A&A 652, A151 (2021)
https://doi.org/10.1051/0004-6361/202141247

Astronomy & Astrophysics

Drifting features: Detection and evaluation in the context of automatic RR Lyrae identification in the VVV

J. B. Cabral[1,2], M. Lares[2], S. Gurovich[2], D. Minniti[3,4,5], and P. M. Granitto[1]

[1] Centro Internacional Franco Argentino de Ciencias de la Información y de Sistemas (CIFASIS, CONICET–UNR), Rosario, Argentina
e-mail: cabral@cifasis-conicet.gov.ar
[2] Instituto De Astronomía Teórica y Experimental – Observatorio Astronómico Córdoba (IATE–OAC–UNC–CONICET), Córdoba, Argentina
[3] Departamento de Física, Facultad de Ciencias Exactas, Universidad Andrés Bello, Av. Fernandez Concha 700, Las Condes, Santiago, Chile
[4] Instituto Milenio de Astrofísica, Santiago 7500912, Chile
[5] Vatican Observatory, 00120 Vatican City State, Italy

Received 4 May 2021 / Accepted 25 May 2021

ABSTRACT

Context. As most of the modern astronomical sky surveys produce data faster than humans can analyse it, machine learning (ML) has become a central tool in astronomy. Modern ML methods can be characterised as highly resistant to some experimental errors. However, small changes in the data over long angular distances or long periods of time, which cannot be easily detected by statistical methods, can be detrimental to these methods.
Aims. We develop a new strategy to cope with this problem, using ML methods in an innovative way to identify these potentially detrimental features.
Methods. We introduce and discuss the notion of drifting features, related with small changes in the properties as measured in the data features. We use the identification techniques of RR Lyrae variable objects (RRLs) in the VVV based on an earlier work and introduce a method for detecting drifting features. For the VVV, each sky observation zone is called a tile. Our method forces the classifier to learn from the sources (mostly stellar 'point sources') which tile the source originated from and to select the features that are most relevant to the task of finding candidate drifting features.
Results. We show that this method can efficiently identify a reduced set of features that contains useful information about the tile of origin of the sources. For our particular example of detecting RRLs in the VVV, we find that drifting features are mostly related to colour indices. On the other hand, we show that even if we have a clear set of drifting features in our problem, they are mostly insensitive to the identification of RRLs.
Conclusions. Drifting features can be efficiently identified using ML methods. However, in our example removing drifting features does not improve the identification of RRLs.

Key words. methods: data analysis – methods: statistical – surveys – catalogs – stars: variables: RR Lyrae – Galaxy: bulge

1. Introduction

Most of the modern astronomical sky surveys are characterised by fast-paced data ingestion, data intensive science cases, or automatic reduction pipelines (e.g., Feigelson & Babu 2012), which often lie on the verge of technological developments and analysis capabilities. This unprecedented availability of observations challenges the traditional approaches for data analysis, leading to a shift in the paradigm for knowledge discovery (Bell et al. 2009), which has notably become dominated by machine learning (ML) techniques (Ball & Brunner 2010). Despite the difficult mathematical and statistical foundations, a complex terminology driven by the confluence of several sciences, and the arduous interpretation of the results, the training of intelligent agents has become an everyday practice in astronomy. The accessibility of easy-to-use free software resources mostly written for *R* (Team R Core 2000) or Python (Van Rossum & Drake 2003) languages was fundamental to this step.

In most cases, ML methods can be separated into two basic steps. First, raw data are converted into a set of useful features that are relevant to the task at hand (e.g., periods or intensities), and then these features are fed to a classifier or a statistical method (see e.g., Mitchell 1997).

Machine learning methods have a number of limitations. For instance, they are highly susceptible to errors produced by the limitations of the datasets (Cai & Zhu 2015). The results can also be hampered by the role of the features, which is not fully understood (Duboue 2020) or by the biases introduced by improperly defined experiments (Domingos 2012). These facts are well known and have not been ignored in the astronomical community (Luo et al. 2020).

Here, we are interested in the role of some sources of noise that are present in commonly used features in astronomical research as well as their impact on the results of ML methods in this context. We use data from the synoptic survey Vista Variables in Via Láctea (VVV; Minniti et al. 2010), observed with the VISTA telescope (Sutherland et al. 2015), which pursues, among its main objectives, the production of a three-dimensional map of a large part of the Galactic centre (bulge) of the Milky Way and a fraction of the internal Galactic disk. The VVV

data are presented in units called 'tiles', which are rectangular areas of the sky surveyed over time. For each tile, the VVV data reduction pipeline (Emerson et al. 2004) provides a preprocessed image and a database of files with the positions, magnitudes, and colour indices of the light sources present in the image, which comprises the 'photometric catalogue'. These catalogues are the main subject of this study.

The images are subject to two noise sources, namely experimental errors and observation conditions. The derived catalogues are also affected since the noise permeates all the survey information, which is composed of a set of features or observables. Atmospheric conditions, moon phases, maintenance of the camera and telescope, or modifications to the software, among many factors, can influence the observation or recording of the data. As a consequence, the derived measurements that are used as features for an ML analysis can also be prone, to different extents, to these errors and conditions.

Random measurement errors are present in all experimental or observational science. They are unavoidable, but each error typically affects only a single observation or a reduced set of observations; in particular, wide-field survey images can be affected by issues such as weather, astronomical conditions, and software updates. Machine learning methods can efficiently cope with these kinds of errors. For a large survey such as the VVV, observational conditions can change slightly (but not randomly) over long periods of time or for different regions in the sky. This problem is more difficult for ML methods. In many situations we want to train an intelligent agent using a well-known portion of the survey and then use it to predict other less-known zones when searching for a given astronomical phenomenon. Given the ML methodology, the agent will work efficiently on training data but will probably fail to generalise to other zones due to this slight change in observational conditions. Due to the diverse nature of the features extracted from the data (intensity, periods, colours, etc.), they will possibly reflect this effect in different proportions. It is thus interesting to ask whether it is possible to automatically detect which of the extracted features are more sensitive to these changes in observational conditions. Hereafter, we refer to the features in a dataset that are sensitive to observational conditions as 'drifting features'. We aim at evaluating their influence over a large-scale ML experiment.

As a working example, we focus on the problem of detecting RR Lyrae variable objects (RRLs) in VVV data. That is, we train classifiers using data from some VVV tiles and evaluate how they conduct the task of identifying RRLs on other tiles.

Drifting features should be consistent within a limited zone of the sky (for instance, one tile or two consecutive tiles) but should show slight changes, almost undetectable by most simple statistics, between tiles that are separated from one another[1]. Those changes could potentially alter the capabilities of the classifier. To detect these features and their effect on automatic classification, we once again propose using ML methods. If we confront an ML method with the task of discriminating data from two tiles, it will be forced to learn the differences between the tiles that are present in the features. We can then use feature selection methods (Guyon et al. 2002) to evaluate the importance of each feature for this classifier that discriminates tiles and to mark highly relevant features as candidate drifting features. In other words, we propose learning a separate task (the tile of origin of a source), not because it is unknown or difficult but as a method for detecting which are the features that are most useful to this task: the features that contain information that changes with the tile of origin.

This work is divided into the following sections: In Sect. 2 we explain our experimental setup (data, feature extraction, model selection, etc.). In Sect. 3 we introduce our procedure for the identification of drifting features, and in Sect. 4 we evaluate the effect of these features on the task of RRL identification. Finally, in Sect. 5 we discuss our results and draw our conclusions.

