Artificial Intelligence Predicts and Explains West Nile Virus Risks Across Europe: Extraordinary Outbreaks Determined by Climate and Local Factors

Albert A Gayle[1]

1. Department of Public Health and Clinical Medicine, Section of Sustainable Health, Umeå University, SE-90187 Umeå, Sweden

Methods

Data Collection.

A literature review was conducted. The purpose of this review was to identify potential covariates of West Nile virus (WNV) emergence. PubMed and Google Scholar were searched. All research published between 1999 and 2019 was identified that included “West Nile” in either the title or abstract. This set of research publications was furthermore filtered. Only those titles or abstracts that referred to geospatial risk associations were included. Search terms for this secondary filter included: “risk”, “model”, “predict”, “estimat”, “associat”, “determinant”, “factor” and “review”. This list was then manually screened to exclude articles that presented neither primary research nor references to the results of such. The determinants presented were then extracted and compiled. Time-lag and other interaction effects were noted separately and in addition to the primary determinants. Determinant specifications with respect to aggregation, normalization, and other processing were also noted. This list was then stratified into eight categories based on class and source of data. Spatial and temporal covariates were included among the eight feature classes.

The selected feature classes included the following broad categories: “climate”, “environmental”, “economic”, “sociodemographic”, “hosts”, and “vectors”. Public data sources for each feature class were identified and selected based on the availability of data that aligned with determinant specifications. The literature on WNV consistently describes it as “hard to predict”. And this is reflected in broad uncertainties surrounding the precise mechanistic associations surrounding even the most basic, and generally accepted, determinants. In many cases, time-lagged climatic features are included in models to serve as proxy for specific features relevant to vector breeding or disease transmission. For example, what does it mean to associate rain that occurred 4 weeks prior to a reported case today? What is the actual biotic or environmental feature that is driving increased risk today: a) precipitation accumulation, b) soil moisture, c) standing water, or d) increased surface reflectivity indicative of water presence (MNDWI)? In order to avoid such potentially confounding effects, we expanded our inclusion criteria to include all climatic or environmental indicators available from a given selected source – this included some features not present in the literature.

All data was downloaded and processed using the R platform. Spatial and temporal covariates were included: one for each of the district-level regions in Europe (NUT3) as well as years 2010-2016 and 2018. Geometries for each of the NUTS3 regions were obtained using the “Eurostats” package in R. Public sources were sought out for each of the remaining feature classes. See *Supplementary Table 1* for a complete summary of all included features, sources, processing specifications, and summary statistics. The “earthEngineGrabR”[1] package in R was used to obtain environmental and climate data from Google Earth Engine. Climatic trend data was obtained directly, in preprocessed form from the International Research Institute for Climate and Society (IRI) at Columbia University (New York, USA)[2]. The “Bioclimatic” feature set is a well-cited set of time-varying geoclimatic metrics applied in the area of ecological niche modeling[3]. They are derived from standard climatic variables such as precipitation and temperature, over time, for a given region. The “dismo” package[4] in R was used to calculate these. In addition, indices for intradecal climatic systems such as the “El Nino South Oscillation” (ENSO) and the “North Atlantic Oscillation” (NAO) were obtained using the “rSOI”[5] package in R. Environmental and host data was obtained directed from the European Environment Agency (EEA)[6]. This included all Corine Land Cover (CLC) and Natura 2000 Bioregions data[7]. Geospatial species distribution data was also obtained from the EEA, by merging the Natura 2000 Habitats and Species tables. This included all species, to allow for

potentially unreported hosts. Economic and sociodemographic data was obtained using the "Eurostat" package[8]. The European Center for Disease Control (ECDC) provided mosquito distribution data; this included all reported vectors in the region. West Nile virus case data was also obtained from the ECDC.

To account for time-lag effects, all climatic variables were aggregated into four quarters as done in similarly scoped studies[9]. This excluded the Bioclimatic variables which are defined annually.

WNV case data was quality-controlled to exclude cases likely to be subject to regional variation in reporting or other biases. Excluded were 1) asymptomatic cases, 2) cases with no syndromatic description, and 3) cases acquired outside the reporting district. This is consistent with previous work, in which all but the most severe neuroinvasive manifestations (WNND) were excluded[10].

This final data set was then divided into a training set (2010-2016) and test/validation set (2018). The statistical effect of large-scale climatic systems and associated downstream covariation may potentially overlap year-to-year. To mitigate, 2017 was omitted from training.

Analysis.

The final data set was thus used to construct an explanatory model. This included the full year's data, including climatic data from periods overlapping and extending beyond the outbreak season (i.e., January - December). This model was therefore not suited for same year prospective early warning, but would serve instead to retrospectively examine the factors driver the 2018 outbreak.

