A&A, 708, A224 (2026)
https://doi.org/10.1051/0004-6361/202553826

Astronomy & Astrophysics

Stellar flare detection in *XMM-Newton* with gradient-boosted trees

Mario Pasquato[1,2,⋆], Martino Marelli[1], Andrea De Luca[1], Ruben Salvaterra[1], Gaia Carenini[3,4], Andrea Belfiore[1], Andrea Tiengo[5,1], and Paolo Esposito[5]

[1] INAF IASF-Milano, Via Alfonso Corti 12, 20133 Milano, Italy
[2] Ciela, Computation and Astrophysical Data Analysis Institute, Montreal, Quebec, Canada
[3] Département d'Informatique, École Normale Supérieure, Université PSL (Paris Sciences & Lettres), Paris, France
[4] Trinity College, University of Cambridge, Cambridge, UK
[5] IUSS Pavia, Piazza della Vittoria 15, 27100 Pavia, Italy

Received 20 January 2025 / Accepted 31 May 2025

ABSTRACT

Context. The EXTraS project, based on data collected with the *XMM-Newton* observatory, provides us with a vast amount of light curves for X-ray sources. For each light curve, EXTraS also provides us with a set of features. From the EXTraS database, we extracted a tabular dataset of 31,832 variable sources based on 108 features. Of these, 13,851 sources were manually labeled as stellar flares or non-flares based on direct visual inspection.

Aims. We employed a supervised learning approach to produce a catalog of stellar flares based on our dataset, subsequently releasing it to the community. We leveraged explainable AI tools and interpretable features to better understand our classifier.

Methods. We trained a gradient-boosting classifier on 80% of the data, which had labels available. We computed the permutation feature importance scores, visualized the feature space using UMAP, and analyzed some false positive and false negative data points with the help of Shapley additive explanations. Specifically, we used it to measure the importance of each feature in determining the classifier's prediction for each instance.

Results. On the test set made up of the remainder 20% of our labeled data, we obtained an accuracy of 97.1%, with a precision of 82.4% and a recall of 73.3%. Our classifier outperforms a simple criterion based on fitting the light curve with a flare template and significantly surpasses a gradient-boosted classifier trained only on model-independent features. False positives appear to be related to flaring light curves that are not associated with a stellar counterpart, while false negatives often correspond to multiple flares or otherwise peculiar or noisy curves.

Conclusions. We applied our trained classifier to currently unlabeled sources, leading to the compilation and release of the largest catalog of X-ray stellar flares to date. We estimated that integrating our classifier into the astronomers' workflow will reduce the time spent on visually inspecting light curves by approximately half, compared to an approach based on flare template fitting. This holds implications for the classification of sources whose variability is less well established within EXTraS as well as for other catalogs and, possibly, forthcoming missions.

Key words. stars: activity – stars: flare – X-rays: binaries – X-rays: bursts – X-rays: general – X-rays: stars

1. Introduction

The EU-FP7 project Exploring the X-ray Transient and variable Sky, (EXTraS, De Luca et al. 2021) characterized the aperiodic and periodic variability of all the sources detected by the soft-X-ray (0.2–12 keV) EPIC camera of the ESA telescope *XMM-Newton*. The project ended in 2016 and resulted in the full variability characterization of about 500 000 detections obtained before 2012. The systematic EXTraS analysis proved to be very powerful both for discovering peculiar phenomena and for studying samples of sources. For instance, Pizzocaro et al. (2016) observed a flare from a very early protostar, while Mereghetti et al. (2018) detected a very short flare of difficult interpretation from the direction of the Galactic globular cluster NGC 6540. Moreover, EXTraS proved to be very sensitive, compared to previous investigations of variability: De Luca et al. (2021) showed that in a sample of 2357 stars compiled by Pye et al. (2015) a much larger number of variable sources can be found in the EXTraS archive[1] with respect to the 3XMM/4XMM catalogs.

At the end of the project, we improved the EXTraS pipelines devoted to aperiodic variability. In particular, we generated light curves with uniform time bins by combining data from the three EPIC cameras, which resulted in a higher sensitivity to variability (De Luca et al. 2022). We ran the updated analysis on all public data up to the end of 2020, obtaining new results such as discovering periodic dips in a peculiar source in M31 (Marelli et al. 2017), determining the orbital period of a fast nova (Marelli et al. 2018), investigating the statistical properties of flares from supergiant fast X-ray transients to constrain accretion models for this peculiar class of binary systems (Sidoli et al. 2019), and discovering an X-ray superflare from a very cool star of spectral class L1 (De Luca et al. 2020).

The systematic analysis of such a large volume of data calls for a machine learning approach, preferably an interpretable one given the need to draw rigorous scientific conclusions from its

⋆ Corresponding author: mario.pasquato@inaf.it

[1] https://extras.inaf.it

results. We previously performed a phenomenological classification of all variable sources from the original EXTraS project with an unsupervised approach, which showed that light curves with similar patterns are clustered based on the temporal features computed by the EXTraS analysis (Kovačević et al. 2022), thus confirming that these features encode useful information for downstream tasks. In this paper, we focus on one such specific classification task, based on a supervised approach: the detection of stellar flares. The study of X-ray flares is a crucial probe of magnetic field generation and dynamics in young stellar objects and, in the main sequence, in cool stars (Pye et al. 2015). It is also very relevant to assess the habitability of extrasolar planets (see, e.g., Kowalski 2024).

A considerable body of work concerning the application of machine learning to time domain astronomy, including the X-ray band, has accrued over the last decade or so. This includes supervised classification tasks on various precomputed features (Lo et al. 2014; Yang et al. 2024; Zhang et al. 2021; Lin et al. 2012; Zuo et al. 2024) as well as unsupervised classification (Kovačević et al. 2022; Pérez-Díaz et al. 2024) and, more traditionally, other approaches to automatically detect transients based on statistical methods (e.g. Quirola-Vásquez et al. 2022, 2023; Yang et al. 2019; Ruiz et al. 2024). In alternative or in addition to statistical learning on human-engineered features, many recent contributions are increasingly focused on learning the relevant features directly from the data by means of deep neural networks (Orwat-Kapola et al. 2022; Ricketts et al. 2023; Song et al. 2025; Dillmann et al. 2025). While we are currently also pursuing such a deep representation learning strategy for light curves and even individual photon detection events (Pasquato et al., in prep.), the present work focuses on a feature-centric approach.

We trained a gradient boosting classifier (Friedman et al. 2000; Friedman 2001) to assign flare versus non-flare labels to light curves (LC) based on a subset of their EXTraS features. Training and validation were conducted on 13 851 manually classified sources. We then applied it to our entire sample of 31 832 variable light curves in the EXTraS catalog, generated from XMM observations collected between 2000 and 2020, obtaining a predicted class and the associated probability for each source. Crucially, we also pursued humanly intelligible explanations for the behavior of our classifier in the context of the eXplainable AI paradigm (XAI; Gunning et al. 2019). Features that are readily understandable by experts in the field allow us to apply the explainability tools more fruitfully. This approach follows the philosophy outlined by Huppenkothen et al. (2023) and exemplified in previous work by some of our team (see, e.g., Pasquato et al. 2024), which establishes explainability as a crucial requirement for machine learning in the natural sciences.

The paper is organized as follows. In Sect. 2, we describe the data and the fiducial sample used to train the model. In Sect. 3, we describe the ML framework. In Sect. 4, we discuss our results, striving to understand the relevance of individual features in the classification and the nature of misclassified instances. We also provide the output catalog of candidate flares. Finally, in Sect. 5, we present our conclusions.

2. Data

In the following, we focus on the features extracted from the intra-observation EXTraS LC (0.2–12 keV flux as a function of time) and derived cumulative distribution functions (CDF; the fraction of time spent by the source below a fixed flux, as a function of the flux itself) of the point-like sources detected by *XMM-Newton/EPIC* between the beginning of 2000 and the end of 2020. These LCs have a bin time of 500 s and last from ∼1 to ∼140 ks, depending on the observation time. Since our aim is to study their variability, we excluded the least variable LCs, defined as the ones that are well-fitted (at 5σ) by a constant model. This resulted in the 31 832 variable LC data sample used in this paper. It should be noted that the ML model presented in this work, even though it was trained on sources of high variability, is applicable to sources of low variability as well, although likely with a decrease in performance. Low variability sources are not amenable to manual classification, however, since the number of sources increases greatly as we lower the cutoff to be considered as variable.

From each LC and associated CDF, we extracted a number of variability indexes; to choose them, we also took into account the ones used in published machine-learning based papers applied on time series mostly in the X-ray domain (Richards et al. 2011; Lo et al. 2014; Farrell et al. 2015). We can divide the variability indexes into three main groups:

- model-independent statistical features: these comprise the most used statistical features (e.g., weighted average, median, skewness, kurtosis) and their associated 1σ errors;
- model-dependent features: we fit the LCs with a set of models and listed the best-fitted value for each parameter and its associated 1σ error and the tail probability; moreover, when possible, the goodness of the fit of different models are compared using an f-test (Bevington 1969);
- model-independent features of the CDF: by renormalizing both the time and the flux between 0 and 1, we give the coordinates of some specific points of the CDF.

From this set of features, we excluded the ones with a straight dependence on the average flux (e.g., the average, the median, the constant feature of every model) since they may bias our model towards brighter sources. This is due to the fact that with the better statistics obtained thanks to higher counts, the type of variability we are looking for is clearer and hence easier to detect. This was already noted during previous work with self-organizing maps (Kovačević et al. 2022). The potential for bias and the steps taken to mitigate it are discussed in Sect. 3.6.

We adopt a final dataset comprising 108 features for each of our 31 832 LCs that are the inputs for the supervised analysis. A brief description of the fitting models and of the features is reported in Appendix A, while a more complete description of the entire work and its results will be presented in Marelli et al. (in prep.).

We trained and tested our machine learning models on a subset of the entire data set (i.e., including all features not depending on the average flux) with human-assigned ground truth: our fiducial sample. We generated it by selecting the sources observed between the beginning of 2012 and the end of 2020. We cross-matched the position with optical and multi-wavelength catalogs to select stars, then performed a visual search for flares in LCs associated to stars.