2. Experimental setup

2.1. Data

One of the main objectives of the VVV is the creation of a three-dimensional map of the bulge and the Galactic centre (Minniti et al. 2010) for which the search for variable stars in general, and RRLs in particular, is important due to their use as standard candles (Bailey 1902). To this end, the survey relies on data from the VISTA Infrared Camera (VIRCAM), mounted on Visible and Infrared Survey Telescope for Astronomy (VISTA) of the European Southern Observatory (ESO; Sutherland et al. 2015), which at the time of its construction was the largest near-infrared camera, with 16 non-contiguous $2k \times 2k$ detectors. To complete a contiguous tile, VIRCAM simultaneously exposes its detectors six times with a suitable offset. Each of the exposures is called a 'pawprint', and the combination of the six overlapping pawprints is a tile. For this reason each pixel is observed in at least two pawprints and the edges are shared with the observations of the adjacent tiles. The survey observation plan was organised in two stages: During the first year the tile was observed in five astronomical filters, Z, Y, J, H, and Ks, separated by a few hours; then, in subsequent years, it was re-observed using the Ks band for variability studies. Only some tiles were observed in multi-band after the first year. The dataset used in this work is the one presented in Cabral et al. (2020), which consists of 62 features extracted with `feets` (Cabral et al. 2018) from light curves that were reconstructed from the photometric catalogues provided by the Cambridge Astronomical Survey Unit (CASU).

From the original dataset we selected eight tiles located at different zones of the bulge, as shown in Fig. 1. For each tile we extracted all the RRLs plus a uniform sample of 2000 unclassified, unsaturated, and non-faint sources (average magnitude between 12 and 16.5). From these selections, sources with invalid values were removed, leaving the final dataset for this work, described in Table 1.

We chose to use a reduced dataset with around 2000 sources for each tile in order to dramatically decrease the computational burden of our experiments. As shown in Cabral et al. (2020), the use of a reduced dataset can lead to optimistic estimations of the accuracy of the detections, but our main objective is to find and characterise the features that best represent the differences between the tiles and not the accuracy of the detection of the RRLs.

2.2. Error measures

We faced two different binary classification problems in this work. First, we tried to separate sources between two tiles; this

[1] Whether two consecutive tiles or, in general, two regions on the sky are similar depends on the survey strategy. For example, regions that are repeatedly observed in quick succession in a survey or that share important observational parameters, such as the exposure time, could show this property even if they have long angular distances. The opposite is also valid: Regions that are close at angular distances can be observed under very diverse conditions.

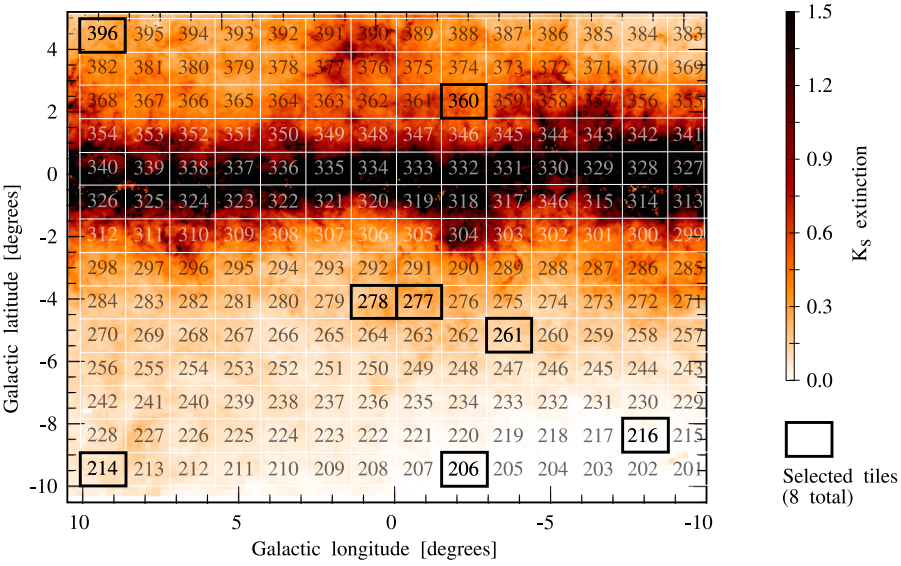

Fig. 1. Map of the bulge tiles of the VVV survey over an extinction map (extinction map adapted from Gonzalez et al. 2012). We highlight the tiles used in this work with red borders.

Table 1. Total number of sources, RRL and sample, taken in each tile used in this work.

Tile	Total	RRL	Sample
b206	407720	47	2047
b214	376822	35	2034
b216	334773	43	2043
b261	735838	253	2252
b277	831323	430	2429
b278	857887	437	2436
b360	1029149	679	2669
b396	729671	15	2015

was done in order to construct a tile classifier (TC) that allows the relevance of the features to be assessed. As stated in the Introduction, we used the TC as an auxiliary method that allows us to detect which features are candidate drifting features. Then, we built a source classifier (SC) that seeks to discriminate RRL sources from unknown sources. In the first problem both classes are nearly balanced in all cases. On the other hand, as discussed in Cabral et al. (2020), the identification of a few variable stars within a large set of unknown sources is usually a highly imbalanced problem that generates several inconveniences, such as those discussed in the recent work by Hosenie et al. (2020), and requires specific error measures.

In the RRL detection problem (SC), we define RRL samples as the positive class and the other sources as the negative class. In the tile identification problem (TC), both classes (the two tiles) are equivalent, so we arbitrarily call one of them positive and the other negative. All positive samples (in this case, either a source or a tile) that are correctly identified by the classifier are called true positives (TP); otherwise, if they are missed by the classifier they are called false negatives (FN). Negative samples that are wrongly classified are called false positives (FP), and those that are correctly identified are called true negatives (TN). Using a combination of these four outcomes, we can define two complementary performance measures, called 'precision' and 'recall', which are adequate to deal with unbalanced problems. The precision is defined as $TP/(TP+FP)$. It measures, for example, the fraction of real RRLs detected over all those retrieved by the classifier. The recall, on the other hand, is defined as $TP/(TP+FN)$. It measures, in the same example, the fraction of all RRLs that are detected by the classifier.

Many classifiers can change their decision outputs by adjusting the probability threshold that considers an observation to be positive or negative. A high threshold increases the precision and decreases the recall since fewer cases are classified as positive, while a low threshold generates the opposite effect. To evaluate precision and recall together, we consider the precision-recall curves (PR), where we plot a set of pairs of values corresponding to different thresholds. A curve that approaches the top-right corner is, in general, considered to represent a better classifier.

For balanced classification problems it is common to find more traditional metrics in the literature. As such, for the tile identification problem we also use 'accuracy', $(TP+TN)/(TP+FP+TN+FN)$, and the 'area under the receiver operating characteristic curve' (ROC-AUC) measures. The ROC curve is equivalent in concept to the precision-recall curve described above, and the area under it is a global measure of the performance of the classifier. The only difference between the two curves is that an ROC curve that approaches the top-left corner represents a better classifier.

2.3. Model selection

For the TC problem we evaluated four classifiers with diverse foundations – support vector machine (SVM) with a linear kernel (Vapnik 2013), SVM with a radial basis function (RBF) kernel, K-nearest neighbours (KNN; Mitchell 1997), and random forest (RF; Breiman 2001) – all implementations from the Scikit-Learn Python package (Pedregosa et al. 2011).

To determine the best hyper-parameters for every model, we executed a grid search of all possible combinations of values for each hyper-parameter over a fixed list. We used a five k-fold setup on a dataset with tiles b278 and b261, using precision as a performance measure. These tiles were chosen because they are not extreme in terms of their location or their balance, unlike b396 or b220.

