Ensembling and calibration in VIEWS, version Fatalities002*

Håvard Hegre[1, 2], Sofia Nordenving[2], Michael Colaresi[3], Mihai Croicu[2], James Dale[2], and Paola Vesco[2]

[1]Peace Research Institute Oslo (PRIO)
[2]Department of Peace and Conflict Research, Uppsala University
[3]University of Pittsburgh

October 16, 2023
VIEWS Version: Fatalities002

Abstract

We describe how individual models are combined into model ensembles in VIEWS, and how model predictions are calibrated.

*The research was funded by the European Research Council, project H2020-ERC-2015-AdG 694640 (ViEWS), the European Research Council Advanced Grant project *ANTICIPATE*, Riksbankens Jubileumsfond project *Societies at Risk*, Uppsala University, the Swedish Research Council project *DEMSCORE*, the UK Foreign, Commonwealth and Development Office, the United Nations High Commissioner for Refugees, and Peace Research Institute Oslo. The simulations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX). For more information on the project see `viewsforecasting.org`.

Contents

1 Ensembling – using the 'wisdom of the crowd'

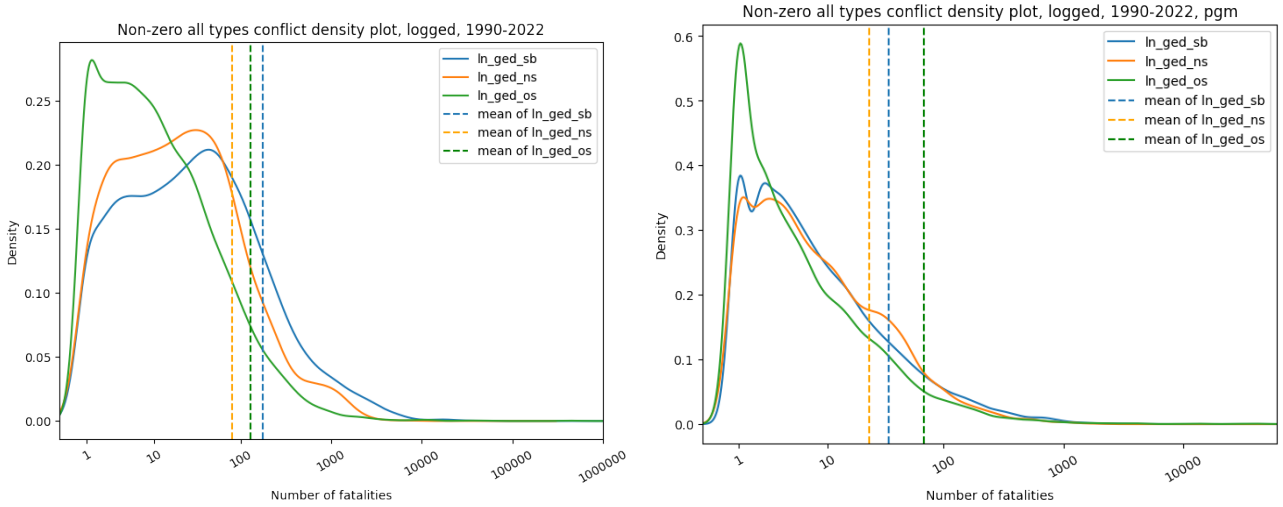
No statistical or machine-learning model or algorithm can perfectly learn the patterns of behavior that link some observable predictors to subsequent observations of the number of fatalities in war. Building on a variety of theoretical and methodological perspectives – the 'wisdom of the crowd' – clearly yields the best foundation for good decisions and high-quality forecasts (Tetlock, 2005). The greater the variance of adequate models available, the better a forecasting and decision-making system performs (Page, 2007). Ensembling – grouping of diverse forecasting models – also work as a means to smooth over problems (Armstrong, Green, and Graefe, 2015). Following the approach in ViEWS (Hegre et al., 2019), we use ensembling of constituent models to aggregate insights from various models, allowing a variety of modeling algorithms and feature sets, and applying state-of-the-art model weighting algorithms (Sivanandam and Deepa, 2008; Scrucca et al., 2013; Montgomery, Hollenbach, and Ward, 2012).

