# Quality Assurance for Home Spirometry using Machine Learning

1st Darcey Gardiner
*School of Computing Sciences*
*University of East Anglia*
Norwich, UK
darcey.gardiner@uea.ac.uk

2nd Harry Rogers
*Department of Engineering Science*
*University of Oxford*
Oxford, UK
harry.rogers@eng.ox.ac.uk

3rd Jason Lines
*School of Computing Science*
*University of East Anglia*
Norwich, UK
j.lines@uea.ac.uk

4th Andrew Wilson
*Norwich Medical School*
*University of East Anglia*
Norwich, UK
a.m.wilson@uea.ac.uk

5th Min Hane Aung
*School of Computing Science*
*University of East Anglia*
Norwich, UK
min.aung@uea.ac.uk

***Abstract*—Spirometry is used for evaluating lung function, playing a crucial role in assessing lung health and monitoring treatment effectiveness. Numerous studies demonstrate the potential of Machine Learning algorithms to match human experts in spirometry classification, although most approaches depend on custom data pre-processing and complex model architectures. Therefore, we apply efficient Time Series (TS) classifiers to quickly and computationally assure spirometry signal quality, enabling real-time deployment. Seven classifiers were implemented to classify spirometry curves as 'Acceptable' or 'Not Acceptable', with performance referenced against results from similar studies. The best-performing classifier was FreshPRINCE, a TS method combining TSFresh feature extraction with Rotation Forest classifier. The FreshPRINCE model achieved an accuracy of 0.9449, precision, recall and F1 Score of 0.9745, 0.9586 and 0.9665, consistently matching and sometimes outperforming more complex models. These findings suggest models, such as FreshPRINCE, could streamline spirometry analysis, reducing computational burden, whilst maintaining classification performance.**

***Index Terms*—spirometry, pulmonary function, machine learning, time series, classification**

## I. Introduction

Spirometry tests measure the flow and volume of air an individual can exhale after maximal inspiration, displayed in Flow-Volume and Volume-Time curves. It is predominantly used to diagnose and manage pulmonary diseases. Traditionally, qualified respiratory physiologists analyse spirometry results, but this process is costly and time-consuming, limiting access to testing. There is a healthcare need for rapid, reliable and affordable community-based spirometry. With the rising global burden of Chronic Respiratory Diseases (CRDs) and pressure on healthcare systems, automating this process provides an opportunity to improve diagnostic efficiency and patient outcomes. Many healthcare professionals struggle with the lack of diagnostic tools, leading to misdiagnosis or delayed diagnoses and higher CRD mortality [1]. Widespread access to spirometry and improved automated quality assurance could reduce mortality by enabling earlier diagnosis and treatment [2].

Machine Learning (ML) techniques offer new approaches to analysing spirometry data, reducing the complexities inherent in lung function testing. Classifying spirometry curve quality, which requires expert knowledge, is prone to human error, something ML can address. A recent study used pre-built Convolutional Neural Networks (CNNs) to classify spirometry curves into three categories: acceptable, early termination, and non-acceptable results, with the VGG16 model achieving 0.939 accuracy but only 0.877 precision (proportion of all the positive classifications that are actually positive) [3]. Other research, such as Das et al. [4], used a custom CNN architecture, yielding 0.87 accuracy for acceptable curves, with a sensitivity of 0.90 (the ability to correctly identify positives) and specificity of 0.85 (the ability to correctly identify negatives [5]). Bonthada et al. [6] used CNNs to detect and classify use-errors, achieving 0.94 accuracy with only 100 samples. This approach is valuable for identifying patterns in spirometry errors (e.g. early termination of the breath, extra intake of breath or submaximal blow), advancing data collection practices.

Another classification task for spirometry results is to classify an individual as having an obstructive or non-obstructive condition. The former describes a disease of the airways in which a patient struggles to get air out of the lungs such as Chronic Obstructive Pulmonary Disease (COPD) and the latter describes a condition with reduced lung volumes. A 2022 study uses supervised learning models to aid in this classification task, achieving 0.837 accuracy using the Multi-Layer Perceptron (MLP) model [7]. A different study, focussing on distinguishing between healthy and COPD affected lungs uses a fusion of ML techniques [8]. By combining a Support Vector Machine (SVM) and K-Nearest Neighbours (KNN), the model performs at a 0.94 accuracy rate and is able to detect patterns in the data for both the 'Normal' and 'Disorder' classes.