With this setup, we selected the following hyper-parameter values:
SVM-linear: $C = 100$.
SVM-RBF: $C = 100$ and $\gamma = 0.003$.
KNN: $K = 56$ with a *Manhattan* metric; also, the importance of the neighbour class was not weighted by distance.

RF: We created 500 decision trees with information gain as a metric; the maximum number of random selected features for each tree is 0.5 the total number of features, and the minimum number of observations in each leaf is five.

Using the optimal values for the hyper-parameters, we compared the four models on the same dataset using a ten-fold cross-validation setup. Table 2 shows the corresponding results using the default threshold (0.5) for all models. For all three metrics considered (precision, recall, and AUC), the SVM-linear classifier clearly outperformed all the other classifiers. More importantly, Fig. 2 shows the corresponding ROC and precision-recall curves, which show that SVM-linear also outperforms the other methods for all possible thresholds. Given these results, we selected SVM-linear as the classifier for the tile identification problem. For the SC problem, Cabral et al. (2020) already determined that RF is the classifier with the best performance for our dataset and general experimental setup.

2.4. Feature selection

Feature selection (Guyon & Elisseeff 2003) is the process of extracting some subsets of features from the entire set in order to optimise the classification performance and/or the computational complexity of the problem. We chose for this work the 'recursive feature elimination' (RFE) algorithm (Guyon et al. 2002). The method is widely adopted and is characterised by its good performance and simplicity. As a backward selection method, RFE starts with all the features and sequentially eliminates the unimportant features using a recursive process.

The RFE algorithm is integrated with a classification method, which provides, at each step of the recursion, the importance score of the features. It iteratively executes the underlying classifier and extracts the score for each variable; then the variable (or group of variables) with a worse performance (according to the score) is eliminated.

The method typically ends when the desired (fixed) number of features to select is reached. Another possibility is to monitor a performance metric for the subsets (for example, the accuracy on an independent validation set) and stop the recursion when the metric is optimal.

In this work we relied on the RFE implementation with k-fold Cross-Validation (RFECV) for the stopping criteria, which are provided by the Scikit-Learn package (Pedregosa et al. 2011). The RFECV produces k replicated experiments ($k = 5$ in our work), each of which selects features over $(k - 1)$ folds and monitors the classification error (1 − accuracy) over the remaining fold. Then it determines the number of features to select, looking for the least average error throughout all the folds. In the last step, RFECV produces a final selection using the entire dataset to select the features, stopping at the previously selected point.

It is worth mentioning that the classifier embedded in the RFE in the feature selection stage may be different from the eventual method in the final classification stage.

3. Finding drifting features

As we stated in the Introduction, we propose using ML methods to detect drifting features, looking for features that are useful for determining the tile of origin of a given source (exclusively from features derived from the pawprint stack photometry, without any other header keyword data). With this goal, in this first experiment we considered all the sources in each tile together (RRLs and unknowns) and trained classifiers to learn the tile of origin of each source and not its astronomic type.

Table 2. Classification metrics of the SVM (with linear and RBF kernels), RF, and KNN models on the sources of tiles b278 and b261.

Model	Precision	Recall	AUC
SVM-Linear	0.8511	0.8511	0.9286
SVM-RBF	0.8003	0.8003	0.8680
RF	0.7707	0.7707	0.8548
KNN	0.6973	0.6973	0.7685

We applied the RFECV method, as described in the previous section, to 28 binary classification problems, each of which consisted in separating a different pair of tiles from the set of eight tiles in our dataset. Thus, for each SC problem (for example, separating tiles b206 and b214), we obtain from RFECV a subset of selected features for that problem. Each subset potentially has a different length, as discussed above.

Figure 3 shows the number of features selected for each problem. It is evident that there are two different behaviours. In some cases RFE selects just a few features, as for example tile b216 with any other except b206; on the other hand, in some other cases the selected subset contains a high number of features (tile b206 against b216, for example, or b396 against b277 or b278). However, the number of selected features by itself is not relevant; what is more important for identify drifting features is how well they separate the two tiles.

We can arbitrarily divide the problem into two categories: 'few features' (four or fewer selected features) and 'high number' (more than four selected features). Figure 4 shows ROC curves for the 12 high-number problems as well as their relative locations in space. The figure shows curves for three classifiers: one trained with all the features in the dataset ('all features'), a second trained using only those selected by RFE (i.e. our candidate drifting features), and a third one trained with those not selected by RFE (we call this the 'stable' subset). All the few-features subset cases (b216-b278 for example) produce trivial ROC curves that are saturated at the top-left corner for the three subsets, with AUC > 0.99, which we do not show.

Analysing the results, the first observation is that, as expected, the classifier trained with the drifting features (those chosen via feature selection) is always very similar in performance to the one trained with all the features. This result is a confirmation that RFECV does its work, selecting a subset of features that are responsible for the separation of the classes. The second result is that the performance of the models for the few-features problems is clearly superior to those shown in the figure (i.e. the high-number problems). This implies that the two behaviours shown in Fig. 3 correspond to problems that are easy to solve – where the separation is almost perfect and can be done with a few features – and problems that are harder – where the tiles cannot be fully separated and RFE selects bigger subsets.

It is interesting to note the different response of the classifiers trained on the stable subset on the 'easy' and 'hard' problems. In the hard problems, RFECV selects a high number of features, which means that there are features with no considerable information about the tile of origin of the source. After several features are selected by RFECV, the remaining features (the stable subset) contain much less information about the origin and produce a classifier with low performance. For the easy problems,

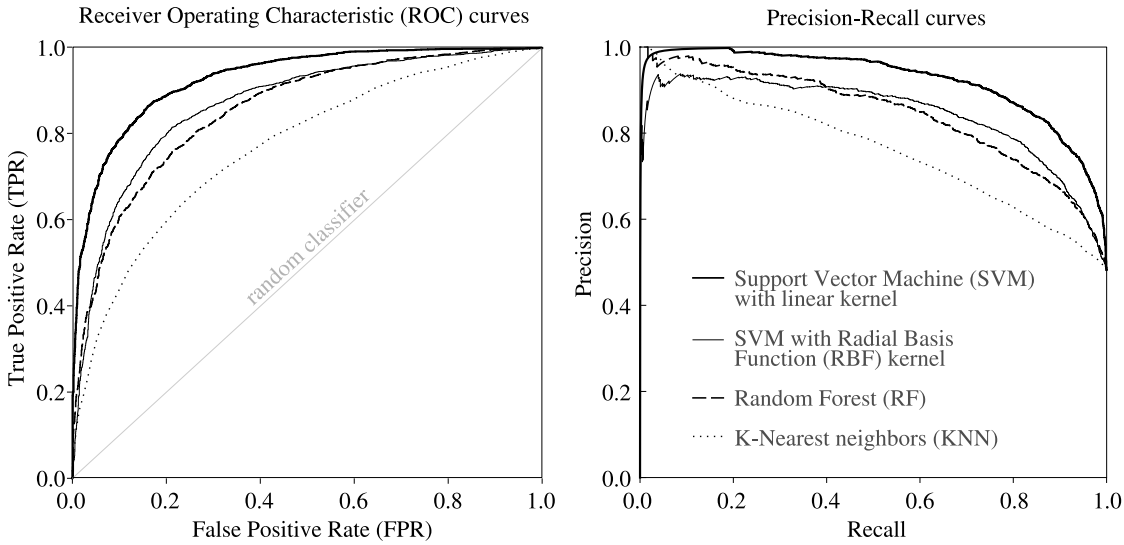

Fig. 2. ROC (*left*) and precision-recall (*right*) curves of the SVM (with linear and RBF kernels), RF, and KNN models for the prediction of the tile of a given source, using ten-fold cross-validation with tiles b278 and b261.

