Danish Resume

Filterbankes effekt på detektering af fjendtlige talegenkendelsesangreb i støj

I en verden hvor stadig flere enheder er internetforbundne og får flere input-modaliteter end tidligere, vil systemer der anvender visuelle og auditive modaliteter for interaktion med verdenen omkring, blive stadig mere eksponeret til fjendtlige angreb. Et nyligt eksempel hvorpå "fjendtlige angreb"er en særlig bekymring, ligger i beslutningsprocessen af selvkørende biler, hvor der er potentiale for tab af menneskelig liv. Auditive systemer der gøre brug af Automatisk Talegenkendelse (ASR) bliver typisk ikke givet ansvar i den kaliber, men nyere ASR-systemer såsom stemmeassistenter bliver ofte udstyret med online shoppingfunktionaliteter og/eller tilsluttet “smart home”-systemer. Angreb på disse auditive systemer kan foregå helt ubevidst for ejeren af systemet, hvilket gør muligt at handle online uden at ejeren ved det eller at styre smart-hjemmeapparater, såsom sikkerhedskameraer, låse eller høj effekts apparater, hvilket hvis uopdaget kan forårsage seriøse ulykker. Ved brug af viden fra psykoakustik og foregående i talegenkendelsessystemer og lydkomprimering har man en idé om, hvordan menneskelig tale typisk udtrykkes i trykbølge frekvenser og effektive repræsentationer af disse udtryk. I dag udnytter vi denne viden i vores ASR-systemer til at få dem til at yde tilfredsstillende, men denne viden kan også gøre disse systemer sårbare over for målrettede fjendtlige angreb. Vi undersøger en række klassifikationsmodeller med varierende filterbanke for at udsøge de bedst egnede cepstral-funktionsrum til en fjendtlig angrebs detektionsmode, der ville gøre ASR-system mere modstandsdygtige.

I vores undersøgelser finder vi at de model der bruge filterbanke der koncentrere opløsning i det højere frekvensrum over det lavere frekvensrum, yder bedre i klassificeringsydelse end dem der gør det modsatte eller vægter ligeligt. Der dog den undtagelse at når klassifikationsmodeller udelukkende er blevet udsat for baggrundsstøj i de højere frekvenser under optimeringer af klassificering, så som i et køkken med klirende tallerkener og bestik, så daler denne klassificering fordel dramatisk.

Filter Bank Effects on Detecting Adversarial Speech Recognition Attacks In Noise

Christian Heider Nielsen*
VGIS Master Student - Study No. 20138103
Aalborg University
Danmark
chrini13@student.aau.dk

Abstract

In an existence of ever more internet connected devices and increasing number of input modalities, systems employing visionary and auditory modes of interaction with the world, become ever so slightly more exposed to adversarial behaviour. A very recent vision example of where "adversarial attacks"are of particular concern are in the decision making of self-driving cars, where there are a potential of human casualties. Although auditory systems such as Automatic Speech Recognition (ASR) systems are typically not responsible for such real-time tasks, recent ASR systems such as voice assistants are often equipped with online shopping features or connected to smart home systems. These auditory attacks may be imperceptible to humans, meaning it would be possible to shop online without the knowledge of the owner, or control smart home devices, such as security cameras, locks or high wattage appliances. Employing knowledge from psychoacoustics and prior works on speech recognition systems and audio compressing, we have an idea of how human speech is typically expressed terms of frequencies, harmonics and efficient representations of these in cepstrums. Today we utilise this knowledge in our ASR system today but this knowledge also make these system vulnerable to targeted attacks if not considered. We survey a range of adversarial classification models with varying filterbanks, to ease out the best suited cepstral feature space for a adversarial attack detection model in terms of classification performance.

1 Introduction

The prevalence of Automatic Speech Recognition systems on the rise, with systems such as Google Assistant, Amazon Alexa, Samsung Bixby, Apple Siri and Microsoft Cortana.. And with them comes an ever increasing attack surface of over the air remote execution possibilities, without the security and control measures of radio protocols such as cryptography and secrets to validate inputs. This attack surface comprises locally present audio playback devices, that may be accessible either directly from breached security or by proxy of user streamed audio content.

Some of the earlier named system are given the responsibilities of home automation, through Internet of Things systems, turning on and off devices, activating and deactivating security measures, locking and unlocking door, opening and closing windows and garage doors. While other of the mentioned systems given access to do administrative tasks for the user that may involve monetary transactions, making calls, sending and forwarding emails, banking and making purchases online.

These are just some immediate system examples from hypothesised with experience from recent years development, and at all exhaustive of the possible attack surface as the these system grow and gain more responsibility. From the the aforementioned examples it is obvious that without a way of

*cnheider.github.io

verifying identity or validity of the request being made by an authorised user, that these functionalities leaves gaping security holes and vulnerabilities in users systems.

If these flaws are left unaddressed it would be possible for an adversary to gain unauthorised access to these voice interfaces by remotely executing adversarial sound bites or injection on a speaker in vicinity of a subject ASR system. Adversarial Attacks are carried out by injecting a signal into an ASR system, either by itself or injected into an another signal.

The nature of the attack can be targeted or untargeted. Targeted meaning the attack made to accomplish a certain outcome and thus being the most dangerous in terms of intrusive behaviour. Untargeted attack are more of a nuisance, however from the point of view of this investigation the nature of the attack is irrelevant.

When generating both targeted and untargeted attacks an iterative optimisation method is used to optimise the signal injections, to respectively maximise and minimise the target prediction of the subject ASR model.

A usual attack will be in the form of small perturbations an ASR input signal, unnoticeable to the any human bystanders. We will refer to these as hidden attacks, what may be incomprehensible noise to the user, but with the important distinction of being structure designed to trick the ASR system.

This kind of attack can utilise the property of human hearing ability to be limited to a frequency region from 20 Hz to 20 kHz, if the ASR in not already circumventing limiting the frequency range to this region. But even still attacking in the higher frequencies will be beneficial for inaccuracies of human hearing which will be unable perceive accurate changes in these regions.

Even still hidden attack may be masked to the user by employing knowledge from audio compression[5], a principle from MP3 encoding, where perturbations under certain magnitude thresholds in the frequency spectrum of signal may be masked other the present frequency stimuli as to be imperceptible to human.

A property of correlation based system such machine learning system is that is fits to anything present correlation even if it is audible to humans but is useful for accurate prediction. Adversarial attacks may exploit this fact to manipulate the machine learned ASR model predictions fully imperceivable to the human ear.

Some disclaimers for the real world (over the air) viability some of the attacks should be mentioned that of course without the knowledge of speaker characteristic where the attack is performed it may proof very difficult to produce successful attacks if audio fidelity is not sufficient. It should also be noted that for through streaming content with varying bit rates and where compression is employed, the attacks will of course be affected which will be make harder to a carry outside successful attacks. However at the same time compression allows some wiggle room for the adversarial signal to be audible as it easily can be disregarded as compression artifacts.

Another motivation to investigate legitimacy of audio signal queries in prevention of unintended queries from other audio source devices, however this will not remain the focus of this article, other methods has been proposed directly addressing these issues. However the ability to separate human sources from electroacoustic sources remains interesting and a key inspiration to the following investigation of this article.

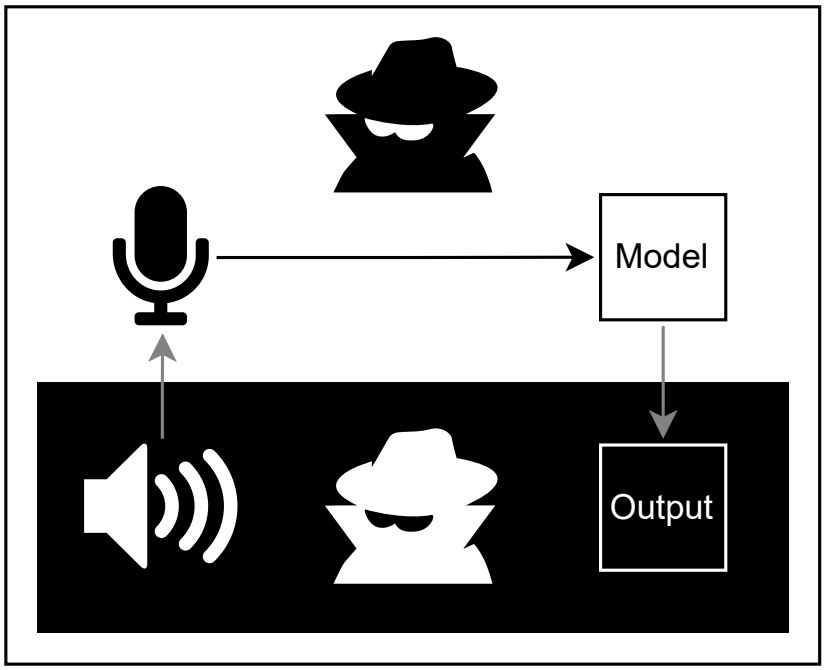
2 Problem Statement

Given that the adversarial attacks we focus on are reproduced non-human playback devices and the feature extraction methods commonly used for Automatic Speech Recognition (ASR) are generally engineered for optimal distinction of signals is the human audible frequency spectrum, it is not a given that is also well fitting for differentiating legitimate human speech from structured adversarial signals. Thus we will seek to survey a range of these filterbanks cepstral feature spaces in the context of ASR systems for validation of utterances as either adversarial or legitimate inputs.

3 Background

Human Audio Perception Adult humans fundamental voice frequency range is between 85Hz to 255Hz (85Hz to 180Hz for male and 165Hz to 255Hz for female). On top of the fundamental frequency there are harmonics of fundamental frequencies. Harmonics are whole multiplications of the fundamental frequency. If for instance the fundamental frequency is 100Hz then its second harmonic will be 200Hz, third harmonic is 300Hz and so on. Humans can hear roughly between 20Hz to 20KHz. The perception of sound is non-linear and you can better distinguish between low frequency sounds than high frequency sounds e.g. humans can clearly hear the difference between 100Hz and 200Hz but not between 15kHz and 15.1kHz.

Types of attacks In a white-box attack setting is an attacker has direct access to the entire parameterisation of the recognition model, where in a black-box setting the attacker only has access to the output of the recognition model, figure 1 serves as a reference for possible attack surfaces.



Figur 1: Illustrative example of black- and white-box attack surfaces, for white-box the model parameterisation is known and the black conversely. This black-box attacker can only query the recognition model and observe the output

while there may be distinctions in the attacks generated for each type of attack, ultimately considering the both attacks has structure the adversarial classification should be agnostic to the type of attack. Based on this knowledge we hypothesise that no noticeable differences will exhibited in adversarial attack classification performance between the two types of attacks, both will be investigated in our experiment for confirmation.

3.1 Cepstral coefficients

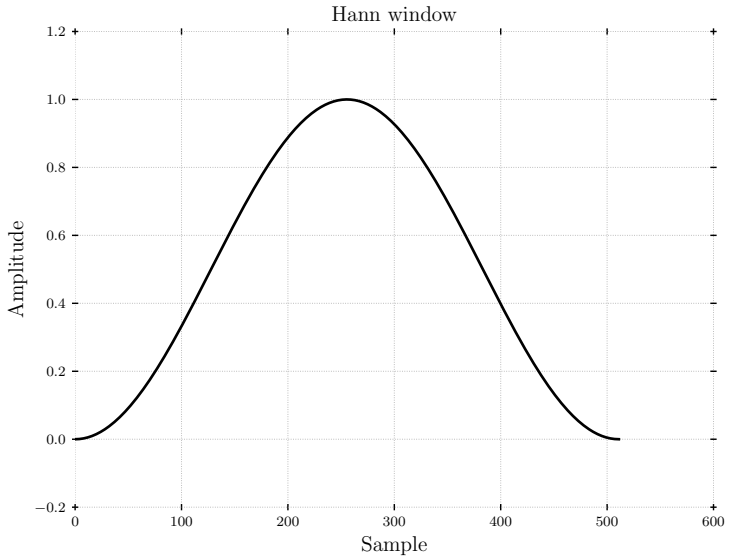
For the generation of the cepstral coefficients the following steps are taken.

Pre-Emphasis We perform pre-emphasis to amplify higher frequencies formants that are otherwise suppressed.

Framing Because audio is a non stationary process, to overcome this we can assume that the audio is a stationary process for short periods of time, divide the signal into short frames. Where each frame

will be the same size as the resulting Fast Fourier Transform (FFT). When letting the frames overlap some amount, the frames will have some correlation between them, which is desirable because at the edges of each frame, after applying a following window function, information is lost. Doing this operation leaves us with a framed audio matrix with the number of frames times the FFT size.

Windowing We convert the audio, which is still in the time domain, to frequency domain. To use the FFT we assume the audio to be periodic and continuous. By framing the signal we assured the audio to be periodic. To make the audio continuous, we apply a window function on every frame. If we wont do that, We will get high frequency distortions. To overcome this, we first need to apply a window function to the framed audio and then subsequently perform FFT. The window function assures that both ends of the signal will be close to zero. For our experiments we chose the Hanning window2.



Figur 2: Hanning window

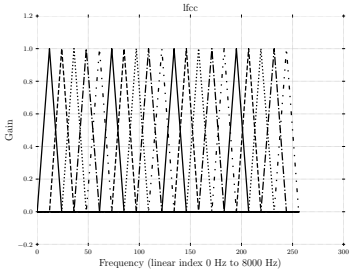
Discrete Fourier Transform and Power Spectrum Convert to frequency domain and compute power spectrum

Filter Banks AND Discrete Cosine Transform We use filterbank to divide the frequency band to sub-bands and then extracts the Cepstral Coefficients using Discrete Cosine Transform (DCT).

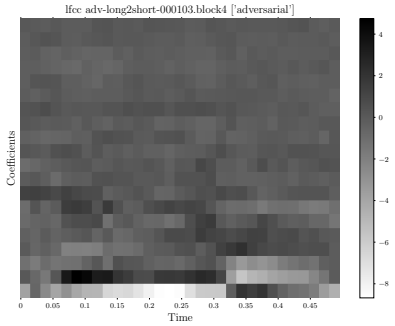
We compute a filterbank and then pass the framed audio through them. This will result in representation of how much the power is in each frequency band.

Linear / Uniform Filter Bank

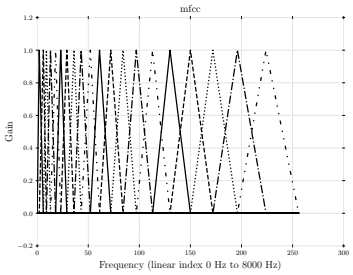
Mel-Scale Filter Bank A Mel-Scale Filter Bank is engineered for mimicking human auditory perception, discriminative at lower frequencies and less discriminative at higher frequencies, it is a bank of triangular filters equally spaced on the Mel scale. The Mel-scale is a scale of pitches judged by listeners to be equidistant one from another. On a linear-scale the spacing between the filters which grows exponentially with frequency.



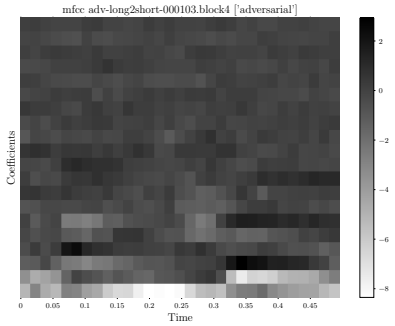
(a) Linear Filter Bank



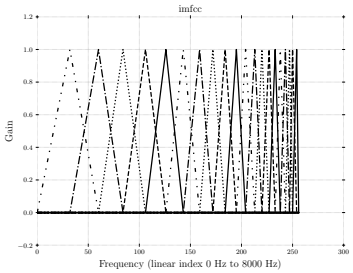
(b) Linear Frequency Cepstral Coefficients (LFCC) of adversarial attack, sample gfcc_adv-long2short-000103.block4



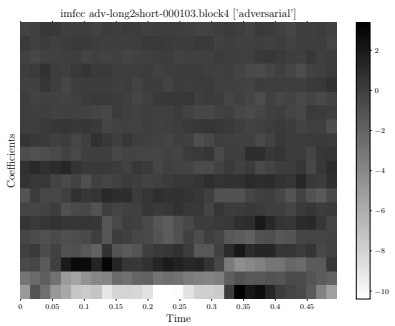
(a) Mel filterbank



(b) Mel Frequency Cepstral Coefficients (MFCC) of adversarial attack, sample gfcc_adv-long2short-000103.block4

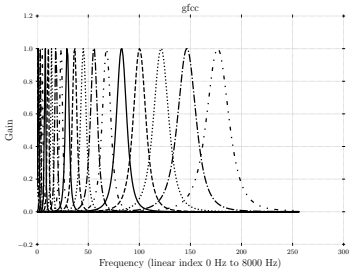


(a) Inverse Mel filterbank

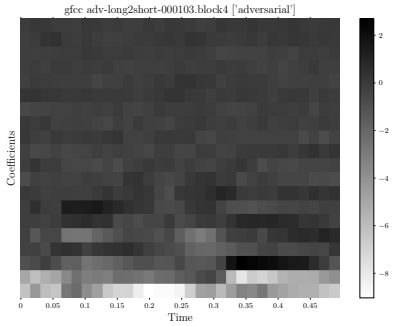


(b) Inverse Mel Frequency Cepstral Coefficients (IMFCC) of adversarial attack, sample gfcc_adv-long2short-000103.block4

Gamma-Tone Filterbank Equivalent Rectangular Bandwidth (ERB) scale from psychoacoustics, is an approximation of the bandwidths of the filters in the human auditory system. A bank of gammatone filters equally spaced on the ERB scale.



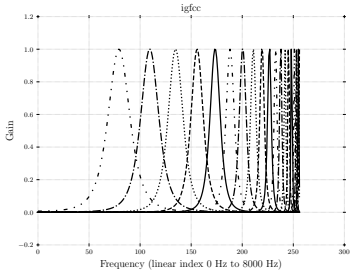
(a) Gamma-Tone Filterbank



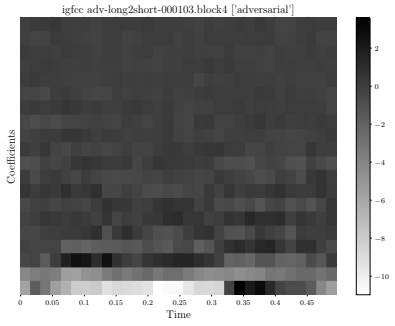
(b) Gamma-tone Frequency Cepstral Coefficients (GFCC) of adversarial attack, sample gfcc_adv-long2short-000103.block4

Figur 6: GFCC

The same idea of inverting the frequency bank from MFCC is also applicable to GFCC, resulting in Inverse Gamma-tone Frequency Cepstral Coefficients (IGFCC).