Building on these ML innovations, Clinical Decision Support Systems (CDSS) have emerged as a critical tool for leveraging automated spirometry analysis in real-time clinical decision-making. Amaral et al. [9] stressed the importance of combining the use of these algorithms into CDSS as they can identify and diagnose indistinguishable patterns in spirometry data that a respiratory clinician may miss. It is clear that the use of this technology is paramount in improving how spirometry data is classified. Time Series (TS) classification offers unique advantages in analysing spirometry data and has been successfully applied in various contexts. It has been used to classify disease types, as demonstrated by Mac et al. [10], and for predicting COPD exacerbation, as shown by Xie et al. [11]. However, to the best of our knowledge, limited research exists on using TS classification to evaluate the acceptability of spirometry recordings, a crucial quality assurance step in spirometry testing. The minimal research into TS-specific classification for spirometry data has allowed us to experiment with models that have not been used previously. Recent studies often focus on Deep Learning (DL) techniques, custom architectures, and model fusion, which frequently require significant data processing, specialised feature extraction and require large datasets. In contrast, our study aims to demonstrate that fast and powerful TS models, which require significantly less data processing, can achieve performance results that are comparable or superior to those more complex and computationally intensive approaches.

The rest of this paper is organised as follows: Section II explains and visualises the dataset. Section III describes the study design. Section IV explains the results from our study, with Section V discussing published studies as a benchmark and how our results compare. Finally, Section VI concludes the paper and discusses future research direction.

## II. The Dataset

The dataset that has been used within this study is a subset of the TIPAL dataset. It is comprised of raw data extracted from the Contec Handheld Bluetooth Spirometer SP80B (Contec Qinhuangdao, Hebei Province, China) obtained from patients with idiopathic pulmonary fibrosis as part of the TIPAL clincal trial [12]. This clinical research, currently ongoing, is designed to investigate how effective lansoprazole is for treating people with Idiopathic Pulmonary Fibrosis (IPF). IPF is a type of chronic and fibrotic Interstitial Lung Disease which destroys the lung parenchyma (lung tissue), it is derived from unknown causes and is severely lacking treatment avenues [13]. Treatment with lansoprazole may reduce the progression of this condition. The subset used for this research includes 98 patients with a total of 3404 spirometry sessions collected between August 2021 and March 2023. All patients that have taken part in this study are aged 40 years or older and all have a diagnosis of IPF, which was agreed upon by following the most up-to-date international guidelines [14].

The dataset is comprised of sequential data (examples in Figure 1). An initial preprocessing stage involved removing erroneous data and each patient is given a unique ID, in order to make them discernible from one another. A timestamp is then used to distinguish the different days and times the recordings were made. From the over 3400 entries a total of 15150 individual recordings are used for this study. Ethical approval to use the data set was granted by the University of East Anglia, Research Ethics Committee (reference ETH2223-1573).

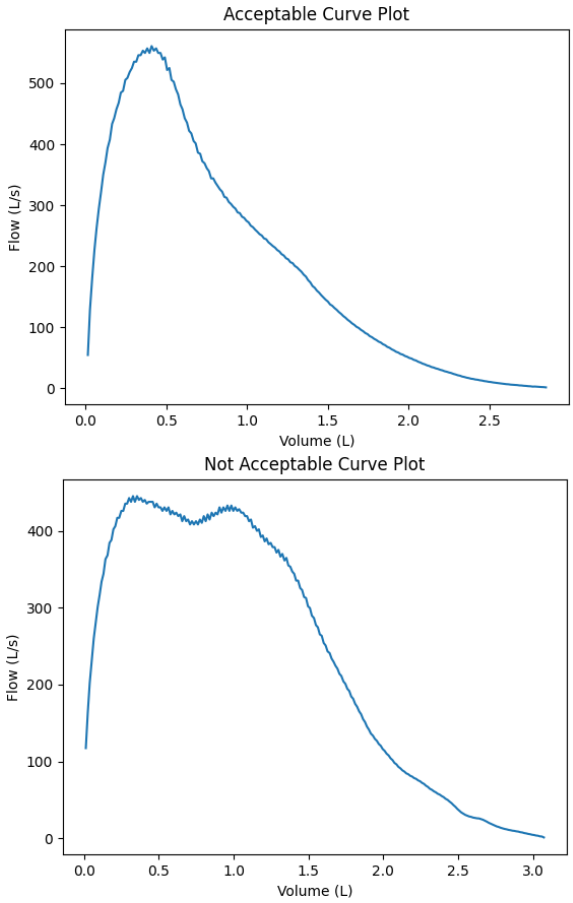


Fig. 1. Examples of an acceptable curve (top) and not acceptable curve (bottom), the latter contains two peaks possibly indicating the presence of a cough. Flow is measured in litres per second (L/S) and Volume is measured in litres (L).

## III. Study Design

The study focuses on using a variety of TS classifiers to build models that can determine if individual spirometric recordings are acceptable or not acceptable. This is the primary label given to individual recordings within the same session for them to be assessed in context. This label is provided using the 2019 American Thoracic Society (ATS) and European Respiratory Society (ERS) Spirometry guidelines [15]. This is explained in Section III-A. An example of the labelled Flow-Volume curves can be seen in Figure 1, each one is a visualisation of a single recording from a patient.

This study involves three main stages: training models with default parameters, applying hyperparameter tuning, and evaluating the optimal model using person-independent k-fold

cross-validation. Person independence ensures generalisability by preventing training and testing on data from the same individual.

