HIVE-COTE 2.0: a new meta ensemble for time series classification

Matthew Middlehurst
University of East Anglia
Norwich, United Kingdom
m.middlehurst@uea.ac.uk

James Large
University of East Anglia
Norwich, United Kingdom
james.large@uea.ac.uk

Michael Flynn
University of East Anglia
Norwich, United Kingdom
michael.flynn@uea.ac.uk

Jason Lines
University of East Anglia
Norwich, United Kingdom
j.lines@uea.ac.uk

Aaron Bostrom
University of East Anglia
Norwich, United Kingdom
a.bostrom@uea.ac.uk

Anthony Bagnall
University of East Anglia
Norwich, United Kingdom
ajb@uea.ac.uk

ABSTRACT

The Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COTE) is a heterogeneous meta ensemble for time series classification. Since it was first proposed in 2016, the algorithm has remained state of the art for accuracy on the UCR time series classification archive. Over time it has been incrementally updated, culminating in its current state, HIVE-COTE 1.0. During this time a number of algorithms have been proposed which match the accuracy of HIVE-COTE. We propose comprehensive changes to the HIVE-COTE algorithm which significantly improve its accuracy and usability, presenting this upgrade as HIVE-COTE 2.0. We introduce two novel classifiers, the Temporal Dictionary Ensemble (TDE) and Diverse Representation Canonical Interval Forest (DrCIF), which replace existing ensemble members. Additionally, we introduce the Arsenal, an ensemble of ROCKET classifiers as a new HIVE-COTE 2.0 constituent. We demonstrate that HIVE-COTE 2.0 is significantly more accurate than the current state of the art on 112 univariate UCR archive datasets and 26 multivariate UEA archive datasets.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Supervised learning by classification**; **Ensemble methods**.

KEYWORDS

Time series classification, Multivariate time series, Heterogeneous ensembles, HIVE-COTE

ACM Reference Format:
Matthew Middlehurst, James Large, Michael Flynn, Jason Lines, Aaron Bostrom, and Anthony Bagnall. 2021. HIVE-COTE 2.0: a new meta ensemble for time series classification. In *MileTS '21: 7th KDD Workshop on Mining and Learning from Time Series, August 14th, 2021, Singapore.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

MileTS '21, August 14th, 2021, Singapore

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Time series classification (TSC) is the problem of predicting a discrete target variable from a (possibly multivariate) time series. The publication of the University of California, Riverside (UCR) TSC archive resulted in an increased interest into algorithmic research for this type of problem. An experimental study, characterised as a bake off [3], facilitated the objective and reproducible comparison of learning algorithm performance on the UCR archive. Since then, new classifiers have been proposed in the literature that have advanced the field by significantly outperforming those used in the bake off. There are currently four algorithms with reasonable claim to being state of the art for TSC based on experimentation on the recently expanded UCR archive [7]. These are: the deep learning approach called InceptionTime [10]; the tree based Time Series Combination of Heterogeneous and Integrated Embedding Forest (TS-CHIEF) [23]; the Random Convolutional Kernel Transform (ROCKET) [8]; and the heterogeneous meta-ensemble Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COTE) [13], the latest version of which is called HIVE-COTE version 1.0 (HC1) [2]. There have also been a range of algorithms proposed for MTSC [19]. Dynamic Time Warping with pointwise multivariate distance and a one nearest neighbour classifier, characterised as dependent dynamic time warping (DTW-D) [24], and multivariate versions of ROCKET, InceptionTime and CIF [15] are our multivariate benchmark.

We propose a new version of HIVE-COTE that is significantly more accurate than all four current state-of-the-art algorithms for univariate time series classification. We call this classifier HIVE-COTE version 2.0, or HC2 for short. The critical difference diagram in Figure 1 summarises the final results of HC2 against the four leading algorithms on 112 equal length UCR archives, using 30 stratified resamples on each dataset (more detail is provided in Section 3). The number associated with each algorithm is the average rank of the classifier on 112 UCR datasets and solid bars group classifiers between which there is no significant difference. HC2 is on average over 1% more accurate per problem than all of the current state of the art.

