Partitioning and time-shifting in VIEWS, fatalities002*†

Håvard Hegre[1, 2], Sofia Nordenving[2], Michael Colaresi[1, 2], 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

May 22, 2023
VIEWS Version: Fatalities002

Abstract

In this paper, we present our procedures for partitioning data and for calibration, model weighting, and out-of-sample evaluation.

*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`.

†Forogh Akbari, Frederick Hoyles, Remco Jansen, Maxine Ria Leis, Angelica Lindqvist-McGowan, Malika Rakhmankulova, David Randahl, and Espen Geelmuyden Rød contributed to earlier versions of this documentation document.

Contents

1 Data partitioning and evaluation

1.1 Data partitioning

As in Hegre et al. (2019) and Hegre et al. (2021), we split all of the available data into three partitions. In our notation, the time periods for these partitions are defined based on the time stamps for the observed outcomes Y. Table 1a summarizes how we currently split our data. Internally and in online documentation, we refer to the month December 2021 as month 504, and will use the numeric month id in examples below.[1]

τ refers to calendar time,[2] but we add subscripts to identify when the partitions start and end. Because the partitions differ between evaluation and true forecasting, we have also added the superscript ^{e} to all notations of our evaluation partitions (see Table 1a).

The first relevant partition is the *training period* from τ_0^e, the first month with data, to τ_t^e (τ_0 to τ_t in the forecasting periodization). We use the notation $Y_{\tau_0:\tau_t^e}$ to define the beginning and ending time-points for the labeled instances in the training data.

The second set of observations is defined as the *calibration period* and is bounded by the cut-points $\tau_t^e + 1$ and τ_c^e ($\tau_t + 1$ and τ_c in the forecasting periodization), and is represented as $Y_{(\tau_t^e+1):\tau_c^e}$.

A third set is added only to the forecasting periodization. The UCDP releases final, carefully vetted events data annually– currently the version covering 1989–2021 is used (Davies, Pettersson, and Öberg, 2022). To ensure maximal consistency, we use only final events data for training, weighting, and calibration, so we set the calibration to the 2018–2021 period. However, the UCDP has provided 'candidate data' for 2022 at a monthly update schedule (Hegre et al., 2020). We make use of these updated data when computing forecasts, in order to allow data input up until the month before the true forecasting period. We call this the *predictor updating period*. It starts immediately upon the end of the calibration period, at $\tau_c + 1$, and runs up until the first month of true forecasts at $s = 1$, to $\tau_c + (k - 1)$. $k - 1$ thus represents the months from which we rely on UCDP-Candidate data as opposed to UCDP-GED data.

The last set is the *testing/forecasting period*, extending from $\tau_c^e + 1$ to τ_f^e in the evaluation periodization, and from $\tau_c + k$ to τ_f in the forecasting periodization.

[1] The VIEWS month id is a counter that started on 1 in January 1980.

[2] This is the reference point from which the project is operating. We are assuming we can observe values less than τ, but not values greater than τ, when computing models.

	Periodization	
	Evaluation	Forecast
Training period	$\tau_0^e = 121$ (January 1990) $\tau_t^e = 408$ (December 2013)	$\tau_0 = 121$ (January 1990) $\tau_t = 456$ (December 2017)
Calibration period	$\tau_t^e + 1 = 409$ (January 2014) $\tau_c^e = 456$ (December 2017)	$\tau_t + 1 = 457$ (January 2018) $\tau_c = 504$ (December 2021)
Predictor updating period	n/a n/a	$\tau_c + 1 = 505$ (January 2022) $\tau_c + (k-1) = 516$ (December 2022)
Testing/forecasting period	$\tau_c^e + 1 = 457$ (January 2018) $\tau_f^e = 504$ (December 2021)	$\tau_c + k = 517$ (January 2023) $\tau_f = 552$ (December 2025)

The 'forecast' periodization is for actual forecasting, the 'evaluation' periodization for testing models and ensembles. We use the training periods to train models and the calibration periods for hyper-parameter tuning and estimating model weights. After calibration, ensemble weighting, and hyper-parameter tuning, we retrain our models using both the training and calibration partitions.

(a) Partitioning of data for estimating model weights, hyper-parameter tuning, evaluation, and forecasting

1.2 Procedures for calibration, model weighting, and out-of-sample evaluation

All VIEWS models are subjected to a careful out-of-sample evaluation of predictive performance. We have developed a setup to maximize the precision of this evaluation, using available data as efficiently as possible.