### A. Data Preprocessing and Labelling

Using the ATS/ERS standards outlined in their report [15], and the key spirometric metrics from the dataset, each breath was evaluated for acceptability. The two primary metrics used for this evaluation are Forced Vital Capacity (FVC) and Forced Expiratory Volume in the first second ($FEV_1$). FVC measures the total volume of air exhaled during a forced breath manoeuvre, while $FEV_1$ quantifies the volume of air exhaled within the first second. If both FVC and $FEV_1$ met the acceptability criteria, the breath was labelled as 'Acceptable'. Conversely, if the criteria were not met for either FVC, $FEV_1$, or both, the breath was labelled as 'Not Acceptable'. While some breaths labelled as 'Not Acceptable' may still have been usable, this aspect was not explored during this study.

It is crucial to use ML to aid in labelling spirometry curves as some of the ATS/ERS guidelines require experts to visually evaluate specific criteria that cannot be assessed purely numerically. For example, identifying artefacts such as evidence of an obstructed mouthpiece, evidence of a leak or hesitation at the start of a breath often depends on a visual inspection by a trained professional. As home spirometry becomes more prevalent, automating this evaluation with ML can enhance consistency and allow for large-scale analysis of spirometry curves without requiring the time and expertise of respiratory specialists. By incorporating ML, it can address some of the limitations of manual review: subjectivity and human error.

All breaths were initially of different lengths as patients blow for varying amounts of time. To make them compatible for batch processing, all breaths were then zero-padded to the same length. This ensures the preservation of sequence order, as padding with zeros does not alter the intrinsic temporal relationships within the data.

### B. Classifiers

The aeon toolkit [16] offers a range of TS specific models for classification tasks. Middlehurst et al. [17] provide a comprehensive evaluation of these classifiers, comparing their performance and speed. This evaluation offers a broad perspective on the strengths and weaknesses of various TS classifiers. Based on this, the following seven classifiers were selected:

*1) Catch22:* CAnonical Time Series CHaracteristics (Catch22) [18] is a feature extraction method integrated into a classifier within the aeon toolkit [16]. Catch22 transforms TS data into a set of 22 descriptive features, derived from the over 7000 initially available within the Highly Comparative Time Series Analysis (HCTSA) toolbox [19]. This reduction of features allows for a more concise evaluation of the input data, reducing its complexity but preserving the key characteristics. The extracted features are then typically paired with a default decision tree-based model, such as Random Forest, to produce predictions. This model achieves faster training times and reduced computational cost compared to models that use raw TS data. The tuning in stage two adjusts Catch22 parameters related to feature selection, outliers and NaN handling and computational efficiency.

*2) FreshPRINCE:* Fresh Pipeline with RotatIoN forest Classifier (FreshPRINCE) [20] is a TS classification model that integrates the TSFresh feature extraction algorithm with a Rotation Forest Classifier. The pipeline begins with TSFresh; used to extract just under 800 features from the TS input data. The features chosen capture a comprehensive set of characteristics designed to perform well on classification tasks. The Rotation Forest Classifier pairs well with the extracted features as it is an ensemble method known for its accuracy in high-dimensional feature spaces. The FreshPRINCE classifier often outperforms basic TS classification methods, such as k-Nearest Neighbours (KNN), but its computational requirements are higher due to the more extensive feature-extraction process. The stage two tuning for FreshPRINCE optimises parameters for feature extraction and the size and robustness of the Rotation Forest classifier.

*3) KNN DTW:* K-Nearest Neighbour TS Classifier, using Dynamic Time Warping distance metric (KNN DTW), is another commonly used classification model. The model uses the k-Nearest Neighbours (KNN) algorithm which classifies a sample based on the majority label of the k-closest neighbours. As an alternative to a standard distance metric, such as Euclidean distance, in this case, Dynamic Time Warping (DTW) is used. DTW is a well-suited distance metric for TS data as it measures the similarity of sequences by aligning them with potential shifts in time or warping. This means it can handle the variations in timing and non-linear distortions between sequences. Although it can perform competitively against other TS classifiers, the computational cost is high for large datasets due to the large number of pairwise distance calculations needed. The stage two tuning for the KNN DTW model optimises parameters to balance bias and variance whilst adjusting the influence of neighbours for better performance.

*4) ROCKET:* RandOm Convolutional KErnel Transform (ROCKET) [21] is a classifier that aims to significantly increase training speed without sacrificing the quality of the results. Using random Convolutional Kernels enables a single streamlined approach to extract multiple features from the raw TS data, which may have previously required multiple, specialised techniques. Once extracted, the features are aggregated to create a fixed-length feature representation, regardless of the original TS input length, and then passed through a linear model. The benefit of using a linear model allows for scalability for large datasets and in turn improves the speed of classification. The stage two tuning for ROCKET changes the number of kernels to balance feature representation, computational cost and the risk of overfitting and underfitting.