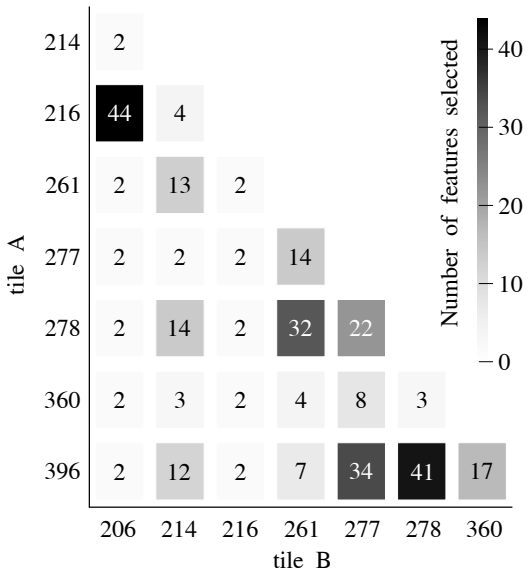

Fig. 3. Number of features selected by RFE for each binary TC problem. Each cell corresponds to the dataset that includes tile A (rows) and tile B (columns).

on the other hand, there are several features with a great deal of information about the tile of origin. Using just two to four features selected by RFECV is enough to produce an almost perfect classifier. The stable subset in this case contains plenty of features with good information about the origin, and it also produces a classifier with almost perfect performance. If we take the relative positions of each pair of tiles for the hard and easy problems into account, no definite pattern emerges. Most problems involving neighbour tiles, such as b277, b278, and b261, are hard, and most problems involving tiles in the bottom-right region are easy.

Another relevant analysis for a feature selection method is an analysis of which features are selected in each case. The upper half of Fig. 5 shows the number of times that each feature was selected by RFECV over the 28 TC problems, for all features that were selected at least two times (Table A.1 shows the list of features selected on each problem). The features related to pseudo-colour (Cabral et al. 2020) were the most frequently selected, appearing in at least half of the cases. Two such features (`c89_hk_color` and `c09_hk_color`) were selected in almost all cases. This information suggests that colour-related information in general is the most important characteristic for distinguishing tiles.

The two very different behaviours of hard and easy problems will complicate the evaluation of the real influence of drifting features on the TC problems because deleting only two features versus half of the features will lead to diverse scenarios. To allow for an easier and fairer comparison, we changed the feature selection method, using RFE with a fixed number of ten selected features.

The list of these selected features and their frequency is shown in the bottom half of Fig. 5. Only 15 features were selected in total over the 28 TC problems, of which the 11 most relevant are related to colour, probably with a high dependence on the location of the tile.

The overall performance of the classifiers trained with the full, drifting, and stable subsets for some exemplars of easy and hard datasets can be seen in Fig. 6. For the easy problems (bottom row), we used fewer features in the stable subset, leading to a lower performance. On the opposite side, for the hard problems (top row) we used more features in the stable subset, leading to a clear improvement in its performance. The rest of the TC problems show the same type of result (data not shown).

Using a fixed selection of features with RFE, we obtain, in all cases, a subset of ten features (the 'drifting' subset) that can discriminate the tile of origin with high accuracy as well as another subset (stable) with much less information about the tile of origin of the sources.

4. Evaluation of the influence of drifting features

In this section we evaluate the influence of the drifting subsets selected in the previous step on the SC problems (i.e. to discriminate between unknown sources and RRL variable stars).

For each pair of tiles we have three datasets, one with all the features ('full'), a second with only ten drifting features selected by RFE, and finally one with the remaining features, the stable subset. Unlike the previous problem, we now have, for the SC

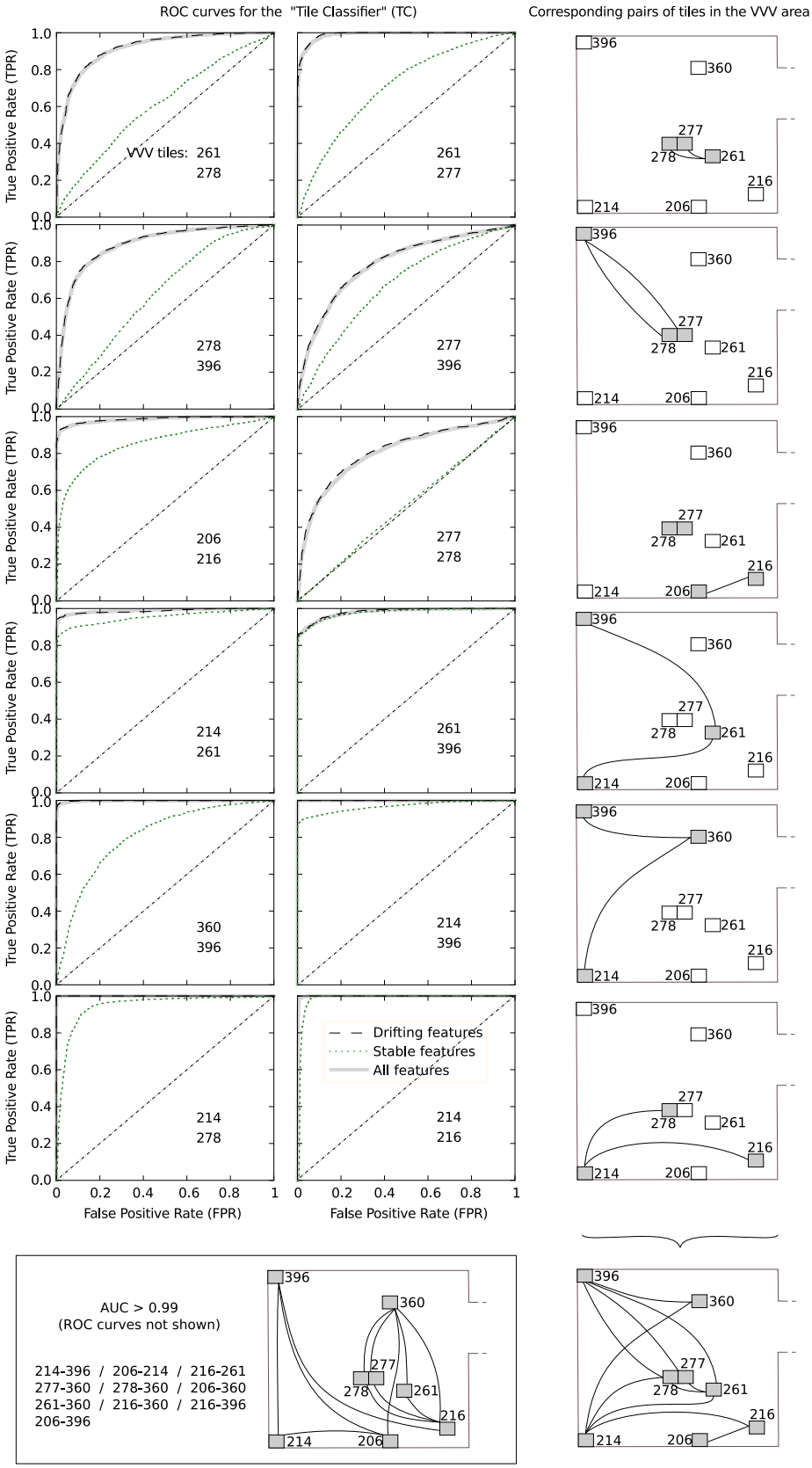

Fig. 4. ROC curves for the combinations of two tiles, selecting the more important features. Each panel has three curves: one using all the features (thick grey lines), the drifting features selected by RFE (dashed lines), and those considered stable and not selected by RFE (dotted lines).

problem, two possibilities for each pair of tiles: First, we can train our classifiers in one of the tiles and searched for RRLs in the other, and second we can invert the tiles, training classifiers in the second tile of the pair and looking for RRLs in the first tile. Thus, for each of the 56 SC problems we trained corresponding RF classifiers and obtained three PR curves for the full, drifting, and stable classifiers.