1.1 Obtaining ensemble weights at the *pgm* level

The ensemble algorithm we use at the *pgm* level is just the equally weighted mean. However, it is clear from the evaluation of results below that some models perform better than others, and we should be able to improve performance by giving these models more weight in our ensembles. At the *cm* level, this is acheived using a genetic algorithm, described in the next section, but this is prohibitively expensive at the *pgm* level.

1.2 Obtaining ensemble weights at the *cm* level

For the *cm* level, we have developed an algorithm to learn these weights from the data. To do this, we split the data into three periods. The first period, the 'training period', include the years 1990–2013. We train the constituent models described above on data for this period, and predict for the 'calibration period'; 2018–2021. Our ensemble weighting and calibration model use these predictions as well as data for the true outcome for the calibration period to obtain weights and calibration parameters. We then retrain all the constituent models for the 1990–2017 period and generate predictions for the 2018–2021 period, and apply the weights and calibration parameters to produce ensemble forecasts for that period. The forecasts



(a) 1990–2022, Country Level

(b) 1990–2022, PRIO-GRID level

Figure 1. Kernel density plots for all country-months/PRIO-GRID-months with non-zero fatality counts 1990–2022. The vertical lines show the mean (non-logged) fatality counts for the non-zero observations. *Source:* UCDP GED, 2022

for the true future will use the 2018–2021 period as calibration period, and generate ensemble predictions for the 2023-2025 period. At the *cm* level, our model weights are obtained using a genetic algorithm (Sivanandam and Deepa, 2008; Russell and Norvig, 2020). These optimize a user-defined performance metric in the calibration data by letting a population of random model weights evolve over a large number of generations to find optimal weights. Genetic algorithms provide a fast, flexible, and intuitive way to optimize the performance metric when the inputs are high-dimensional or when there are complex restrictions on the available inputs.

The genetic ensembling algorithm, as implemented, works like this: 100 random ensembles are chosen with a random set of weights (genes), under the sole condition that the sum of those weights is between 0.5 and 3. Each of the 100 ensembles are then computed using the assigned weights and then evaluated using a mean squared error fitness function ($1/e^{mse}$) against the data in the calibration period. Pairs of the ensembles are then sampled through a weighted sampling procedure based on the fitness scores. These pairs are recombined in a simulation of genetic reproduction - a random subset of weights from one ensemble in the pair is combined with the remaining weights from the other ensemble in the pair. Then, with a probability of .2, a random subset of weights from the resulting ensemble is replaced with random weights. 100 such recombination/mutation processes are carried out, leading to 100 new organisms that form a new generation, to be put, again, through the same process as above.[1] This is repeated 500 times (generations), with the best ensembles from the last (500th) generation being used.

2 Calibration

Predictions should ideally have roughly the same distributions as the observed outcome. Given the distribution of the outcome variable (Figure 1), few algorithms would get this right without post-prediction

[1] As a complication, to improve performance, at each generation, the 10 best models of the previous generations are 'spared and cloned', i.e. preserved intact, without recombination and mutation for the next generation. The ten worst performers in the mutation and recombination steps are discarded, so that the population of ensembles remains 100.