The official implementation for XGBoost in R[11] was used. Cross-validation was used to set the initial iteration limits. A training data set was generated for each using data from years 2010-2016, according to the specification of each model. For cross validation, a hold out subset is specified and then used to evaluate the performance of a model trained with the remaining data. Here, given the cyclical nature of our predictive task, the hold out set was specified to be a single year. For each model, the training set was therefore split according to year and a model was trained using six of the seven years. The seventh, hold out year was then used to test model performance. This process was repeated seven times, using each of the included years once as the hold out set. This process generates several benchmarks, one of which is an optimal number of iterations. For classification tree models, iteration limitations are commonly set so as to avoid over-fitting. A second model was trained using the entire seven years of training data, with the iteration limit set based on the limit assessed by the cross-validation run. An additional criteria was set to stop model training after 10 iterations with no reduction in log-loss error. Once this second training round was complete, covariates determined to be insignificant were dropped and the model was rerun. This mode of feature selection is consistent with a common use case for XGBoost[12] and it is furthermore recommended for this class of model[13]. In our case, covariates found to be insignificant with respect to three importance criteria – marginal contribution to predictive output ("gain"), internal predictive prevalence ("frequency"), and case-wise predictive relevance ("cover") – were removed.

A final model was therefore generated that included an independently determined subset of features. In a departure from established machine learning norms[14,15], model hyperparameters were not subject to automated optimization procedures. Models such as XGBoost are notoriously sensitive to hyperparameter settings[16] – a problem we wished to avoid outright. Inspired by the work of Blagus and Lusa (2017)[17] and Brieman (2001)[13], hyperparameters were therefore set to ensure internal experimental robustness: 1) 95% was set as the internal discriminatory threshold, 2) each submodel was limited to half of the total feature space, and 3) maximum tree size was set to twice the number of included feature classes (i.e., 16). Consistent with guidance put forth in the seminal paper by Elith J. et al (2008), step size was reduced to 0.03 (10% of the default) to ensure feature interactions were robustly assessed within the ensemble.

This model was applied to the data set from 2018 to generate a probability of event. This was conducted on an out-of-sample basis, in prospective time: no data used to train the model was used to evaluate the model and all data corresponded to time points preceding the event.

Two sets of binary predictions (presence or absence) were derived using the standard 50% classification threshold and a more discriminating 10% threshold. The choice of classification threshold directly affects predictive performance and several methods exist to optimize[18]. However, as with manual or automated tuning of hyperparameters in XGBoost, the effect of such methods on out-of-sample performance is often unpredictable[19]. Out-of-sample performance was therefore evaluated at the specified thresholds. In-sample performance was likewise estimated using the 2010-2016 training set. For each model, four sets of evaluations were therefore produced: 1) out-of-sample at 50%, 2) out-of-sample at 10%, 3) in-sample at 50%, and 4) in-sample a 10%. Each included the following (reported) metrics: 1) AUC, 2) sensitivity, 3) specificity, and 4) balanced accuracy. AUC is the probability that a randomly selected positive case will be assigned a higher probability than a randomly selected negative case. It indicates how well the model discriminates between positive and negative classes irrespective of classification threshold. The remaining three metrics are threshold-dependent and vary accordingly. Sensitivity, also referred to as true positive rate, measures the probability of detecting event occurrence. Specificity, also referred to true negative rate, measures the probability of correctly identifying non-occurrence. Balanced accuracy is the average of the prior two metrics; it measures the probability of detection assuming negative and positives classes are equally represented in the data set. In the present study, positive classes represent only 0.05% of the total sample. A model that optimizes balanced accuracy irrespective of threshold is therefore preferable.

The “SHapley Additive exPlanations” (SHAP) engine is an AI-driven deductive framework for imputing local feature effects[20]. The underlying algorithm is derived from a military intelligence application based on cooperative game theory. This algorithm determines individual contributions for group outcomes and has been mathematically proven to be optimal with respect to his task[21]. The SHAP implementation applies this framework to an XGBoost model to determine the degree to which individual features influence the computed event probability for a given case. The output of this process is a meta-model, consisting of an effect matrix dimensionally identical to the original, nominal value data (matrix): one complete set of feature effect estimations for each case. Each term reflects the local change in log relative risk associated with a given feature with respect to the global, baseline probability of event. This global baseline risk is computed along with the effect matrix and is the same for all cases. This transformation effectively recenters the predictive output at zero, implying a predictive threshold equivalent to the computed baseline probability. Values greater than zero indicate increased relative risk of event and *vice versa*. This transformation provides for several useful analytical modalities that are presented later and in the main text. The inputs for this process are the computed XGBoost model and the original, nominal value data (set).

The SHAP engine was therefore applied to this model to generate a SHAP meta-model for 2018. Predictive performance was evaluated in a manner identical to that described for the XGBoost model. However, as previously described, threshold selection is implicit to the SHAP generation process and was therefore omitted. The SHAP-implied classification threshold was likewise computed and applied to the original XGBoost model to confirm additive and predictive equivalence. For each of the SHAP model, one set of evaluations was therefore produced.