The changes from HC1 to HC2 relate to the component classifiers and a redefinition of the underlying data representations used (see Section 2). HC2 is contractable (i.e. you can give the classifier a maximum run time), checkpointable (i.e. you can restart the classifier from a previous run) and works with multivariate time

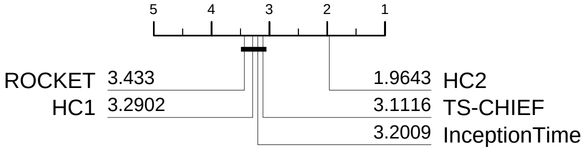


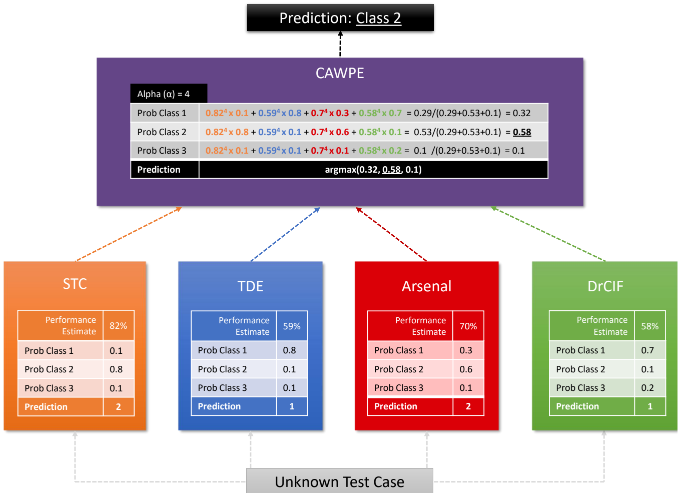
Figure 1: Critical difference diagram for HC2 against the current state of the art on 112 UCR TSC problems. The average rank for each classifier is shown, and solid lines group classifiers between which there is no significant difference. It demonstrates that there is no difference between HC1 [2], InceptionTime [10], ROCKET [8] and TS-CHIEF [23], but HC2 is significantly higher ranked than all of them. More details are given in Section 3.

series classification (MTSC). A recent study [19] concluded that that MTSC is at an earlier stage of development than univariate TSC. The only algorithms significantly better than the standard TSC benchmark, one nearest neighbour with dynamic time warping (DTW), were HC1, ROCKET, InceptionTime and CIF [15]. HC2 is significantly more accurate than all these algorithms on the UEA MTSC archive [1]. HC2 is available in two open source toolkits, sktime[1] and tsml[2]. There are more comprehensive and downloadable results are on the accompanying website[3] including an easy guide to reproducing the results.

2 HIVE-COTE 2.0 (HC2)

HIVE-COTE 2.0 replaces three of the four classifiers that make up HIVE-COTE 1.0. The component modules are: the shapelet based Shapelet Transform Classifier [5]; the convolution based ensemble of ROCKET classifiers we call the Arsenal; the dictionary based representation Temporal Dictionary Ensemble (TDE) [16]; and the interval based Diverse Representation Canonical Interval Forest (DrCIF) [15]. An overview of the updated HC2 structure is displayed in figure 2. Each component is trained independently and in addition to the final model, it is required to produce an estimate of its accuracy on unseen data. For new data, each module produces a probability estimate for each class. The controller constructs a tilted distribution through exponentiation (using $\alpha = 4$ by default) to extenuate differences in classifiers when weighting the accuracy estimate. Each module of HC2 contains new features and improvements over previous versions. These include novel algorithm improvements, multivariate extensions and contracting improvements. In addition, the method for estimating the accuracy from the train data has been improved. Rather than build 11 total models for ten fold cross validation or a single model using bagging, we construct one bagged model to estimate accuracy, and a full model to predict new cases.

Temporal Dictionary Ensemble (TDE). HIVE-COTE alpha contains the dictionary based classifier BOSS [20], which was updated to cBOSS [17] in HC1. HC2 uses the Temporal Dictionary Ensemble (TDE) (first introduced in Middlehurst et al. [16]), which draws on



Alpha (α) = 4	
Prob Class 1	0.82⁴ x 0.1 + 0.59⁴ x 0.8 + 0.7⁴ x 0.3 + 0.58⁴ x 0.7 = 0.29/(0.29+0.53+0.1) = 0.32
Prob Class 2	0.82⁴ x 0.8 + 0.59⁴ x 0.1 + 0.7⁴ x 0.6 + 0.58⁴ x 0.1 = 0.53/(0.29+0.53+0.1) = **0.58**
Prob Class 3	0.82⁴ x 0.1 + 0.59⁴ x 0.1 + 0.7⁴ x 0.1 + 0.58⁴ x 0.2 = 0.1 /(0.29+0.53+0.1) = 0.1
Prediction	**argmax(0.32, 0.58, 0.1)**