*5) Time Series Forest:* Time Series Forest (TSF) classifier is an ensemble-based model built specifically for TS data [22]. It is derived from the original concept of decision trees but

extended for TS data by focusing on random intervals. For each interval, three key features are extracted: mean, standard deviation and slope. These features summarise the TS interval and compile it into an understandable feature, as opposed to the raw TS data. Due to the simplistic nature of the feature extraction, the TSF classifier is computationally efficient compared to models with more complex feature extraction methods. The stage two tuning for TSK adjusts parameters controlling the number of decision trees, interval extraction settings and time constraints, whilst enhancing computational efficiency.

*6) SVC:* TS specific Support Vector Classifier (TSSVC) is a Support Vector Machine (SVM) based classifier that is built specifically for TS data [23]. SVM classifies data based on finding the optimal hyperplane separating classes in a high-dimensional space. It is adapted for TS data by applying the SVM to pairwise similarity measures rather than the raw TS data. It uses the Global Alignment Kernel (GAK), which is derived from DTW, and quantifies the similarity between two TS sequences. It computes a kernel matrix, which can be mapped to a high-dimensional feature space, outlined by the GAK. The model can learn the hyperplane, between classes from this transformed feature space. Whilst it can offer robustness and strong classification performance, SVC can be computationally expensive and parameter sensitive - significantly affecting results. The stage two tuning for TSSVC optimises parameters to balance pattern discovery, overfitting and underfitting, whilst adjusting training time to reduce computational cost.

*7) Shapelet Transform:* Shapelet Transform Classifier (STC) makes use of shapelets; discriminative sub-sets of TS sequences, to transform the input data into a new feature space [24]. Shapelets are specifically chosen based on their ability to differentiate between two classes, leveraging their powerful ability to detect patterns most relevant for classification. The algorithm employs a single-scan method to identify the best k shapelets. These shapelets are then used to transform the dataset, in which each of the k features is a representation of the distance between a TS sequence and a corresponding shapelet. Shapelet transform can provide insights into the patterns that separate classes, providing a degree of interpretability, however discovering these shapelets is computationally expensive. The stage two tuning for STC adjusts parameters that control the quantity and quality of extracted shapelets, impacting pattern recognition and classification, whilst also optimising computational efficiency and processing speed.

### C. Evaluation Metrics

To assess the quality of the models, several evaluation metrics were used. The purpose of using multiple metrics is to provide justification for the reliability and true effectiveness of the model. Thus, the metrics are as follows;

*1) Accuracy:* A standard evaluation metric, but does not account for class imbalances.

*2) Precision:* A metric used to evaluate the proportion of all positive classifications that are actual positives. A high precision score, approaching or equalling 1.0, indicates that almost all samples predicted as a particular class truly belong to that class.

*3) Recall:* Also known as the True Positive Rate (TPR), it evaluates the proportion of actual positives that were correctly classified as positive. A recall score close to or equal to 1.0 signifies the model's ability to correctly classify almost all samples from a particular class.

*4) F1 Score:* This metric provides a balance between precision and recall, it works as a better evaluation metric for class-imbalanced data than accuracy.

*5) Area Under the Receiver Operating Characteristic Curve (AUROC):* Provides a representative probability of how well the classifier can distinguish between the positive and negative classes. The closer the score is to 0.5, the closer the results are to random chance.

### D. Hyperparameter Tuning

In the second stage of the study hyperparameter tuning was implemented using the Optuna library [25]. This framework is flexible and efficient, using Bayesian optimisation to allow for focus on promising regions rather than exhaustive grid or random searches. This is done by using a probabilistic model, such as Tree-structured Parzen Estimator (TPE), which guides the search as a surrogate model. The purpose of using a surrogate model is to approximate the results of the objective function. This is the primary evaluation metric, for this we chose F1 Score.

The surrogate model aids in approximating the performance of the current set of hyperparameters, allowing the optimiser to make informed decisions. This involves deciding between two main strategies: exploration (searching new areas within the hyperparameter space) or exploitation (refining the search within areas that have already demonstrated promising results). TPE uses Kernel Density Estimation (KDE) to approximate the distribution of good and bad hyperparameter configurations [26]. The distributions are applied to two categories of objective function values: low values (indicating good results) and high values (indicating poor results).

For KDE, the low-value sets are denoted as $X_{low} = \{x_1, x_2, ..., x_{n_{low}}\}$ and high-value sets as $X_{high} = \{x_1, x_2, ..., x_{n_{high}}\}$. The general formula for KDE is:

$$\hat{p}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{1}$$

Where $\hat{p}(x)$ is the estimated probability density at point $x$, $n$ is the number of data points (either $n_{low}$ or $n_{high}$) and $h$ is used to determine how much weight is given to each data point. $K$ is a kernel function used to represent how much influence a data point $x_i$ has on the density estimate at location $x$. This is typically the Gaussian kernel, which can be expressed as:

$$K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right) \tag{2}$$

Where $u$ is the scaled distance between the location $x$ and each data point $x_i$. The likelihood ratio of the two densities is used to calculate the TPE:

$$\text{ratio}(x) = \frac{\hat{p}_{\text{low}}(x)}{\hat{p}_{\text{high}}(x)} \tag{3}$$

If $\hat{p}_{\text{low}}(x)$ is high, it indicates that the objective function returns better results for those hyperparameters. Conversely, if $\hat{p}_{\text{high}}(x)$ is high, the objective function returns worse results for those parameters.