We matched the position of our X-ray sources, as reported in the XMM serendipitous source catalog[2], with the position of the stars reported in the SIMBAD catalog[3] and in the optical *Gaia*-DR3 (epoch 2016) catalog (Gaia Collaboration 2016, 2023). Using SIMBAD, we considered a star each source cataloged as such, with the exception of the known X-ray binaries (also labeled as HXB, LXB, and/or Pulsar). Using *Gaia* data, we considered a star each optical source with either a confirmed parallax or a confirmed proper motion at three sigma. We

[2] http://xmm-catalog.irap.omp.eu/
[3] https://simbad.cds.unistra.fr/simbad

conservatively considered as a star each X-ray source that falls within 5″ – the mean positional error of *XMM-Newton/EPIC* (Strüder et al. 2001; Turner et al. 2001) – from a star, as previously defined. The 5″ radius we considered for matching is, in fact, generously accounting for both the error attributable to *XMM-Newton*/EPIC and for the much smaller errors in optical catalogues we matched against. This might lead to concerns that matches to stellar objects may occur serendipitously within the 5″ radius even in the absence of a physical association. To check how sensitive our results are to the specific choice of 5″, we reran a cross-match against Gaia with a 2.5″ radius. The number of matches (before vetting for parallax) decreases from 24 415 to 21 367, suggesting that most matches occur well within the 5″ tolerance, so the exact value of the radius does not make a major difference. Nonetheless, this conservative approach may lead to certain flares being erroneously tagged as stellar. Since this limitation is impossible to fully overcome even with perfect positional matching, as the example of a star-compact object binary illustrates, we are currently accepting it as a known shortcoming of the training data.

We visually inspected the LC of stars defined as above in search for flares. Each LC was independently inspected by two different co-authors; each case of disagreement was then discussed, also with the aid of a literature search, if possible (e.g., low-statistics LC containing eclipses only partially covered by the observation could mimic a flare). We conservatively consider a flare each sudden (within few ks) increase in the LC followed by a sudden or gradual decline. Unfortunately, the low statistics of most of the LC make it impossible to differentiate fast-rise exponential-decay (FRED) flares from different types of flares. During the manual labeling phase, the expert coauthors further investigated controversial cases with the help of binning at different time resolutions, as well as adaptively binned LCs following the Bayesian block prescription (Scargle et al. 2013). A curve is marked as "flaring" if the variability is dominated by one or more flares. In a fraction of cases, the flare is not entirely comprised within the LC. Such LCs were, however, tagged as flaring when clearly recognized as such.

Out of 13 851 LCs, we labeled 953 LCs as flaring and 163 of them display more than one flare. Our fiducial sample is shown in terms of the observation duration and counts per bin in the histogram in Fig. 1.

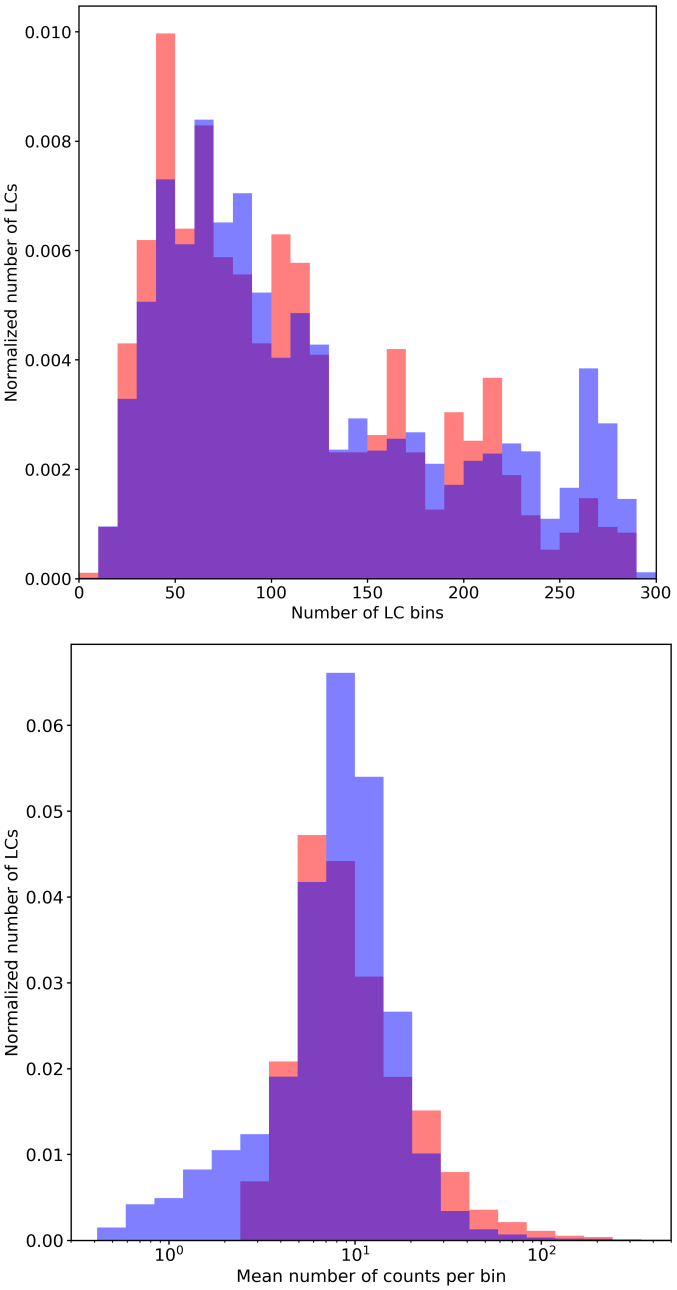

Fig. 1. Histograms of the number of bins (*top*) and mean counts per bin (*bottom*) of our sample of 13 851 light curves. The areas of histograms are normalized to 1. In blue, we show the light curves labeled as "not flaring" and in red the "flaring" ones.

3. Methods

3.1. Train and test split

We divided our initial data set into a training and a test set, comprising 80 and 20% of the full dataset, respectively. The split was stratified on our label, namely, the fraction of sources containing a flare was enforced to be roughly the same in the two data sets. We evaluated the performance of our models on unseen test data.

3.2. Models

We trained a gradient-boosted decision-tree model using `scikit-learn` (Pedregosa et al. 2011) `GradientBoostingClassifier`. Gradient boosting iteratively builds an additive ensemble of weak learners (in our case, shallow decision trees) by fitting each new tree to the negative gradient (residual) of a specified loss function (Friedman et al. 2000; Friedman 2001).

The procedure starts with a scalar baseline predictor that is constant for every sample. For binary classification with logistic loss, this constant equals the log-odds of the prevalence of the positive class,

$$f^{(0)} = \log\left(\frac{1-p}{p}\right), \qquad p = \frac{\#\text{positives}}{\#\text{samples}}. \tag{1}$$

At this stage no tree is grown; the model outputs the same probability for all observations. Each subsequent iteration fits a decision tree $h^{(m)}(\mathbf{x})$ to the current residuals and updates the ensemble prediction,

$$f^{(m)}(\mathbf{x}) = f^{(m-1)}(\mathbf{x}) + \nu\, h^{(m)}(\mathbf{x}),$$

where $\nu \in (0, 1]$ is the learning-rate (shrinkage) factor. Regularization through ν, maximum tree depth, and subsampling controls complexity and mitigates overfitting.

Gradient-boosted trees remain state-of-the-art for tabular data; for instance, the XGBoost implementation (Chen & Guestrin 2016) has repeatedly outperformed more complex deep-learning models on diverse benchmarks (Shwartz-Ziv & Armon 2021).

We stuck to the default hyperparameters, avoiding an optimization step that would be computationally intensive and likely not particularly valuable at this stage. The only parameter we changed from the defaults was the number of estimators (individual trees grown), which we set to 1000 (rather than the default value of 100) as a trade-off between improved performance and computing time requirements.

3.3. Data augmentation

Our classification task was carried out on an imbalanced dataset, with the positive class representing less than 10% of the instances both in the training and (presumably) in the final deployment data. We considered mitigating this issue using the Synthetic Minority Oversampling Technique (SMOTE; Chawla et al. 2011). This consists of creating new synthetic instances of the minority class as convex combinations of a given instance and its nearest neighbors. Whether this approach proves successful ultimately depends on the topology of the data. If minority class instances cluster into two or more disconnected, but otherwise adjacent regions in feature space, the approach can be detrimental to the performance of a classifier trained on the augmented data. This happens because synthetic instances may end up in a region predominantly populated by instances of the majority class, degrading the precision of the classifier by causing an increase in false positive classifications. For our data set, this issue was empirically confirmed using the SMOTE implementation provided by the *imbalanced-learn* library (Lemaitre et al. 2016), which we used with the default settings. We found that a model trained on SMOTE-augmented data where the frequency of the minority class was increased to 20% lost approximately 10% precision compared to the reference model (see Table 1). As a consequence, we avoided using SMOTE to create our final training dataset.

3.4. Performance metrics

We evaluated our models in terms of the predictions on our test set. The metrics we considered are accuracy, precision, recall, and F-score (the harmonic mean of precision and recall; see Rijsbergen 1979), as well as the whole precision-recall curve. Accuracy is the fraction of correctly classified samples over the total. Raw accuracy is not a particularly useful metric in this context, given how unbalanced the classes are in our data set: as we discuss below, an accuracy exceeding 90% can be obtained merely be predicting the majority class (no flare). We still reported our accuracy for each model for the sake of completeness.

Precision and recall are computed for the flare class. The former corresponds to the fraction of actual flares (true positives) over the total number of sources classified as flares. This metric is also called "purity" in the context of astronomical catalogues. Recall refers to the fraction of true positives over the total number of sources that are actually flares. This metric is also called "completeness" in the context of astronomical catalogues. Unlike the accuracy, precision is crucially important for the goals of our classification, which is to produce a sample of candidate sources that are enriched in actual flares, to reduce the time devoted to human visual inspection for a given number of confirmed flares. Recall is also important, but secondary to precision, since the former immediately translates to savings in human classification work.

Typically, and in our case in particular, a classifier returns the predicted probability for each data-point to be a member of a given class. The final binary classification requires setting a threshold on such probability so that a given data-point is predicted as a member if its predicted probability of membership exceeds the threshold. Only then can a value of precision, recall, and accuracy be obtained. The choice of the threshold results in a tradeoff between the risk of false positive results and that of false negative results. As such this choice depends on the relative costs of false positives and false negatives. It is thus common practice to evaluate a classifier at multiple thresholds, obtaining a precision-recall curve that can serve as an overall quantification of the performance of the classifier.