In the following, we will refer to a specification as a 'model' m^j. When we use input data up to December 2022, we generate forecasts for each of the 36 months from January 2023 to December 2025. We refer to these as steps $s \in [1, 36]$. We train each of these models specifically for each s. Model $m^{(j,1)}$ is trained to predict $s = 1$ month into the future, $m^{(j,6)}$ $s = 6$ months forward, and so on. Our procedure can be summarized as follows:

For each model $m^{(j)}$ and step $s \in [1, 36]$ in the evaluation periodization we do the following:

1. Train model $m^{(j,s)}$ on monthly data from τ_0^e to τ_t^e
2. Generate predictions from $m^{(j,s)}$ trained in (1) for all months i in the calibration period $(\tau_t^e + 1, \tau_c^e)$, using data up to s months before i
3. Calibrate models, obtain ensemble weights, and tune hyper-parameters using the predictions from (2) along with the actuals for all months in the calibration period
4. Retrain model $m^{(j,s)}$ using both the training and calibration periods (τ_0^e, τ_c^e)
5. Generate predictions for the testing/forecasting period $(\tau_c^e + 1, \tau_f^e)$ from the new model $m^{(j,s)}$ re-trained in (4)

In step 5, the procedure is different when we evaluate models than when we generate true forecasts. The two variants of the procedure are summarized in Figure 1. The horizontal bar labeled $s = 1$ indicates how we produce forecasts one month into the future. The bars are divided into three blocks, each of which consists of a number of rectangles representing months. The start and end months τ of these periods, only numerically referenced in the figure, are specified in Table 1a.

Each rectangle in Figure 1 is divided into data for the outcome (Y) and for predictors (X). The green block represents the training data partition, stretching from month τ_0 to month τ_t. The arrows show how our models use input data X available at a given month to learn the relationship to the outcome Y for the subsequent s-month(s) ahead.[3]

When *evaluating* the models, we generate predictions for each month in the testing period, just as in step 2. We then match $s = 2$ forecasts for January 2018 (based on input data up to November 2017) with what actually happened in January 2018, $s = 2$ forecasts for February 2018 with actuals for February 2018, etc, for all s. This means that a model $m^{(j,2)}$ targeting $s = 2$ is evaluated against all 36 months in our calibration or testing period.

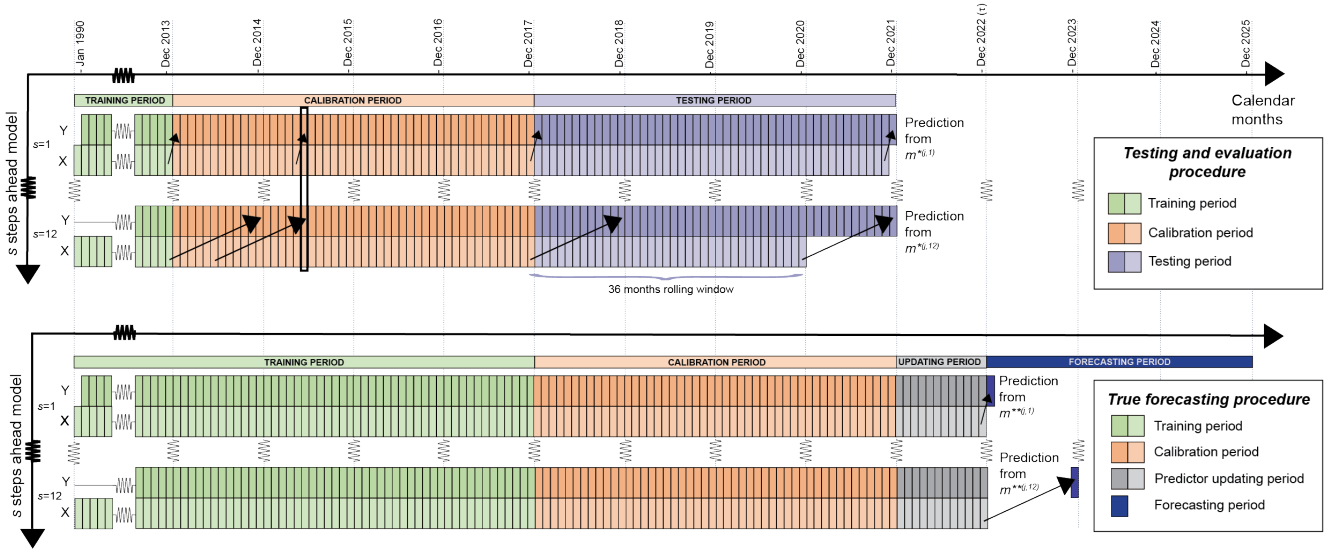
When we generate *true forecasts*, however, we only make forecasts based on the most recent input data. For the forecasts presented below, we have data up to December 2022. We make one set of forecasts at $s = 1$ for January 2023, one at $s = 2$ for February 2023, etc.