The complete results are presented in Appendix B, while a summary of some representative cases can be seen in Fig. 7. A first result is that, clearly, the drifting subset shows lower per-

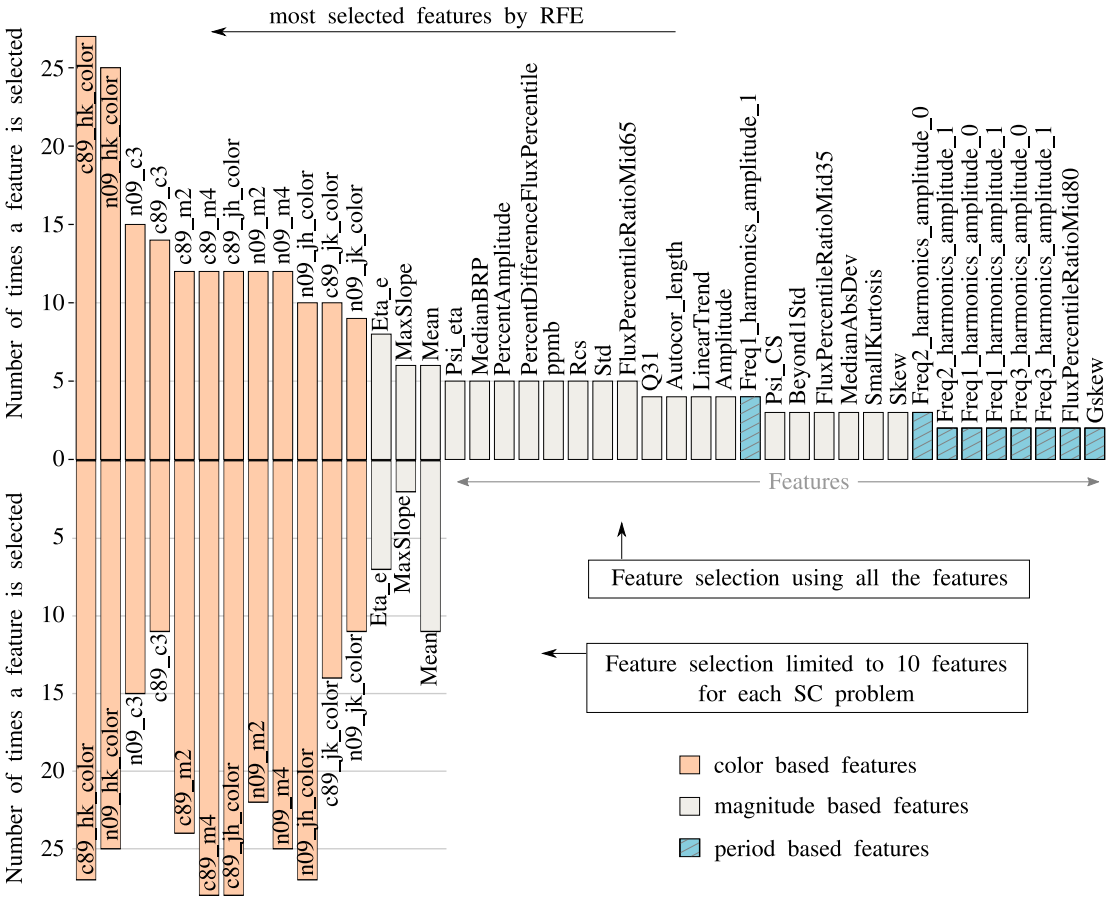

Fig. 5. Total times that each feature was selected by RFECV over the 28 SC problems considered in this work. The different colours identify which of the three groups each feature belongs to: orange for those based on colour, blue for those based on period, and white for those based only on magnitude.

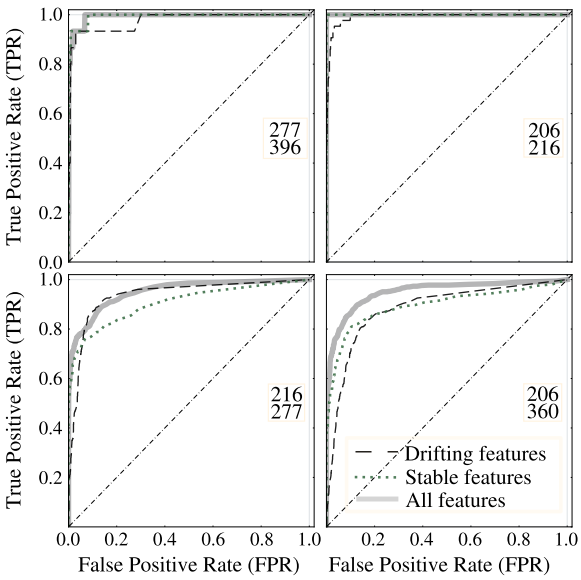

Fig. 6. Same as Fig. 4 but for a fixed RFE selection of ten features for the drifting subset. The *top row* shows two easy datasets, and the *bottom row* shows two hard datasets.

formance in all cases. This was expected as most drifting features are related to colour and Cabral et al. (2020) demonstrated that colour alone cannot clearly identify RRLs. More interesting, if we compare the performance of the full datasets with the stable datasets, we can see that there is no clear advantage in eliminating the drifting features from the datasets. Full and stable curves are very similar in all cases. Differences are small, and there is no clear pattern indicating when eliminating drifting features would improve the performance of the ML methods.

5. Discussion

In this work we have introduced and discussed the concept of drifting features, in relation to the small changes in the properties measured by those features, which can potentially distort the results of ML methods in astronomy. Using the identification of RRLs in VVV as a working example, we have introduced a method for detecting drifting features using an indirect ML method. We forced a classifier to learn the tile of origin of diverse sources and to select the features most relevant to this task as candidate drifting features. We have shown that this method can efficiently identify a reduced set of features that contains useful information about the tile of origin of the sources. We have also shown that, for our particular example of detecting RRLs in the VVV, drifting features are mostly related to colour. On the other hand, we showed in Sect. 4 that even if we have a clear set of drifting features in our problem, they are almost harmless for the identification of RRLs.