3.5. Explainability

Gradient-boosting models, being based on an ensemble of decision trees, are not inherently interpretable. Using this kind of models is justified given our goal of sifting a large amount of data to reduce manual classification labour, since the ultimate decision on follow-up observations and similarly expensive actions will be still guided by human visual inspection of the relevant light curves. However, seeking explanations for the behavior of our models is helpful in, on the one hand, building confidence in their predictions and on the other, iteratively improving the models.

We thus leveraged XAI techniques to gain insights into our model's behavior. In particular, we used a permutation feature importance (PFI; Altmann et al. 2010) and Shapley additive explanations (SHAP Lundberg & Lee 2017). The former measures the impact of each feature on the model's prediction accuracy by randomly permuting the feature (thus making it useless for prediction) and observing the resulting decrease in accuracy; the latter is a game theoretic strategy introduced originally by Shapley (1953) to equitably divide earnings in cooperative games, which has been applied to assign a value to each feature based on its contribution to the prediction on a given data point when it is combined with other features. This method considers all possible combinations[4] of features as player coalitions and computes the marginal contribution of each feature to the prediction outcome by comparing the prediction with and without the given feature. The Shapley value for a feature is then calculated as the weighted average of its marginal contributions across all possible coalitions. This approach ensures a fair attribution of the prediction outcome to individual features, adhering to properties such as efficiency (the total contribution of all features equals the total change in prediction from a baseline prediction), symmetry (features with identical contributions receive identical Shapley values), dummy (features that do not change the prediction receive a Shapley value of zero), and additivity (the Shapley values for a model that is a sum of several models equal the sum of the Shapley values of each model). Shapley values, unlike a PFI, can deal with interactions between features because it considers all possible coalitions.

We also visualize the space of the most important features (according to a PFI) using Uniform Manifold Approximation and Projection (UMAP; McInnes et al. 2018). UMAP leverages a graph representation of the data in the high-dimensional input space, by connecting each point to its nearest neighbors. The

[4] Computational considerations permitting.

number of points considered in this step is adjustable by the user and influences the final outcome by privileging global over local structure as it is increased. Having built a suitable graph, UMAP then seeks to embed this graph in a lower-dimensional space in a way that best preserves faithfulness to the original data. This process involves minimizing the cross-entropy between two similar fuzzy sets representing the high-dimensional and low-dimensional spaces, respectively.

We run PFI 50 times, thus obtaining mean importances and their standard deviations over the 50 runs. This allows us to select the features that are important at one standard deviation, which yields six features. UMAP was run on the space of these six features after each one has been scaled using the robust scaler preprocessing tool (from *scikit-learn* preprocessing), so that the first quartile of the original feature corresponds to 0 in the scaled feature and the third quartile to 1. A tedious manual exploration of the results of adopting different hyperparameters for UMAP was carried out while searching for insight into our data set. In particular, we tweaked the *n_neighbors* and *min_dist* hyperparameters in umap-learn's implementation of UMAP to avoid the formation of "horseshoes", winding, elongated one-dimensional structures that appear when a dimension reduction method recognizes an essentially one-dimensional structure in a section of the data (Diaconis et al. 2008). These may arise even in the absence of a genuinely one-dimensional manifold in the data-generating process; for instance, due to overly local connectivity. Furthermore, we varied UMAP's hyperparameters to confirm that certain features on the map, in particular, islands and peninsulas to which we attached a meaning as discussed in Sect. 4.2, were stable. As an additional way to check that the features selected by PFI are indeed capturing most of the useful signal for our classification task, we retrained a model on these features only, obtaining a limited reduction in performance.

Finally, we used individual conditional expectation (ICE) curves to understand how the model predictions depend on each feature. The ICE plots are a visualization technique used to analyze the effect of a single feature on the predictions made by a machine learning model, for a sample of observations from the dataset. An ICE plot does this by varying the value of the feature of interest across its range and observing how the prediction changes, while keeping all other features fixed. This process is repeated for every LC, resulting in a series of lines on the plot: one for each LC. These lines show how the predictions would change if only that one feature were altered, providing insight into the feature's individual effect on the model output. The ICE plots are particularly useful for identifying how the relationship between a feature and the prediction varies across different instances in the dataset, showcasing the heterogeneity in the model dependence on that feature.

3.6. Bias mitigation

Machine learning models will use any information available in the data to predict the labels, even when a human would understand that certain information is better left out. Several features are correlated with overall source brightness, but we do not want our model to more confidently classify a bright source as a flare just because the data show that a flare pattern is more easily spotted by human experts in a bright source than in a dim one. The appropriate theoretical framework for discussing this kind of biasing effect introduced by certain features is that of fairness in machine learning (see Caton & Haas 2024, for a review).

Fairness in classification is commonly formalized with respect to a protected attribute, A, that the classifier should not discriminate on. A simple statistical notion of fairness is demographic parity, requiring that the distribution of predictions be the same between groups, independently of the true label, expressed as

$$\forall s,\ \forall a, b: \quad P(S = s \mid A = a) = P(S = s \mid A = b). \tag{2}$$

For a given definition of fairness, departure from these objectives counts as evidence of bias. There are several mitigation strategies, the most intuitive of which is so-called "fairness by unawareness", consisting of removing the protected attribute from the training features. In our case, features such as AVE were not included in the training set. However this approach in not sufficiently effective: a machine learning model will latch onto any other feature that correlates with the protected attribute if that leads to improved predictions.

We have thus defined our protected attribute as the new binary feature BRIGHT, obtained by thresholding AVE on its median. The hand-labeled data show a mild but significant difference in the prevalence of flares among BRIGHT=True and BRIGHT=False sources: 8.3% against 7.0%, respectively. We used the fairlearn library (Bird et al. 2020) to test a correlation-removal pre-processing mitigation strategy for our dataset in addition to fairness by unawareness.

4. Results

4.1. Performance on test data

The accuracy, precision, recall, and F-score for our models on our test set (withheld in training) are presented in Table 1. We compare our full model (first row in the table), trained on all available features, to a model trained on a subset of features that excludes features obtained from model-dependent and computationally intensive fits of physically motivated templates to LC (second row) and to a model trained only on the most important features according to PFI (listed in Table 2; last row). To compare the performance of our model with a simple human-based method, we do not have a standard baseline. To search for flares, we usually make a cut on F_NSIGMA_FLCON[5], which is the probability coming from an f-test (Bevington 1969) performed by comparing a constant+FRED model with a constant model[6]. Thus, in order to search for real flares, we cut out every LC with a F_NSIGMA_FLCON below a certain threshold, usually 5. This is expected to be the best parameter to cut, since, by definition, it indicates the improvement by adding a flare model even in the case of a multi-component LC, while the parameter associated with the goodness of the constant+FRED flare (FL_NSIGMA) would fail. Thus, we compare our models with this cut (third row in Table 1). We notice how the full model outperforms every other in terms of accuracy and precision, while the baseline flare fit yields a high recall at the expense of precision, which is an abysmal 33.4%. Moreover, it is possible to change the classification threshold of our full model to obtain the same level of recall as the baseline flare fit, while keeping a precision of 66.0%, which is still roughly double the baseline.

The tradeoff between precision and recall is best appreciated in Fig. 2, where we present three precision-recall curves obtained

[5] A multi-component cut using other flare model parameters does not result in a significant improvement, or does introduce clear biases.

[6] We report the probability in terms of number of sigmas in order to normalize the distribution of the values, thus a low probability turns in a high value of F_NSIGMA_FLCON that show that the improvement by adding the flare model to the constant is hardly by chance.

Table 1. Metrics for our models.

Model	Accuracy	Precision	Recall	F-score
Full	97.1%	82.4%	73.3%	77.6%
No fit	95.0%	69.4%	48.7%	57.2%
Cutoff	87.0%	33.4%	89.5%	48.6%
Majority	93.1%	NA	0.0%	NA
Important feat.	96.2%	77.8%	70.9%	74.2%
SMOTE	95.6%	69.2%	77.9%	73.3%

Notes. Performance metrics for the different models considered. The full model is trained on all available features. The restricted model ("No fit") is trained on a subset of model-independent features. The cutoff model relies on a flare template fit. The majority model always predicts the majority class. The "Important feat" model uses only the most PFI-important features. The SMOTE model is trained on a dataset augmented using the SMOTE algorithm.

Table 2. Permutation feature importance.

Feature	Importance	Model-dependent?
F_NSIGMA_FLCON	0.059 ± 0.004	Yes
FL_NSIGMA	0.010 ± 0.002	Yes
FL_DT_ERR	0.009 ± 0.002	Yes
MEDMAXOFF	0.005 ± 0.002	No
FLUX_P50	0.0006 ± 0.0004	No
TFRAC_BEL1S	0.0005 ± 0.0003	No

Notes. We included the features that are important at least at one sigma. Features are marked as model dependent if they are the result of fitting a physically motivated template to the LC.

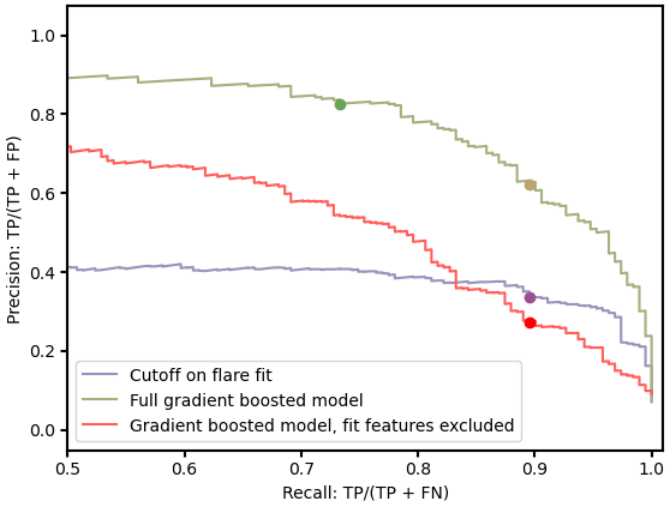

Fig. 2. Precision-recall curves for setting a cutoff on F_NSIGMA_FLCON (purple), with the brighter purple dot corresponding to a five-sigma cutoff. The green curve corresponds to our best gradient boosting model (including all features), with the brighter green dot corresponding to a threshold of 0.5. The brown dot on this curve corresponds to the level of precision that would be reached at the higher recall reached by the five-sigma cutoff on F_NSIGMA_FLCON. The red curve corresponds to a gradient boosting model including only features that do not derive from fitting physically meaningful templates to the light curve (such as flares, dips, eclipses, etc.), and the brighter red dot corresponds to the level of precision that would be reached at the recall reached by the five-sigma cutoff on F_NSIGMA_FLCON.

by varying the relevant thresholds for each classification method, while the results reported in Table 1 correspond to a conventional threshold of 0.5. The green curve represents our gradient boosted classifier including all 108 features, the red curve the classifier trained only on model-independent features and the purple curve the cut on the F_NSIGMA_FLCON parameter (in this case, the thresholds are the different possible values at which we apply the cut).