In Hegre et al. (2019), we used the procedure we now use for forecasts also when evaluating, calibrating, and estimating model weights. The current setup provides us with much more data for testing and calibration as we are able to reuse multiple times the set of actuals Y for a given month, e.g. month 414 (June 2014), marked off in Figure 1.[4]

The changes do not affect training, but have other important benefits. Most importantly, using data for actual conflicts for all months in the calibration period for each step s means we are now able to estimate ensemble weights specifically for each s. We also have more data for hyper-parameter tuning, allowing us to introduce new algorithms. In addition, our evaluation of individual models in the ensemble yields more precise results, as we can allow similar model specifications to perform differently for different s. We may now capture that some models are more important for forecasting the immediate future and others for the more distant ones.

[3]Note that the first months i with observed outcomes Y cannot be related to any features since we do not have data before τ_0, illustrated by the missing rectangles in the green blocks in Figure 1.

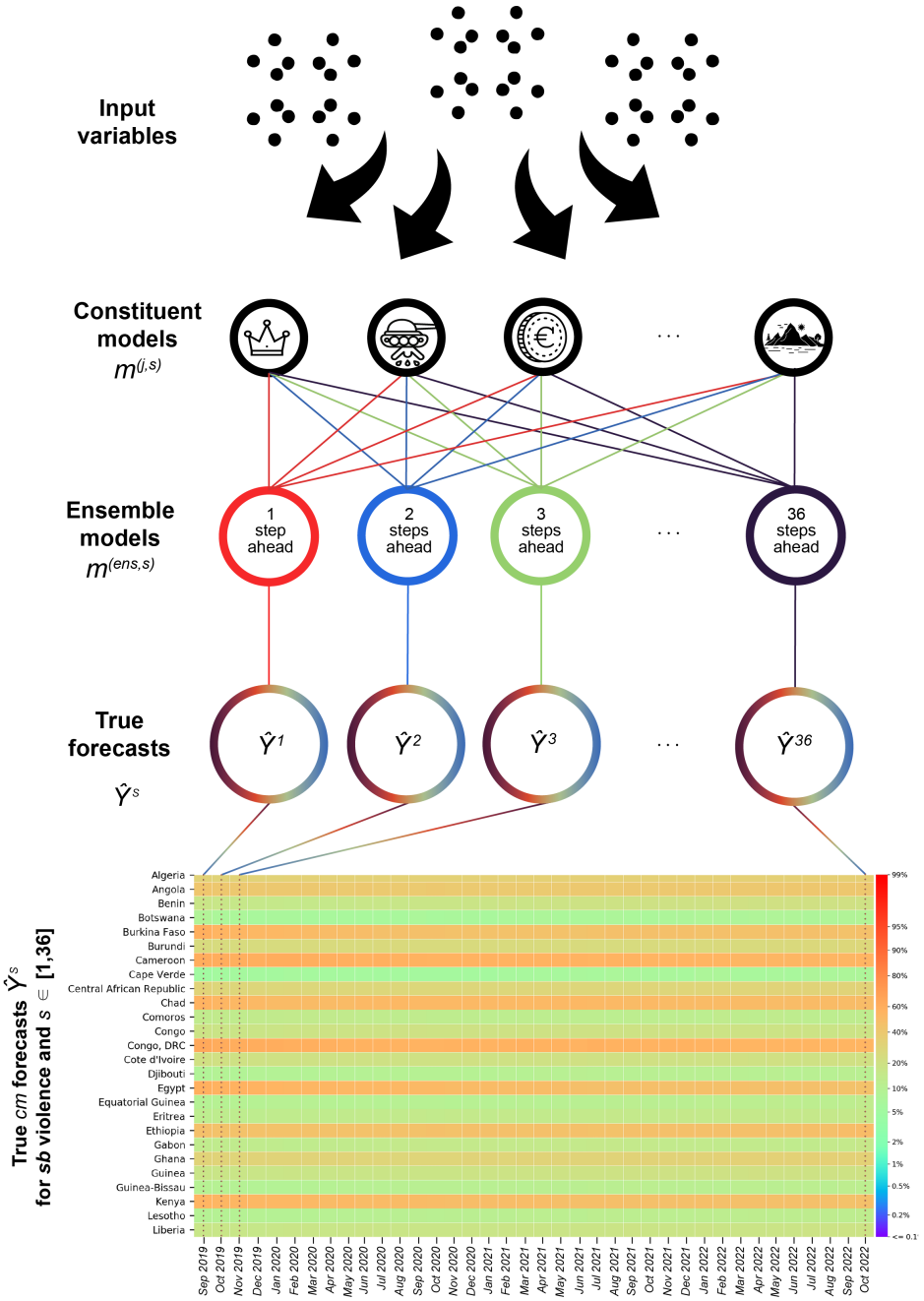
[4]E.g., when evaluating, a $s = 1$ model prediction based on data up to and including May 2014 can be compared to the the actuals of June 2014, as can a step-12 forecast based on data up to May 2013.