The SHAP effect matrix was visually summarized and geospatially mapped. We furthermore conducted means tests ($p < 0.05$) and calculated standardized means differences to determine which effects most strongly discriminated between between a) regions with positive and negative outbreak risk and b) 2018 and the control period (2010-2016) with respect to regions with positive outbreak risk. Supervised clustering is the application of a traditional unsupervised cluster analysis to effect estimates derived from a supervised learning process. This was introduced by Lundberg et al (2020)[20] who used it to demonstrate case-wise interaction effects. Previous studies have suggested underlying non-climatic risk factors to be strong predictors of WNV risk, provided suitable climate.

We therefore applied a revised methodology to a subset of features corresponding with environmental, host, and vector related covariates. The resulting geospatial clusters and associated outbreak risk statistics were visually summarized and geospatially mapped.

The "XGBoost"[22] package in R was used for all modeling, including generation of the local SHAP meta-models. The "ggplot2"[23] package in R was used for visual summarization and the "tmap"[24] package in R was used for geospatial mapping. Geospatial polygons were obtained from the Statistical Office of the European Commission using the R package, "eurostat"[8]. All computations as well as as necessary data processing were conducted in R, using the "data.table"[25] package.

Limitations

Several *a priori* assumptions and modeling choices were required. Such choices naturally influence the final model and output in various ways, not limited to overall predictive performance and imputed associations with respect to individual features. Most notably, the final feature set size for any given model is in no way deterministic and could very well be a function of model hyperparameters, namely "tree-depth" and "eta". These two hyperparameters effectively control the degree to which the model explores causal alternatives and complexity of causation, and it is plausible to presume some effect. Furthermore, the data set used to derive this model was in no way comprehensive or statistically refined and outputs are sure to change given additional fine-tuning in this regard. That having been said, exploring the performance of these methods with data of uncertain or uneven quality was an implicit goal of this work.

Citations

1. Simplify the acquisition of remote sensing data. https://jesjehle.github.io/earthEngineGrabR/.
2. IRI Climate and Society Map Room. index.html.
3. Kriticos, D. J., Jarošik, V. & Ota, N. Extending the suite of BIOCLIM variables: a proposed registry system and case study using principal components analysis. *Methods Ecol. Evol.* **5**, 956–960 (2014).
4. Hijmans, R. J., Phillips, S. & Elith, J. L. and J. *dismo: Species Distribution Modeling*. (2017).
5. Import Various Northern and Southern Hemisphere Climate Indices. https://boshek.github.io/rsoi/.
6. European Environment Agency's home page — European Environment Agency. https://www.eea.europa.eu/.
7. Natura 2000 - Environment - European Commission. https://ec.europa.eu/environment/nature/natura2000/index_en.htm.
8. Lahti [aut, L. *et al. eurostat: Tools for Eurostat Open Data*. (2020).
9. Keyel, A. C. *et al.* Seasonal temperatures and hydrological conditions improve the prediction of West Nile virus infection rates in Culex mosquitoes and human case counts in New York and Connecticut. *PLOS ONE* **14**, e0217854 (2019).
10. Laperriere, V., Brugger, K. & Rubel, F. Simulation of the seasonal cycles of bird, equine and human West Nile virus cases. *Prev. Vet. Med.* **98**, 99–110 (2011).
11. Chen, T. & Guestrin, C. Xgboost: A scalable tree boosting system. in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* 785–794 (2016).
12. He, X. *et al.* Practical Lessons from Predicting Clicks on Ads at Facebook. in *Proceedings of 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining - ADKDD'14* 1–9 (ACM Press, 2014). doi:10.1145/2648584.2648589.
13. Breiman, L. Random forests. *Mach. Learn.* **45**, 5–32 (2001).
14. Wang, Y. & Ni, X. S. A XGBoost risk model via feature selection and Bayesian hyper-parameter optimization. *ArXiv190108433 Cs Stat* (2019).
15. Xia, Y., Liu, C., Li, Y. & Liu, N. A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert Syst. Appl.* **78**, 225–241 (2017).
16. Mayr, A., Binder, H., Gefeller, O. & Schmid, M. The evolution of boosting algorithms. From machine learning to statistical modelling. *Methods Inf. Med.* **53**, 419 (2014).
17. Blagus, R. & Lusa, L. Gradient boosting for high-dimensional prediction of rare events. *Comput. Stat. Data Anal.* **113**, 19–37 (2017).
18. Cortes, C. & Mohri, M. AUC optimization vs. error rate minimization. in *Advances in neural information processing systems* 313–320 (2004).
19. Yang, K., Yu, R., Wang, X., Quddus, M. & Xue, L. How to determine an optimal threshold to classify real-time crash-prone traffic conditions? *Accid. Anal. Prev.* **117**, 250–261 (2018).

20. Lundberg, S. M. *et al.* From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2**, 56–67 (2020).

21. Lundberg, S. M., Erion, G. G. & Lee, S.-I. Consistent Individualized Feature Attribution for Tree Ensembles. *ArXiv180203888 Cs Stat* (2019).

22. XGBoost. https://xgboost.ai/.

23. Wickham, H. *et al. ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. (2020).

24. Tennekes, M. *et al. tmap: Thematic Maps*. (2020).

25. Dowle, M. *et al. data.table: Extension of 'data.frame'*. (2019).