Performance Estimate	82%
Prob Class 1	0.1
Prob Class 2	0.8
Prob Class 3	0.1
Prediction	**2**

Performance Estimate	59%
Prob Class 1	0.8
Prob Class 2	0.1
Prob Class 3	0.1
Prediction	**1**

Performance Estimate	70%
Prob Class 1	0.3
Prob Class 2	0.6
Prob Class 3	0.1
Prediction	**2**

Performance Estimate	58%
Prob Class 1	0.7
Prob Class 2	0.1
Prob Class 3	0.2
Prediction	**1**

Figure 2: An overview of the ensemble structure of HIVE-COTE 2.0 for a three class problem. Each module is trained independently and produces an estimate of the probability of membership of each class for unseen data. The control unit (CAWPE) combines these probabilities, weighted by an estimate of the quality of the module found on the train data.

more recent work on dictionary classifiers [11, 22] and includes several novel features.

TDE is an ensemble of 1-NN classifiers that transforms each series into a histogram of word counts. A sliding window of length w is run along each series, and the subseries is discretised into a word of length l from an alphabet of size α. TDE transforms the window using the Symbolic Fourier Approximation (SFA) [21] transform proposed for BOSS [20]. Distance between histograms is found using histogram intersection. In addition to word frequencies, TDE also captures the frequencies of bigrams found from non overlapping windows. Thus a transformed instance includes a histogram of word counts and bigram counts for a given trio of parameters (w,l,α). TDE also includes some spatial information by the utilisation of spatial pyramids [12].

Diverse Representation Canonical Interval Forest (DrCIF). The Diverse Representation Canonical Interval Forest (DrCIF) is an interval based ensemble and an extension of its prototype version, the Canonical Interval Forest (CIF) [15]. Interval based classifiers extract phase-dependent subseries, aiming to find discriminatory features over different intervals. For time series of length m there are $m(m-1)/2$ possible intervals that can be extracted. The original interval based classifier, the Time Series Forest [9], is a component of HC1. It selects multiple intervals for each decision tree base classifier, then concatenates derived features (mean, standard deviation and slope) to form a diverse training set for each ensemble member. The other interval based classifier in HC1, RISE, selects a single interval for each base classifier, then derives spectral features (periodogram and auto-regressive terms) over the single interval. DrCIF replaces both these interval based classifiers, combining and

[1] https://github.com/alan-turing-institute/sktime
[2] https://github.com/uea-machine-learning/tsml
[3] https://www.timeseriesclassification.com/HC2.php

enhancing both feature spaces. It draws on recent ideas presented in the STSF interval based classifier [6] and the feature set method defined as the canonical time series characteristics (catch22) [14].

The base classifier for DrCIF is a simple information gain based tree used in TSF, called the time series tree [9]. Features from the tree are derived from multiple intervals taken from the base series, the first order difference series and the periodograms of the whole series. Intervals from each are randomly selected. a out of 29 possible features (seven summary statistics and 22 catch22 features) are randomly selected for each tree. For each of the three representations, This is repeated for k randomly selected phase dependent intervals and features are concatenated into a $3 \cdot k \cdot a$ length vector for each series, and the new dataset is used to build the tree. Diversity is achieved by providing each base classifier with different intervals and a different subset of the 29 features.

The Arsenal: A ROCKET Ensemble. The Random Convolutional Kernel Transform (ROCKET) [8] uses a large number of randomly parameterised convolution kernels applied to each instance. As each kernel is applied to a series, the max value and proportion of positive values are recorded and concatenated into a feature vector. These features are then used to build a linear ridge regression classifier with built in cross-validation to select the alpha parameter.

