, M. A. H. Kafi, M. A. Drebot, L. R. Lindsay, G. U. Ahsan, and W. A. Brooks. Risk factors for the presence of dengue vector mosquitoes, and determinants of their prevalence and larval site selection in Dhaka, Bangladesh. *PloS One*, 13(6):e0199457, 2018.
55. G. Quantum. Development Team.(2013). Quantum GIS geographic information system. Open Source Geospatial Foundation Project, 2013.
56. A. Roth, A. Mercier, C. Lepers, D. Hoy, S. Duituturaga, E. Benyon, L. Guillaumot, and Y. Souares. Concurrent outbreaks of dengue, chikungunya and Zika virus infections–an unprecedented epidemic wave of mosquito-borne viruses in the Pacific 2012–2014. *Eurosurveillance*, 19(41):20929, 2014.
57. J. M. Scavuzzo, F. C. Trucco, C. B. Tauro, A. German, M. Espinosa, and M. Abril. Modeling the temporal pattern of Dengue, Chicungunya and Zika vector using satellite data and neural networks. In *Information Processing and Control (RPIC), 2017 XVII Workshop on*, pages 1–6. IEEE, 2017.
58. J. M. Scavuzzo, F. Trucco, M. Espinosa, C. B. Tauro, M. Abril, C. M. Scavuzzo, and A. C. Frery. Modeling Dengue vector population using remotely sensed data and machine learning. *Acta Tropica*, 185:167–175, 2018.
59. P. Siriyasatien, S. Chadsuthi, K. Jampachaisri, and K. Kesorn. Dengue epidemics prediction: A survey of the state-of-the-art based on data science processes. *IEEE Access*, 6:53757–53795, 2018. ISSN 2169-3536.
60. P. Siriyasatien, S. Chadsuthi, K. Jampachaisri, and K. Kesorn. Dengue Epidemics Prediction: A Survey of the State-of-the-Art Based on Data Science Processes. *IEEE Access*, 6:53757–53795, 2018.
61. A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.
62. R. Standish. SciDAVis. *Free Application for Scientific Data Analysis and Visualization*, 2016.
63. L. M. Stolerman, P. D. Maia, and J. N. Kutz. Forecasting dengue fever in brazil: An assessment of climate conditions. *PLoS One*, 14(8):e0220106, 2019.
64. Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014.
65. J. A. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.
66. P. Thakur and S. Kaur. An intelligent system for predicting and preventing Chikungunya virus. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pages 3483–3492. IEEE, 2017.

67. R. Tosepu, K. Tantrakarnapa, S. Worakhunpiset, and K. Nakhapakorn. Climatic Factors Influencing Dengue Hemorrhagic Fever in Kolaka District, Indonesia. *Environment and Natural Resources Journal*, 16(2):1–10, 2018.
68. P. F. d. C. Vasconcelos. Doença pelo vírus Zika: um novo problema emergente nas Américas? *Revista Pan-Amazônica de Saúde*, 6(2):9–10, 2015.
69. D. Verstraeten. *Reservoir computing: computation with dynamical systems*. PhD thesis, Ghent University, 2009.
70. W. E. Villamil-Gómez, O. González-Camargo, J. Rodriguez-Ayubi, D. Zapata-Serpa, and A. J. Rodriguez-Morales. Dengue, chikungunya and Zika co-infection in a patient from Colombia. *Journal of Infection and Public Health*, 9(5):684–686, 2016.
71. I. H. Witten and E. Frank. *Data Mining: Pratical Machine Learning Tools and Technique*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2005.
72. I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.
73. Y. Yusof and Z. Mustaffa. Dengue outbreak prediction: A least squares support vector machines approach. *International Journal of Computer Theory and Engineering*, 3(4): 489, 2011.
74. C. Zanluca, V. C. A. d. Melo, A. L. P. Mosimann, G. I. V. d. Santos, C. N. D. d. Santos, and K. Luz. First report of autochthonous transmission of Zika virus in Brazil. *Memórias do Instituto Oswaldo Cruz*, 110(4):569–572, 2015.
75. G. Zhang, B. E. Patuwo, and M. Y. Hu. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14(1):35–62, 1998.
76. G. P. Zhang. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50:159–175, 2003.
77. N. Zhao, K. Charland, M. Carabali, E. O. Nsoesie, M. Maheu-Giroux, E. Rees, M. Yuan, C. G. Balaguera, G. J. Ramirez, and K. Zinszer. Machine learning and dengue forecasting: Comparing random forests and artificial neural networks for predicting dengue burden at national and sub-national scales in Colombia. *PLoS Neglected Tropical Diseases*, 14(9):e0008056, 2020.