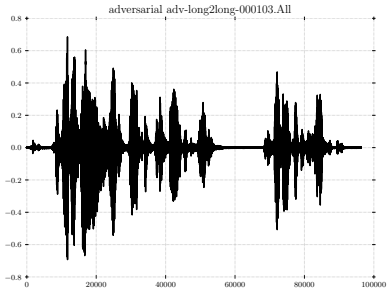
(a) Inverse Gamma-tone Filterbank



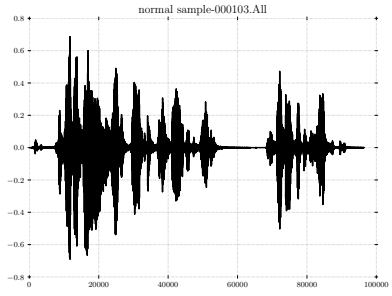
(b) Inverse Gamma-tone Frequency Cepstral Coefficients (IGFCC) of adversarial attack, sample gfcc_adv-long2short-000103.block4

Figur 7: IGFCC

Example transformation from raw waveform signal to mel frequency cepstral coefficients (MFCC) A sample is given as raw waveform8



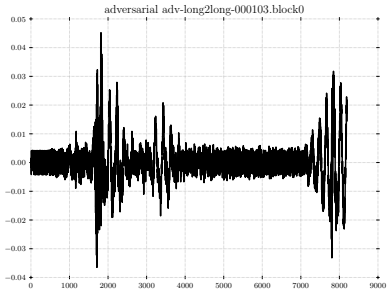
(a) Adversarial (category=1), adv_long2long attack



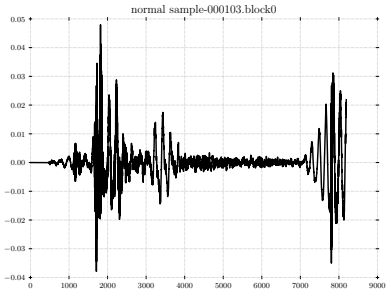
(b) Legimate (category=0)

Figur 8: Raw waveform signal of sample-000103

We chop it up in to smaller blocks9



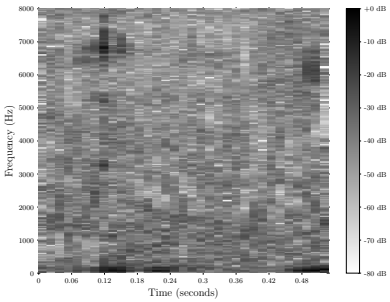
(a) Adversarial (category=1), adv_long2long attack



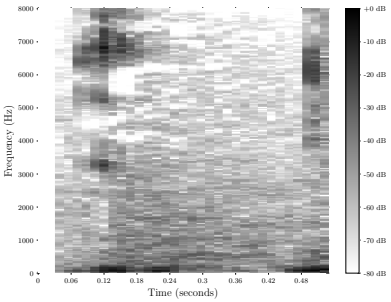
(b) Legimate (category=0)

Figur 9: raw waveform signal block of sample-000103

A fast Fourier transformation is applied to the block 10



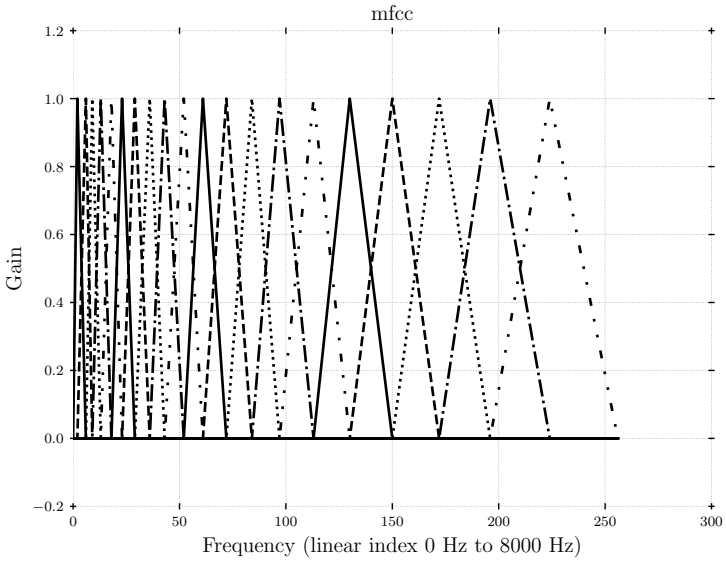
(a) Adversarial (category=1), adv_long2long attack



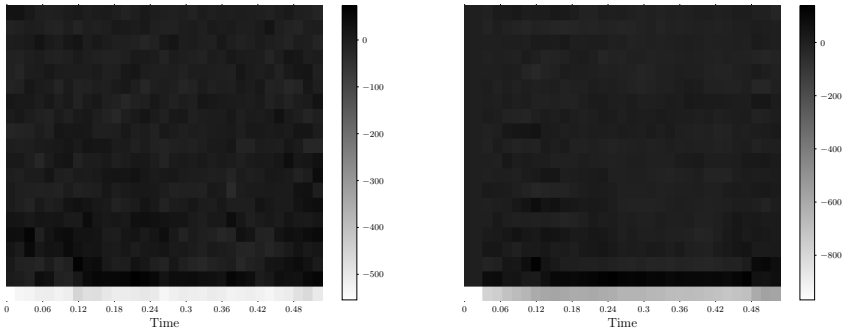
(b) Legimate (category=0)

Figur 10: STFT of sample-000103

And finally apply the mel filterbank11 and Discrete Cosine Transform12



Figur 11: mel filterbank



(a) Adversarial (category=1), adv_long2long attack (b) Legimate (category=0)

Figur 12: MFCC of sample-000103

4 Related Work

This contribution builds on the prior results from [4]. Expanding on the chosen the convolutional model, by augmenting it by varying the cepstral preprocessing rather than defaulting to the de facto speech recognition industry standard of using Mel-Frequency Cepstral Coefficients (MFCC).

5 Methodology

A main pillar of this survey is to provide reproducible results, thus a heavy emphasise is taken on producing comparable experiments and providing an exhaustive review of cepstral feature, model employment setting combinations. A full catalogue of the combinations and results is found in appendix B, appendix C and appendix D. For each experiment which repeated it 5 times varying the model weight initialisation, see appendix A, and we calculated a confidence interval for each evaluation metric.

The adversarial attack datasets we will be using is provided by [4]. We will refer to this article for detail on the generation of the adversarial attack. We will also be using the same CNN architecture, however an important change we made is that we chopped the examples into small block of 512 milliseconds and pass those through the CNN classifier model.

And the specific filterbanks variant we will consider are

- Linear frequency cepstral coefficients (LFCC)
- Mel frequency cepstral coefficients (MFCC)
- Inverse Mel frequency cepstral coefficients (IMFCC)
- Gamma-tone frequency cepstral coefficients (GFCC)
- Inverse Gamma-tone frequency cepstral coefficients (IGFCC)

Experiment settings We will testing out models the follow settings

- subject to white-box attacks
- subject to black-box attacks
- subject to both white-box and black-box attacks
- attacks in silence
- attacks in speech
- attacks in noise
- attacks in mixed noise, one noise left unknown

These experiments were chosen to represent a wide variety of authentic real-world configurations where attacks might occur. Common for all the experiment, the utilised datasets are split into a training set, a validation set and testing set with respectively 70%, 10%, 20%, truncated, untruncated or merged.

The datasets in numbers The A(white-box) and B(black-box) datasets provided by [4] has respectively 1470 and 2660 samples. Dataset A with 620 adversarial samples and 850 non-adversarial(legitimate) samples, and Dataset B with 1252 adversarial samples and 1408 legitimate samples, which should be a reasonably balanced dataset. For reference when our block chopping and splitting is performed these block samples amount to $(2737, 405, 649)$ samples for training, validation and tests sets for A dataset, with $(1187, 183, 272)$ adversarial attack samples and $(1550, 222, 377)$ legitimate samples.

The chosen hyperparameters is mostly borrowed from [4], however as mentioned there is significant change to the resulting models, instead of zero padding the samples to the longest sample like the authors, we instead chopped the samples into blocks of length 512. This block length remains consistent throughout all experiments.

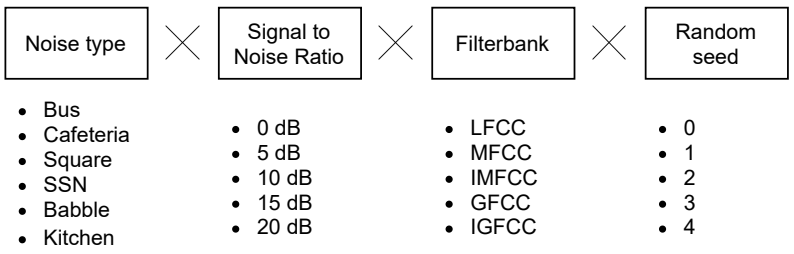
For the transformation of the labelling of the samples in the dataset into a numeral domain for the numerical optimisation procedure, we will let the categorical value 1 denote an adversarial example and the value 0 a legitimate example.

Subject to white-box attacks and black-box attacks separately The goal of this experiment is test how the attacks types affect the performance of the adversarial classifier. To make the experiment results comparable across attack type, we truncate the number of examples in each dataset to be the same and the lesser. We also test across attack-types to see if a model trained for adversarial classification on white-box attacks generalises to adversarial classification black-box attacks and vice-versa.

Subject to both white-box and black-box attacks In this experiment we evaluate how the models hold up against attacks of both white- and black-box nature at once, we train the models using both the untruncated training-split of the white- and block-box datasets. Similarly we validate and test respectively of the untruncated validation and test-splits of both attack modes. Similarly to the latter experiment we additionally test individually of the test set of the white-box attacks and black-box attacks separately.

Attacks in silence and Attacks in speech As mentioned earlier in the introduction, attacks might also occur without any legitimate human speech being present. This experiment sets out to compare performance in both scenarios, we extract. again to make the results comparable we truncate the size of the speech and silence dataset. Again we also test across speech and silence to see if a model trained for adversarial classification on white-box in speech attacks generalises to adversarial classification white-box attacks in silence and vice-versa.

Attacks In Noise models For this experiment we are evaluating the various filterbank models performances, when subjected to a range of noise types. With silence being present is the raw speech samples, in order to mix the noises at appropriate level and calculate and reasonable root mean square (RMS) value of the signal, we separately split the samples into speech and silence parts prior to splitting into the samples into the smaller blocks and mixing occurs. The noises is additively mixed at $0dB$, $5dB$, $10dB$, $15dB$, $20dB$ SNR ratios and sequential pieces of the noise-type is randomly sampled. Additionally each noise-type is also split into training, validation and test subparts to ensure leakage occurs between the mixed splits.



Figur 13: Combinations of models to be trained and evaluated

Figure 13 show the combinations of models in this experiment, this results is quite a number of experiments with numerous models, which is not fitting to embed into this article, we will refer the reader to the appendix to look up specific results.

The additive noise type we consider are

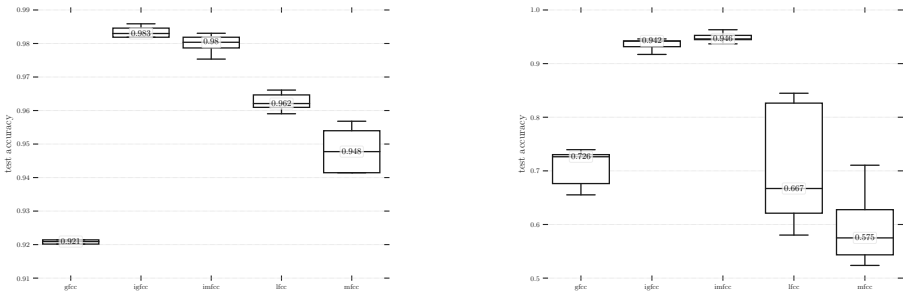
- cafeteria noise, $PCAFETER\_16k$
- bus passenger noise, $TBUS\_16k$
- city square noise, $SPSQUARE\_16k$
- kitchen noise, $DKITCHEN\_16k$
- speech shaped noise, ssn
- babble noise, *bbl*

$DKITCHEN\_16k$, $TBUS\_16k$, $SPSQUARE\_16k$, $PCAFETER\_16k$ are from [7], while ssn, *bbl* are from [2].

Attacks in mixed noise, mixed SNR, One Noise Left unknown Again using the noise types from the latter experiment, we observe how leaving a single noise type unseen during training time affect performance. We are using all of the rest of the noise types, with duplicate speech sample blocks but with differing noise types, for the training of the model. Note that there is still not overlap between the sample blocks in the training, validation and test sets. The noise chosen to be left out is the cafeteria noise, $PCAFETER\_16k$. Finally we also test across different SNR test set separately and all mixed.

6 Experiment Results

As the number of experiments performed is overwhelming to here include here, we will only include the accuracy figures each of experiments, apart from the in noise experiment where we will only put emphasis on a single finding, that contradicted from the rest.



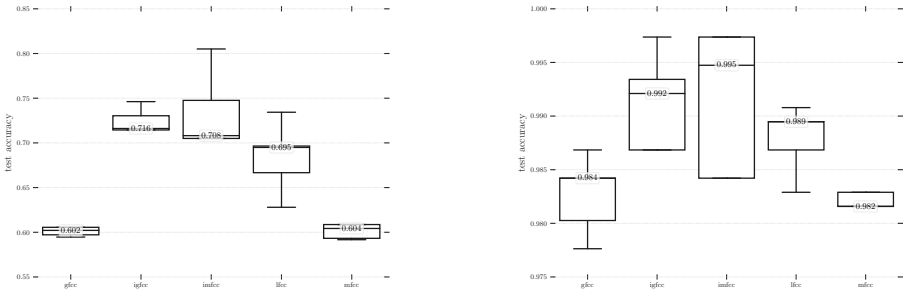
(a) Testing white-box attack accuracy with model trained on white-box attacks

(b) Testing black-box attack accuracy with model trained on white-box attacks

Figur 14: Model trained on white-box attacks

White-box (truncated dataset A) Figure 14 show the accuracy of the model trained on the truncated white-box attack dataset, see the follow appendices for the rest of the performance metrics.

- Precision recall curve, see appendix B.1
- Confusion Matrix, see appendix D.1
- Other metrics, see appendix C.2



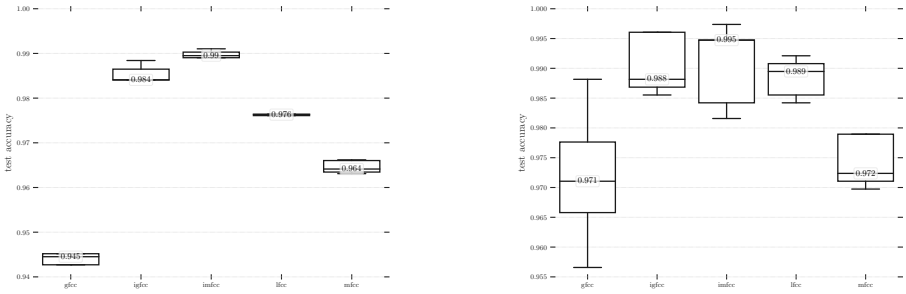
(a) Testing white-box attack accuracy with model trained on black-box attacks

(b) Testing black-box attack accuracy with model trained on black-box attacks

Figur 15: Model trained on black-box attacks

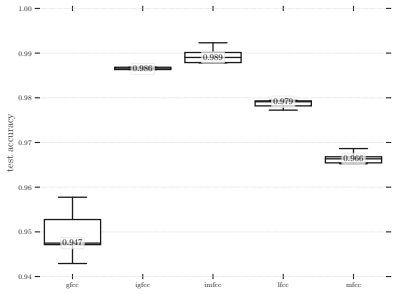
Black-box (truncated dataset B) Figure 15 show the accuracy of the model trained on the truncated black-box attack dataset, see the follow appendices for the rest of the performance metrics.

- Precision recall curve, see appendix B.2
- Confusion Matrix, see appendix D.2
- Other metrics, see appendix C.3



(a) Testing white-box attack accuracy with model trained on both white- and black-box attacks

(b) Testing black-box attack accuracy with model trained on both white- and black-box attacks



(c) Testing both white- and black-box attack accuracy with model trained on both white- and black-box attacks

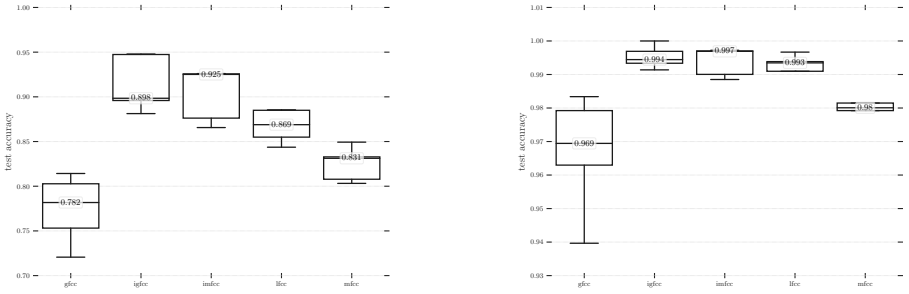
Figur 16: Model trained on black-box attacks

White and Black-box (datasets A and B merged) Figure 16 show the accuracy of the model trained on the merged white- and black-box attack datasets, see the follow appendices for the rest of the performance metrics.

- Precision recall curve, see appendix B.3
- Confusion Matrix, see appendix D.3
- Other metrics, see appendix C.1

The rest of the experiments are solely on the white-box adversarial attack dataset to limit the scope of the survey, also for producing comparable and comprehensible results varying only a single component of preprocessing was chosen.

We conducted an experiment, where the examples were split into speech parts and silence parts using the rVAD of [6]. Also in this case with truncated the dataset sizes to the smaller one of speech and silence split datasets.



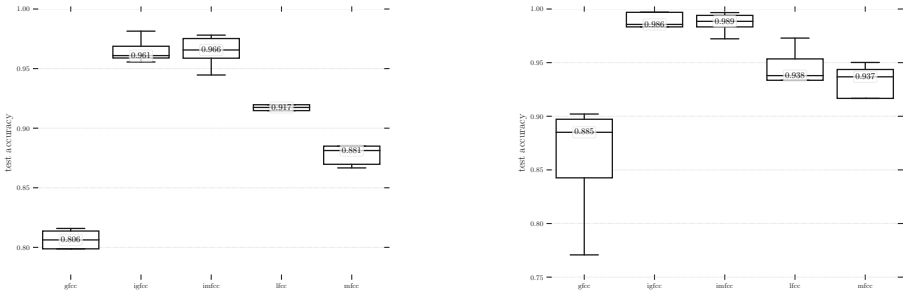
(a) Testing white-box in speech attack accuracy with model trained on in silence attacks

(b) Testing white-box in silence attack accuracy with model trained on in silence attacks

Figur 17: Model trained on white-box in silence attacks

Silence Split (truncated datasets A silence segments) Figure 17 show the accuracy of the model trained on the truncated in silence attack dataset, see the follow appendices for the rest of the performance metrics.

- Precision recall curve, see appendix B.4
- Confusion Matrix, see appendix D.4
- Other metrics, see appendix C.5



(a) Testing white-box in speech attack accuracy with model trained on in speech attacks

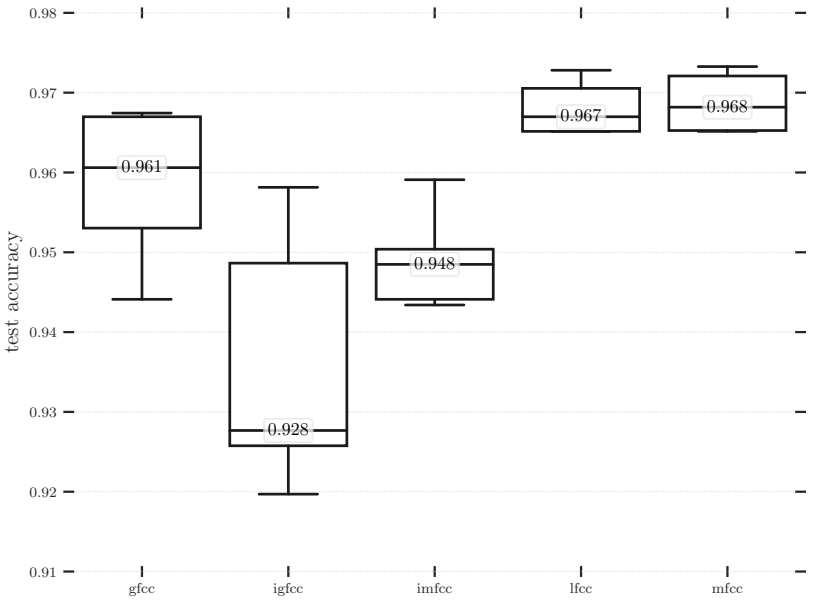
(b) Testing white-box in silence attack accuracy with model trained on in speech attacks

Figur 18: Model trained on white-box in speech attacks

Speech Split (truncated datasets A speech segments) Figure 18 show the accuracy of the model trained on the truncated in speech attack dataset, see the follow appendices for the rest of the performance metrics.