ROCKET is a very fast classifier that has state-of-the-art accuracy, and we believe it is the most important recent development in the field. It represents a different class of approach, and as such is a candidate for assimilation into the collective. However, an issue arises when trying to include ROCKET in HIVE-COTE: the ridge regressor used by ROCKET is hard to configure to produce useful probability values for each class when making predictions. The CAWPE ensemble structure of HIVE-COTE uses weighted probabilities, and relies on classifiers to produce a distribution representative of the classifiers strength of belief in predictions. One solution would be to replace the ridge regressor with a classifier that does produce representative probability estimates. However, our experimentation with suitable replacement classifiers did not yield a candidate algorithm that was as accurate as the ridge regressor for ROCKET.

To solve this problem, the version of ROCKET we use in HIVE-COTE is an ensemble of smaller ROCKET classifiers. We refer to this fusillade of ROCKETs as the Arsenal. New cases are classified using a weighted majority vote. Arsenal is slower to build than ROCKET, but its improved probabilities make it a better candidate for HC2.

Shapelet Transform Classifier (STC). Shapelets are phase independent subseries found in the training data. The STC approach to classification using shapelets is to construct a pipeline where the search for high quality shapelets is followed by a transformation where the new features represent distances to retained shapelets. A rotation forest [18] is constructed on the transformed features. The shapelet transform is highly configurable: it can use a range of sampling/search techniques in addition to alternative quality measures. We present the default settings and direct the interested reader to the `tsml` code. The original shapelet based algorithms performed an exhaustive search of all possible shapelets. This of course is very slow. However, subsequent work [5] identified that exhaustive search can actually lead to over fitting and is never necessary. Instead, we randomly search for shapelets for a given amount of time, which is now a parameter (defaults to one hour). Our version of STC is essentially the same as that used for HC1 [2], so we direct the interested reader there for more details. The multivariate version searches dimensions independently and is the same version used in the MTSC bake off [19].

3 RESULTS

We perform our experiments on the 112 equal length univariate TSC problems from the UCR time series archive [7] and 26 multivariate TSC problems from the UEA archive [1]. For each dataset we present performance as an average over 30 resamples. Both archives provide a default split into train and test sets which we use for the first resample. The remaining 29 are randomly resampled from the original split in a stratified manner. We seed each classifier and data resample using the fold index to ensure out results are reproducible. All of our non-deep learning experiments were run using the Java `tsml` toolkit implementations. For deep learning approaches we use Python `sktime` companion package `sktime-dl`[4]. Further experimental detail, including the configuration for each algorithm, is provided in the accompanying website. Our core result is that HC2 is significantly better than the current state of the art on the 112 UCR equal length datasets. Figure 1 in the Introduction shows the accuracy performance of HC2 vs the four baseline approaches. These pattern of results are also observed using negative log likelihood and area under the receiver operator curve to assess performance. Table 1 summarises the differences in test accuracy between HC2 and the four baselines.

Table 1: Summary of the differences between HC2 and the benchmarks. A negative value means the HC2 is better. Wins and losses against HC2 is in brackets. So, for example, ROCKET beat HC2 11 times but lost 97.

Classifier	Mean	Median	Max	Min	StDev
TS-CHIEF (29/77)	-1.36%	-0.41%	-22.99%	5.50%	3.64%
IncTime (32/74)	-1.69%	-0.37%	-31.04%	9.46%	5.13%
HC1 (25/85)	-1.06%	-0.69%	-10.47%	6.33%	2.10%
ROCKET (11/97)	-2.49%	-0.72%	-76.31%	3.64%	7.92%

We observe from Table 1 that there is lower variance between HC1 and HC2, but that HC2 consistently outperforms HC1 with an average accuracy of more than 1%. The variation in difference to HC2 is greater with the other three classifiers, in particular ROCKET. The median difference is lower than the mean in all cases. This suggests skew, which supports the core hypothesis that the heterogenous ensemble can compensate for the shortcomings of its components. It also suggests that HC2 has a higher representational power, in that it can find a more diverse set of features. Figure 3 shows the accuracy scatter plot of HC2 against a representative baseline classifier and Table 1 summarises the differences in test accuracy between HC2 and the four baselines.