In Optuna, model tuning is referred to as a 'Study' that finds optimal hyperparameters through iterative trials. Each trial evaluates a different set of parameters, and unpromising trials are pruned early to reduce training time. For our tuning, we selected to use 10 trials, a number chosen based on the dataset size and the relatively small hyperparameter space. As most parameters were categorical, fewer trials were sufficient to achieve meaningful exploration. Furthermore, for computationally expensive models like KNN DTW and TSSVC, 10 trials struck a balance between exploration and training efficiency. Optuna's dynamic search space feature further enhanced efficiency by adapting parameter suggestions based on prior trial results, ensuring that even a limited number of trials provide valuable results.

## IV. Study Results

As explained previously, the first stage compared the baseline performance of all models using the evaluation metrics above, without any parameter tuning. FreshPRINCE performed best (0.925 accuracy, 0.956 precision, 0.953 recall, 0.954 F1, 0.862 AUROC), closely followed by Catch22 (0.924 accuracy, 0.954 precision, 0.954 recall, 0.954 F1, 0.861 AUROC). These results indicate that both models consistently outperformed the others in accuracy, precision, and F1 score.

After each classifier underwent hyperparameter tuning, the results were compared and Figure 2 summarises the results of the best trial for each model. All classifiers demonstrated improved performance after hyperparameter tuning, as expected. FreshPRINCE remained the best-performing model, marginally outperforming Catch22 in all metrics.

The final stage involved retraining using the best parameters obtained in the previous stage, on a person-independent 5-fold cross-validated model, to eliminate potential patient-level bias. Table I presents the evaluation metrics resulting from testing on unseen data. The top three models remain consistent, as shown in Figure 3, which visually compares their performance across all metrics. FreshPRINCE outperformed Catch22 in most metrics, but Catch22 achieved a higher recall, demonstrating its strength in correctly identifying positive samples. ROCKET consistently underperformed compared to both FreshPRINCE and Catch22 across all metrics except recall. However, its lower performance in the other metrics suggest that its recall advantage could be due to a higher false positive rate rather than true effectiveness.

Inference times (including signal pre-processing and feature processing stages) were calculated using a test set of 20 subjects. The Shapelet model was the fastest at 2.92 seconds, followed by Catch22 (6.61 seconds) and TSForest (17.3 seconds). While run-time testing was not the primary focus, the results suggest potential for future mobile deployment. Reimplementation into more efficient languages for final app production could reduce latency, but is beyond the scope of this study. Given the resource constraints of mobile devices, balancing computational efficiency and accuracy is crucial.

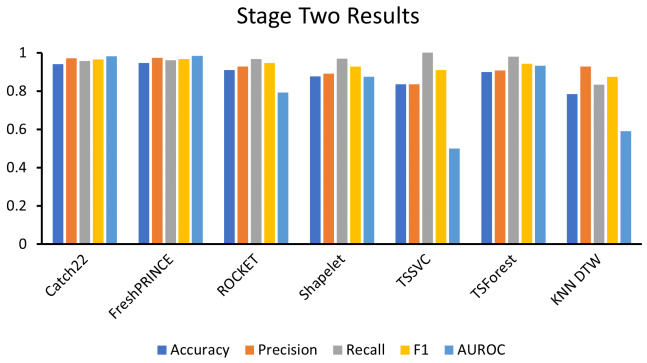


Fig. 2. Results from each classifier at the second stage of the study.

At all three stages, TSSVC consistently achieved the highest recall, classifying nearly all 'Not Acceptable' spirometry tests correctly. However, this resulted in a high false positive rate, as reflected in its low precision score. The AUROC score (0.566) further highlights TSSVC's weak ability to distinguish between 'Acceptable' and 'Not Acceptable' tests, leading to near-random classification. Although FreshPRINCE has a slightly lower recall (0.959) compared to TSSVC, it is a more balanced model overall. With higher precision (0.975) and a stronger AUROC (0.918), FreshPRINCE offers better distinction between 'Acceptable' and 'Not Acceptable' tests, making it more reliable with fewer false positives.