- Precision recall curve, see appendix B.5
- Confusion Matrix, see appendix D.5
- Other metrics, see appendix C.4

Individual noise types (datasets A with individual additive noise type injection) This experiment is the only one that has an obvious conflicting result to the rest of the experiments, only for the kitchen type noise is the inverse filterbank model performing worse. Already at a $20dB$ SNR the degrading of effects the kitchen noise type becomes overtaking and plummet the performance of the inverse filterbank models relative to the rest, see figure 19.



Figur 19: Testing white-box in silence attack accuracy with model trained on in speech attacks

- Precision recall curve, see appendix B.1
- Confusion Matrix, see appendix D.1
- Other metrics, see appendix C.2

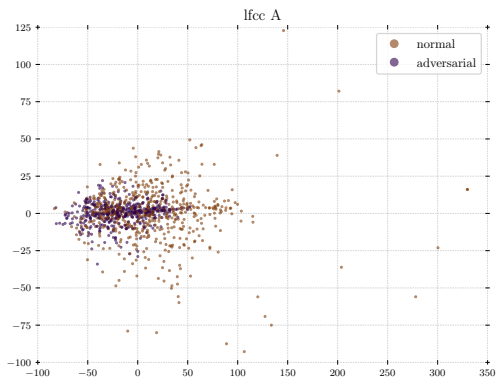
All noise types to cafeteria noise (datasets A with all individual additive noise type injection merge into a training set and validation set, test on cafeteria, no sample overlap) No hard indicative results were retrieved from this experiment.

- Precision recall curve, see appendix B.6
- Confusion Matrix, see appendix D.6
- Other metrics, see appendix C.6

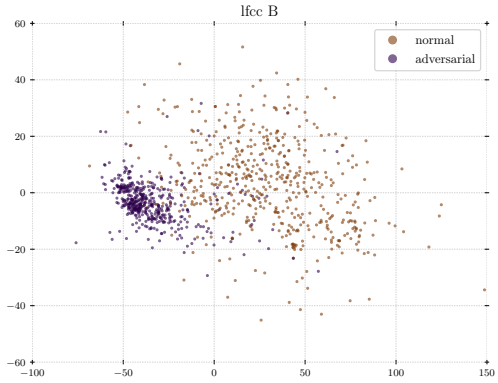
In general for majority of the experiments the consensus is that the inverse filterbank models are performing better than their counterparts.

7 Cepstral Space Projection

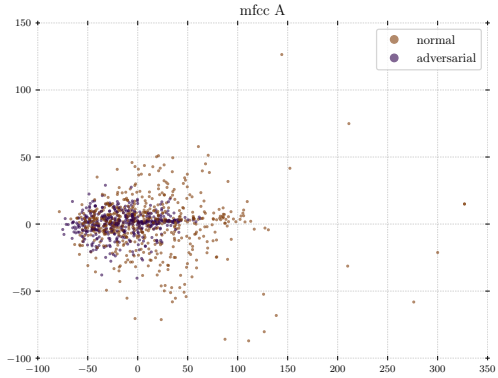
We perform a 2 dimensional projection on the cepstral representations, to visually inspect if any low dimensional structure is present. We limited the number of samples included in the projection to 1000 for both Principal Component Analysis (PCA), see figure 20 and 21, and T-distributed Stochastic Neighbor Embedding(TSNE)[3], see figure 22 and 23.



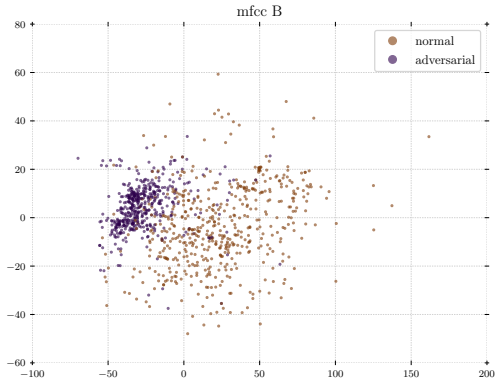
(a) 2 dimensional PCA projection for white-box lfcc features



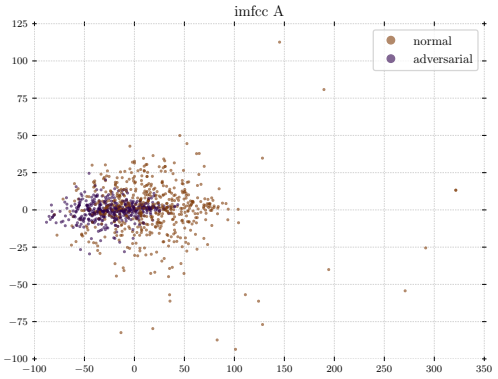
(b) 2 dimensional PCA projection for black-box lfcc features



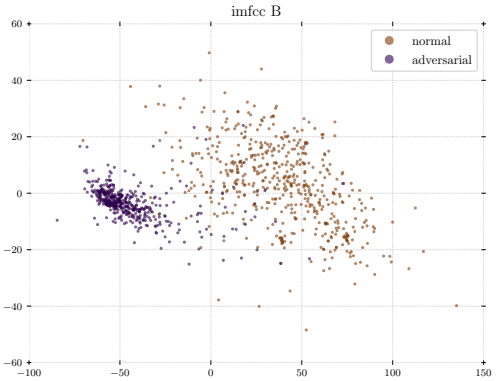
(c) 2 dimensional PCA projection for white-box mfcc features



(d) 2 dimensional PCA projection for black-box mfcc features

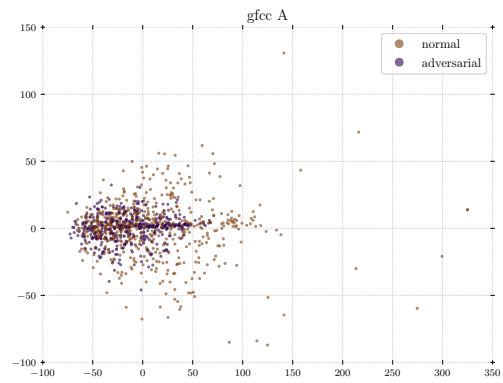


(e) 2 dimensional PCA projection for white-box imfcc features

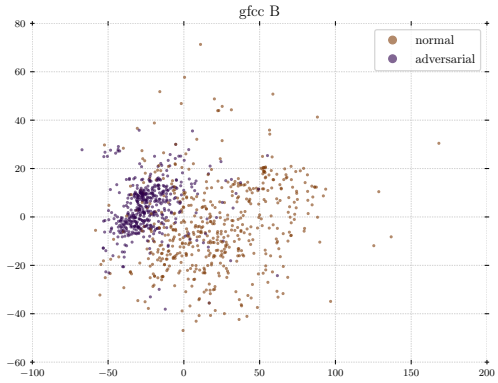


(f) 2 dimensional PCA projection for black-box imfcc features

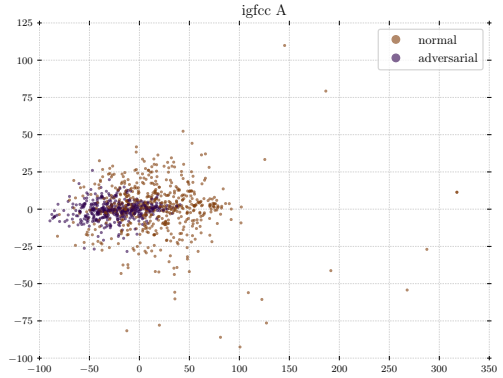
Figur 20: 2 dimensional PCA projection



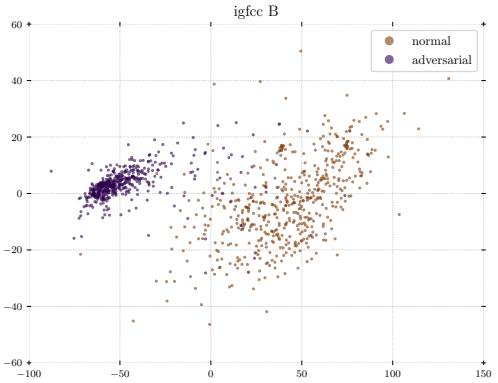
(a) 2 dimensional PCA projection for white-box gfcc features



(b) 2 dimensional PCA projection for black-box gfcc features

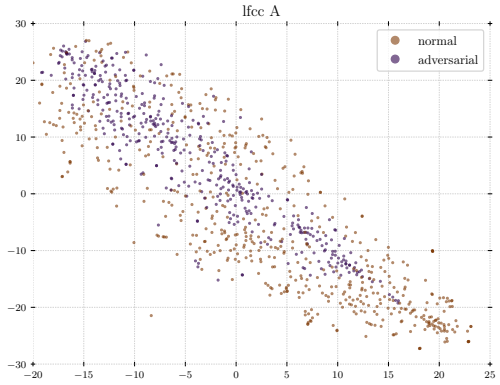


(c) 2 dimensional PCA projection for white-box igfcc features

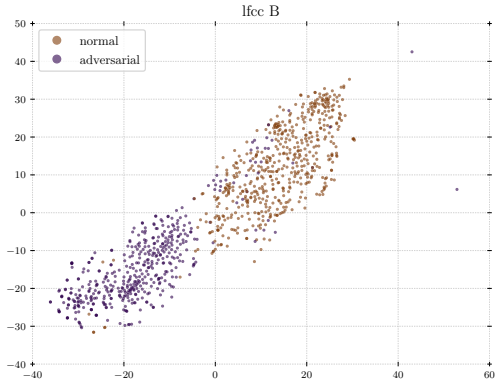


(d) 2 dimensional PCA projection for black-box igfcc features

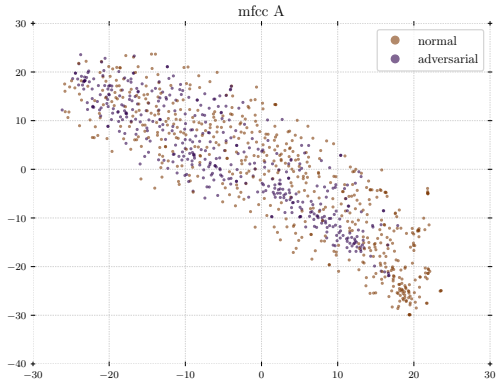
Figur 21: 2 dimensional PCA projection



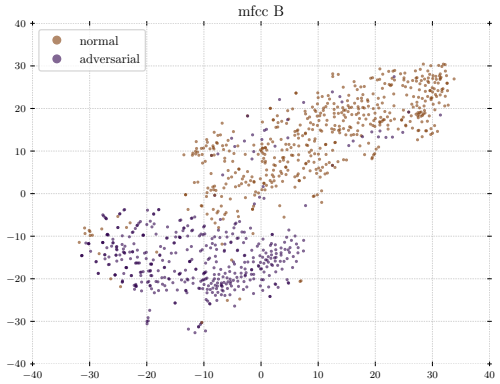
(a) 2 dimensional TSNE projection for white-box lfcc features



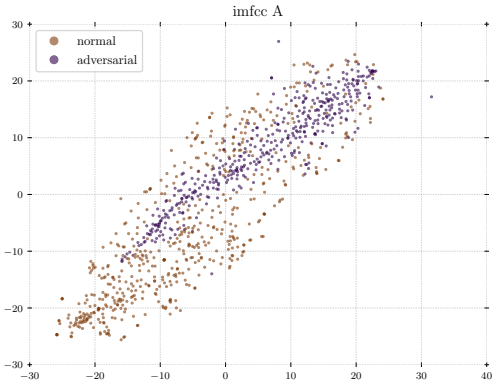
(b) 2 dimensional TSNE projection for black-box lfcc features



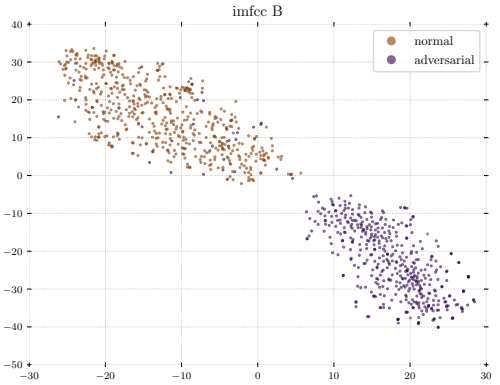
(c) 2 dimensional TSNE projection for white-box mfcc features



(d) 2 dimensional TSNE projection for black-box mfcc features

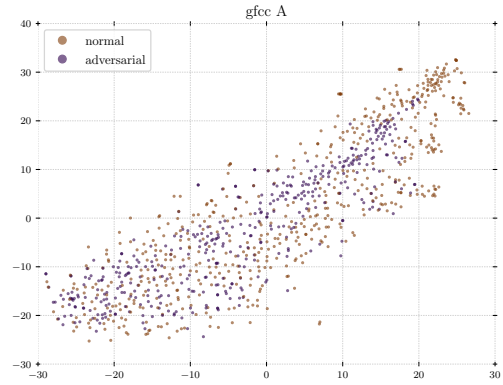


(e) 2 dimensional TSNE projection for white-box imfcc features

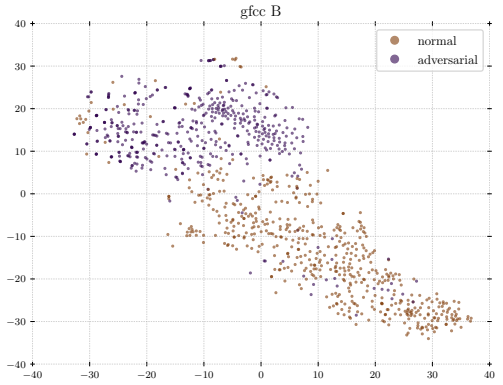


(f) 2 dimensional TSNE projection for black-box imfcc features

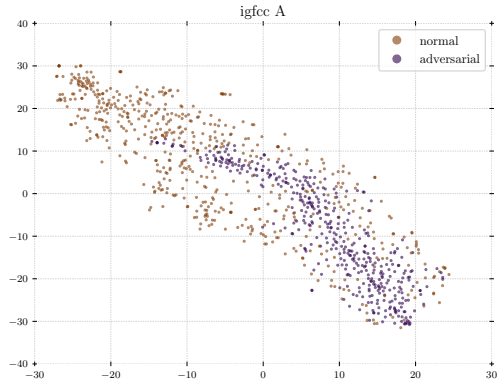
Figur 22: 2 dimensional TSNE projection



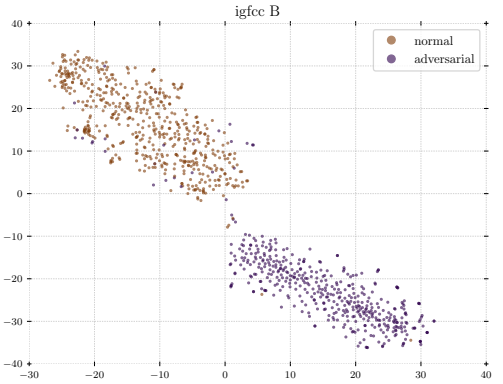
(a) 2 dimensional TSNE projection for white-box gfcc features



(b) 2 dimensional TSNE projection for black-box gfcc features



(c) 2 dimensional TSNE projection for white-box igfcc features



(d) 2 dimensional TSNE projection for black-box igfcc features

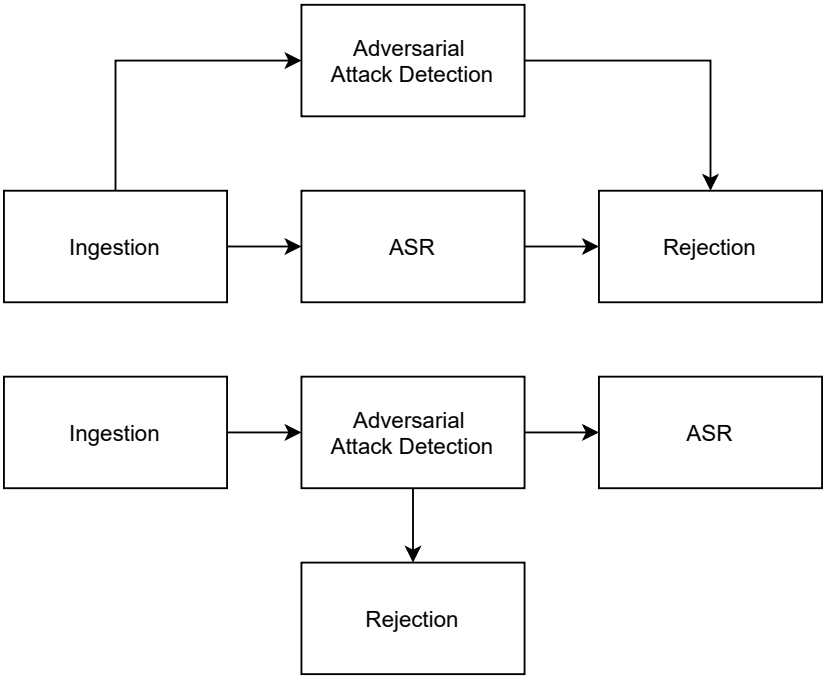
Figur 23: 2 dimensional TSNE projection

Surprisingly although there is only a small discrepancy between the white-box and black-box attack classifier performances, these projection algorithms seems to have a much easier finding dichotomic projections of the black-box attacks in most cepstral spaces but especially in the inverse filterbank features spaces.

8 Discussion

We observe tendency of the inverse filter bank model performing better on out evaluation, a likely explanation might lie in fundamentals psychoacoustics and the reason choosing to the Mel-scale filter bank in the first place. The resolution in the higher frequencies of sound waves are diminished thus any small scale distinguishing characteristics in signal the of adversarial attacks are weakened. This may unintentionally occlude the adversarial signal as regular noise to a adversarial attack detection model by minimising finely detailed frequency variations.

Now for the contradicting result with the "In Noise, Kitchen Noise Type"experiment, it will need further investigation, but a likely explanation is that the high pitch nature of the noise type is messing with the models capability of selecting out adversarial attacks from the mainly cutlery on plates noises present. While this finding and likely explanation might not be surprising it is worthwhile considering for adversarial attack counter measures.



Figur 24: Suggested placement of the detection model in ASR system architectures

Use Case Implementation For a use case of a model capable detecting adversarial attacks like the ones presented, an obvious setting is in ASR systems. The adversarial classification may be performed in incoming frame in parallel or prior to forwarding the audio signal the ASR model. Some consideration should be taken in terms of selecting different feature transformations for the adversarial transformation and the ASR model as it will inevitably increase computation time and resource usage.