Accuracy is not the only consideration. Table 2 summarises the run time and memory requirements for the classifiers compared in Figure 3. There are a few caveats to these results. Firstly, all of

[4] https://github.com/sktime/sktime-dl

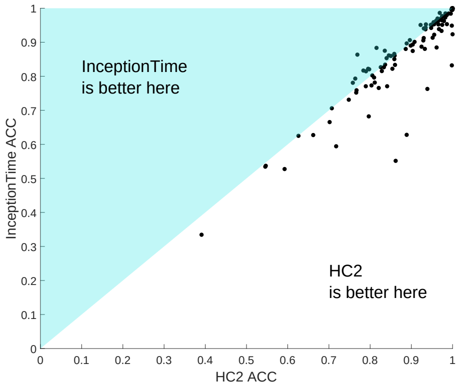


Figure 3: Scatter plot of the accuracy of HIVE-COTE 2.0 against InceptionTime.

Table 2: Median run time and memory requirements to train single resample of 112 UCR problems.

Algorithm	Total (hrs)	Average (mins)	Max Mem (MB)
ROCKET	2.85	1.53	4349
Arsenal	27.91	14.95	1683
DrCIF	45.40	24.32	920
TDE	75.41	40.40	6565
InceptionTime	86.58	46.38	-
STC	115.88	62.08	4219
HC2	340.21	182.26	6677
HC1	427.18	228.84	4876
TS-CHIEF	1016.87	544.75	26052

the results except InceptionTime are run in a single thread on a CPU. Thus InceptionTime time experiments are not really directly comparable, since it runs on a GPU. ROCKET and HC2 are forced to run in a single thread, despite being threadable. The times for the HC2 components are without the time to estimate performance, but these are included in the HC2 times. Memory is the maximum memory used throughout the run, as obtained from the Java garbage collector, and should be considered approximate.

With this in mind, we can make the following observations. ROCKET lives up to its name and can build models on all 112 data in under 3 hours, even when not threaded. If speed is your main criteria, ROCKET is a good starting point in any analysis. STC is the slowest component, but this is caused by the configuration rather than an inherent problem: STC defaults to a one hour shapelet search or a full evaluation of the shapelet search if this will take less than an hour. For the very small problems, it takes a lot longer than the other algorithms (although still less than an hour). HC2 is faster than HC1, primarily because of improvements to STC and the change in classifiers. TS-CHIEF is the slowest algorithm by far, and seems to scale less well than the others. On the slowest five problems it takes ten times longer than HC2, but the difference is minimal on smaller problems. All of the classifiers are within reasonable bounds for memory. TS-CHIEF has the highest memory requirement, with a max requirement of 26GB on HandOutlines. As with run time, it seems to scale worse than the others. HC2 requires more memory than HC1, but it is is not unreasonable. ROCKET has a worse max memory case (ElectricDevices) than Arsenal. Overall, ROCKET tends to use less memory than Arsenal but appears to scale worse for larger datasets with many cases. Arsenal uses a smaller amount of kernels for each individual classifier, meaning that each transformed set of data is smaller in size and discarded before the next is built. ROCKET on the other hand must transform using its larger amount of kernels at a single point. Figure 4 summarises the accuracy and run time results by plotting the log of the train time against the rank.

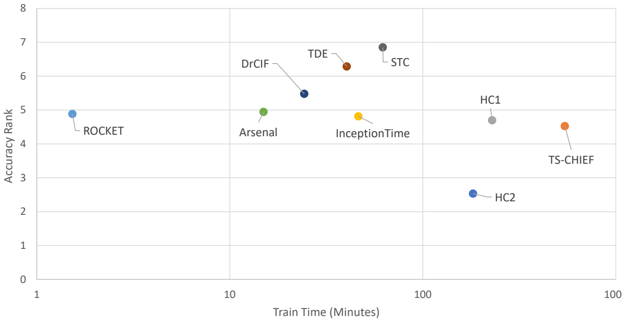


Figure 4: A comparison of classifiers in terms of accuracy rank and train time. The time and accuracy are averaged over 112 UCR problems. The train time is on a log scale.

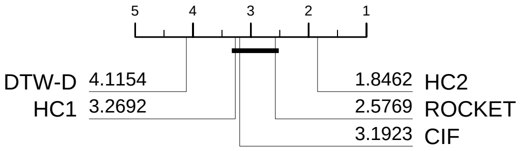


Figure 5: Test accuracy critical difference diagram for five classifiers, averaged over 30 resamples for each of the 26 UEA MTSC problems.

Figure 5 shows that HC2 is significantly more accurate than DTW-D, ROCKET, HC1 and CIF on the 26 MTSC datasets. Table 3 summarises the differences of HC2 against the benchmarks. We think these results strongly support the assertion that HC2 represents a new state of the art for multivariate time series classification.