Each point on any of these lines represents a value of precision and recall obtained by changing said threshold, without altering the underlying model. A curve situated towards the upper right of the plot represents a higher overall performance.

The classifier trained on all 108 features clearly outperforms the others at all recall levels. Excluding model-dependent features strongly reduces the performance, although at sufficiently low recall (below ≈0.835) it still performs better than a naive cutoff on F_NSIGMA_FLCON.

4.2. The role of important features

Table 2 presents the PFI importance for the most important features. These have been selected by repeating the permutation procedure 50 times and obtaining mean and standard deviation values of the PFI for each feature, including in the final table only those for which the mean differs from zero by at least one standard deviation.

The features deemed important include F_NSIGMA_FLCON and FL_NSIGMA, two parameters that are related to the goodness of fit with a flare template, as well as FL_DT_ERR, quantifying the error on the decline time of the flare fit. Additionally, MEDMAXOFF, FLUX_P50, and TFRAC_BEL1S further characterize the variability of the source in a model independent way. It is unsurprising that F_NSIGMA_FLCON and FL_NSIGMA appear as important features, given the fact that these statistics are designed precisely to select the curves that most resemble a flare template. Similarly, FL_DT_ERR represents the error in the decay time of the flare template fit; it is clear that a large error may indicate a poor fit, and may also appear in cases where the rise in flux happens close to the end of the observation. Those are most likely spurious. The role of MEDMAXOFF is also immediately evident: a large difference between the maximum and the median of a light curve signifies a sudden spike that decays rapidly.

To gain further insight into the role of this subset of most important features with respect to the others, we trained a new model only on them. As shown in Table 1, the full model outperforms the restricted model, but not by a huge margin: raw accuracy drops by about one percentage point. While the six most important features are sufficient to constrain predictions on the bulk of the data, it appears likely that edge cases have to rely on less frequently used features. In fact this is confirmed by our subsequent analysis of instance-level predictions, on false-positives and false-negatives.

To better understand the relationships between features and their interactions in prediction, we explored the entire six-dimensional space of the features that are important at one sigma to identify where flares are preferentially located with respect to relevant substructures such as clustering of similar sources. This was carried out by means of visualization with UMAP, which allowed us to represent our six-dimensional data points on the plane, as shown in Fig. 3, where we also mark actual

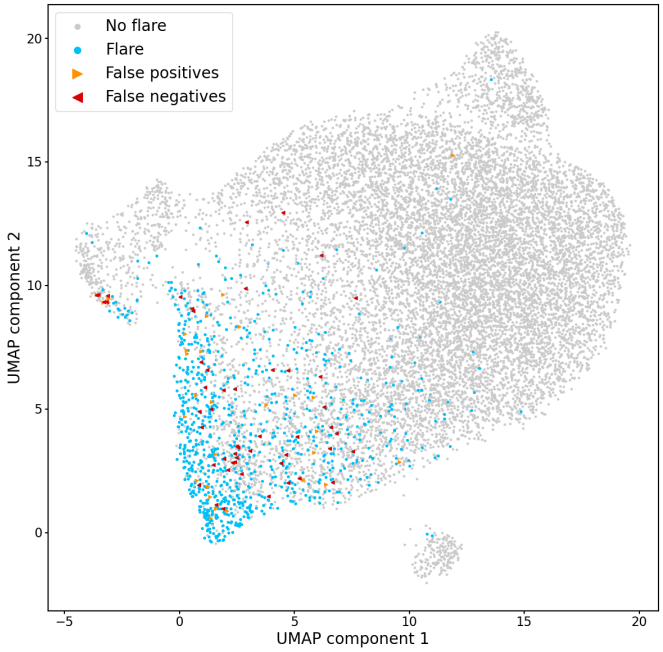

Fig. 3. UMAP embedding calculated on the important features from the training set (flares in cerulean blue, non-flares in light gray), test set displayed in the same coordinates. False positives within the test set are shown in orange and false negatives in red.

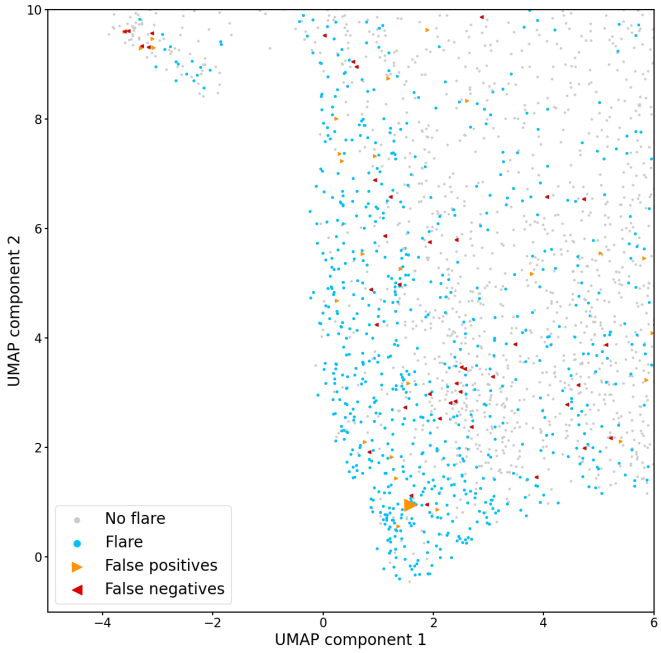

Fig. 4. UMAP embedding calculated on the important features from the training set (flares in cerulean blue, non-flares in light gray). The test set is displayed in the same coordinates. False positives within the test set are shown in orange and false negatives in red. Zoom on flareland, showing the false positive source 0722360301/5 (discussed below) with a bigger triangle.

flares, false positives, and false negatives. Individual features are visualized on the same plane in Appendix B. Two of the most prominent features, the fin-like peninsula at the top right and the small island at the bottom, are related to extreme values of FL_DT_ERR: unrealistically large (much longer than the typical duration of an observation) in the former, and exactly zero in the latter. Both values indicate that a flare fit essentially failed, and in fact the top peninsula contains no flares and the bottom island only two. The peninsula at the top left is associated to high values of FL_N_SIGMA, even though it does not contain many flares: these are sources that are mostly not stellar flares, yet have been well fit by a flare template. These may be curves with a step increase in flux, which end up well fitted by flare models with a long decay time. At the bottom-left we find flareland: there F_N_SIGMA_FLCON and MEDMAXOFF are high, while FL_DT_ERR and FLUX_P50 are low. Flareland is shown at a closer level of zoom in Fig. 4.

A complementary way to understand how our machine learning model is using our features is ICE curves. Fig. 5 shows that the full model is strongly relying on the flare fit result to find flares, as expected. The predicted probability of being predicted as a flare increases following a sigmoid-like curve, that is slowly at first, then roughly linearly with FL_NSIGMA_FLCON up until almost 1, where it flattens out. The ICE curves are mostly non-intersecting and similar in shape, suggesting that FL_NSIGMA_FLCON does not have major interactions with other features.

As a second example of feature whose meaning gets clarified by an ICE plot, we show MEDMAXOFF in Fig. 6. A flare is a sudden, isolated increase in flux, so a large offset between the median and the maximum of the light curve is a signal that a source may be flaring. This is however per se not a sure indication of a flare, hence for most data points the response shown by the ICE curve is flat (bottom part of the image), while the model relies on MEDMAXOFF only for points that, due to other

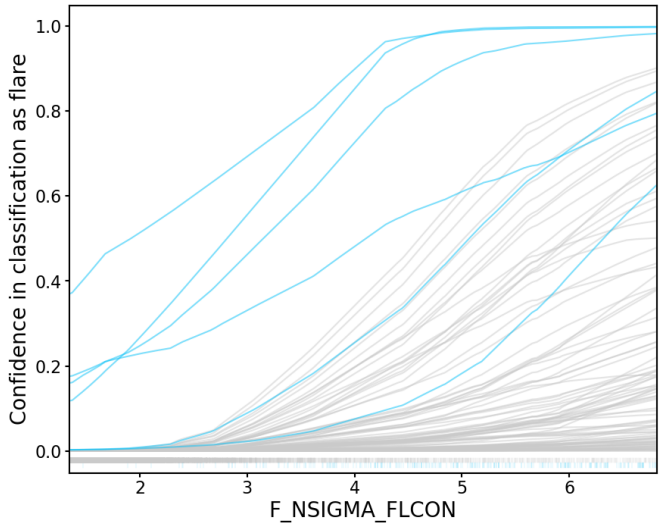

Fig. 5. ICE plot for F_NSIGMA_FLCON. Curves for 100 randomly chosen sources are shown. Non-flare sources are shown in gray, flares in cerulean blue. At the bottom a rug plot shows the actual values of the feature taken on by flares (cerulean blue) and non-flares (gray).

features, are already suspect flares. This can be understood as an example of interaction between features.

4.3. Understanding misclassified instances

Our classifier misclassifies 81 LC out of 2771. It is natural to wonder why such a misclassification would occur. In Fig. 7, we show a paradigmatic misclassified source among the false positives for which the model predicts the flare class but the ground truth label has no flare. Ground truth labels were assigned by

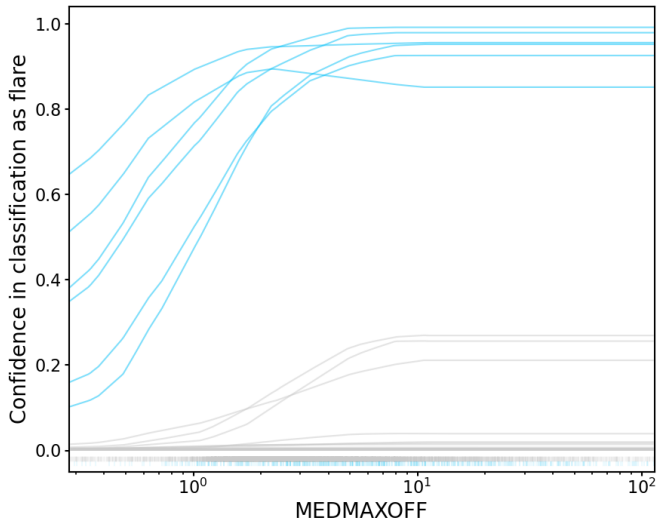

Fig. 6. ICE plot for MEDMAXOFF. Curves for 100 randomly chosen sources are shown. Non-flare sources are shown in gray, flares in cerulean blue. At the bottom a rug plot shows the actual values of the feature taken on by flares (cerulean blue) and non-flares (gray).