9 Conclusion

We have presented a range experiments all indicating a tendency of the Inverse Filter bank models outperforming their counterparts and the linearly spaced filterbank model. Consistently through near to all trials we observe that IMFCC and IGFCC feature models are showing higher precision and recall measures than the rest of cepstral features spaces, indicating that in order to better detect adversarial attacks one should prefer the inverse filter banks over their counterparts. Only footnote to attach is that of the Kitchen noise type, which consist mainly of high pitch noise does severely affect this conclusion in a direction of further nuance to be investigated, Specific what frequency components are reasonable for this behaviour, as it almost flips the results in the favour of the regular non-inverted filterbank.

Acknowledgments and Disclosure of Funding

We wish to thank Saeid Samizade et al. for providing the generated adversarial examples, based on the Mozilla Common Voice dataset[1] and Google Speech Commands dataset[8], thank you to the authors of these dataset for making them publicly available as well.

Earlier first steps has been made to sneak out conclusive results utilising this knowledge has been done by students at Aalborg University, Amalie V. Petersen, Jacob T. Lassen and Sebastian B. Schiøler.

I Christian Heider Nielsen would also personally like to thank Zheng-Hua Tan for the guidance, conversations and patience.

References

[1] *Common Voice by Mozilla*. da. URL: https://commonvoice.mozilla.org/ (bes. 22.04.2021).

[2] Morten Kolbœk, Zheng-Hua Tan og Jesper Jensen. "Speech enhancement using Long Short-Term Memory based recurrent Neural Networks for noise robust Speaker Verification". I: *2016 IEEE Spoken Language Technology Workshop (SLT)*. 2016, s. 305–311. DOI: 10.1109/SLT.2016.7846281.

[3] Laurens van der Maaten og Geoffrey Hinton. "Visualizing Data using t-SNE". I: *Journal of Machine Learning Research* 9 (2008), s. 2579–2605. URL: http://www.jmlr.org/papers/v9/vandermaaten08a.html.

[4] Saeid Samizade m.fl. "Adversarial Example Detection by Classification for Deep Speech Recognition". I: *arXiv:1910.10013 [cs, eess, stat]* (okt. 2019). arXiv: 1910.10013 version: 1. URL: http://arxiv.org/abs/1910.10013 (bes. 18.03.2021).

[5] Lea Schönherr m.fl. "Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding". I: *arXiv:1808.05665 [cs, eess]* (okt. 2018). arXiv: 1808.05665. URL: http://arxiv.org/abs/1808.05665 (bes. 15.04.2021).

[6] Zheng-Hua Tan, Achintya kr Sarkar og Najim Dehak. "rVAD: An unsupervised segment-based robust voice activity detection method". I: *Computer Speech & Language* 59 (2020), s. 1–21. ISSN: 0885-2308. DOI: https://doi.org/10.1016/j.csl.2019.06.005. URL: https://www.sciencedirect.com/science/article/pii/S0885230819300920.

[7] Joachim Thiemann, Nobutaka Ito og Emmanuel Vincent. *DEMAND: a collection of multi-channel recordings of acoustic noise in diverse environments*. Supported by Inria under the Associate Team Program VERSAMUS. Jun. 2013. DOI: 10.5281/zenodo.1227121.

[8] Pete Warden. *Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition*. 2018. arXiv: 1804.03209 [cs.CL].

Appendices

A Model Hyper Parameters

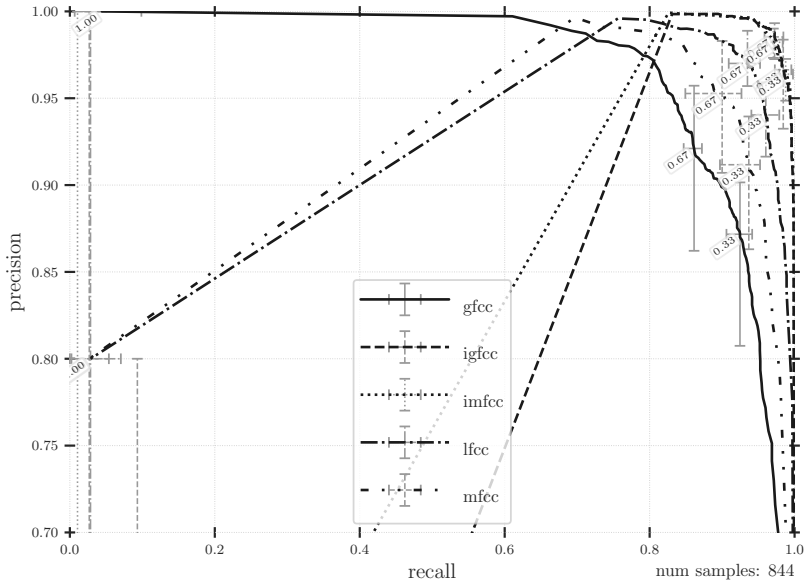
If not listed below it is left to default, refer to the code for missing items.

- $blockwindowsizems = 512$
- $blockwindowstepsizems = 512$
- $cepstralwindowlengthms = 32$
- $nfcc = 20$
- $nfft = 512$
- $samplerate = 16000$
- $valinterval = 1$
- $numruns = 5$
- $batchsize = 64$
- $numepochs = 99$
- $optimiser = Adam$
- $lr = 6e - 4$
- $betas = (0.9, 0.999)$
- $trainpercentage = 0.7$
- $validationpercentage = 0.1$
- $testpercentage = 0.2$
- $projectionnumsamples = 1000$
- $tsnelearningrate = 1000$
- $tnseperplexity = 50$
- $snrratios = [0, 5, 10, 15, 20]$
- $randomseeds = [0, 1, 2, 3, 4]$

B Model Results PR

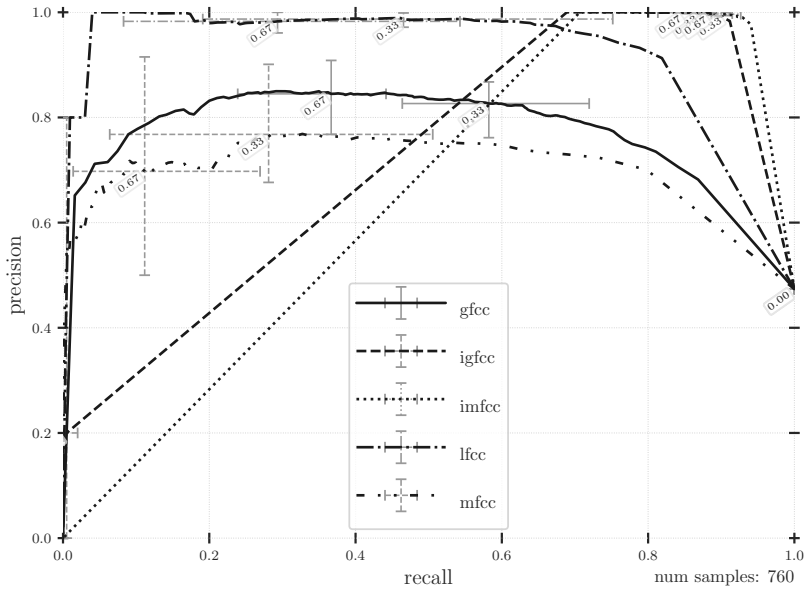
B.1 truncated a model a a

B.1.1 test a



Figur 25: Precision-Recall curve for truncated a model a a, test a

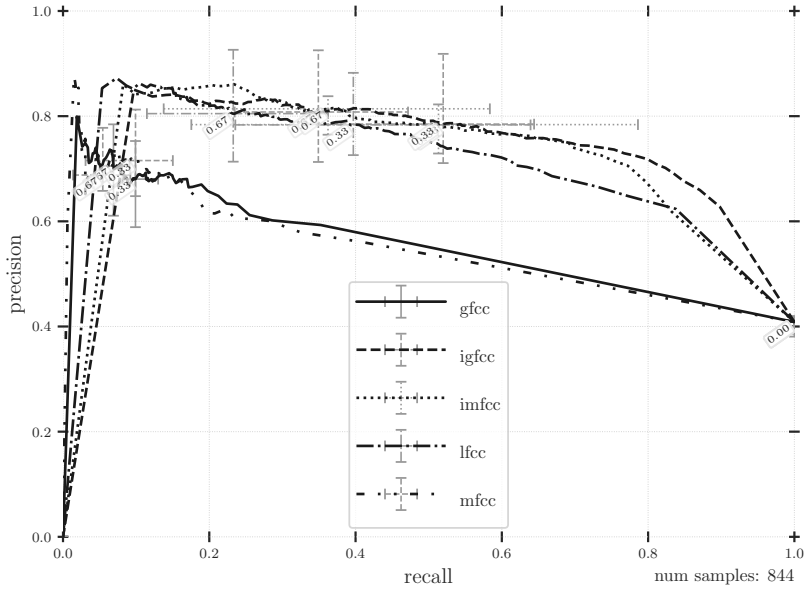
B.1.2 test b



Figur 26: Precision-Recall curve for truncated a model a a, test b

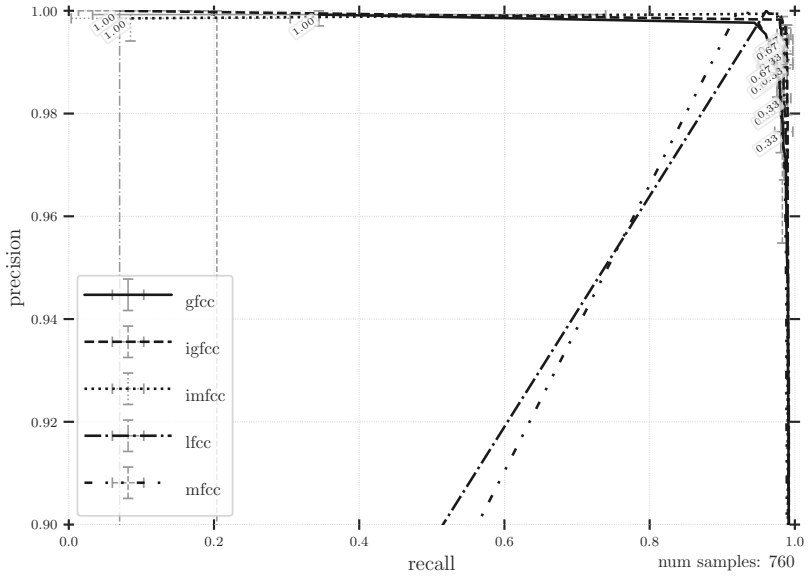
B.2 truncated b model b b

B.2.1 test a



Figur 27: Precision-Recall curve for truncated b model b b, test a

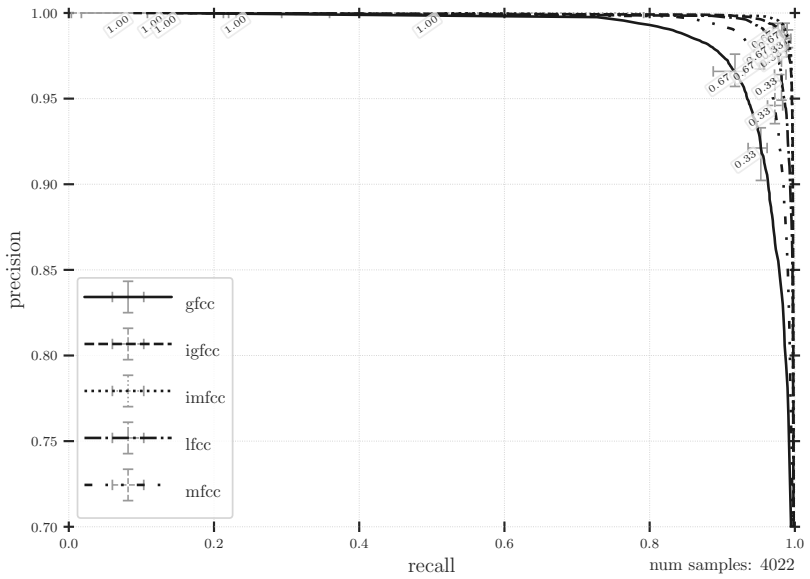
B.2.2 test b



Figur 28: Precision-Recall curve for truncated b model b b, test b

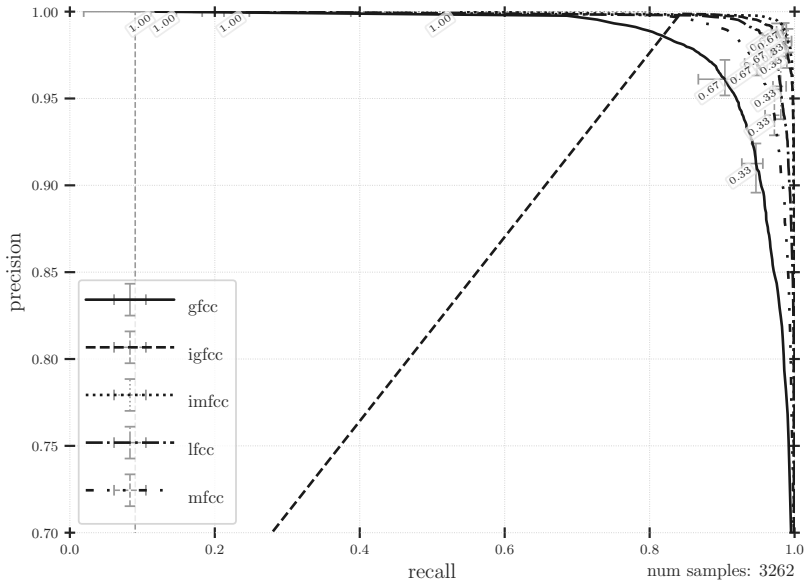
B.3 merge ab test ab model ab ab

B.3.1 test a b



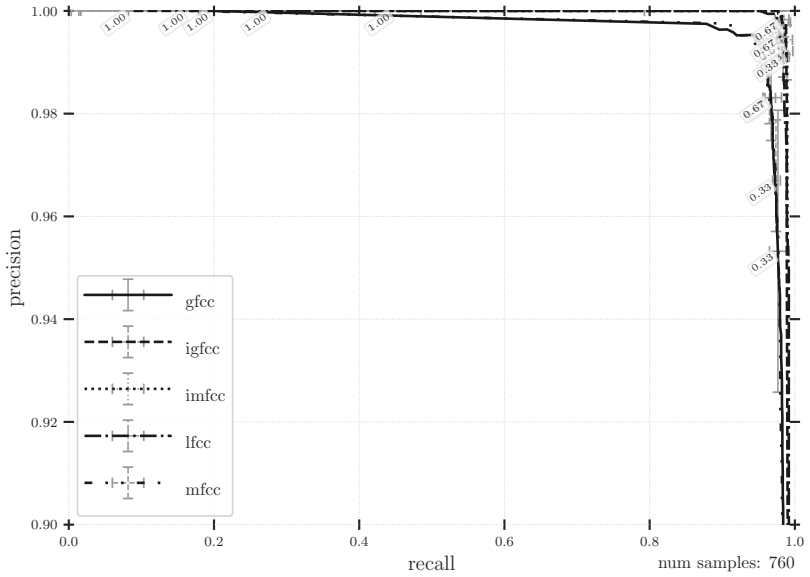
Figur 29: Precision-Recall curve for merge ab test ab model ab ab, test a b

B.3.2 test a



Figur 30: Precision-Recall curve for merge ab test ab model ab ab, test a

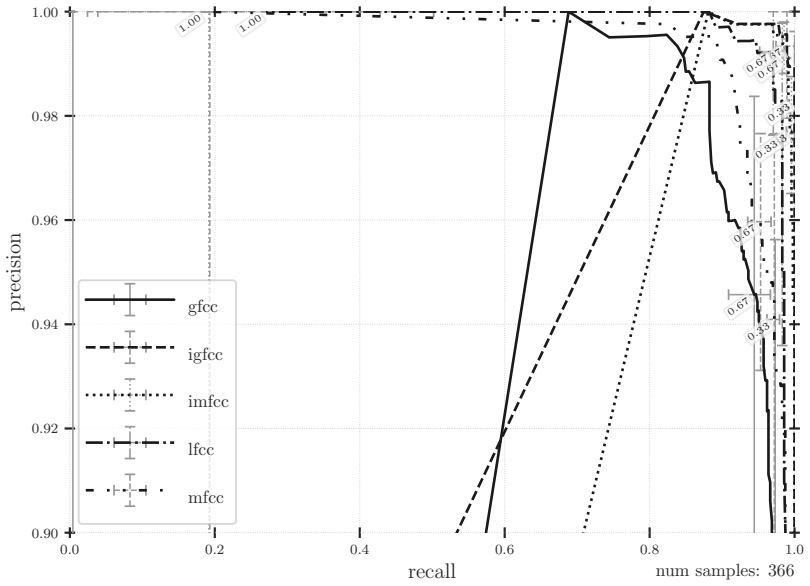
B.3.3 test b



Figur 31: Precision-Recall curve for merge ab test ab model ab ab, test b

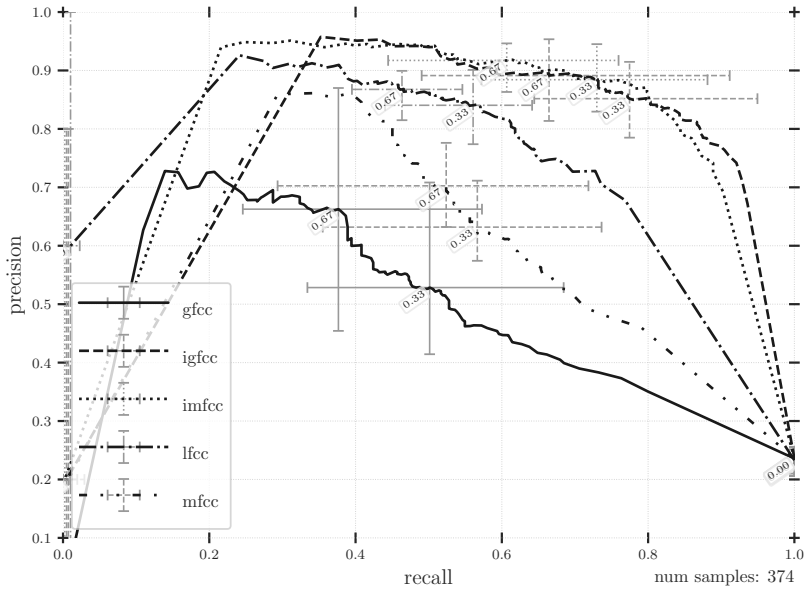
B.4 truncated a silence model silence silence

B.4.1 test silence



Figur 32: Precision-Recall curve for truncated a silence model silence silence, test silence

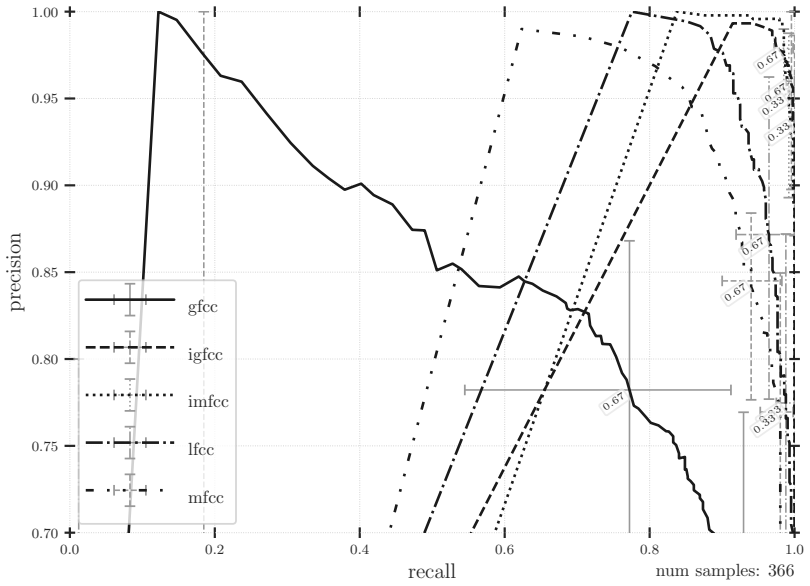
B.4.2 test speech



Figur 33: Precision-Recall curve for truncated a silence model silence silence, test speech