In future work we will explore the influence of drifting features on the detection of other types of variable sources and other

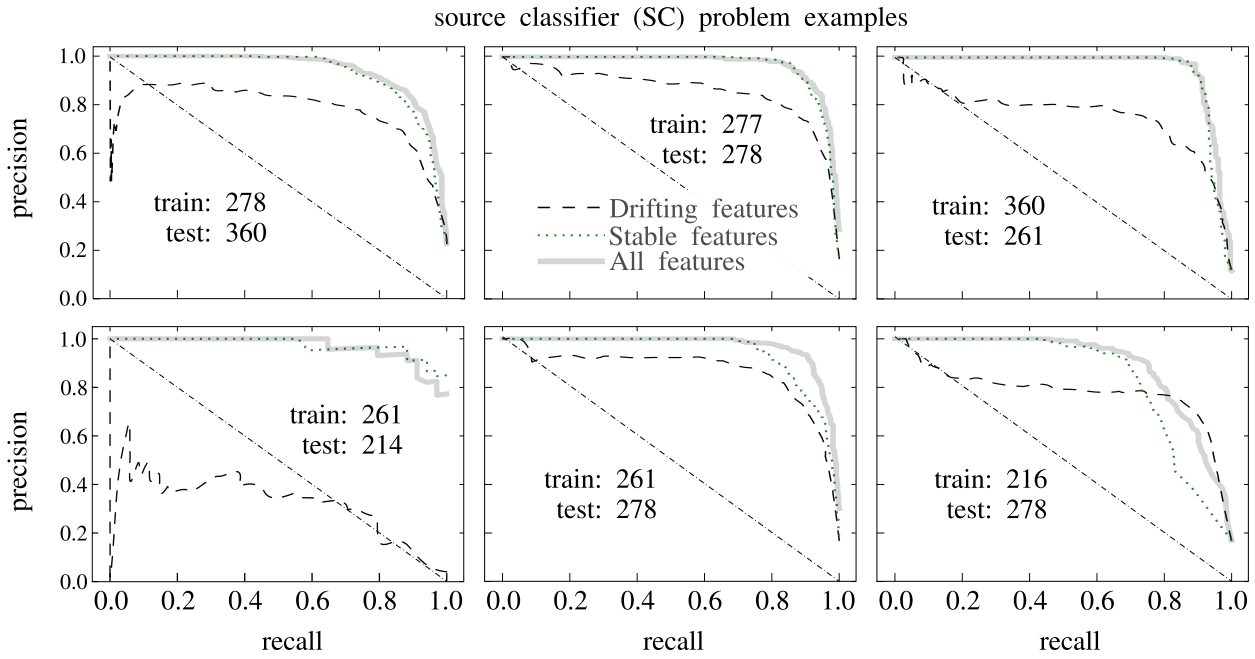

Fig. 7. Precision-recall curves for the SC problems. We show results for six combinations of train-test tiles using three classifiers trained with the full, drifting, and stable subsets of features.

large-scale ML experiments. We will also explore a different way of setting the number of selected features by RFE, considering all features that are relevant to the problem and not only the subset that shows the best performance for some metric or a fixed-length subset.

Acknowledgements. The authors would like to thank their families and friends, and also IATE astronomers for useful comments and suggestions. This work was partially supported by the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina) and the Secretaría de Ciencia y Tecnología de la Universidad Nacional de Córdoba (SeCyT-UNC, Argentina). J. B. C, are supported by a fellowship from CONICET. Some processing was achieved with Argentine VO (NOVA) infrastructure, for which the authors express their gratitude. We gratefully acknowledge data from the ESO Public Survey program ID 179.B-2002 taken with the VISTA telescope and products from the Cambridge Astronomical Survey Unit (CASU). J. B. C. thanks to Maren Hempel by creating the template for the creation of the template on which the Fig. 1 is based, and finally Bruno Sánchez and Martín Beroiz for the continuous support and friendship. This research has made use of the `http://adsabs.harvard.edu/`, Cornell University xxx.arxiv.org repository, adstex (`https://github.com/yymao/adstex`), astropy and the Python programming language.

References

Bailey, S. I. 1902, Ann. Harvard College Obs., 38, 1
Ball, N. M., & Brunner, R. J. 2010, Int. J. Mod. Phys. D, 19, 1049
Bell, G., Hey, T., & Szalay, A. 2009, Science, 323, 1297
Breiman, L. 2001, Mach. Learn., 45, 5
Cabral, J. B., Sánchez, B., Ramos, F., et al. 2018, Astron. Comput., 25, 213
Cabral, J. B., Ramos, F., Gurovich, S., & Granitto, P. 2020, A&A, 642, A58
Cai, L., & Zhu, Y. 2015, Data Sci. J., 14
Domingos, P. 2012, Commun. ACM, 55, 78
Duboue, P. 2020, The Art of Feature Engineering: Essentials for Machine Learning (Cambridge University Press)
Emerson, J. P., Irwin, M. J., Lewis, J., et al. 2004, in Proc. SPIE, eds. P. J. Quinn, A. Bridger, SPIE Conf. Ser., 5493, 401
Feigelson, E. D., & Babu, G. J. 2012, Significance, 9, 22
Gonzalez, O., Rejkuba, M., Zoccali, M., et al. 2012, A&A, 543, A13
Guyon, I., & Elisseeff, A. 2003, J. Mach. Learn. Res., 3, 1157
Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. 2002, Mach. Learn., 46, 389
Hosenie, Z., Lyon, R., Stappers, B., Mootoovaloo, A., & McBride, V. 2020, MNRAS, 493, 6050
Luo, S., Leung, A. P., Hui, C. Y., & Li, K. L. 2020, MNRAS, 492, 5377
Minniti, D., Lucas, P. W., Emerson, J. P., et al. 2010, New A, 15, 433
Mitchell, T. 1997, Machine Learning (McGraw-hill New York)
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, J. Mach. Learn. Res., 12, 2825
Sutherland, W., Emerson, J., Dalton, G., et al. 2015, A&A, 575, A25
Team R Core 2000, Vienna, Austria: R Foundation for Statistical Computing
Van Rossum, G., & Drake, F. L. 2003, Python Language Reference Manual (Network Theory United Kingdom)
Vapnik, V. 2013, The Nature of Statistical Learning Theory (Springer science & business media)

Appendix A: Finding drifting features

Table A.1. RFE results without a minimum feature limit in an attempt to identify the tile of a source.