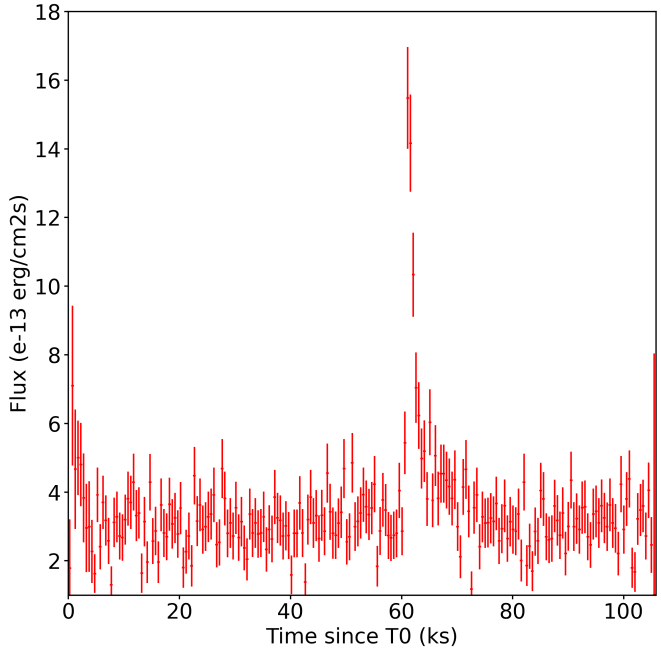

Fig. 7. Light curve for 0722360301/5. An apparent flare, but it is not associated with a star. T0 is the time of the first photon of the observation.

human inspection to flare patterns that corresponded to a stellar counterpart. However flare-like activity can be associated also to non-stellar sources or to stars that are undetected in current surveys and this is likely often the case for false positives: Fig. 7 in particular appears to be a clear cut flare; in fact, Fig. 4 shows it deeply embedded in flareland, but it does not have a stellar counterpart. Consequently, it has not been labeled as a stellar flare. Interestingly, SHAP shows (Fig. 8) the source was in fact classified as such because of variables related to the flare template fit, whose role in classification we have clarified above. Shapley values thus suggest that false positives may

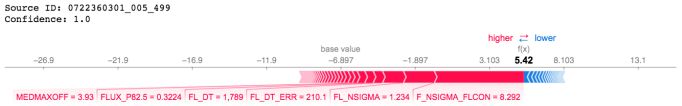

Fig. 8. Shapley values for a paradigmatic false positive source, 0722360301/5.

Table 3. First lines of the online material fits file, reporting the catalog of candidate flares.

OBS_ID	SRC_NUM	RA	Dec	ST_FLARE	PROB
0000110101	1	64.926	55.999	F	0.0006
0000110101	6	64.996	56.225	F	0.0002
0001730201	1	263.677	–32.582	F	0.02
0001730201	9	263.649	–32.596	T	0.99

Notes. OBS_ID is the observation identifier of the *XMM-Newton* observation and SRC_NUM the source number; these two parameters define the "name" of the source. Then, we report the fk5 celestial coordinates, in degrees (RA, Dec). The last two colums are the results of our model: ST_FLARE show if the source is predicted to be a flare, on the basis of the threshold of 0.5 on the resulting probability, PROB.

be flares not associated to a stellar source. This is a limitation of the training set rather than of the ML model: a purely phenomenological definition of flare would likely have resulted in a lower incidence of false positives, but we chose to label only confirmed stellar flares as positive instances in our training data to maximize the usefulness of the catalog for scientific purposes.

We carried out a similar analysis for false negative instances. In Fig. 9, we show three sources confidently classified as not having a flare while the ground truth class was flare. Clearly noise, non-flaring variability, and the presence of multiple flares (top panel) played a role in the misclassification of these LCs. The Shapley plots are shown in Fig. 10. Here, we see that the contribution to the final decision of the classifier comes mostly from features related to eclipse and dip fits. These are not among the important features we discussed above, but are playing an important role for these sources, possibly because the usual features did not provide a clear cut result.

4.4. The catalog of candidate flares

We applied the trained model to the 31 832 EXTraS variable LC with uniform time bin (500 s) generated from *XMM-Newton* data collected between 2000 and 2020. We considered as candidate flares all LC predicted as flares with probability over 0.5, corresponding to a precision of 82.4% and recall of 73.3%. We obtain 2088 candidate flares, of which we expect 1721 ± 18 to be actual stellar flares. In the online material we report the probability of "stellar flareness" coming from our model for each of the 31, 832 sources together with their classification as a flaring LC, which is simply based on a threshold of 0.5 on the probability, as already discussed. Each source is defined by a combination of observation id and source number, as well as the associated celestial coordinates reported in the XMM catalog. Here, we note that we run the EXTraS tools using the XMM catalog version available at the moment of the run or (if not available for a given observation) we used the standard results from the

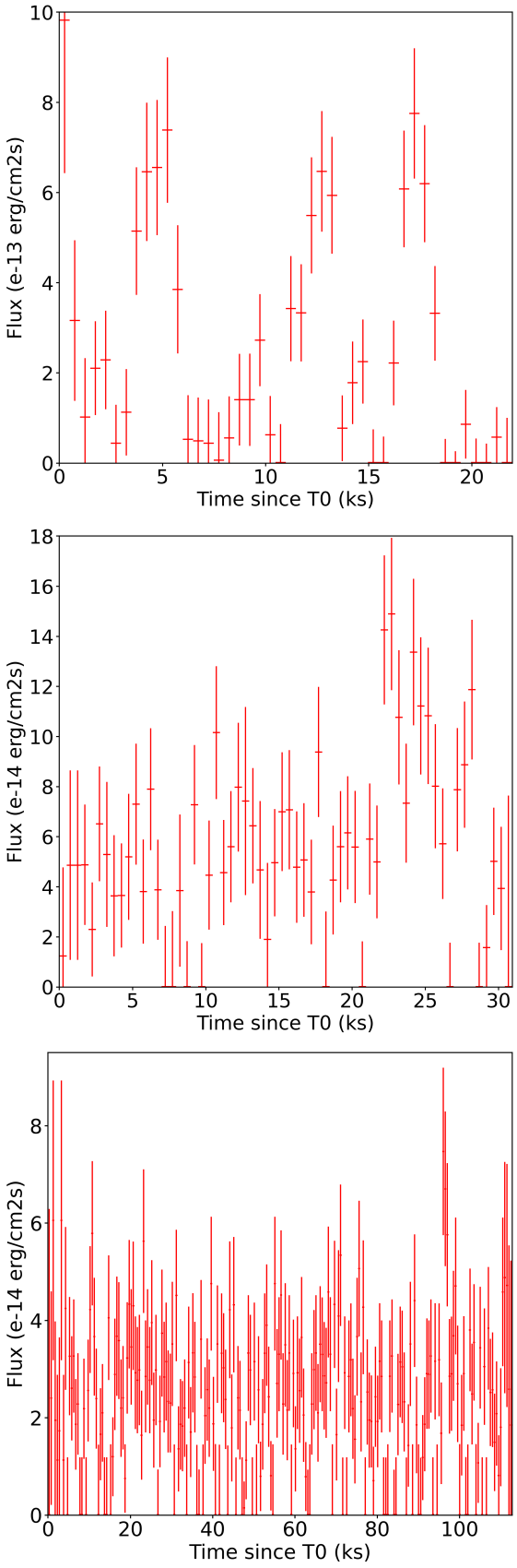

Fig. 9. Light curves for false negative sources 0728560301/4, 0841320101/2, 0822200101/6. The variability of the first LC has been ascribed to three random flares; the second LC shows a feature around t ∼ 22 ks; the third LC shows a probable short (∼1 ks) flare at t ∼ 95 ks. T0 is the time of the first photon of the observation.

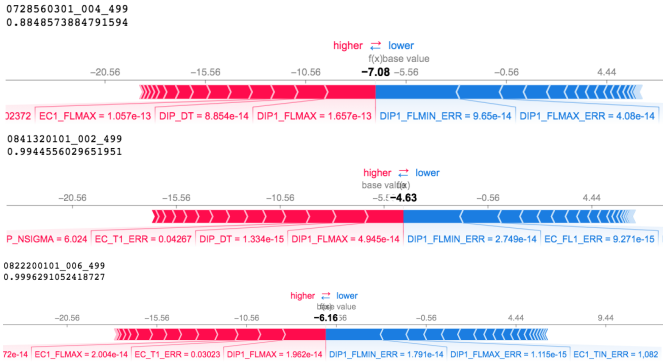

Fig. 10. Shapley values for false negative source 0728560301/4 (top), 0841320101/2 (mid), 0822200101/6 (bottom).

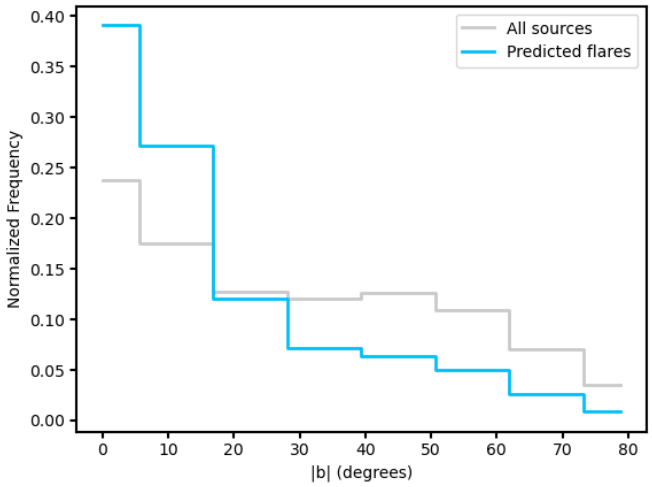

Fig. 11. Histogram of the modulus of Galactic latitude (in degrees) of sources that have been tagged as flares in the final catalog (cerulean blue) superimposed onto the histogram of all sources in the catalog. The two histograms share the same normalization.