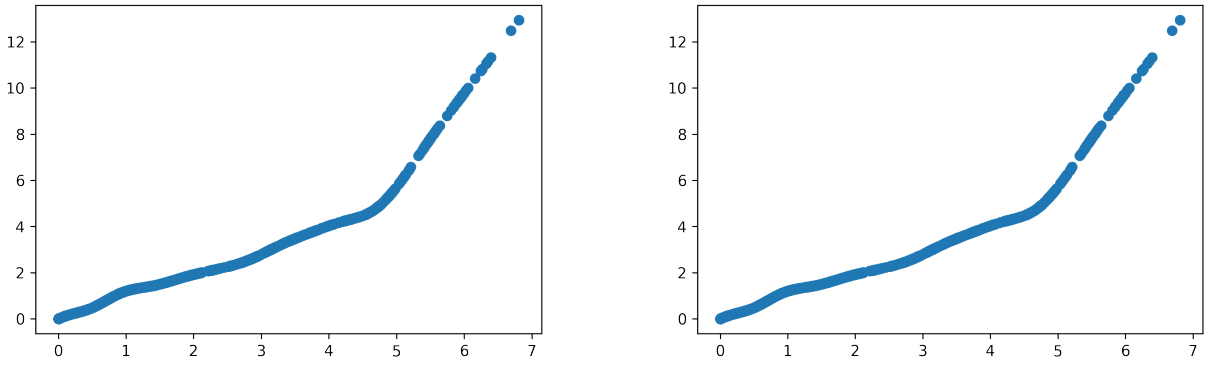


Figure 2. **GAM calibration function, fat_topics_histgbm model (left) and fat_hh20_xgb model (right),** $s = 3$

Source: ViEWS, 2022

calibration. In particular, models tend to yield distributions with smaller variance than the outcome, reducing our ability to correctly predict the total number of fatalities. Moreover, they tend to have means that are lower than the true mean, and some models produce negative values even though these do not exist in the training data.

We have explored two ways to calibrate predictions. The first is to multiply the predicted number of fatalities with the constant that gives the same variance (in the calibration partition) as the outcome – in practice, this is equivalent to estimating a no-constant linear regression model with the outcome in the calibration partition as the dependent variable and the prediction as the independent variable. The estimated parameter is in most cases larger than 1, expanding the variance of the predictions and increasing the mean. We estimate this calibration model separately for each constituent model and each step, and apply the calibration parameter (the β coefficient in the OLS model) to the predictions for the test partition. We also explored including an intercept in the calibration model. This, however, in most cases yields non-zero predictions for a majority of the cases where the true outcome is zero, significantly hurting performance. Without an intercept term, however, the calibration works poorly for models like the XGBoost model that can yield negative predictions – multiplying a negative prediction with a number larger than one obviously does not help calibration.

To counter these challenges, we have moved to using a generalized additive linear model (a GAM), using the PYGAM package (servén, Brummitt, and Abedi, 2020). GAM models fit the relationship between the dependent and independent variables as a very flexible function. To avoid overfitting, we constrained the model to yield calibrated predictions that are monotonically increasing in the non-calibrated predictions – if the original model ranks one case higher than another, the calibrated model also ranks it as at least as high. We set the parameters of the function so that the calibrated transformation is quite smooth, retaining most of the original prediction.

This model was estimated separately for each constituent model and for each step. Figure 2 illustrates how the function works. The calibration function works well, typically decreasing MSE by about 10% relative to the uncalibrated predictions, mostly removes zero predictions, and increases the variance.

Figure 2 shows two example calibrations from the fatalities 001 version. The y axis shows the calibrated predicted number of fatalities as a function of the original prediction (x axis). In the case of the

fat_topics_histgbm model (left), the GAM function does not alter the original predictions much except for predictions above 4 (about 50 deaths), but pull the remaining predictions considerably upwards. In the case of fat_hh20_xgb model (right), which yields a number of large negative predictions, all negative predictions are calibrated to zero, and the remaining predictions are mostly unchanged.

The genetic algorithm can in principle yield weights that sum to less than or more than one. This, then, serves as a second calibration step.

Calibration at the *pgm* level

The issue of calibration is particularly acute for the pgm models. Africa and the Middle East together comprise approximately 13,000 PRIO-GRID cells, which when multiplied by the 356 time–steps under consideration here, yields around five million units of analysis. In any of the conflict datasets, the vast majority of values (i.e. the numbers of fatalities) associated with these units of analysis are zero.