Feature	b261–b277	b261–b278	b261–b360	b277 b278	–	b277–b360	b278–b360	b206–b216	b206 b214	–	b206–b261	b206–b277	b206–b278	b206–b360	b206–b396	b216–b214	b216–b261	b216–b277	b216 b278	–	b216–b360	b216–b396	b214–b261	b214–b277	b214–b278	b214–b360	b214–b396	b261–b396	b360 b396	–	b278–b396	b277–b396	Selected
c89_hk_color	✓	✓	✓	✓		✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	·	27
n09_hk_color	✓	✓	✓	✓		✓	·	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	·	✓	·	✓		✓	✓	25
n09_c3	✓	✓	✓	✓		✓	✓	✓	·		·	·	·	·	·	·	·	·	·		·	·	✓	·	✓	✓	✓	✓	✓		✓	✓	15
c89_c3	✓	✓	✓	✓		✓	✓	✓	·		·	·	·	·	·	·	·	·	·		·	·	✓	·	✓	✓	✓	·	✓		✓	✓	14
c89_m2	✓	✓	·	✓		✓	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	✓	·	✓	·	✓	✓	✓		✓	✓	12
c89_m4	✓	✓	·	✓		✓	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	✓	·	✓	·	✓	✓	✓		✓	✓	12
c89_jh_color	✓	✓	·	✓		·	·	✓	·		·	·	·	·	·	✓	·	·	·		·	·	✓	·	✓	·	✓	✓	✓		✓	✓	12
n09_m2	✓	✓	·	✓		✓	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	✓	·	✓	·	✓	✓	✓		✓	✓	12
n09_m4	✓	✓	·	✓		✓	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	✓	·	✓	·	✓	✓	✓		✓	✓	12
n09_jh_color	✓	✓	·	✓		·	·	✓	·		·	·	·	·	·	✓	·	·	·		·	·	✓	·	✓	·	✓	·	✓		✓	·	10
c89_jk_color	✓	✓	·	✓		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	✓	·	✓	·	✓	·	✓		✓	✓	10
n09_jk_color	✓	✓	·	✓		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	✓	·	✓	·	·	·	✓		✓	✓	9
Eta_e	·	✓	·	✓		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	✓	·	✓	·	·	·	✓		✓	✓	8
MaxSlope	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	✓	·	✓	·	✓		✓	✓	6
Mean	·	✓	·	✓		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	✓		✓	✓	6
Psi_eta	·	✓	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	✓		✓	✓	5
MedianBRP	·	✓	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	✓		✓	✓	5
PercentAmplitude	✓	✓	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	5
PercentDifferenceFluxPercentile	·	✓	·	✓		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	5
ppmb	·	✓	·	✓		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	5
Rcs	✓	✓	·	✓		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	5
Std	·	✓	·	✓		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	5
FluxPercentileRatioMid65	·	✓	·	✓		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	5
Q31	·	✓	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	4
Autocor_length	·	✓	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	4
LinearTrend	·	✓	·	✓		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	4
Amplitude	·	✓	·	✓		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	·	4
Freq1_harmonics_amplitude_1	·	✓	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	4
Psi_CS	·	✓	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	3
Beyond1Std	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	3
FluxPercentileRatioMid35	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	3
MedianAbsDev	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	3
SmallKurtosis	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	3
Skew	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	3
Freq2_harmonics_amplitude_0	·	✓	·	✓		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	3
Freq2_harmonics_amplitude_1	·	·	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	2
FluxPercentileRatioMid80	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	·	2
Freq1_harmonics_amplitude_2	·	✓	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	2
Freq1_harmonics_rel_phase_1	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	·	2
Gskew	·	·	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	✓	2
Freq3_harmonics_amplitude_1	·	✓	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	2
Freq3_harmonics_amplitude_0	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	✓	2
PeriodLS	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	1
Freq1_harmonics_rel_phase_2	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	1
FluxPercentileRatioMid50	·	✓	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	1
Freq3_harmonics_amplitude_2	·	·	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	·	1
Freq1_harmonics_amplitude_0	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	1
Period_fit	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	1
Freq2_harmonics_rel_phase_3	·	·	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	·	1
Freq3_harmonics_amplitude_3	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	1
Freq1_harmonics_rel_phase_3	·	·	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		✓	·	1
Freq3_harmonics_rel_phase_1	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	1
Freq2_harmonics_amplitude_3	·	·	·	·		·	·	✓	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	1
Freq2_harmonics_rel_phase_2	·	·	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	0
PairSlopeTrend	·	·	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	0
Freq2_harmonics_rel_phase_1	·	·	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	0
Freq2_harmonics_amplitude_2	·	·	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	0
Freq3_harmonics_rel_phase_3	·	·	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	0
FluxPercentileRatioMid20	·	·	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	0
Con	·	·	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	0
Freq3_harmonics_rel_phase_2	·	·	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	0
Freq1_harmonics_amplitude_3	·	·	·	·		·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	·	·	·	·	·	·	·		·	·	0

Notes. The rows contain the name of the features ordered from highest to lowest by the frequency with which they are selected, the columns are the datasets, and a check mark indicates that a dataset selected a given feature.

Table A.2. RFE result selecting the ten most important features in an attempt to identify a source tile.

Feature	b360–b396	b278–b396	b278–b360	b277–b396	b277–b360	b277–b278	b261–b396	b261–b360	b261–b278	b261–b277	b261–b396	b261–b360	b261–b278	b261–b277	b261–b261	b216–b396	b216–b360	b216–b278	b216–b277	b216–b261	b216–b214	b206–b396	b206–b360	b206–b278	b206–b277	b206–b261	b206–b214	b206–b216	Selected
c89_m4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	28
c89_jh_color	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	28
n09_jh_color	✓	✓	✓	·	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	27
c89_hk_color	✓	✓	✓	·	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	27
n09_m4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	·	✓	·	✓	✓	✓	✓	·	25
n09_hk_color	✓	✓	✓	·	✓	✓	·	✓	·	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	25
c89_m2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	·	✓	✓	✓	✓	✓	✓	·	✓	·	·	✓	✓	✓	24
n09_m2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	·	·	✓	✓	✓	✓	·	·	·	·	✓	✓	✓	22
n09_c3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	·	·	·	·	·	·	·	·	·	·	·	·	·	15
c89_jk_color	·	✓	·	·	·	·	✓	·	✓	·	·	·	·	·	·	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	·	✓	·	14
n09_jk_color	·	·	·	✓	·	·	✓	·	✓	·	·	·	·	·	·	✓	✓	·	✓	·	✓	✓	✓	✓	✓	·	·	·	11
c89_c3	✓	·	✓	·	✓	✓	·	✓	·	✓	✓	✓	✓	✓	✓	·	·	·	·	·	·	·	·	·	·	·	·	·	11
Mean	·	·	·	✓	·	·	·	·	·	·	·	·	·	·	·	✓	✓	✓	·	✓	·	✓	✓	✓	✓	✓	✓	·	11
Eta_e	·	·	·	·	·	·	·	·	·	·	·	·	·	·	✓	·	·	·	·	·	·	✓	✓	✓	✓	✓	·	✓	7
MaxSlope	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	✓	·	·	·	·	·	·	✓	2
PercentDifferenceFluxPercentile	·	·	·	✓	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	1
Std	·	·	·	✓	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	1
SmallKurtosis	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	✓	1
Skew	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Rcs	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Q31	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Psi_eta	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Psi_CS	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Period_fit	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
PeriodLS	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
PercentAmplitude	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
PairSlopeTrend	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
MedianBRP	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
MedianAbsDev	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
LinearTrend	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Amplitude	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Gskew	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Autocor_length	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq3_harmonics_rel_phase_3	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Beyond1Std	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Con	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
FluxPercentileRatioMid20	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
FluxPercentileRatioMid35	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
FluxPercentileRatioMid50	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
FluxPercentileRatioMid65	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
FluxPercentileRatioMid80	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq1_harmonics_amplitude_0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq1_harmonics_amplitude_1	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq1_harmonics_amplitude_2	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq1_harmonics_amplitude_3	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq1_harmonics_rel_phase_1	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq1_harmonics_rel_phase_2	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq1_harmonics_rel_phase_3	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq2_harmonics_amplitude_0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq2_harmonics_amplitude_1	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq2_harmonics_amplitude_2	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq2_harmonics_amplitude_3	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq2_harmonics_rel_phase_1	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq2_harmonics_rel_phase_2	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq2_harmonics_rel_phase_3	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq3_harmonics_amplitude_0	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq3_harmonics_amplitude_1	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq3_harmonics_amplitude_2	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq3_harmonics_amplitude_3	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq3_harmonics_rel_phase_1	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
Freq3_harmonics_rel_phase_2	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0
ppmb	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	0

Notes. The rows contain the name of the features ordered from highest to lowest by the frequency with which they are selected, the columns show the combination of dataset tiles, and the check marks indicate if a dataset selected a given feature.