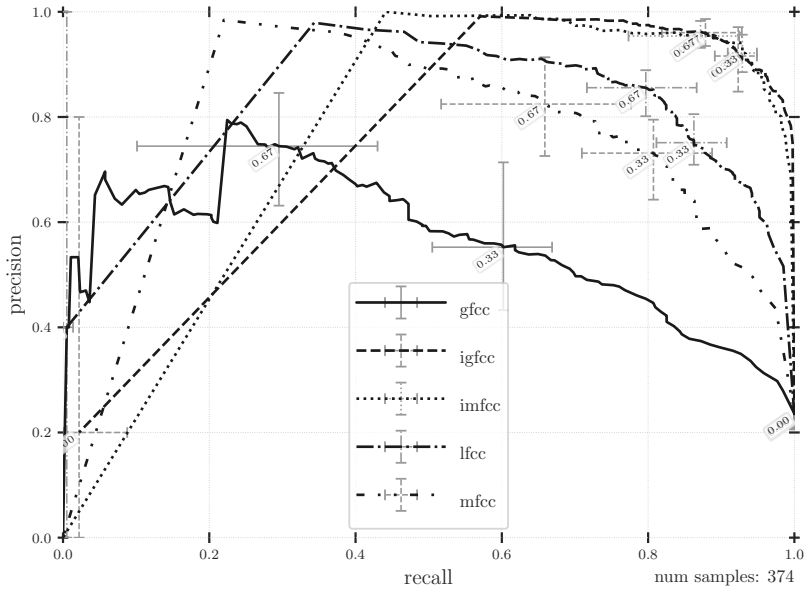
B.5 truncated a speech model speech speech

B.5.1 test silence



Figur 34: Precision-Recall curve for truncated a speech model speech speech, test silence

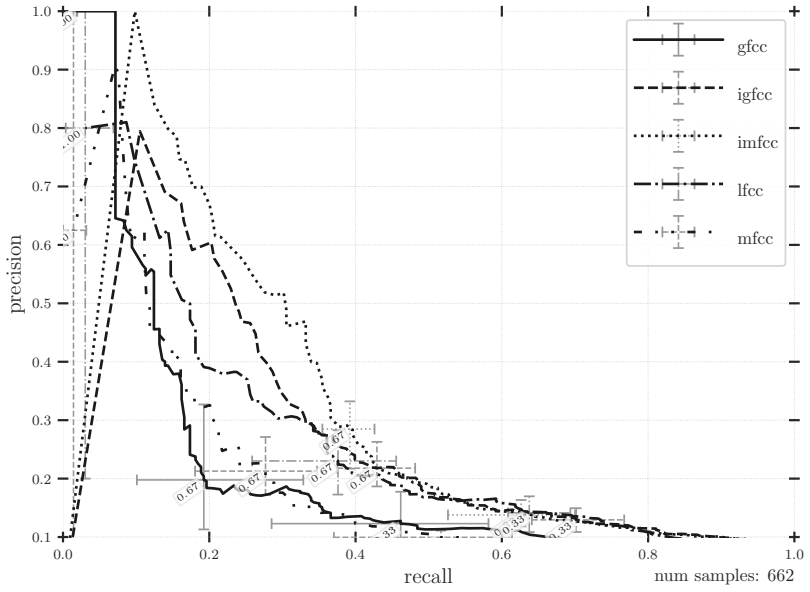
B.5.2 test speech



Figur 35: Precision-Recall curve for truncated a speech model speech speech, test speech

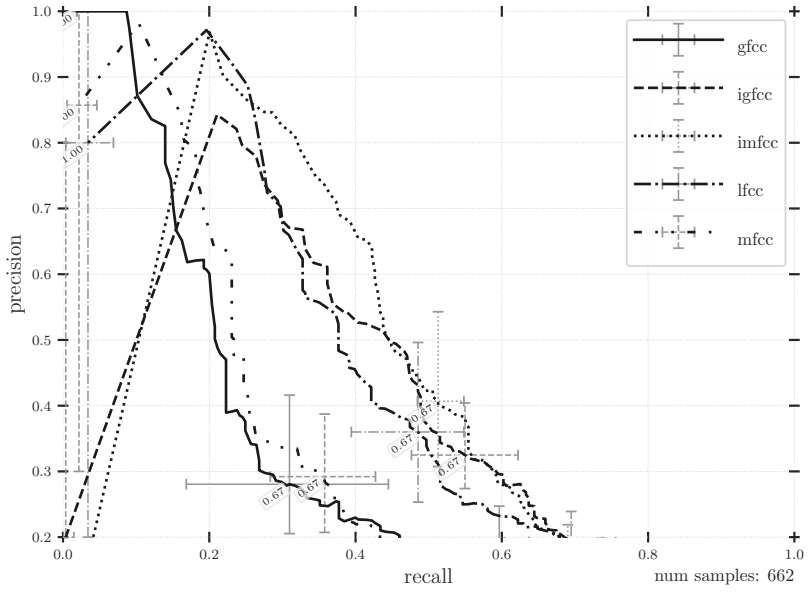
B.6 noises all snr to pcafeter model all noise snr all noise snr

B.6.1 test pcafeter snr 0db



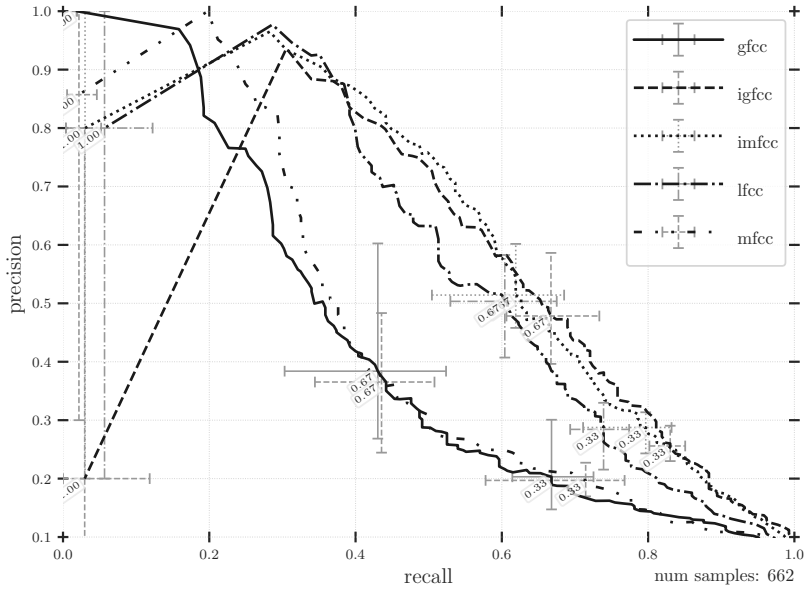
Figur 36: Precision-Recall curve for noises all snr to pcafeter model all noise snr all noise snr, test pcafeter snr 0db

B.6.2 test pcafeter snr 5db



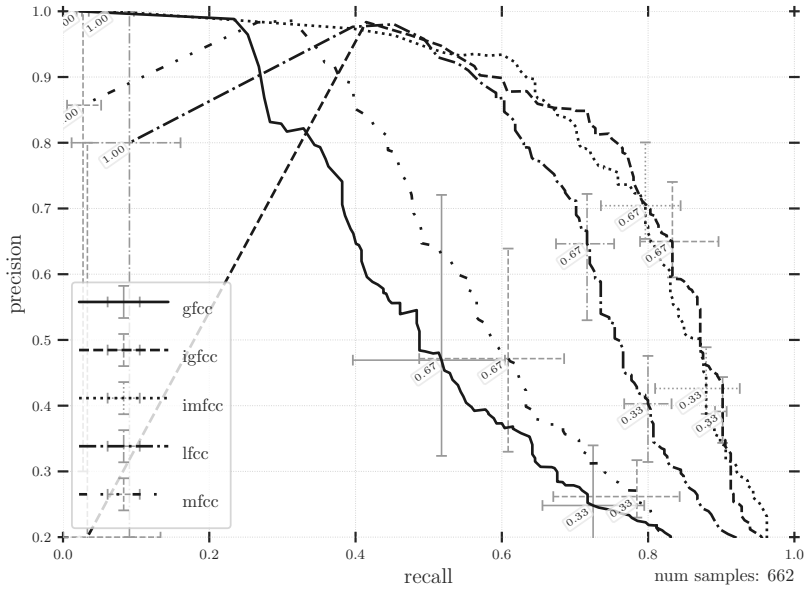
Figur 37: Precision-Recall curve for noises all snr to pcafeter model all noise snr all noise snr, test pcafeter snr 5db

B.6.3 test pcafeter snr 10db



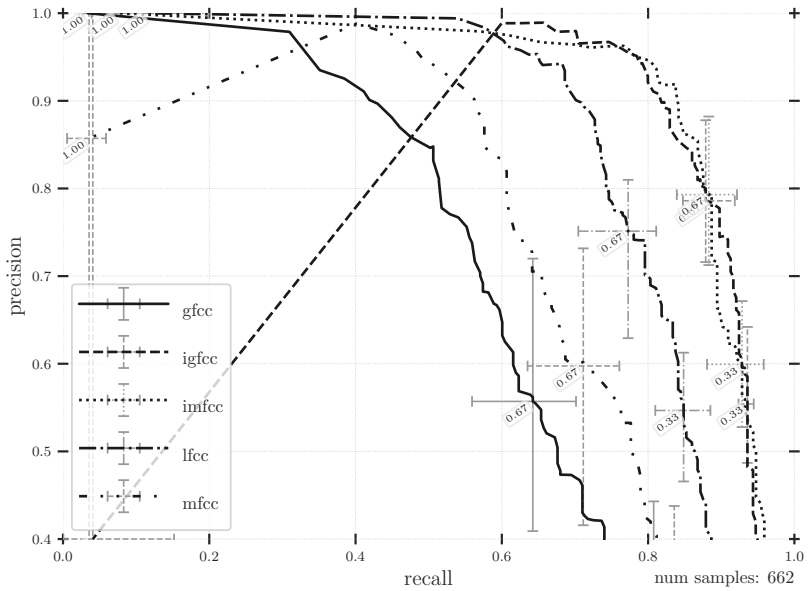
Figur 38: Precision-Recall curve for noises all snr to pcafeter model all noise snr all noise snr, test pcafeter snr 10db

B.6.4 test pcafeter snr 15db



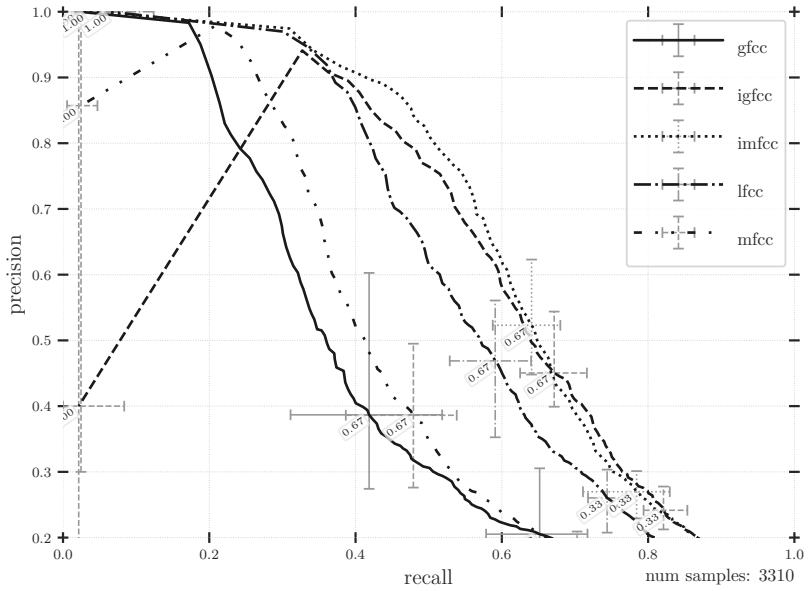
Figur 39: Precision-Recall curve for noises all snr to pcafeter model all noise snr all noise snr, test pcafeter snr 15db

B.6.5 test pcafeter snr 20db



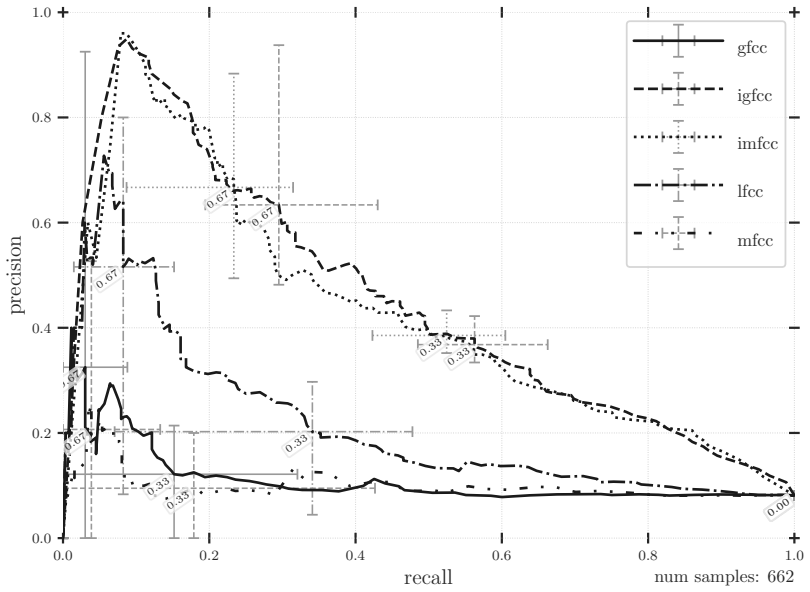
Figur 40: Precision-Recall curve for noises all snr to pcafeter model all noise snr all noise snr, test pcafeter snr 20db

B.6.6 test pcafeter snr 0db pcafeter snr 5db pcafeter snr 10db pcafeter snr 15db pcafeter snr 20db



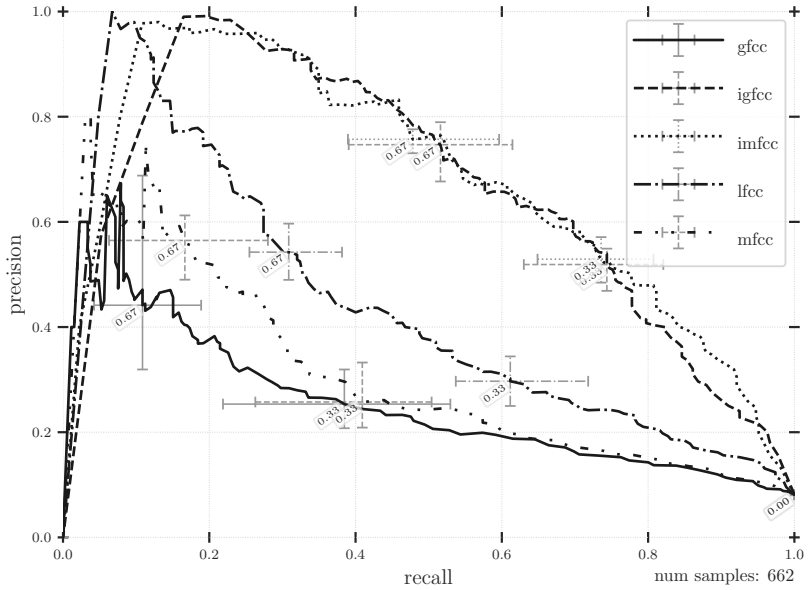
Figur 41: Precision-Recall curve for noises all snr to pcafeter model all noise snr all noise snr, test pcafeter snr 0db pcafeter snr 5db pcafeter snr 10db pcafeter snr 15db pcafeter snr 20db

B.7 bbl morten



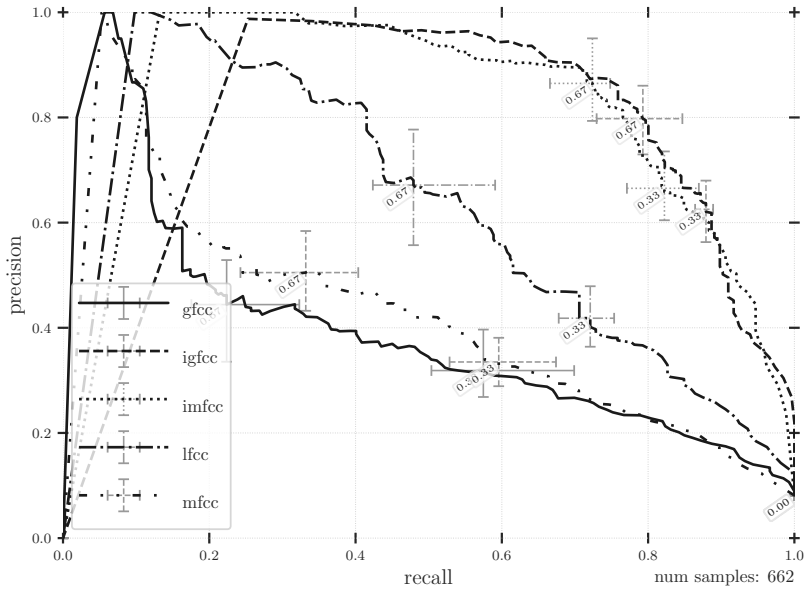
Figur 42: Precision-Recall curve for bbl morten snr 0db

B.8 bbl morten



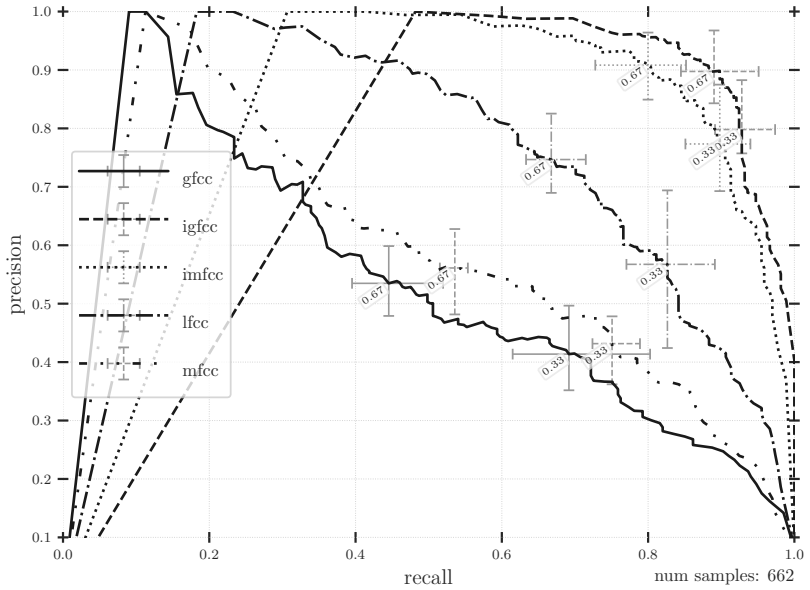
Figur 43: Precision-Recall curve for bbl morten snr 5db