XMM-Newton Pipeline Processing System (PPS)[7]. Table 3 shows the first lines of the online material fits, as an example. As reported in Sect. 4.3, among the false positives, there are a number of real flares that are not associated to a *Gaia* or *SIMBAD* optical counterpart. Meanwhile, the main source of false negatives seems to be the presence of a large number of flares in the same LC or, more often, a noisy, non-flaring variability.

A simple check that our results are consistent with the expectation based on the stellar origin of flares for the majority of our sources, we show the histogram of the absolute value of Galactic latitude for flaring sources (as defined by our model and threshold) in Fig. 11. We compare it to that of all sources and it is clear that candidate flares are more concentrated towards the Galactic plane, suggesting that indeed they are mostly genuine stellar flares. A two-sample Kolmogorov-Smirnoff test rejects the null hypothesis that candidate flares share the same distribution as the other sources in our catalog with $p = 1.1 \times 10^{-139}$.

[7] Since minor changes to the detection tools take place across different versions of the Science Analysis System (SAS) and different releases of the catalog not only add new entries but also re-run the detection for some observations, we warn that the reported coordinates can slightly change if compared to a given catalog distribution, as well as the source number (that is ordered based on the predicted count rate, from the highest to the lowest).

Table 4. Correlation removal pre-processing.

Model	Accuracy	Precision	Recall	Deviation from dem. parity
Imp. feat.	96.2%	77.8%	70.9%	13.7%
Corr. Rem.	96.2%	77.4%	70.9%	0.07%

Notes. Performance of the model trained on PFI-important features (top row) compared with an identical model trained on the same features preprocessed with CorrelationRemover (bottom row). The deviation from demographic parity was measured as the percent difference in the fraction of objects predicted as flares on the test set by either classifier.

4.5. Bias mitigation

The final adopted model was not based on any pre-processing or post-processing intended to enforce demographic parity with respect to source brightness or signal-to-noise ratio. We did however explore bias mitigation strategies such as pre-processing to remove correlations with a protected attribute defined as a binary variable BRIGHT corresponding to thresholding AVE against its median. This was carried out using CorrelationRemover in fairlearn on the PFI important features only. It allowed us to reach an approximately equal selection rate (equal fraction of predicted flares among bright sources and dull sources) at the expense of some precision (about 0.4%). This is shown in Table 4.

5. Discussion and conclusions

We have shown that a gradient-boosted classifier trained on a set of 108 variability features derived from X-ray source light curves is able to classify stellar flares with an accuracy of 97.1%, a precision of 82.4%, and a recall of 73.3% on an unseen test set. This is a good result for a classifier that works only on summary features, without direct access to light curves.

We used our classifier to compile and release to the public the largest unbiased (in terms of sky coverage) catalog of stellar flares to date. Further applications of our work may involve classification of sources from the EXtraS archive that are characterized by less of a clear variability in terms of goodness of fit with a constant, as well as an extension of our approach to other datasets, such as those produced by the forthcoming ATHENA (Nandra et al. 2013) and AXIS (Mushotzky et al. 2019) missions, as well as those of Chandra or eROSITA, for instance. Clearly, this is feasible as long as comparable features are computed on the relevant light curves and after checking for significant distribution shifts. This is likely more feasible for a feature-based approach such as ours than for a deep learning model, where features are learned automatically, thereby reducing our control on their meaning.

Our results show that even a simple machine learning approach can save a considerable amount of work for human expert annotators, who are nonetheless still likely to be required in a data analysis pipeline. We can quantitatively estimate the impact of our work, given N sources of which there are pN actual flares; namely, p is the actual prevalence of flares.

Given a model with recall, r, it will correctly tag rpN sources as flares: noting that recall is

$$r = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{3}$$

Additional $s(1-p)N$ sources, where s is the false positive rate, were also tagged as flares.

$$s = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = \frac{rp(1/P - 1)}{1 - p}, \tag{4}$$

where P is precision,

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{5}$$

So in total a human expert will have to look at

$$(rp + s(1-p))N = \frac{rp}{P}N, \tag{6}$$

sources out of at total of N to catch rpN actual flares.

We can now compare two models with precision of P and $P' > P$ in terms of how many more sources a human expert will have to look at when using the worse model with respect to the better model. With the model that has precision, P, to identify M actual flares the expert will need to look at M/P sources; with the model that has precision P' the expert will need to look at M/P' sources.

For our model, $P' = 82.4\%$. However, if we force it to have the same recall as the simple cutoff based on template fitting, we get $P' = 66.0\%$, compared to $P = 33.4\%$ for this method. Even in the latter case, the amount of manual labor saved is roughly a factor two. It should be noted that these benefits are not present if we insist to train our classifier on model-independent features only, suggesting that such features are inadequate (in isolation) for capturing flares. This justifies further work directly on light curves, possibly using deep learning to learn features directly from the data.

However, one of the drawbacks of this method is its opaque nature. To alleviate this issue, we applied a variety of explainable AI techniques to our trained classifier. We used permutation feature importance scores to identify the most important features among those used by our classifier, which are shown in Table 2.

Additionally, we visualized the space of these important features using UMAP and obtained ICE curves for each feature. ICE curves visualize how our model prediction on a given data point changes by varying only one feature while keeping the others fixed. ICE curves are mostly non-intersecting for the features we considered, suggesting the lack of interactions between features. This is an important clue for building better models: in particular generalized additive models (GAM; Hastie & Tibshirani 1986) may be suited to our problem. GAM are inherently interpretable since they predict their outcomes by modeling the relationship between the response variable and individual predictors as a sum of smooth functions. This is viable only in the context of limited interactions between variables.

We also analyzed the false positive and false negative data points with the help of Shapley values and visualized their light curves. In hindsight, most false positives appear to be issues with the labels and we note that a human expert would classify those as flares as well. Moving forward, it seems crucial to develop a standardized protocol for the visual inspection of flares to be used as training instances; however, this is beyond the scope of the current paper. False negatives are the result of genuine mistakes made by our model (which are often peculiar curves, such as multiple flares).

Data availability

The code used to train the models is available at https://github.com/m-a-r-i-o/supervised-flare-detection. The catalogue is available at the CDS via https://cdsarc.cds.unistra.fr/viz-bin/cat/J/A+A/708/A224 .

Acknowledgements. The authors thank the anonymous referee for pointing out several issues in the original version of the manuscript. This work was funded in part by the INAF large grant for the project Astronomy with Natively Interpretable MAchine learning (ANIMA, PI Mario Pasquato).

References

Altmann, A., Toloşi, L., Sander, O., & Lengauer, T. 2010, Bioinformatics, 26, 1340
Bevington, P. R. 1969, Data Reduction and Error Analysis for the Physical Sciences (Columbus: McGraw-Hill Science Engineering)
Bird, S., Dudík, M., Edgar, R., et al. 2020, Fairlearn: A toolkit for assessing and improving fairness in AI, Technical Report MSR-TR-2020-32, Microsoft
Caton, S., & Haas, C. 2024, ACM Comput. Surv., 56, 7
Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. 2011, arXiv e-prints [arXiv:1106.1813]
Chen, T., & Guestrin, C. 2016, arXiv e-prints [arXiv:1603.02754]
De Luca, A., Stelzer, B., Burgasser, A. J., et al. 2020, A&A, 634, L13
De Luca, A., Salvaterra, R., Belfiore, A., et al. 2021, A&A, 650, A167
De Luca, A., Israel, G. L., Salvaterra, R., et al. 2022, Mem. Soc. Astron. It., 93, 122
Diaconis, P., Goel, S., & Holmes, S. 2008, arXiv e-prints [arXiv:0811.1477]
Dillmann, S., Martínez-Galarza, J. R., Soria, R., Stefano, R. D., & Kashyap, V. L. 2025, MNRAS, 537, 931
Farrell, S. A., Murphy, T., & Lo, K. K. 2015, ApJ, 813, 28
Friedman, J., Hastie, T., & Tibshirani, R. 2000, Annal. Statis., 28, 337
Friedman, J. H. 2001, Annal. Statis., 29, 1189
Gaia Collaboration (Prusti, T., et al.) 2016, A&A, 595, A1
Gaia Collaboration (Vallenari, A., et al.) 2023, A&A, 674, A1
Gunning, D., Stefik, M., Choi, J., et al. 2019, Sci. Robot., 4, eaay7120
Hastie, T., & Tibshirani, R. 1986, Statis. Sci., 1, 297
Huppenkothen, D., Ntampaka, M., Ho, M., et al. 2023, arXiv e-prints [arXiv:2310.12528]
Kovačević, M., Pasquato, M., Marelli, M., et al. 2022, A&A, 659, A66
Kowalski, A. F. 2024, Liv. Rev. Sol. Phys., 21, 1
Lemaitre, G., Nogueira, F., & Aridas, C. K. 2016, arXiv e-prints [arXiv:1609.06570]
Lin, D., Webb, N. A., & Barret, D. 2012, ApJ, 756, 27
Lo, K. K., Farrell, S., Murphy, T., & Gaensler, B. M. 2014, ApJ, 786, 20
Lundberg, S. M., & Lee, S.-I. 2017, Adv. Neural Information Process. Syst., 30
Marelli, M., Tiengo, A., De Luca, A., et al. 2017, ApJ, 851, L27
Marelli, M., De Martino, D., Mereghetti, S., et al. 2018, ApJ, 866, 125
McInnes, L., Healy, J., & Melville, J. 2018, ArXiv e-prints [arXiv:1802.03426]
Mereghetti, S., De Luca, A., Salvetti, D., et al. 2018, A&A, 616, A36
Mushotzky, R., Aird, J., Barger, A. J., et al. 2019, Bull. Am. Astron. Soc., 51, 107
Nandra, K., Barret, D., Barcons, X., et al. 2013, arXiv e-prints [arXiv:1306.2307]
Orwat-Kapola, J. K., Bird, A. J., Hill, A. B., Altamirano, D., & Huppenkothen, D. 2022, MNRAS, 509, 1269
Pasquato, M., Trevisan, P., Askar, A., et al. 2024, ApJ, 965, 89
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, J. Mach. Learn. Res., 12, 2825
Pérez-Díaz, V. S., Martínez-Galarza, J. R., Caicedo, A., & D'Abrusco, R. 2024, MNRAS, 528, 4852
Pizzocaro, D., Stelzer, B., Paladini, R., et al. 2016, A&A, 587, A36
Pye, J. P., Rosen, S., Fyfe, D., & Schröder, A. C. 2015, A&A, 581, A28
Quirola-Vásquez, J., Bauer, F. E., Jonker, P. G., et al. 2022, A&A, 663, A168
Quirola-Vásquez, J., Bauer, F. E., Jonker, P. G., et al. 2023, A&A, 675, A44
Richards, J. W., Starr, D. L., Butler, N. R., et al. 2011, ApJ, 733, 10
Ricketts, B. J., Steiner, J. F., Garraffo, C., Remillard, R. A., & Huppenkothen, D. 2023, MNRAS, 523, 1946
Rijsbergen, C. v. 1979, Information Retrieval (Massachusetts: Butterworth-Heinemann)
Ruiz, A., Georgakakis, A., Georgantopoulos, I., et al. 2024, MNRAS, 527, 3674
Scargle, J. D., Norris, J. P., Jackson, B., & Chiang, J. 2013, ApJ, 764, 167
Shapley, L. S. 1953, The Shapley Value (Princeton: Princeton University Press)
Shwartz-Ziv, R., & Armon, A. 2021, arXiv e-prints [arXiv:2106.03253]
Sidoli, L., Postnov, K. A., Belfiore, A., et al. 2019, MNRAS, 487, 420
Song, Y., Villar, V. A., Martinez-Galarza, J. R., & Dillmann, S. 2025, ApJ, 988, 143
Strüder, L., Briel, U., Dennerl, K., et al. 2001, A&A, 365, L18
Turner, M. J. L., Abbey, A., Arnaud, M., et al. 2001, A&A, 365, L27
Yang, G., Brandt, W. N., Zhu, S. F., et al. 2019, MNRAS, 487, 4721
Yang, H., Hare, J., & Kargaltsev, O. 2024, ApJ, 971, 180
Zhang, Y., Zhao, Y., & Wu, X.-B. 2021, MNRAS, 503, 5263
Zuo, X., Tao, Y., Liu, Y., et al. 2024, Res. Astron. Astrophys., 24, 085016