TABLE I
Results from each classifier at the final stage of the study
(Bold indicates best score for each metric)

| | Accuracy | Precision | Recall | F1 | AUROC |
|---|---|---|---|---|---|
| **Catch22** | 0.943 | 0.971 | 0.961 | 0.966 | 0.909 |
| **FreshPRINCE** | **0.945** | **0.975** | 0.959 | **0.967** | **0.918** |
| **ROCKET** | 0.903 | 0.916 | 0.972 | 0.943 | 0.769 |
| **Shapelet** | 0.862 | 0.876 | 0.972 | 0.921 | 0.649 |
| **TSSVC** | 0.837 | 0.839 | **0.995** | 0.910 | 0.566 |
| **TSForest** | 0.893 | 0.900 | 0.980 | 0.939 | 0.725 |
| **KNN DTW** | 0.752 | 0.889 | 0.818 | 0.852 | 0.555 |

## V. Study Results versus Benchmark Reference

Table II evaluates published results on spirometry-related classification. While direct comparison is limited through differences in methods and datasets (not publicly available), our accuracy aligns with the results in Table II. Many of these studies rely primarily on accuracy as the main evaluation metric, without addressing the potential impact of class imbalance on their results. As highlighted by Rezvani et al. [29], this can be

TABLE II
A TABLE COVERING PUBLISHED PAPERS, THEIR MODELS USED AND BEST RESULTS.

| | **Martins et al. 2024 [3]** | **Das et al. 2024 [4]** | **Wang et al. 2022 [27]** | **Bhattacharjee et al. 2022 [7]** | **Taloba et al. 2025 [8]** | **Viswanath et al. 2018 [28]** | **Mac et al. 2024 [10]** | **Our Study Results** |
|---|---|---|---|---|---|---|---|---|
| **Size/Type of Dataset** | 5287 flow-volume curves. | 36873 flow-volume curves. | 16502 flow-volume curves. | 1314 spirometry reports. | 920 annotated audio recordings. | 36161 annotated audio recordings. | 2871 Pulmonary Function Tests (PFTs) (TS data). | 15150 individual spirometry recordings (TS data). |
| **Model/s Used** | 6 CNN models. | Custom CNN-NN. | DL, ResNet50. | Supervised learning models. | Supervised learning models. | A variety of ML and DL models. | Novel algorithm built using cascaded MiniRocket classifiers. | TS classifiers. |
| **Data Preprocessing** | Data augmentation – resizing images and colour channel separation. | Processing of raw data into pixel matrices and calculating ATS/ERS criteria. | Extraction of numerical information, creation of curves. | Not described. | Z-score normalisation, generating MFCC and Forward Feature Selection. | Cleaning data, multiple feature extraction and generating Mel-spectrogram features. | Not described. | Zero-padding to fixed length. |
| **Best Results** | VGG16 – 0.939 accuracy, with highs of 0.977 Precision, 0.968 Recall and 0.952 F1 Score. | 0.87 accuracy for acceptability, 0.92 accuracy for usability. 0.92 sensitivity and 0.96 specificity for usability. | 0.951 accuracy for FEV1 acceptability and 0.943 accuracy for FVC usability. | MLP – accuracy of 0.837 and Matthew's correlation coefficient of 0.682. | SVM-KNN – accuracy of 0.94. | Gradient Boost – 0.982 precision, 0.866 recall. GCRNN – 0.983 precision and 0.880 recall. | Mean accuracies for each of the classes range from 0.91-0.94. | FreshPRINCE - accuracy 0.945, precision 0.975, recall 0.959, F1 0.967 and AUROC 0.918. |

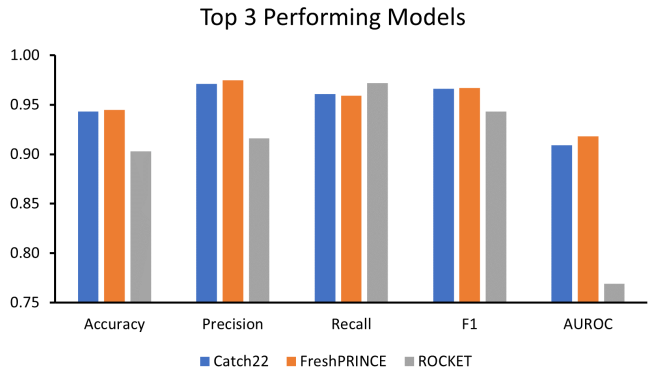


Fig. 3. A visual comparison of the best three classifiers, Catch22, FreshPRINCE and ROCKET using results from the final stage.

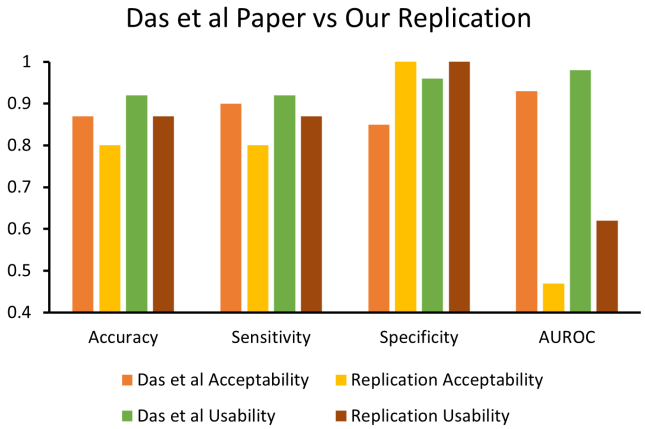


Fig. 4. Das et al. [4] model Vs Our Replication.

problematic because a high accuracy can still be achieved even if all minority class samples are misclassified. To address this issue, it is essential to incorporate additional metrics designed to account for class imbalance, such as precision, recall and F1 Score, which we have used throughout this study.