Appendix B: Evaluation of the drifting features

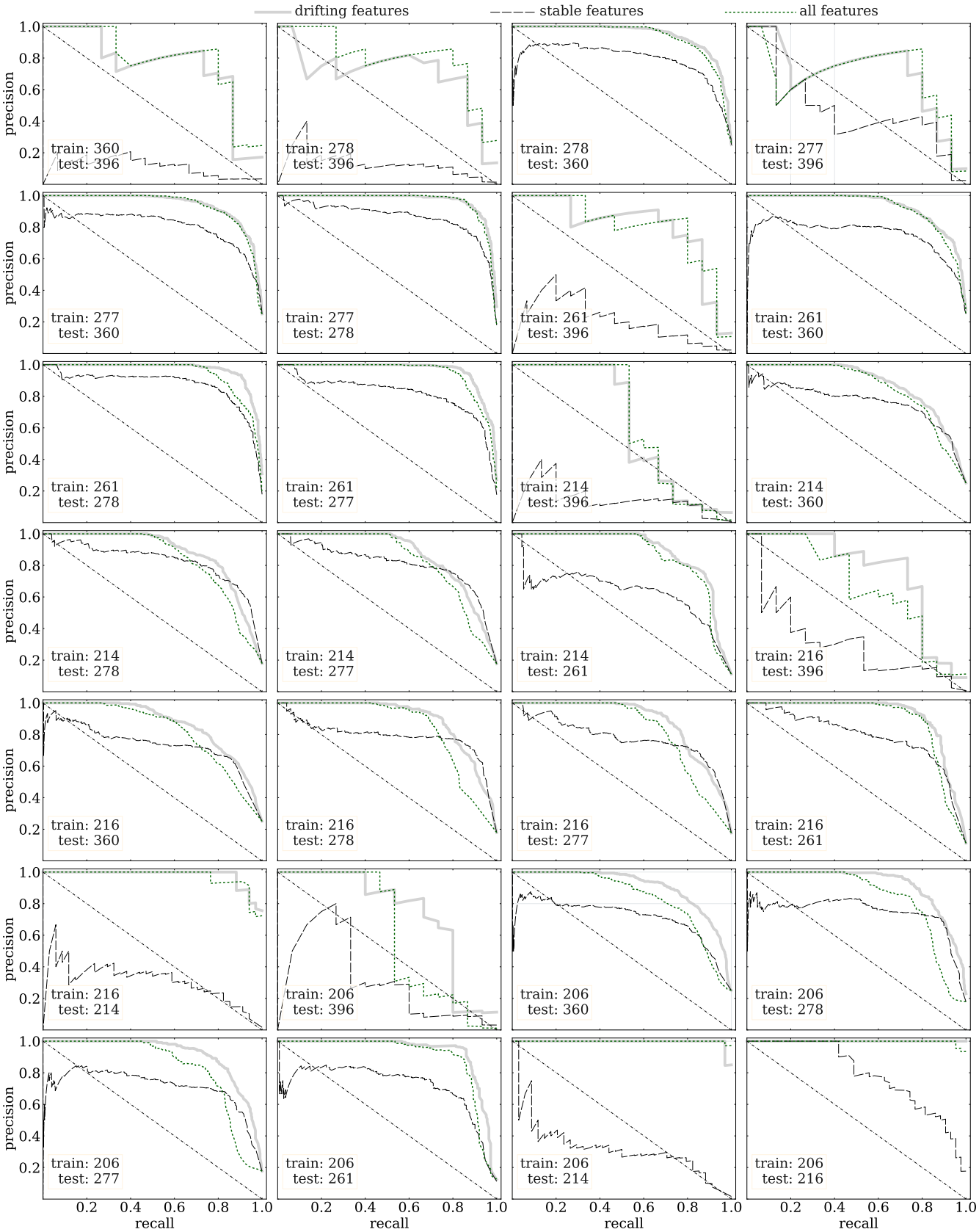

Fig. B.1. First 28 precision-recall curves for all combinations of train-test tiles for three different classifiers: one with all the features (full), another only with the ten drifting features found in that combination of train-test tiles (RFE), and one using all the remaining features with the exception of the drifting features (No-RFE).

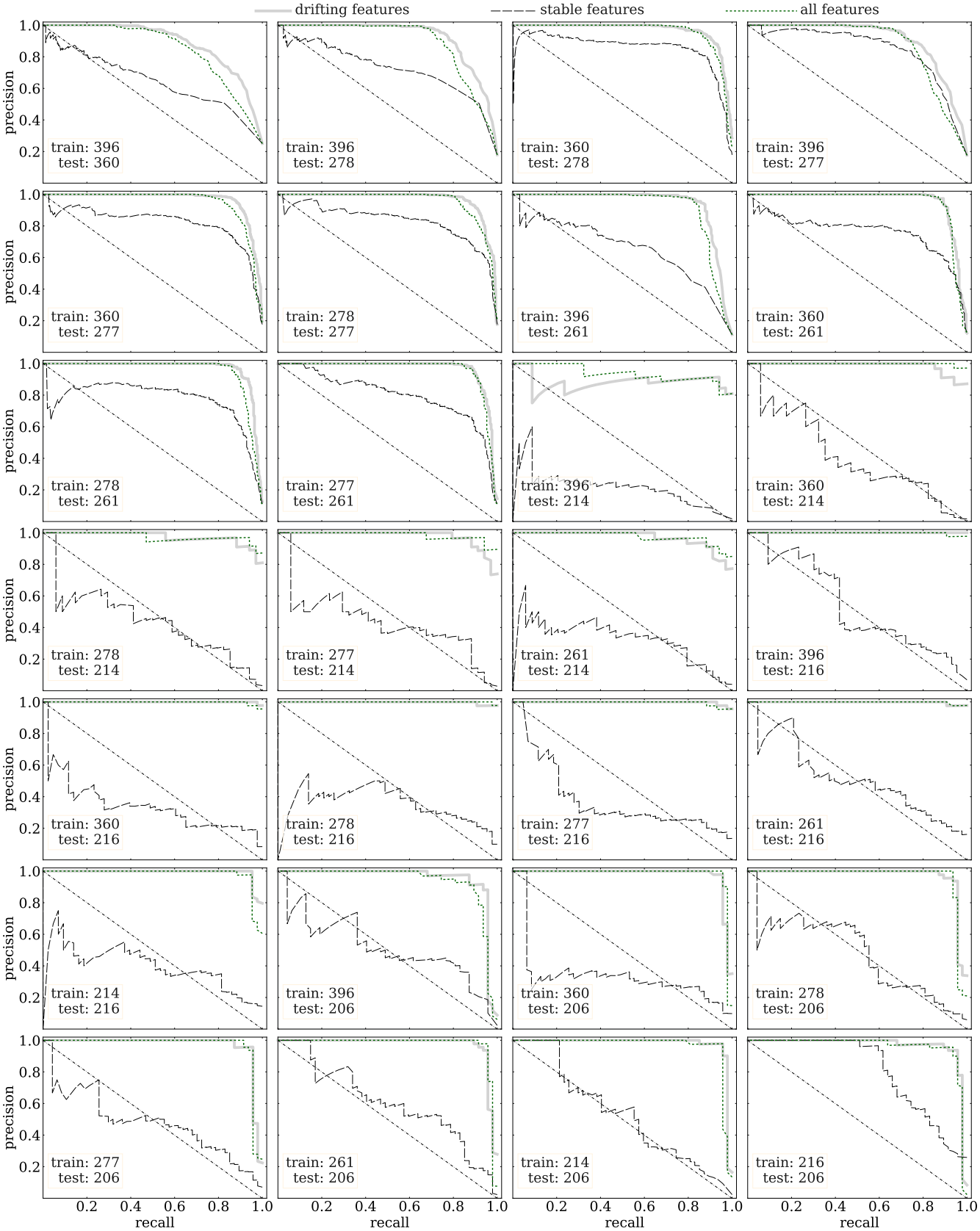

Fig. B.2. Last 28 precision-recall curves for all combinations of train-test tiles for three different classifiers: one with all the features (full), another with only the ten drifting features found in that combination of train-test tiles (RFE), and one using all the remaining features with the exception of the drifting features (No-RFE).