Appendix A: Data analysis and labels definition

As described in Section 1, recently we extended the search for aperiodic variability of *XMM-Newton/EPIC* sources that is part of the EXTraS project (De Luca et al. 2021). While a complete description of the work will be presented in a future paper (Marelli et al. in prep.) and the results will be publicly available through a dedicated site (https://www.iasf-milano.inaf.it/extras/), we report here the main implementations that concern this work.

The new EXTraS-Aperiodic project extends the original data set of EXTraS to all the public *XMM-Newton* observation from its launch to the end of 2020.

Apart from minor changes, e.g. script optimization and bug fixes, the exposure light curves (LC) extraction is basically the same, taking as inputs the raw ODF *XMM-Newton* data and the sources and their parameters from the XMM serendipitous source catalog[8].

We merge the (uniform bin) LC in count rate from different cameras (pn and the two MOSs) into a single LC in flux. For each time bin, the count rate was converted to 0.2–12 keV flux using information from the XMM-Newton source catalog. The contemporaneous bins from the three cameras were then merged by computing a weighted mean, with the associated uncertainties propagated accordingly. This allows us to take into account the different characteristics of each camera to produce a consistent LC for the entire observation, even in the periods in which one or more cameras are off.

From the merged LC, we extracted 130 parameters and flags to characterize their variability. These can be divided into two main groups: the model-independent statistical features and the model-dependent features. The former include standard statistical features like the weighted average, standard deviation, skewness, and kurtosis; most of them are listed in table A. The latter include the goodness and best fit parameters for a number of models; in this case also most of them are listed in table A. We fitted constant, linear, quadratic and exponential models. We also implemented some more complex models in order to search in the LC for features of astrophysical interest, such as flares, dips (single and periodic) and eclipses (single and periodic). Simple visualizations of the features of these models are shown in Figure A.1. In addition to the features from the models themselves, we also compared the results of different models using f-tests (Bevington 1969), in particular comparing more complex models to a constant fit. We run a comparison of the likelihood for nested models, using the f-test. The asymptotic distribution under the null hypothesis that the nested model is correct was then converted in sigma units. This way, the statistical gain across all sources has a compact distribution, more robust towards outliers.

Based on the results of many papers in the literature (see e.g. Richards et al. 2011; Lo et al. 2014; Farrell et al. 2015), the cumulative distribution function (CDF) offers several features that are useful for describing the associated light curve (LC) variability. Thus, from each merged LC, we extracted the corresponding CDF. Unlike what we did in the original EXTraS, we extracted a continuous, smoothed CDF. Each bin is treated as a normalized (by its integral) Gaussian on the flux axis of the CDF, with the Gaussian μ at the bin value and the (2x)bin error as its σ; the Gaussians are then merged to produce a continuous CDF. Once smoothed, we normalize the flux axis so that the flux at 1% of the cumulative time is 0 and the flux at 99% of the cumulative time is 1; this way, the shape of the CDF (and thus the extracted features) is both time and flux independent. We show an example in Figure A.2.

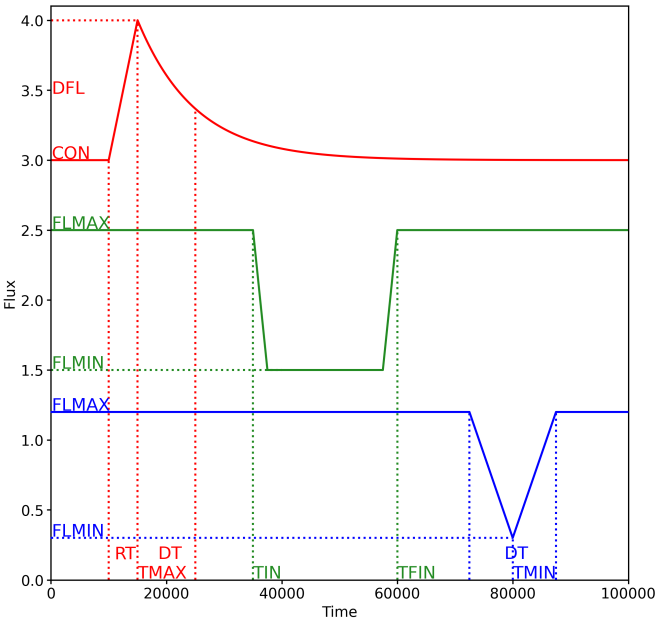

Fig. A.1: Example of three of the models we fit, with the associated features. We show in red the flare model, in green the single eclipse model and in blue the single dip model.

From each smoothed CDF, we extracted 53 features to describe its shape (plus 3 for the normalization), each one being the normalized flux at (or flux range between) certain cumulative time(s) or the cumulative time at certain normalized flux. Our features are chosen in order to reproduce the ones reported in Richards et al. (2011); Lo et al. (2014); Farrell et al. (2015) through a simple linear combination. Most of these features are reported in Table A.

[8] http://xmm-catalog.irap.omp.eu/

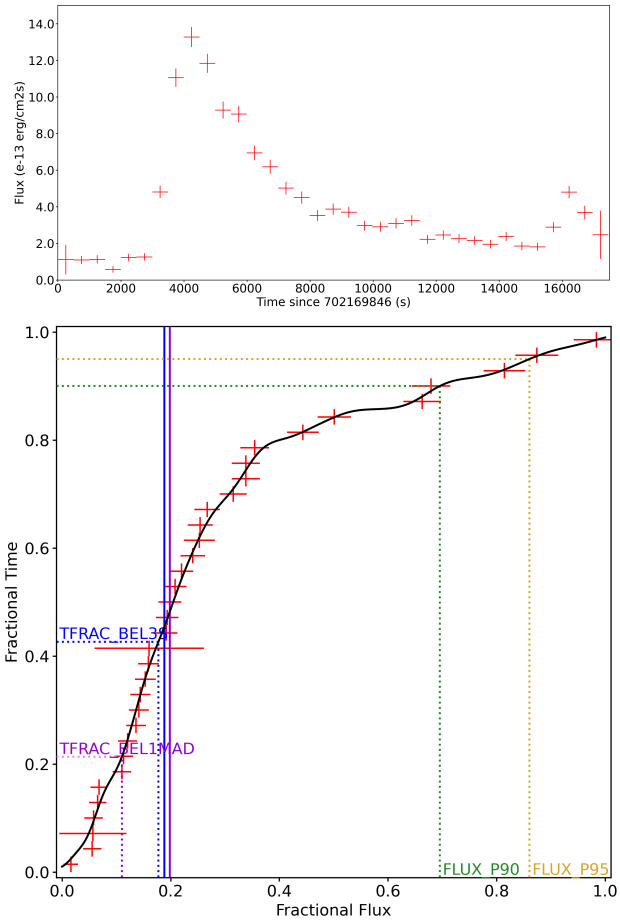

Fig. A.2: Example of a flaring LC (*top*) with the associated CDF (*bottom*). The original discrete CDF is shown in red, the continuous CDF in black; in blue and violet we show, respectively, the weighted average and the median of the LC. We also show some of the CDF features we use: TFRAC_BEL1MAD, TFRAC_BEL3S, FLUX_90 and FLUX_95 (respectively, in violet, blue, green, and orange).