In our study, the best-performing model achieved a precision score of 0.9745, comparable to other results in Table IV. In another example, Martins et al. [3] report a recall score of 0.968, while our best model achieved a recall score of 0.959, demonstrating comparable performance.

For a direct comparison, we replicated the methodology from Das et al. [4] as it closely mirrors how respiratory experts assess spirometry curves, looking at both the plots and spirometric metrics. Other methods were excluded due to missing pre-processing details [7], [10], reliance on audio data [8], [28], or smaller datasets [3], [27].

Figure 4 compares the Das et al [4] study results (green and orange) with our replication (yellow and brown). The original consistently outperformed our replication across all four metrics: accuracy, sensitivity, specificity and AUROC. While the first three remained relatively close, AUROC showed a notable drop in our replication, indicating weaker model generalisation. This likely stems from differences in dataset size and diversity - Das et al. [4] used 36,873 flow-volume curves, from participants aged 6-79, while our dataset contained only 15,150 breaths from participants over 40. Moreover, class imbalance in our dataset led to frequent misclassification, further impacting performance.

## VI. Discussion and Conclusion

The significance of FreshPRINCE and Catch22 providing the best results, indicate that the spirometry data has meaningful patterns that are captured well by statistical and domain-specific features. However, ROCKET and Shapelet Transform traditionally work well on TS classification, but they are prone to overfitting on smaller datasets, which could be the case for this study [21], [30].

Future work will focus on implementing the next step in the ATS/ERS spirometry guidelines by labelling 'Not Acceptable'

breaths as 'Usable' and 'Not Usable' and providing real-time feedback with teaching tips for improving unacceptable breaths. This labelling will simplify grading and help identify the reasons for unacceptable or unusable breaths, enabling real-time detection of use-errors.

Additionally, expanding the dataset beyond TIPAL, which only includes IPF patients, to include healthy subjects and those with other pulmonary conditions will allow a more comprehensive evaluation of the method. Another area to explore in the future is improving inference times and optimising models to support deployment onto mobile devices. Applying lightweight time-series models to portable spirometers and smartphone apps can enable efficient, real-time, at-home spirometry with minimal resource demands, making the technology more accessible and scalable for a wider population.

## REFERENCES


[1] R. Louis, I. Satia, I. Ojanguren, F. Schleich, M. Bonini, T. Tonia, D. Rigau, A. Ten Brinke, R. Buhl, S. Loukides *et al.*, "European respiratory society guidelines for the diagnosis of asthma in adults," *European Respiratory Journal*, vol. 60, no. 3, 2022.

[2] M. Xie, X. Liu, X. Cao, M. Guo, and X. Li, "Trends in prevalence and incidence of chronic respiratory diseases from 1990 to 2017," *Respiratory research*, vol. 21, no. 1, pp. 1–13, 2020.

[3] C. Martins, H. Barros, and A. Moreira, "Transfer learning in spirometry: Cnn models for automated flow-volume curve quality control in paediatric populations," *Computers in Biology and Medicine*, vol. 184, p. 109341, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482524014264

[4] N. Das, K. Verstraete, S. Stanojevic, M. Topalovic, J.-M. Aerts, and W. Janssens, "Deep-learning algorithm helps to standardise ats/ers spirometric acceptability and usability criteria," *European Respiratory Journal*, vol. 56, no. 6, 2020.

[5] A. Swift, R. Heale, and A. Twycross, "What are sensitivity and specificity?" *Evidence-Based Nursing*, vol. 23, no. 1, pp. 2–4, 2020.

[6] S. Bonthada, S. P. Perumal, P. P. Naik, M. A. Padukudru, and J. Rajan, "An automated deep learning pipeline for detecting user errors in spirometry test," *Biomedical Signal Processing and Control*, vol. 90, p. 105845, 2024.

[7] S. Bhattacharjee, B. Saha, P. Bhattacharyya, and S. Saha, "Classification of obstructive and non-obstructive pulmonary diseases on the basis of spirometry using machine learning techniques," *Journal of Computational Science*, vol. 63, p. 101768, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877750322001508

[8] A. I. Taloba and R. Matoog, "Detecting respiratory diseases using machine learning-based pattern recognition on spirometry data," *Alexandria Engineering Journal*, vol. 113, pp. 44–59, 2025.

[9] J. L. M. do Amaral and P. L. de Melo, "Clinical decision support systems to improve the diagnosis and management of respiratory diseases," in *Artificial intelligence in precision health*. Elsevier, 2020, pp. 359–391.