The imbalance between the numbers of zero and non-zero data points in problems like conflict prediction is sometimes redressed by randomly discarding a (possibly very large) fraction of the zero-valued units of analysis, so that the regression algorithms which search for patterns in the data are exposed to more equal numbers of zero and non-zero values. This is known as *downsampling* and the hope in doing this is that the algorithms are not swamped by zero values and do not, therefore, tend to simply predict zeros or very small values everywhere. Downsampling also reduces the runtime of fitting procedures, since algorithms have less data to deal with.

We examined the effect of downsampling by discarding between 70 and 98 per cent of the zero-valued data points. Very large discard fractions significantly worsened predictive performance and more moderate fractions yielded no appreciable performance improvement, while runtimes required to dispense with downsampling entirely were not prohibitive. We therefore abandoned downsampling as a data-engineering strategy.

However, with or without moderate downsampling, the predictive performance of the random forest-, LGBM- and gradient-boosting algorithms were all quite poor, as measured by examining MSE values, or more subjectively by comparing maps of predictions from the test partition with observations from the same timestep.

The clearest problem is that, while able to predict the presence or absence of conflict in roughly the correct geographic locations (although with some spread around the locations of observed conflict events), the numbers of predicted fatalities were almost always very small. This is indicative of a normalisation or calibration problem, as described in the previous section.

A partial solution to this problem might be to use another evaluation metric than MSE at the *pgm* level, one that gives credit for predictions of the right magnitude, but not quite at the correct geographic location or time. The pEMDiv metric (Greene et al., 2019) is the best candidate we are aware of. Our current evaluation procedures are described in (Hegre et al., 2023a).

Another issue is that it is desirable that the sum of predicted fatalities at the *pgm* level over the grid cells that make up a given country are similar to the predicted fatalities for that country generated at the *cm* level. In the modeling system, we calibrate the predictions along this line of logic. Each pgm cell is assigned to the country that contains most or all of the cell and the fatalities in all the cells belonging

to a give country in a given month are summed and multiplied by a factor such that the sum is equal to the forecast fatalities for the same country in the same month. The MSE scores we report, then, have a different interpretation – they should be read as ranking models in terms of the model's ability to capture the distribution of the total fatalities suggested by the *cm* ensemble.

3 Models in the ensemble

Table 1 lists the models in the fatalities002 *cm* ensemble. Further descriptions of the current models are described in (Hegre et al., 2023b).

Model name	Weight, step 3	Weight, step 12	Weight, step 36
fatalities002_baseline_rf	0.0	0.0	0.08
fatalities002_conflicthistory_rf	0.2	0.18	0.07
fatalities002_conflicthistory_gbm	0.015	0.2	0.3
fatalities002_conflicthistory_hurdle_lgb	0.08	0.16	0.0
fatalities002_conflicthistory_long_xgb	0.0	0.07	0.015
fatalities002_vdem_hurdle_xgb	0.01	0.0	0.015
fatalities002_wdi_rf	0.0	0.01	0.015
fatalities002_topics_rf	0.015	0.0	0.015
fatalities002_topics_xgb	0.01	0.02	0.07
fatalities002_topics_hurdle_lgb	0.0	0.04	0.015
fatalities002_joint_broad_rf	0.09	0.01	0.02
fatalities002_joint_broad_hurdle_rf	0.0	0.02	0.0
fatalities002_joint_narrow_xgb	0.03	0.0	0.01
fatalities002_joint_narrow_hurdle_xgb	0.025	0.02	0.01
fatalities002_all_pca3_xgb	0.14	0.015	0.01
fatalities002_aquastat_rf	0.025	0.0	0.0
fatalities002_faostat_rf	0.02	0.0	0.0
fatalities002_faoprices_rf	0.02	0.0	0.015
fatalities002_imfweo_rf	0.01	0.01	0.025
fatalities002_Markov_glm	0.3	0.18	0.3
fatalities002_Markov_rf	0.16	0.18	0.25

Table 1. Models in the fatalities002 *cm* ensemble

4 Change history

4.1 Fatalities002

The Fatalities002 version does not introduce any changes in ensembling or calibration with respect to Fatalities001.