Table A.1: Complete list of the features we used

-	Name	Description
1	N_BINS	Number of time bins
2	STDEV	Standard deviation
3	SKEW	Skewness
4	KURT	Kurtosis
5	AMPLIT	Amplitude $(max(LC) - min(LC)/2$
6	MEDABSDEV	MAD, Median deviation from median $median(\|LC - median(LC)\|)$
7	MEDMAXOFF	Max deviation from median on median $max(\|LC - median(LC)\|)/\|median(LC)\|$
8	CON_NSIGMA	Inverse survival function (constant model)
9	LIN_NSIGMA	Inverse survival function (linear model)
10	LIN_CON	Constant term (linear model)
11	LIN_CON_E^a	1σ error in LIN_CON
12	LIN_LIN	Linear term (linear model)
13	LIN_LIN_E^a	1σ error on LIN_LIN
14	QU_NSIGMA	Inverse survival function (quadratic model)
15	QU_CON	Constant term (quadratic model)
16	QU_CON_E^a	1σ error on QU_CON
17	QU_LIN	Linear term (quadratic model)
18	QU_LIN_E^a	1σ error on QU_LIN
19	QU_QU	Quadratic term (quadratic model)
20	QU_QU_E^a	1σ error on QU_QU
21	DIP1_NSIGMA	Inverse survival function (s.dip model)
22	DIP1_TMIN	Time of the minimum (s.dip model)
23	DIP1_TMIN_E^a	1σ error on DIP1_TMIN
24	DIP1_DT	Time duration of the dip (s.dip model)
25	DIP1_DT_E^a	1σ error on DIP1_DT
26	DIP1_FLMIN	Flux ad the minimum (s.dip model)
27	DIP1_FLMIN_E^a	1σ error on DIP1_FLMIN
28	DIP1_FLMAX	Constant term (s.dip model)
29	DIP1_FLMAX_E^a	1σ error on DIP1_FLMAX
30	DIP_NSIGMA	Inverse survival function (m.dip model)
31	DIP_TMIN	Phase of the minimum (m.dip model)
32	DIP_TMIN_E^a	1σ error on DIP_TMIN
33	DIP_DT	Phase duration of the dip (m.dip model)
34	DIP_DT_E^a	1σ error on DIP_DT
35	DIP_FLMIN	Flux ad the minimum (m.dip model)
36	DIP_FLMIN_E^a	1σ error on DIP_FLMIN
37	DIP_FLMAX	Constant term (m.dip model)
38	DIP_FLMAX_E^a	1σ error on DIP_FLMAX
39	DIP_PER	Time period (m.dip model)
40	DIP_PER_E^a	1σ error on DIP_PER
41	EC1_NSIGMA	Inverse survival function (s.ecl. model)
42	EC1_TIN	Begin time of eclipse (s.ecl. model)
43	EC1_TIN_E^a	1σ error on EC1_TIN
44	EC1_TFIN	End time of eclipse (s.ecl. model)
45	EC1_TFIN_E^a	1σ error on EC1_TFIN
46	EC1_FLMIN	Flux in the eclipse (s.ecl. model)
47	EC1_FLMIN_E^a	1σ error on EC1_FLMIN
48	EC1_FLMAX	Constant term (s.ecl. model)
49	EC1_FLMAX_E^a	1σ error on EC1_FLMAX
50	EC_NSIGMA	Inverse survival function (m.ecl. model)
51	EC_T1	Begin phase of eclipse (m.ecl. model)
52	EC_T1_E^a	1σ error on EC_T1
53	EC_T2	End phase of eclipse (m.ecl. model)
54	EC_T2_E^a	1σ error on EC_T2
55	EC_FL1	Flux level 1 (m.ecl. model)
56	EC_FL1_E^a	1σ error on EC_FL1
57	EC_FL2	Flux level 2 (m.ecl.model)
58	EC_FL2_E^a	1σ error on EC_FL2
59	EC_PER	Time period (m.ecl. model)
60	EC_PER_E^a	1σ error on EC_PER
61	FL_NSIGMA	Inverse survival function (flare model)
62	FL_CON	Constant term (flare model)
63	FL_CON_E^a	1σ error on FL_CON
64	FL_DFL	ΔFlux at maximum (flare model)
65	FL_DFL_E^a	1σ error on FL_DFL
66	FL_TMAX	Time of maximum (flare model)
67	FL_TMAX_E^a	1σ error on FL_TMAX
68	FL_DT	Decay time (flare model)
69	FL_DT_E^a	1σ error on FL_DT
70	FL_RT	Rising time (flare model)
71	FL_RT_E^a	1σ error on FL_RT
72	F_NSIGMA_LINCON	f-test prob. in σs (linear-const.)
73	F_NSIGMA_QUCON	f-test prob. in σs (quadratic-const.)
74	F_NSIGMA_QULIN	f-test prob. in σs (quadratic-linear)
75	F_NSIGMA_DIP1CON	f-test prob. in σs (s.dip-const.)
76	F_NSIGMA_DIPCON	f-test prob. in σs (m.dip-const.)
77	F_NSIGMA_DIP1DIP	f-test prob. in σs (m.dip-s.dip)
78	F_NSIGMA_EC1CON	f-test prob. in σs (s.ecl.-const.)
79	F_NSIGMA_ECCON	f-test prob. in σs (m.ecl.-const.)
80	F_NSIGMA_EC1EC	f-test prob. in σs (m.ecl.-s.ecl.)

81	F_NSIGMA_FLCON	f-test prob. in σs (flare-const.)
82	TFRAC_MID20	%time between 0.9-1.1 of MEDIAN
83	TFRAC_BEL1S	%time below AVE-AVE_ERR[b]
84	TFRAC_ABO1S	%time above AVE+AVE_ERR[b]
85	TFRAC_BEL3S	%time below AVE-3AVE_ERR[b]
86	TFRAC_ABO3S	%time above AVE+3AVE_ERR[b]
87	TFRAC_BEL5S	%time below AVE-5AVE_ERR[b]
88	TFRAC_ABO5S	%time frac. above AVE+5AVE_ERR[b]
89	TFRAC_BEL1MAD	%time below MEDIAN-MAD[c]
90	TFRAC_ABO1MAD	%time above MEDIAN+MAD[c]
91	TFRAC_BEL3MAD	%time below MEDIAN-3MAD[c]
92	TFRAC_ABO3MAD	%time above MEDIAN+3MAD[c]
93	TFRAC_BEL5MAD	%time below MEDIAN-5MAD[c]
94	TFRAC_ABO5MAD	%time above MEDIAN+5MAD[c]
95	FLUX_P05	%flux at 5% of cumulative time
96	FLUX_P10	%flux below 10% of cumulative time
97	FLUX_P17.5	%flux below 17.5% of cumulative time
98	FLUX_P25	%flux below 25% of cumulative time
99	FLUX_P32.5	%flux below 32.5% of cumulative time
100	FLUX_P40	%flux below 40% of cumulative time
101	FLUX_P50	%flux below 50% of cumulative time
102	FLUX_P60	%flux below 60% of cumulative time
103	FLUX_P67.5	%flux below 76.5% of cumulative time
104	FLUX_P75	%flux below 75% of cumulative time
105	FLUX_P82.5	%flux below 82.5% of cumulative time
106	FLUX_P90	%flux below 90% of cumulative time
107	FLUX_P95	%flux below 95% of cumulative time
108	MEDIAN_01	Normalized MEDIAN

Notes. Complete list of the features we used with a brief explanation for each one and its formula, if necessary. We divided the list in three sublists separated by horizontal lines to show the model-independent statistical features, model-dependent features and CDF features, respectively.
[a] _ERR has been written as _E to fit the table.
[b] Weighted average and its 1σ error.
[c] MEDIAN and MEDABSDEV.

Appendix B: UMAP plots for the most important features

Here, we report (Fig. B.1) the detailed UMAP views for each one of the six most important features. For interpretation, see Sect. 4.2.

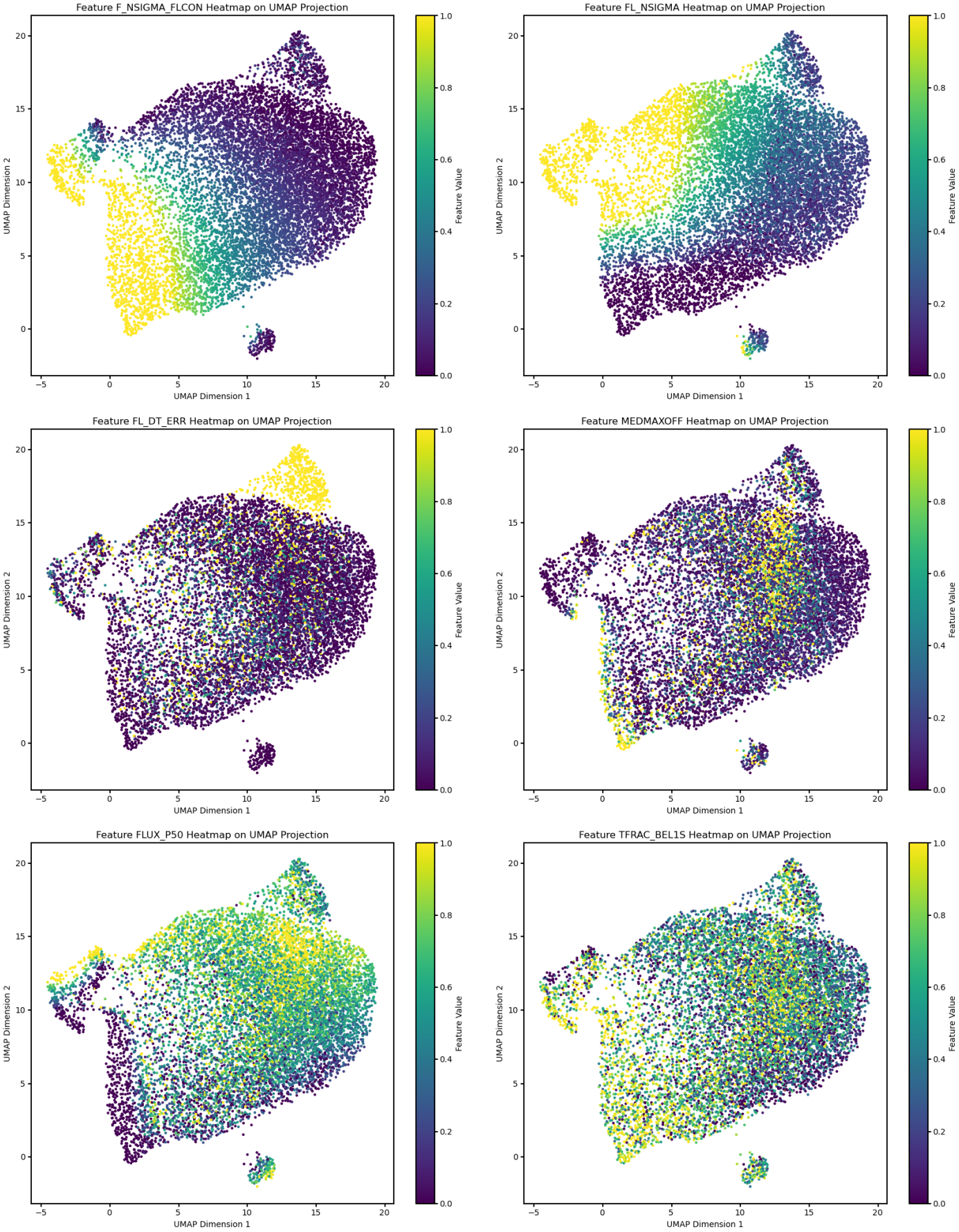

Fig. B.1: Feature by feature visualization on the UMAP plane.