Table 3: Summary of the differences between HC2 and the benchmarks. A negative value means the HC2 is better.

Classifier	Mean	Median	Max	Min	StDev
ROCKET (5/19)	-2.52%	-0.64%	-31.65%	4.73%	6.75%
CIF (7/19)	-1.71%	-1.58%	-21.21%	12.43%	5.56%
HC1 (6/20)	-2.25%	-1.92%	-11.27%	3.44%	3.30%
DTW-D (3/23)	-8.22%	-4.66%	-48.94%	4.87%	11.39%

4 CONCLUSION

HIVE-COTE version 2.0 is a meta ensemble of four very different classifiers, each of which is designed to capture different discriminatory features. It represents a new state of the art in terms of time series classification, significantly outperforming the previous best on both univariate and multivariate problems in terms of accuracy. We believe its strength lies in the fact that many problems have discriminatory features in multiple data domains; a shapelet might be indicative of one class value, whereas a repeating pattern may characterise another. HC2 uses a simple yet highly effective ensemble scheme to combine this information.

HC2 is available in two open source toolkits and has improved usability features such as contracting, which allow the user to specify an approximate maximum run time. Our experiments are easily reproducible, and an accompanying website contains complete results and more information on how to use HC2.

REFERENCES

[1] A. Bagnall, H. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh. 2018. The UEA multivariate time series classification archive, 2018. *ArXiv e-prints* arXiv:1811.00075 (2018). http://arxiv.org/abs/1809.06705

[2] A. Bagnall, M. Flynn, J. Large, J. Lines, and M. Middlehurst. 2020. On the usage and performance of HIVE-COTE v1.0. In *proceedings of the 5th Workshop on Advances Analytics and Learning on Temporal Data (Lecture Notes in Artificial Intelligence)*, Vol. 12588.

[3] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery* 31, 3 (2017), 606–660.

[4] G. Batista, E. Keogh, O. Tataw, and V. deSouza. 2014. CID: an efficient complexity-invariant distance measure for time series. *Data Mining and Knowledge Discovery* 28, 3 (2014), 634–669.

[5] A. Bostrom and A. Bagnall. 2017. Binary Shapelet Transform for Multiclass Time Series Classification. *Transactions on Large-Scale Data and Knowledge Centered Systems* 32 (2017), 24–46.

[6] N. Cabello, E. Naghizade, J. Qi, and L. Kulik. 2020. Fast and Accurate Time Series Classification Through Supervised Interval Search. In *proceedings of the IEEE International Conference on Data Mining*.

[7] H. Dau, A. Bagnall, K. Kamgar, M. Yeh, Y. Zhu, S. Gharghabi, C. Ratanamahatana, A. Chotirat, and E. Keogh. 2019. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica* 6, 6 (2019), 1293–1305.

[8] Angus Dempster, François Petitjean, and Geoffrey Webb. 2020. ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery* 34 (2020), 1454–1495.

[9] H. Deng, G. Runger, E. Tuv, and M. Vladimir. 2013. A time series forest for classification and feature extraction. *Information Sciences* 239 (2013), 142–153.

[10] H. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. Schmidt, J. Weber, G. Webb, L. Idoumghar, P. Muller, and F. Petitjean. 2020. InceptionTime: finding AlexNet for time series classification. *Data Mining and Knowledge Discovery* 34, 6 (2020), 1936–1962.

[11] J. Large, A. Bagnall, S. Malinowski, and R. Tavenard. 2019. On Time Series Classification with Dictionary-Based Classifiers. *Intelligent Data Analysis* 23, 5 (2019).

[12] S. Lazebnik, C. Schmid, and J. Ponce. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2. IEEE, 2169–2178.

[13] J. Lines, S. Taylor, and A. Bagnall. 2018. Time Series Classification with HIVE-COTE: The Hierarchical Vote Collective of Transformation-based Ensembles. *ACM Transactions Knowledge Discovery from Data* 12, 5 (2018), 1–36.

[14] C. Lubba, S. Sethi, P. Knaute, S. Schultz, B. Fulcher, and N. Jones. 2019. catch22: canonical time-series characteristics. *Data Mining and Knowledge Discovery* 33, 6 (2019), 1821–1852.