4.2 Fatalities001

The current ensembling procedures were introduced together with the fatalities model thanks to funding from the UK FCD Hegre et al. (2022).

4.3 ViEWS-ESCWA

The VIEWS system was expanded to cover the Middle East (including Turkey and Iran) thanks to funding from the UN ESCWA (Theisen et al., 2021).

4.4 ViEWS2020

ViEWS2020 is presented in (Hegre et al., 2021). In this version, thematic constituent models were trained and fitted separately before combined into two broader ensemble models, one for country level predictions and one for PRIO-GRID.

4.5 ViEWS2018

The first version of the ViEWS early warning system, the 'ViEWS2018' version was launched in July 2018 (Hegre et al., 2019).

References

Armstrong, J. Scott, Kesten C. Green, and Andreas Graefe (2015). "Golden rule of forecasting: Be conservative". In: *Journal of Business Research* 68.8. Special Issue on Simple Versus Complex Forecasting, pp. 1717–1731. DOI: https://doi.org/10.1016/j.jbusres.2015.03.031.

Greene, Kevin et al. (2019). *Move It or Lose It: Introducing Pseudo-Earth Mover Divergence as a Context-sensitive Metric for Evaluating and Improving Forecasting and Prediction Systems*. Presented to the 2019 Barcelona GSE Summer Forum, workshop on Forecasting political and economic crisis: Social science meets machine learning'.

Hegre, Håvard et al. (2019). "ViEWS: A political Violence Early Warning System". In: *Journal of Peace Research* 56.2, pp. 155–174. DOI: 10.1177/0022343319823860.

Hegre, Håvard et al. (2021). "ViEWS$_{2020}$: Revising and evaluating the ViEWS political Violence Early-Warning System". In: *Journal of Peace Research* 58.3, pp. 599–611. DOI: 10.1177/0022343320962157. eprint: https://doi.org/10.1177/0022343320962157.

Hegre, Håvard et al. (2022). *Forecasting Fatalities*. Forthcoming.

Hegre, Håvard et al. (2023a). *Evaluation of VIEWS forecasts*. Appendix Fatalities002 Documentation.

Hegre, Håvard et al. (2023b). *Models in VIEWS*. Appendix Fatalities002 Documentation.

Montgomery, Jacob M, Florian M Hollenbach, and Michael D Ward (2012). "Improving predictions using ensemble Bayesian model averaging". In: *Political Analysis* 20.3, pp. 271–291.

Page, Scott E. (2007). *The difference: how the power of diversity creaetes better groups, firms, schools, and societies*. Princeton, NJ: Princeton University Press.

Russell, Stuart and Peter Norvig (2020). *Artificial Intelligence: A Modern Approach*. 4th ed. Prentice Hall.

Scrucca, Luca et al. (2013). "GA: a package for genetic algorithms in R". In: *Journal of Statistical Software* 53.4, pp. 1–37.

servén, Daniel, Charlie Brummitt, and Hassan Abedi (2020). *pyGAM: Generalized Additive Models in Python*. Zenodo. DOI: 10.5281/zenodo.1208723.

Sivanandam, SN and SN Deepa (2008). "Genetic algorithms". In: *Introduction to genetic algorithms*. Springer, pp. 15–37.

Tetlock, Philip E. (2005). *Expert Political Judgment: How good is it? How can we know?* Princeton: Princeton University Press.

Theisen, Ole Magnus et al. (2021). *Understanding the Potential Linkages between Climate Change and Conflict in the Arab Region*. E/ESCWA/CL6.GCP/2021/TP.9. UN ESCWA, Beirut, Lebanon.