The gray sections in the lower figure display the gap between the calibration and forecasting windows in the true forecasting procedure due to missing UCDP-GED data post-2021. UCDP-GED data are compiled on an annual basis, giving rise to this gap. The first dark blue rectangle appearing in the forecasting set of the bottom figure is month $\tau + 1$, here January 2023 (517).

Figure 1. Partitioning and time-shifting in the current pipeline for testing (top) and true forecasting (bottom).

Figure 2 summarizes the data handling procedures for the forecasting application.



The solid color connector lines (red, blue, green and purple) between the constituent and ensemble models in the figure above show how each step-specific ensemble model makes use of step-specific features from relevant constituent models in order to produce its forecasts. What constituent models are relevant for a given ensemble depends on the combination of type of violence (**sb, ns, os**) and unit of analysis (*pgm, cm*) that it is to forecast. The constituent models are in turn informed by a selection of themed input variables, relating e.g. to REIGN data (`https://www.oefresearch.org/datasets/reign`), conflict history, GDP, or geography (as pictured above). At the bottom of the figure, we see an example of predictions as compiled for a subset of countries for the period September 2019 to October 2022, as predicted by our old ensemble in September 2019. This particular heatmap thus displays the predicted probabilities of at least 1 battle-related death occurring as a result of **sb** violence at the *cm* level for all steps $s \in [1, 36]$.

Figure 2. Simplified illustration of ViEWS' true forecasting procedure, exemplified for **sb** violence at the *cm* level.

2 Change history

2.1 Fatalities002

No change in partitioning and time shifting with the exception of time period for calibration. Previously this included the years 2016, 2017 and 2018. Now we extend it to four years, including 2018, 2019, 2020, and 2021.

2.2 Fatalities001

The current setup for partitioning and time shifting was introduced with the Fatalities001 version, thanks to funding from the UK FCDO. As part of these model revisions, we decided to change the length of the calibration and test windows to 48 months, to allow more data for calibration, hyper-parameter tuning, and training of ensemble weights.

The revised partition scheme was first described in Hegre et al. (2022).

2.3 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).

2.4 ViEWS2020

ViEWS2020 was introduced (Hegre et al., 2021).

2.5 ViEWS2018

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

References

Davies, Shawn, Therése Pettersson, and Magnus Öberg (2022). "Organized violence 1989–2021 and drone warfare". In: *Journal of Peace Research* 59.4, pp. 593–610. DOI: 10.1177/00223433221108428. eprint: https://doi.org/10.1177/00223433221108428.

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. (2020). "Introducing the UCDP Candidate Events Dataset". In: *Research & Politics* 7.3, p. 2053168020935257. DOI: 10.1177/2053168020935257. eprint: https://doi.org/10.1177/2053168020935257.

Hegre, Håvard et al. (2022). *Forecasting Fatalities*. Uppsala: working paper.

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.