B.9 bbl morten



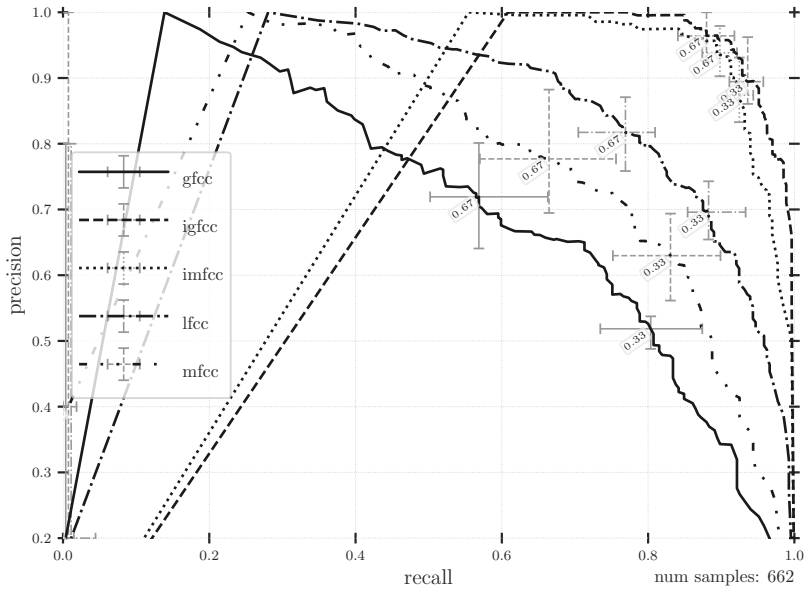
Figur 44: Precision-Recall curve for bbl morten snr 10db

B.10 bbl morten



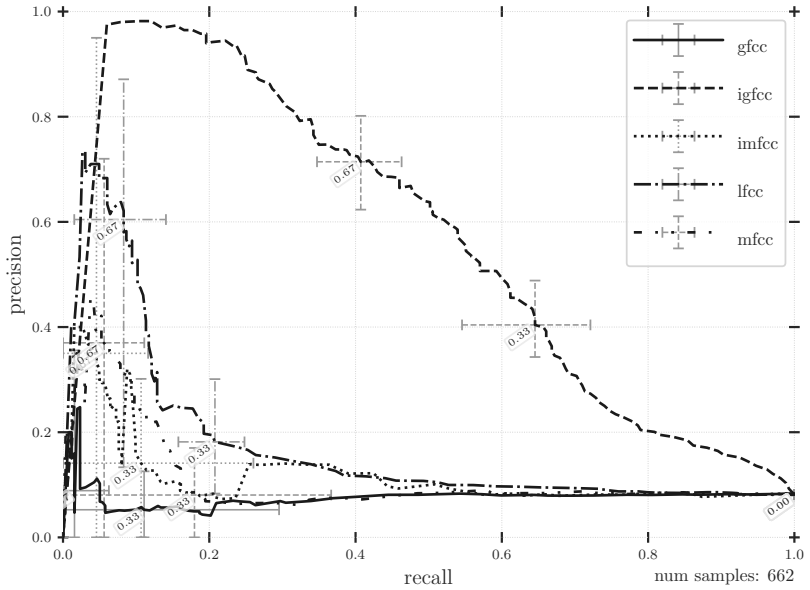
Figur 45: Precision-Recall curve for bbl morten snr 15db

B.11 bbl morten



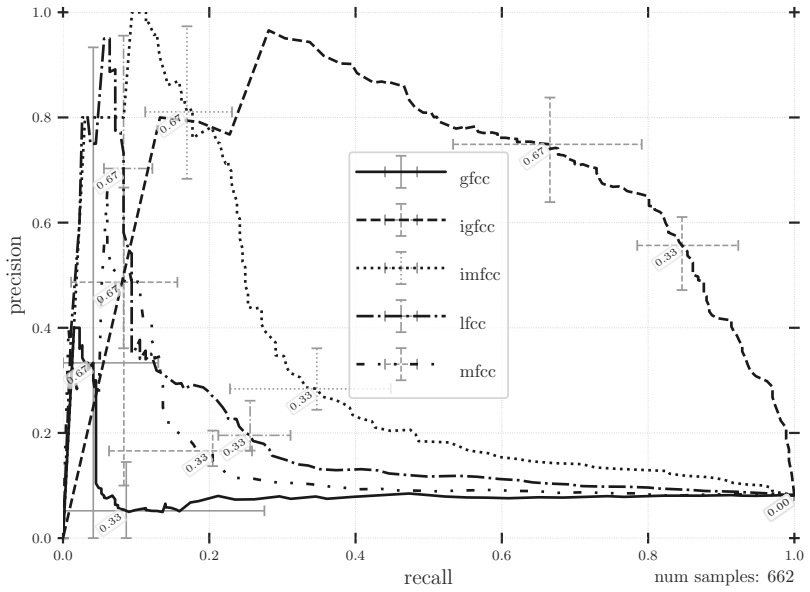
Figur 46: Precision-Recall curve for bbl morten snr 20db

B.12 ssn morten



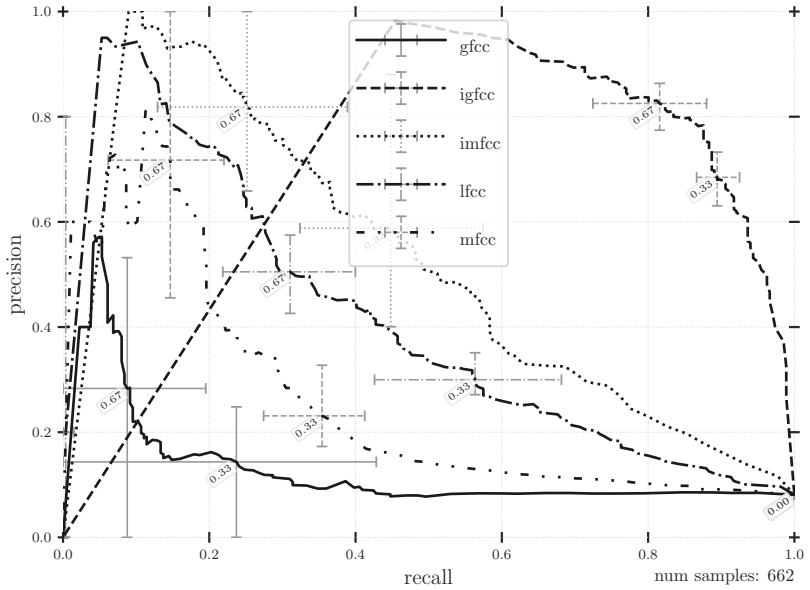
Figur 47: Precision-Recall curve for ssn morten snr 0db

B.13 ssn morten



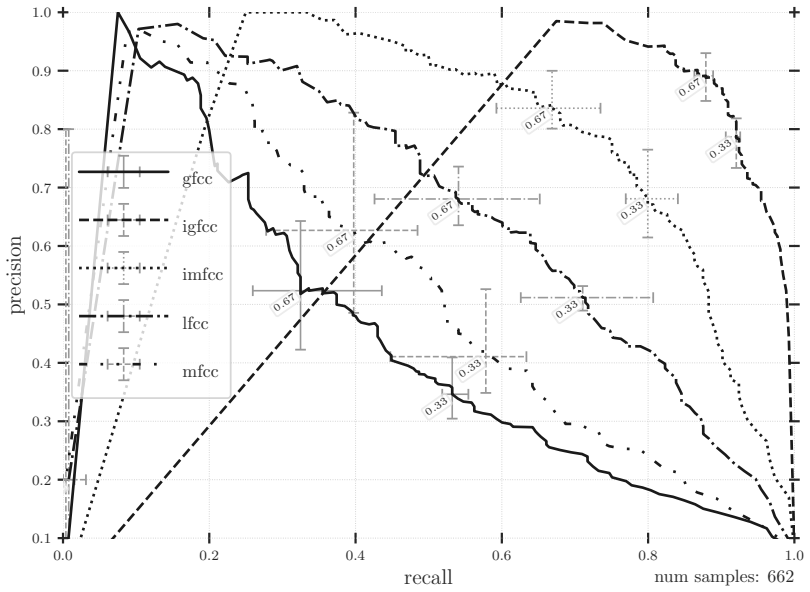
Figur 48: Precision-Recall curve for ssn morten snr 5db

B.14 ssn morten



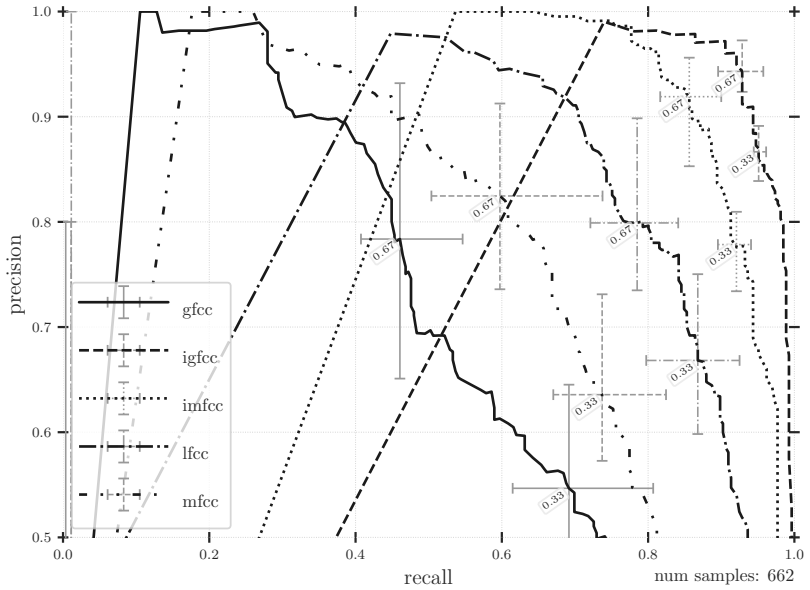
Figur 49: Precision-Recall curve for ssn morten snr 10db

B.15 ssn morten



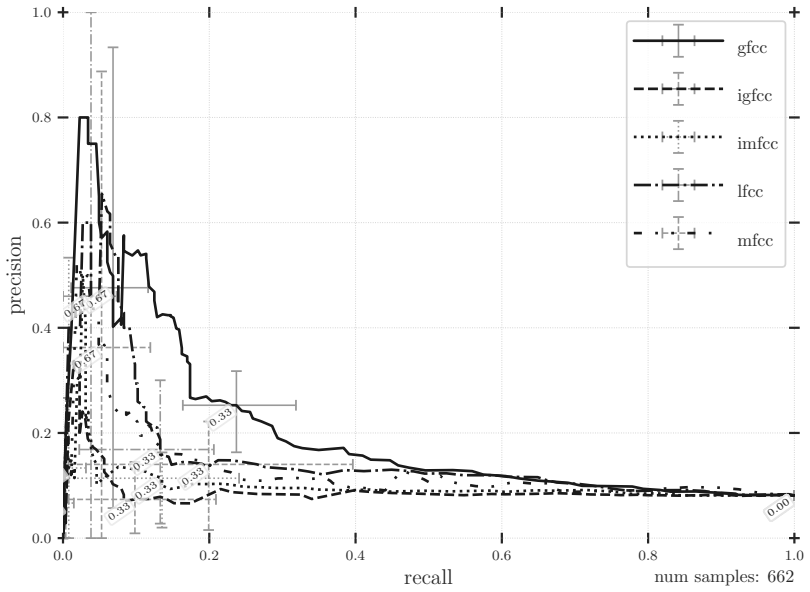
Figur 50: Precision-Recall curve for ssn morten snr 15db

B.16 ssn morten



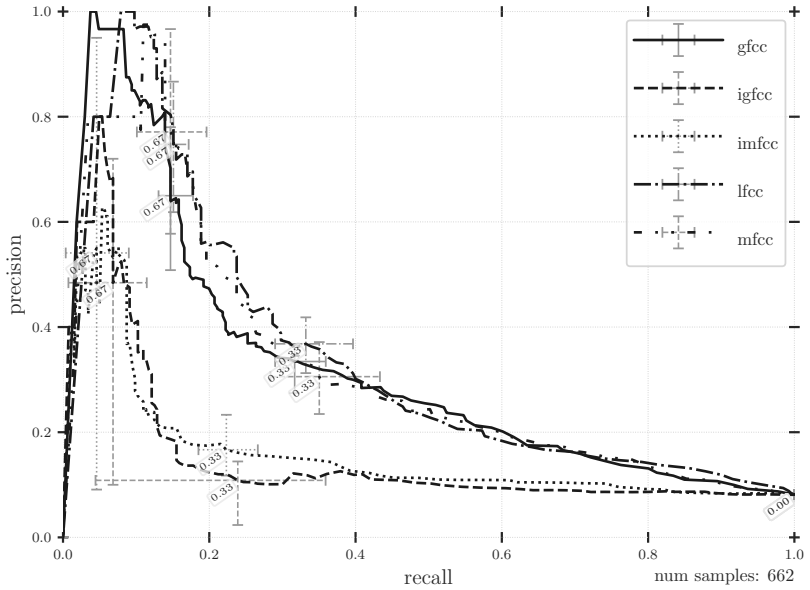
Figur 51: Precision-Recall curve for ssn morten snr 20db

B.17 dkitchen



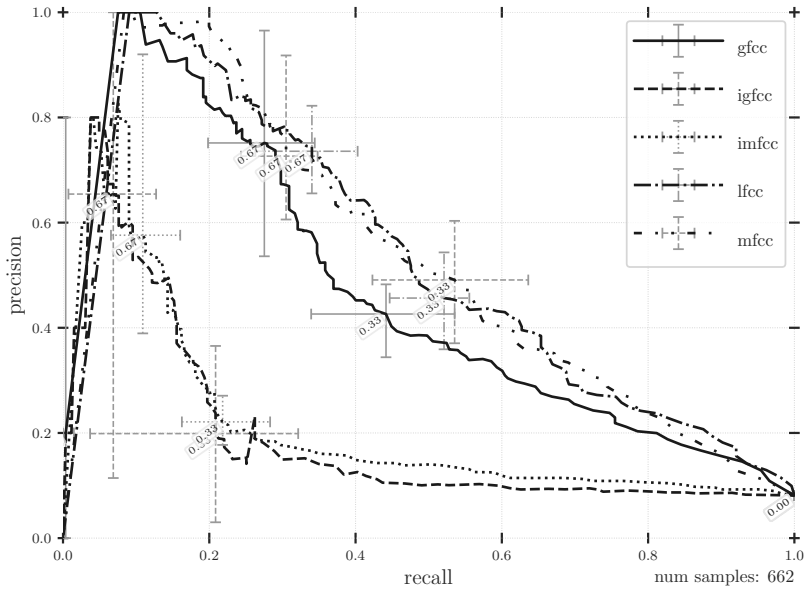
Figur 52: Precision-Recall curve for dkitchen snr 0db

B.18 dkitchen



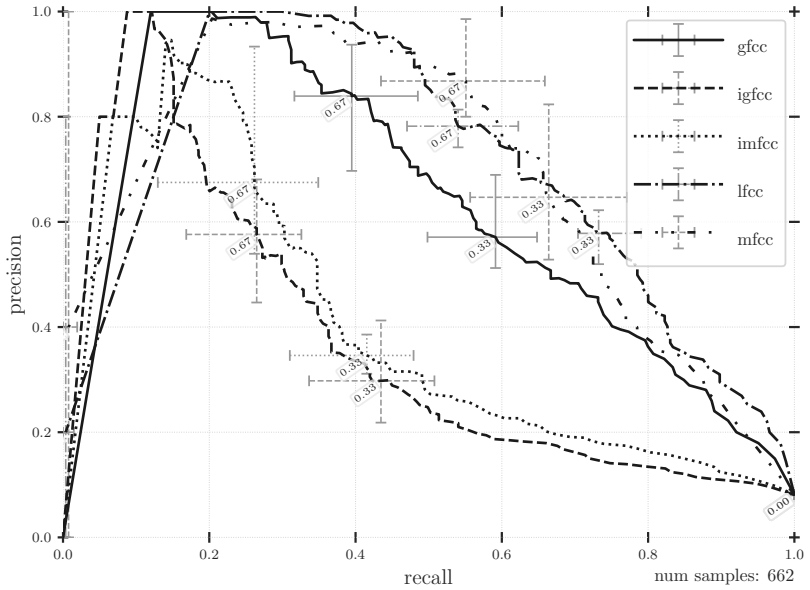
Figur 53: Precision-Recall curve for dkitchen snr 5db

B.19 dkitchen



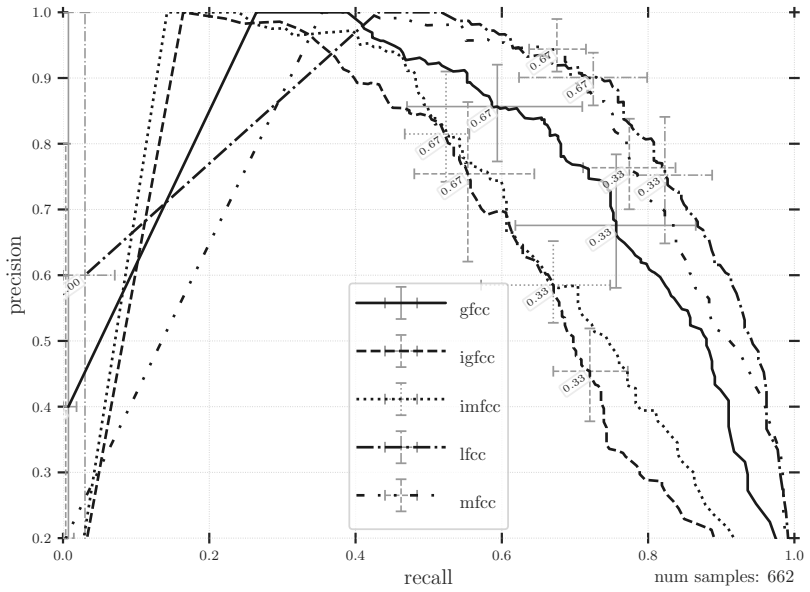
Figur 54: Precision-Recall curve for dkitchen snr 10db

B.20 dkitchen



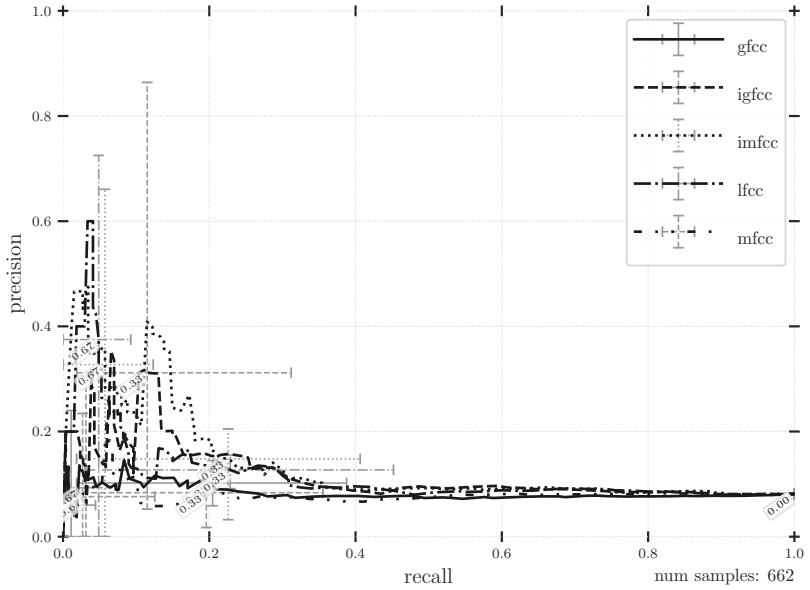
Figur 55: Precision-Recall curve for dkitchen snr 15db