[10] A. Mac, J. Wu, C. Ryan, S. Valaee, and C.-W. Chow, "Time-series machine learning model classifies lung function using spirometry flow-volume loops alone," in *A75. REVISITING CURRENT METHODS IN PULMONARY FUNCTION TESTING*. American Thoracic Society, 2024, pp. A2624–A2624.

[11] Y. Xie, S. J. Redmond, M. S. Mohktar, T. Shany, J. Basilakis, M. Hession, and N. H. Lovell, "Prediction of chronic obstructive pulmonary disease exacerbation using physiological time series patterns," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2013, pp. 6784–6787.

[12] M. Jones, A. Cahn, N. Chaudhuri, A. B. Clark, I. Forrest, M. Hammond, S. Jones, T. M. Maher, H. Parfrey, G. Raghu *et al.*, "The effectiveness and risks of treating people with idiopathic pulmonary fibrosis with the addition of lansoprazole (tipal): study protocol for a randomised placebo-controlled multicentre clinical trial," *BMJ open*, vol. 15, no. 2, p. e088604, 2025.

[13] D. S. Glass, D. Grossfeld, H. A. Renna, P. Agarwala, P. Spiegler, J. DeLeon, and A. B. Reiss, "Idiopathic pulmonary fibrosis: Current and future treatment," *The clinical respiratory journal*, vol. 16, no. 2, pp. 84–96, 2022.

[14] G. Raghu, M. Remy-Jardin, L. Richeldi, C. C. Thomson, Y. Inoue, T. Johkoh, M. Kreuter, D. A. Lynch, T. M. Maher, F. J. Martinez *et al.*, "Idiopathic pulmonary fibrosis (an update) and progressive pulmonary fibrosis in adults: an official ats/ers/jrs/alat clinical practice guideline," *American Journal of Respiratory and Critical Care Medicine*, vol. 205, no. 9, pp. e18–e47, 2022.

[15] B. L. Graham, I. Steenbruggen, M. R. Miller, I. Z. Barjaktarevic, B. G. Cooper, G. L. Hall, T. S. Hallstrand, D. A. Kaminsky, K. McCarthy, M. C. McCormack *et al.*, "Standardization of spirometry 2019 update. an official american thoracic society and european respiratory society technical statement," *American journal of respiratory and critical care medicine*, vol. 200, no. 8, pp. e70–e88, 2019.

[16] M. Middlehurst, A. Ismail-Fawaz, A. Guillaume, C. Holder, D. Guijo-Rubio, G. Bulatova, L. Tsaprounis, L. Mentel, M. Walter, P. Schäfer *et al.*, "aeon: a python toolkit for learning from time series," *Journal of Machine Learning Research*, vol. 25, no. 289, pp. 1–10, 2024.

[17] M. Middlehurst, P. Schäfer, and A. Bagnall, "Bake off redux: a review and experimental evaluation of recent time series classification algorithms," *Data Mining and Knowledge Discovery*, pp. 1–74, 2024.

[18] C. H. Lubba, S. S. Sethi, P. Knaute, S. R. Schultz, B. D. Fulcher, and N. S. Jones, "catch22: Canonical time-series characteristics: Selected through highly comparative time-series analysis," *Data Mining and Knowledge Discovery*, vol. 33, no. 6, pp. 1821–1852, 2019.

[19] B. D. Fulcher and N. S. Jones, "hctsa: A computational framework for automated time-series phenotyping using massive feature extraction," *Cell Systems*, vol. 5, no. 5, pp. 527–531.e3, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405471217304386

[20] M. Middlehurst and A. Bagnall, "The freshprince: A simple transformation based pipeline time series classifier," in *International Conference on Pattern Recognition and Artificial Intelligence*. Springer, 2022, pp. 150–161.

[21] A. Dempster, F. Petitjean, and G. I. Webb, "Rocket: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, 2020.

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

[23] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, and E. Woods, "Tslearn, a machine learning toolkit for time series data," *Journal of Machine Learning Research*, vol. 21, no. 118, pp. 1–6, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-091.html

[24] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall, "Classification of time series by shapelet transformation," *Data mining and knowledge discovery*, vol. 28, pp. 851–881, 2014.

[25] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[26] S. Watanabe, "Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance," *arXiv preprint arXiv:2304.11127*, 2023.

[27] Y. Wang, Y. Li, W. Chen, C. Zhang, L. Liang, R. Huang, J. Liang, D. Tu, Y. Gao, J. Zheng *et al.*, "Deep learning for spirometry quality assurance with spirometric indices and curves," *Respiratory Research*, vol. 23, no. 1, p. 98, 2022.

[28] V. Viswanath, J. Garrison, and S. Patel, "Spiroconfidence: Determining the validity of smartphone based spirometry using machine learning," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2018, pp. 5499–5502.

[29] S. Rezvani and X. Wang, "A broad review on class imbalance learning techniques," *Applied Soft Computing*, vol. 143, p. 110415, 2023.

[30] P. Kidger, J. Morrill, and T. Lyons, "Generalised interpretable shapelets for irregular time series," *arXiv preprint arXiv:2005.13948*, 2020.