[15] M. Middlehurst, J. Large, and A. Bagnall. 2020. The Canonical Interval Forest (CIF) Classifier for Time Series Classification. In *proceedings of the IEEE International Conference on Big Data*. 188–195.

[16] M. Middlehurst, J. Large, G. Cawley, and A. Bagnall. 2020. The Temporal Dictionary Ensemble (TDE) Classifier for Time Series Classification. In *proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (Lecture Notes in Computer Science)*, Vol. 12457. 660–676.

[17] M. Middlehurst, W. Vickers, and A. Bagnall. 2019. Scalable dictionary classifiers for time series classification. In *proceedings of Intelligent Data Engineering and Automated Learning*. Lecture Notes in Computer Science, Vol. 11871. 11–19.

[18] J. Rodriguez, L. Kuncheva, and C. Alonso. 2006. Rotation Forest: A New Classifier Ensemble Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 10 (2006), 1619–1630.

[19] A. Pasos Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall. 2021. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery* 35, 2 (2021), 401-–449.

[20] P. Schäfer. 2015. The BOSS is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery* 29, 6 (2015), 1505–1530.

[21] P. Schäfer and M. Högqvist. 2012. SFA: a symbolic Fourier approximation and index for similarity search in high dimensional datasets. In *proceedings of the 15th International Conference on Extending Database Technology*. 516–527.

[22] P. Schäfer and U. Leser. 2017. Fast and accurate time series classification with WEASEL. In *proceedings of the ACM on Conference on Information and Knowledge Management*. 637–646.

[23] A. Shifaz, C. Pelletier, F. Petitjean, and G. Webb. 2020. TS-CHIEF: A scalable and accurate forest algorithm for time series classification. *Data Mining and Knowledge Discovery* (2020), 1–34.

[24] Mohammad Shokoohi-Yekta, Bing Hu, Hongxia Jin, Jun Wang, and Eamonn Keogh. 2017. Generalizing DTW to the multi-dimensional case requires an adaptive approach. *Data mining and knowledge discovery* 31, 1 (2017), 1–31.

APPENDIX: ABLATIVE STUDY OF HC2

We address the question of why HC2 works so well, and evaluate design decisions made in the change from HC1 to HC2. HC1 uses cross validation to estimate the test accuracy from the train data for each component. HC2 modules are all ensembles, and so it was natural to attempt to use bagging and the out of bag accuracy estimate to speed up HIVE-COTE training. However, whilst this produces good estimates of the test accuracy, the models were less accurate on unseen data for every module. Hence, we made the decision to fit a separate bagging model for the estimation stage for those that need it, thus providing an order of magnitude speed up compared to cross validation. DrCIF and Arsenal both create separate models with bagging to generate their estimates. STC builds a new Rotation Forest model with bagging for its estimate, but uses the same transformed shapelet data for both. TDE naturally takes a 70% subsample when creating its ensemble, as such a new model is not required to generate its out-of-bag error. However, we were concerned that these estimates may be biased and/or not consistent. Table 4 summarises the distributions of the differences between estimated and observed test accuracy for HC2 and its components.

Table 4: Summary of the difference between estimated and observed test accuracy for HC2 and its components. A positive figure means that the classifier is overestimating accuracy from the train data.

Classifier	Mean	Median	Min	Max	MSE
DrCIF	-2.15%	-0.93%	-46.78%	9.64%	0.40%
Arsenal	-1.17%	-0.40%	-23.13%	9.23%	0.15%
STC	1.24%	0.78%	-55.54%	26.88%	0.86%
TDE	-1.14%	-0.77%	-18.89%	10.02%	0.14%
HC2	0.47%	0.11%	-19.81%	19.17%	0.13%

Whilst there is small bias for each component, HC2 ensemble method compensates for this and has the lowest average deviation (and MSE deviation) between estimated and observed test accuracy.