B.21 dkitchen



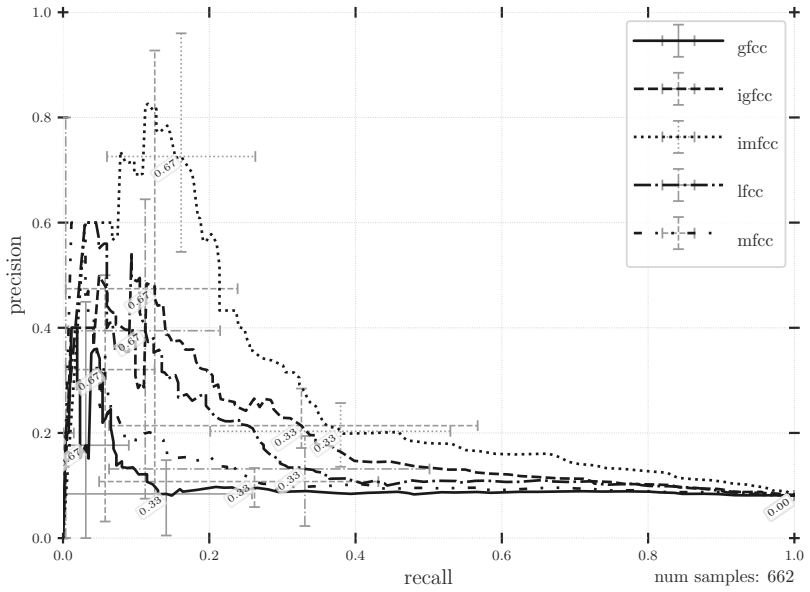
Figur 56: Precision-Recall curve for dkitchen snr 20db

B.22 pcafeter



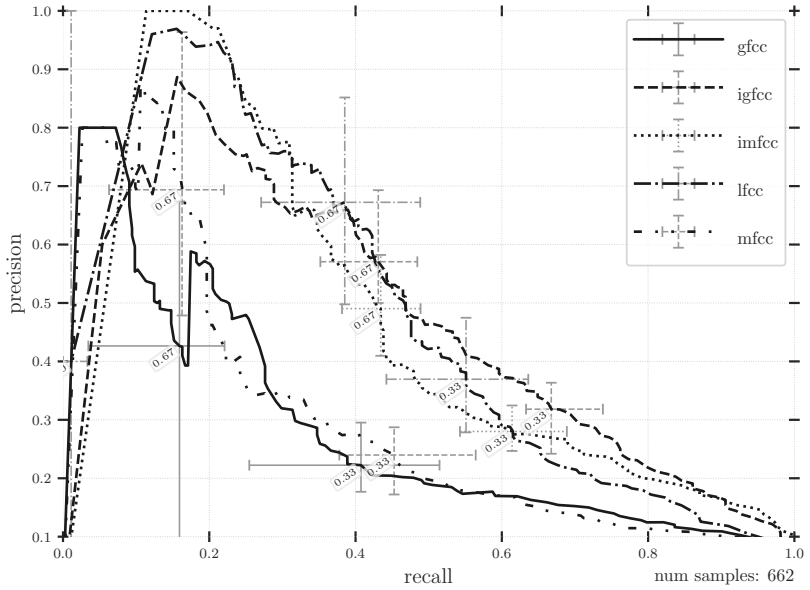
Figur 57: Precision-Recall curve for pcafeter snr 0db

B.23 pcafeter



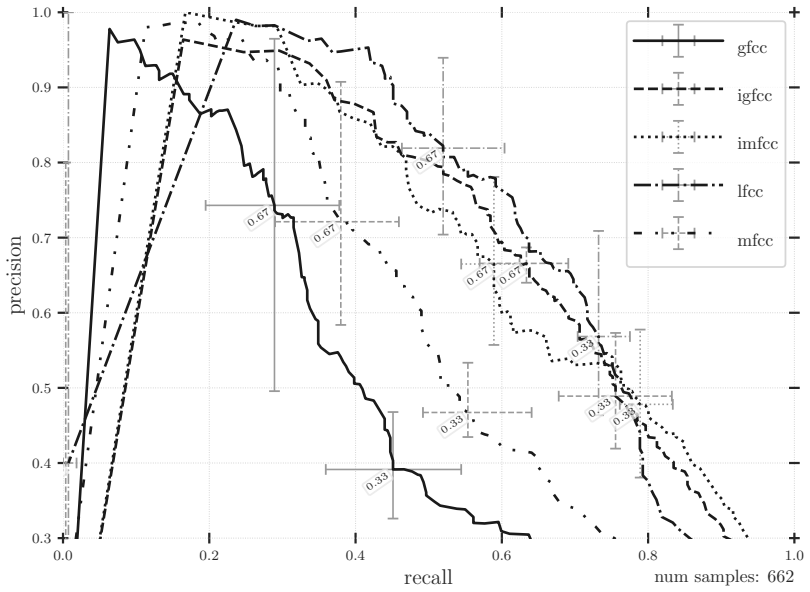
Figur 58: Precision-Recall curve for pcafeter snr 5db

B.24 pcafeter



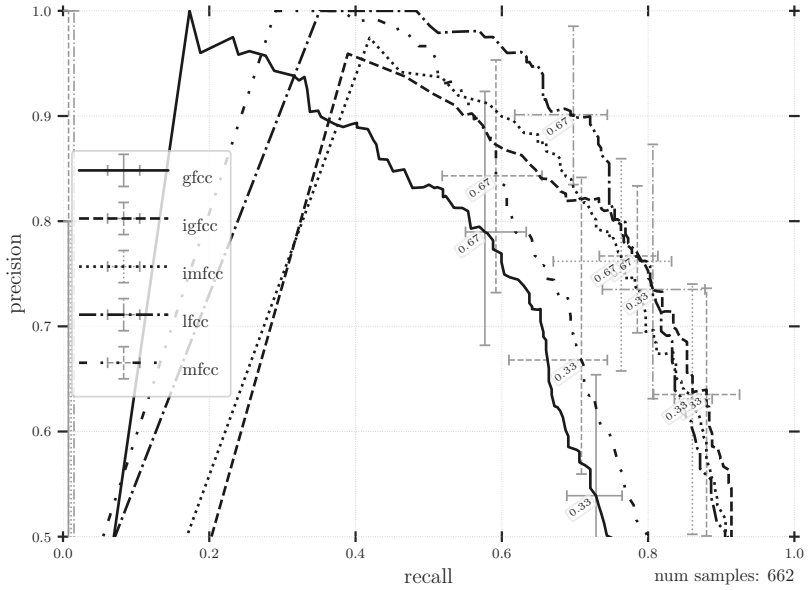
Figur 59: Precision-Recall curve for pcafeter snr 10db

B.25 pcafeter



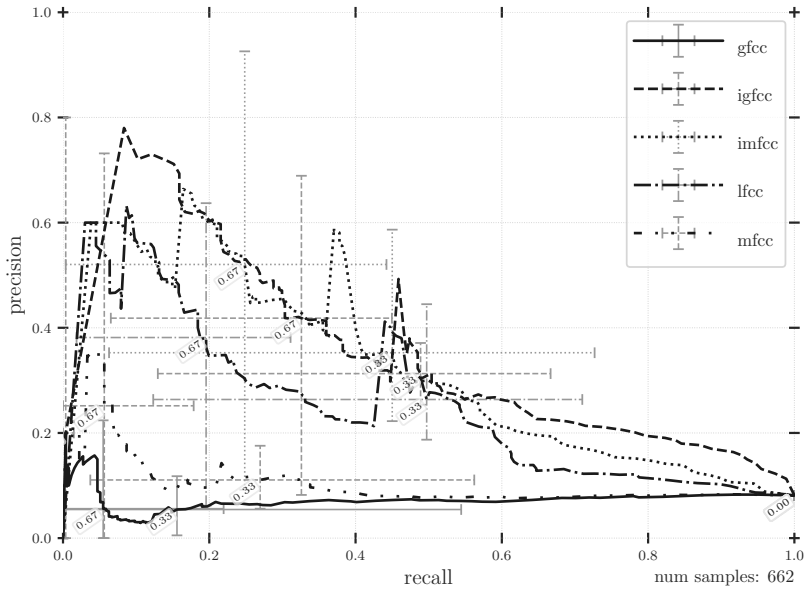
Figur 60: Precision-Recall curve for pcafeter snr 15db

B.26 pcafeter



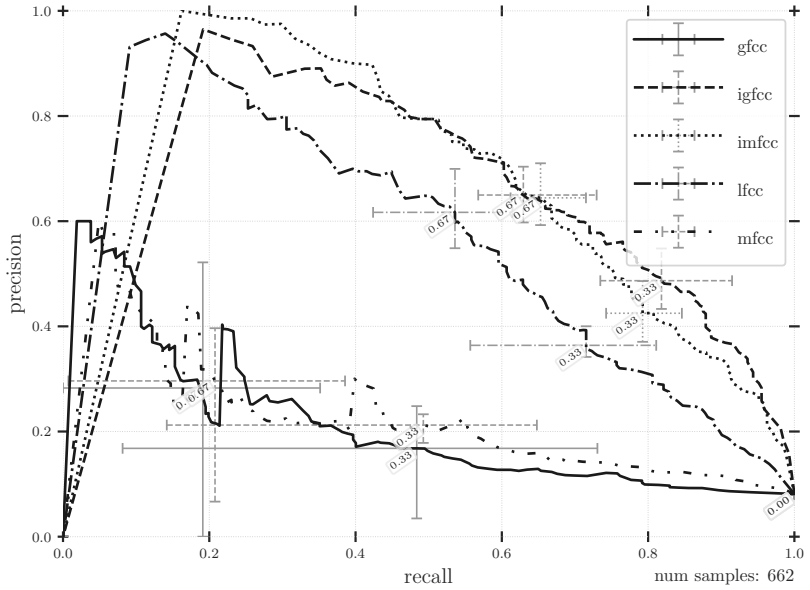
Figur 61: Precision-Recall curve for pcafeter snr 20db

B.27 spsquare



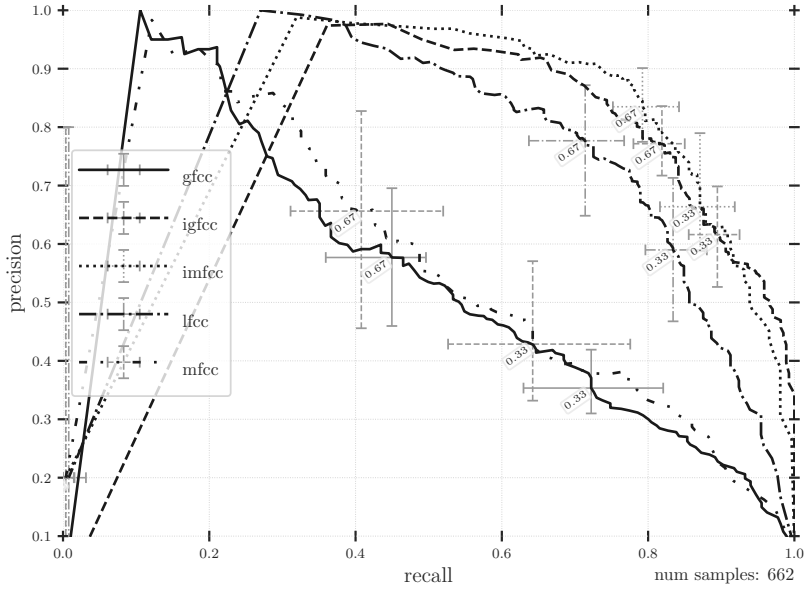
Figur 62: Precision-Recall curve for spsquare snr 0db

B.28 spsquare



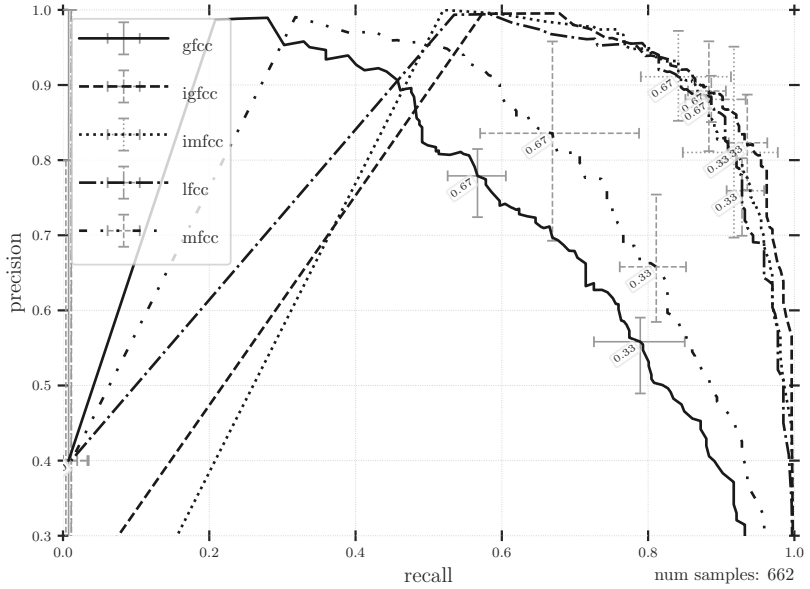
Figur 63: Precision-Recall curve for spsquare snr 5db

B.29 spsquare



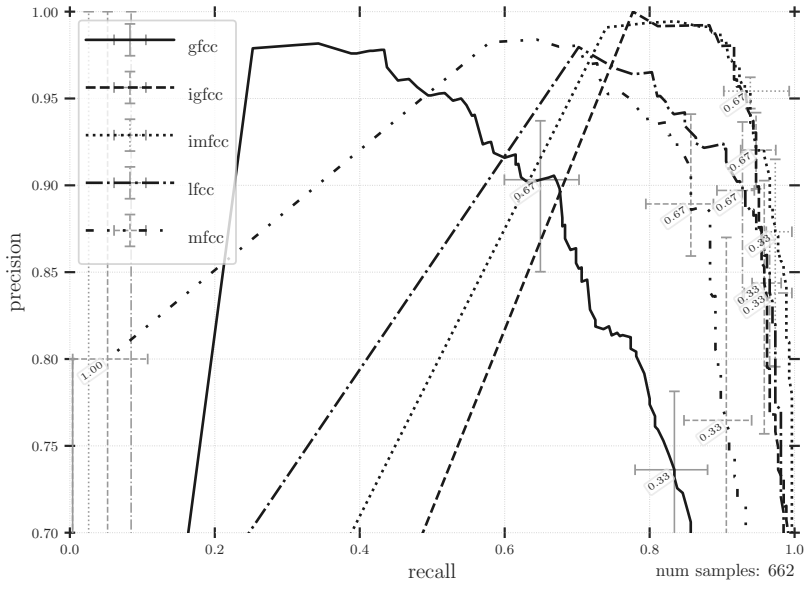
Figur 64: Precision-Recall curve for spsquare snr 10db

B.30 spsquare



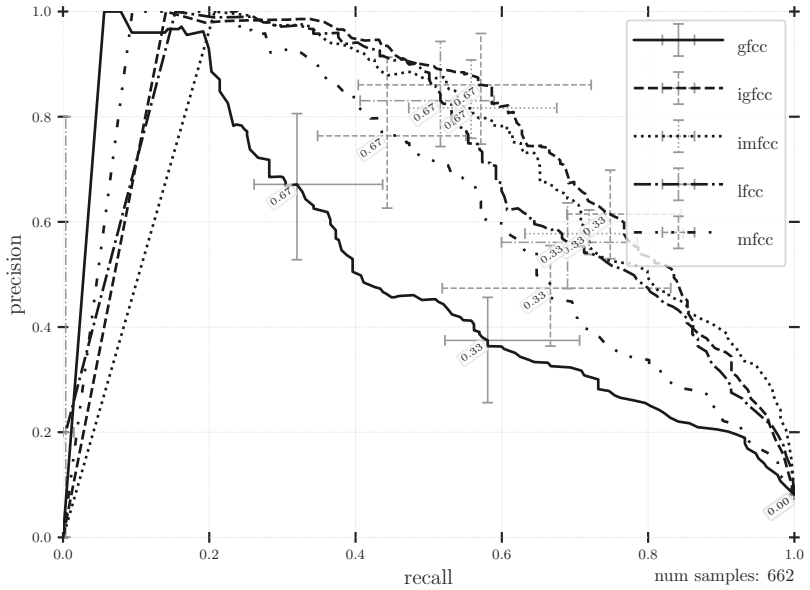
Figur 65: Precision-Recall curve for spsquare snr 15db

B.31 spsquare



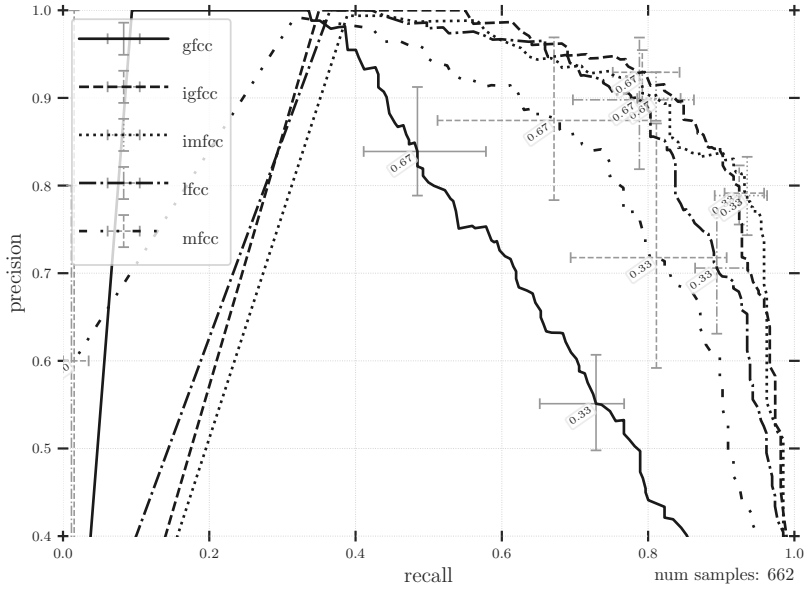
Figur 66: Precision-Recall curve for spsquare snr 20db

B.32 tbus



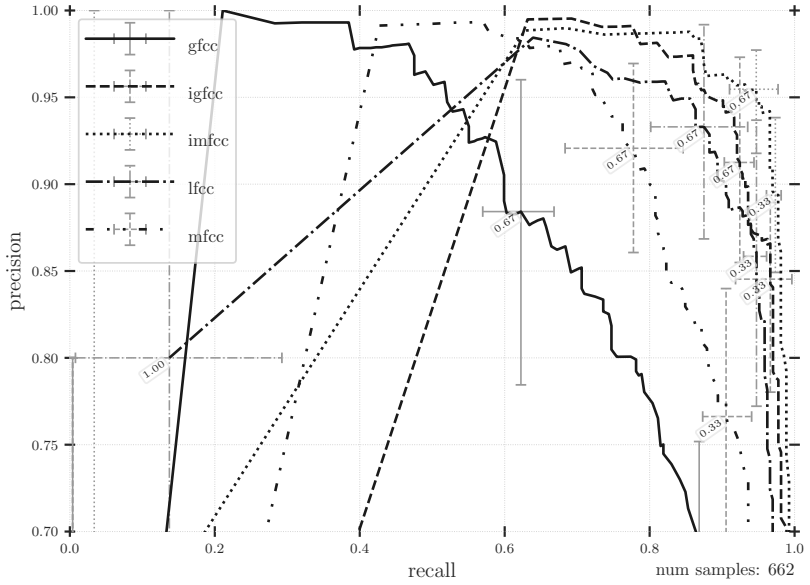
Figur 67: Precision-Recall curve for tbus snr 0db