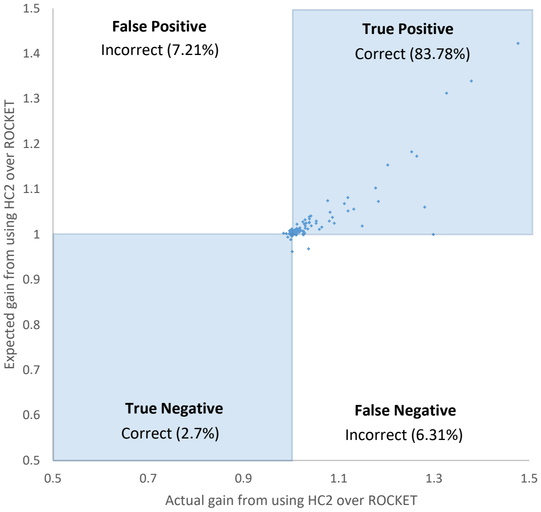


Figure 7: Texas sharpshooter plot for HC2 vs ROCKET. Each point represents a single dataset. The x-axis is the ratio of HC2 and ROCKET actual test accuracy and the y-axis is the ratio or predicted test accuracy.

Table 5: Possible variants of HC2 components.

Variant	DrCIF	Arsenal	STC	TDE
HC-1	X	X		
HC-2	X		X	
HC-3	X			X
HC-4		X	X	
HC-5		X		X
HC-6			X	X
HC-7	X	X	X	
HC-8	X	X		X
HC-9	X		X	X
HC-10		X	X	X

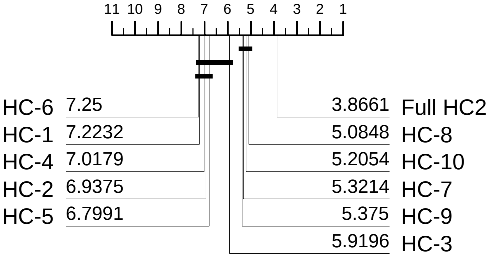


Figure 8: Critical difference diagram for 11 variants of HIVE-COTE 2.0 described in Table 5. Full HC2 contains all four components and is referred to as simply HC2 elsewhere.

This is due to the averaging ensemble effect, and the biasing effect of reusing estimates from the components: a full nested cross validation estimate would be computationally demanding and is not necessary. STC is the only component that is over optimistic. This is to be expected. STC performs a random search on the whole train data then bags rotation forest. This introduces bias, and is a possible area for future improvement. The min and the max show that there are some very large differences between estimate and observed. These primarily arise in problems where there are very few cases per class, such as PigAirwayPressure, PigCVP and PigArtPressure, which each have only two cases per class. Every classifier underestimates the test accuracy by over 10% on these problems. Figure 6 shows the difference in the test accuracy estimate and actual plotted against the log of the train set size for HC2. The picture is not conclusive, but it could be argued that the variance of the difference is decreasing, which is encouraging evidence for the consistency of the HC estimate.

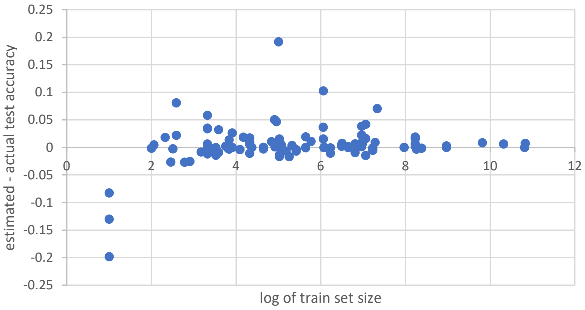


Figure 6: Difference in estimated and observed test set accuracy against the log of the train set size for 112 UCR datasets.

Another benefit of accurate estimates from the train data is that they can be used to compare classifiers with a Texas Sharpshooter plot [4]. These compare two classifiers by comparing the ratio of estimates from the train data with those of the test data to form a kind of contingency table. Computing train estimates through cross validation for TS-CHIEF and InceptionTime is unpractical due to run times. However, it is easy with ROCKET, since it is so fast. Figure 7 shows the plot for ROCKET vs HC2. Using the train estimates would lead to the correct decision of choosing HC2 on 94 of the 112 datasets.

The next issue is to quantify what impact each component has on the overall performance. Ignoring single component variants there are 11 possible combinations, identified as HC-1 to HC-10 in Table 5, with the eleventh being the Full HC2, referred to as just HC2 elsewhere. Figure 8 shows the relative performance of the 11 possible variants. The two component models (HC-1 to HC-6) form a clear clique, followed by another clique of three component versions. However, the full four component classifier is significantly more accurate than all of the other variants. This demonstrates that each element contributes to the overall whole.