B.33 tbus



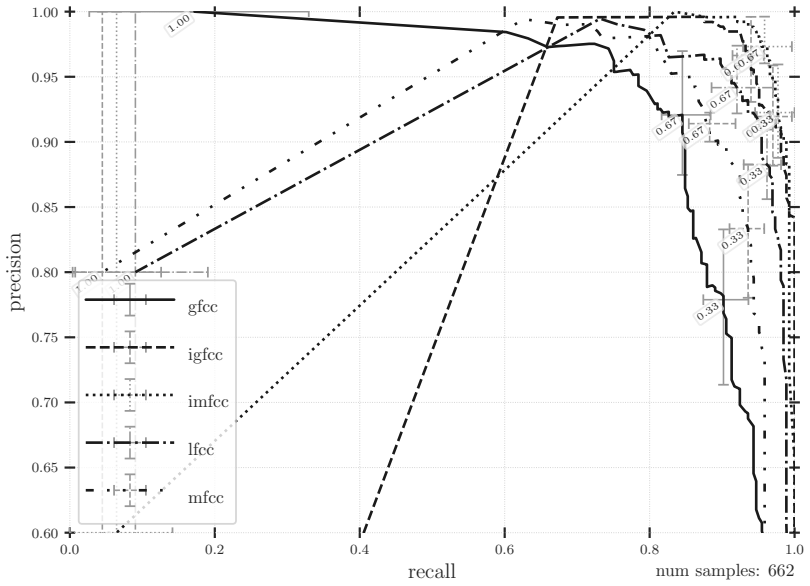
Figur 68: Precision-Recall curve for tbus snr 5db

B.34 tbus



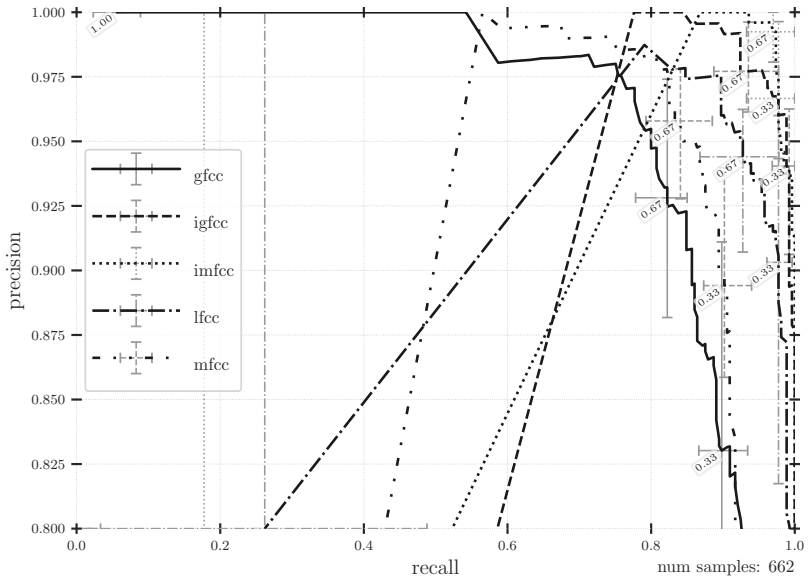
Figur 69: Precision-Recall curve for tbus snr 10db

B.35 tbus



Figur 70: Precision-Recall curve for tbus snr 15db

B.36 tbus

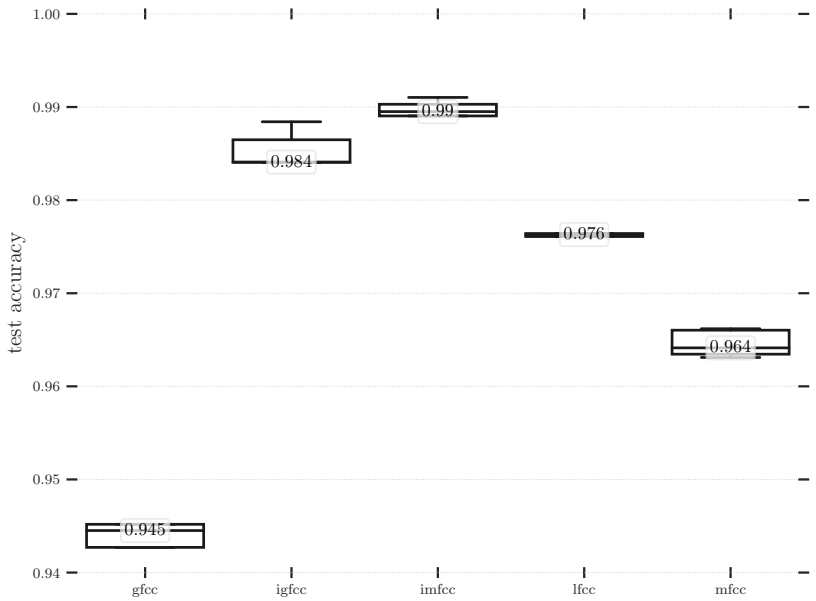


Figur 71: Precision-Recall curve for tbus snr 20db

C Model Results OM

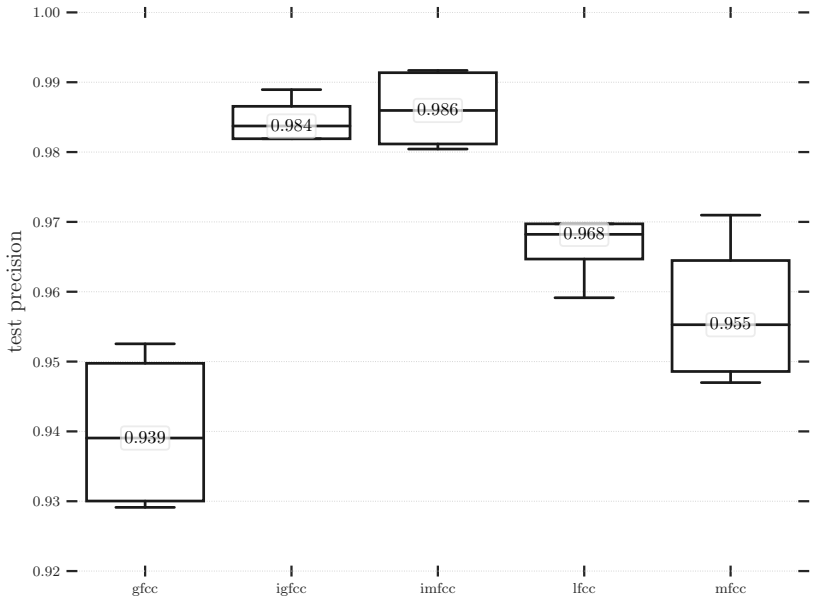
C.1 merge ab test ab model ab ab

C.1.1 test accuracy



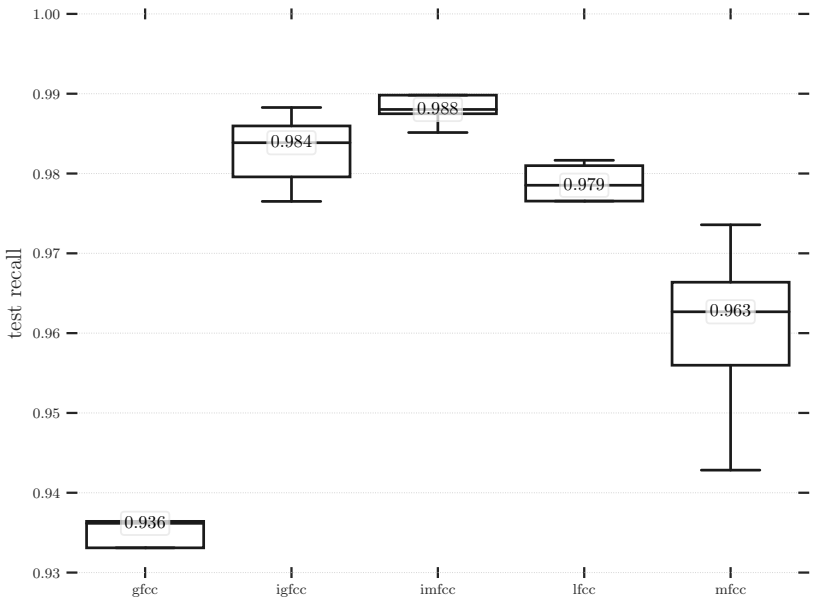
Figur 72: *test accuracy* for merge ab test ab model ab ab, *test a*

C.1.2 test precision



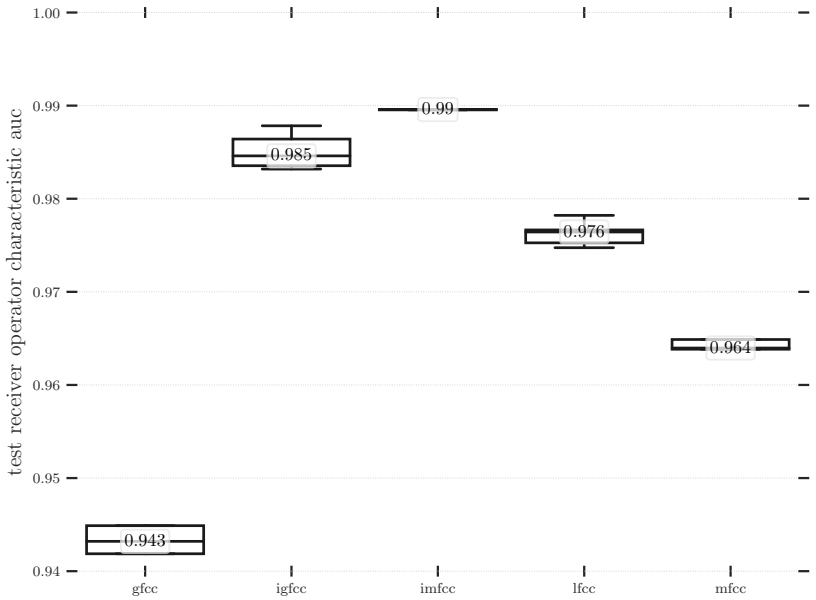
Figur 73: *test precision* for merge ab test ab model ab ab, *test a*

C.1.3 test recall



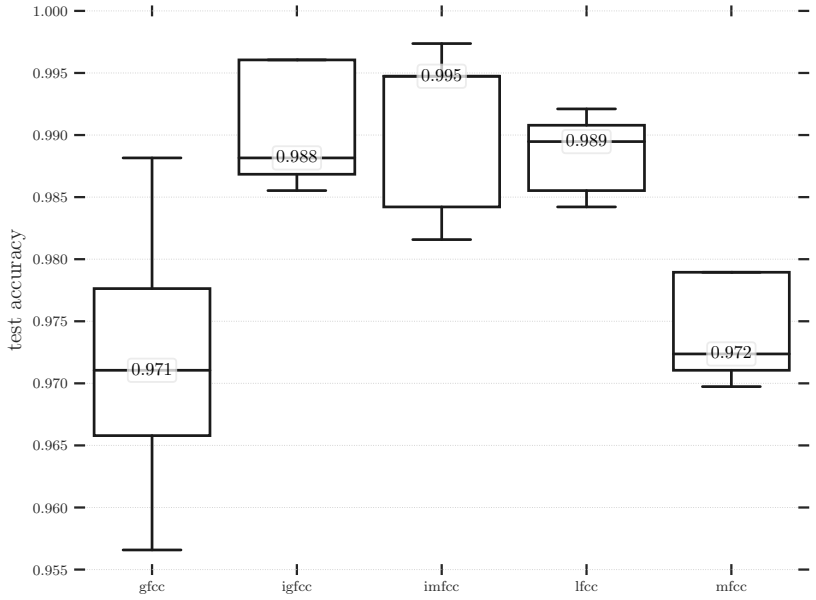
Figur 74: *test recall* for merge ab test ab model ab ab, *test a*

C.1.4 test receiver operator characteristic auc



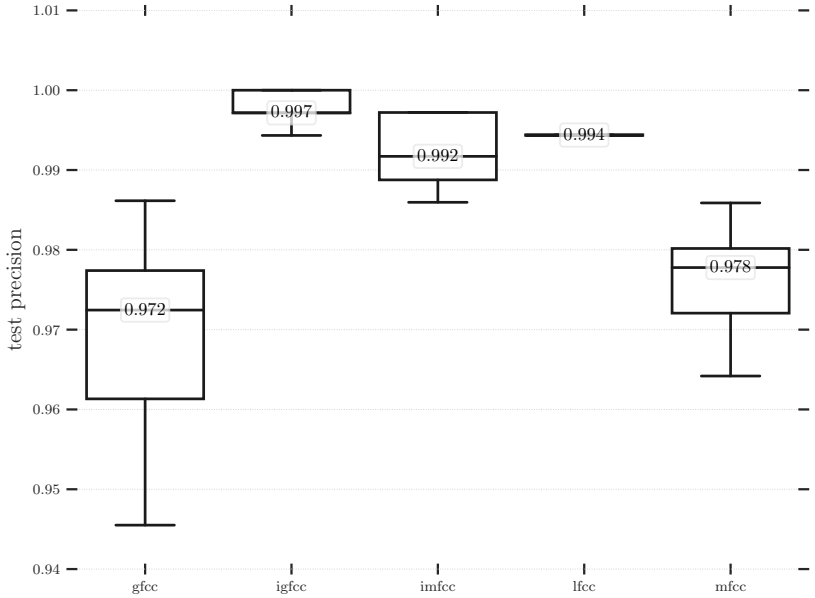
Figur 75: *test receiver operator characteristic auc* for merge ab test ab model ab ab, *test a*

C.1.5 test accuracy



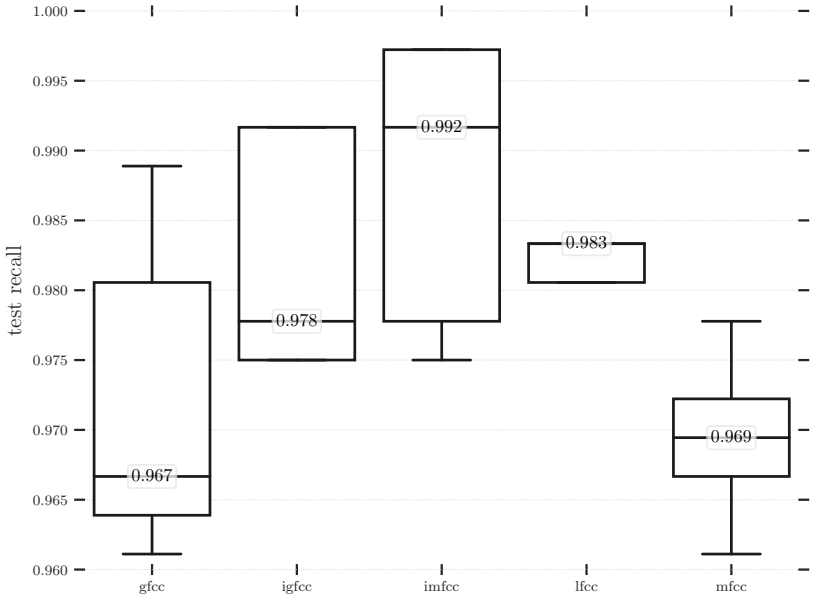
Figur 76: *test accuracy* for merge ab test ab model ab ab, *test b*

C.1.6 test precision



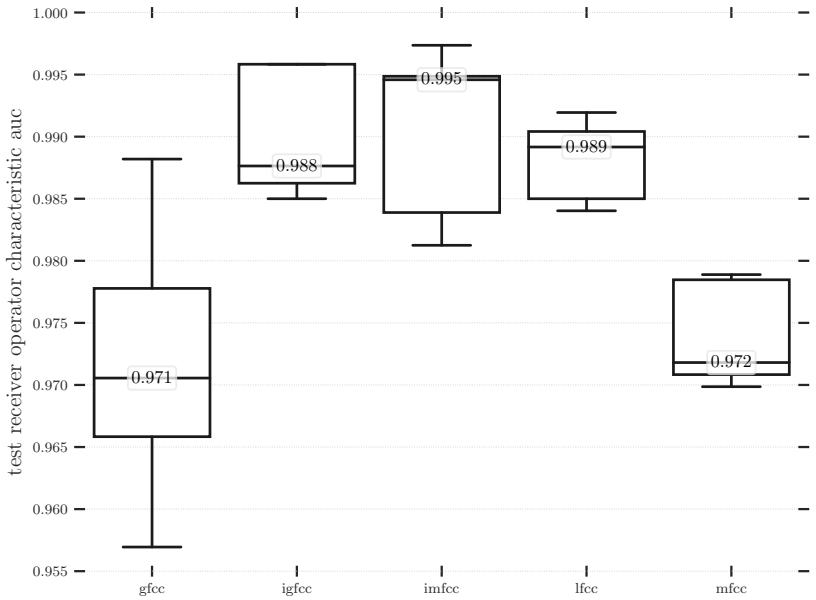
Figur 77: *test precision* for merge ab test ab model ab ab, *test b*

C.1.7 test recall



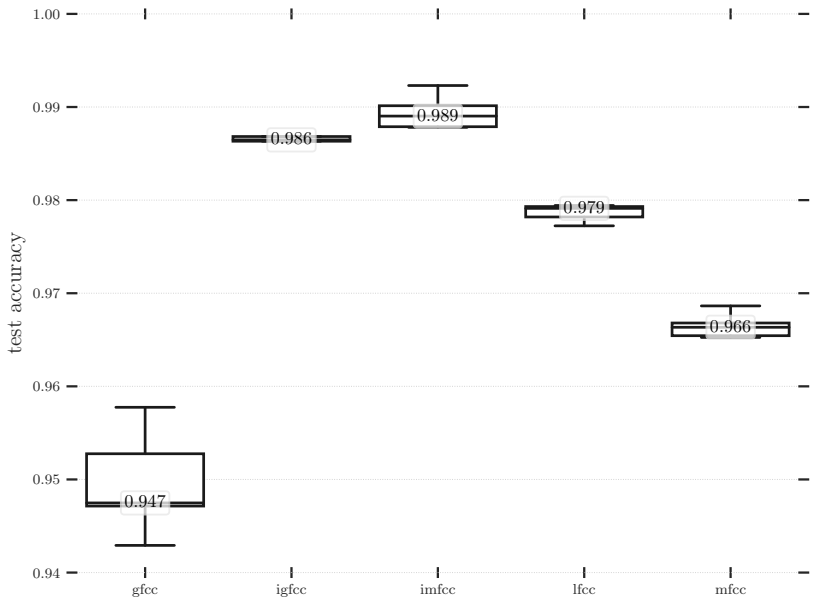
Figur 78: *test recall* for merge ab test ab model ab ab, *test b*

C.1.8 test receiver operator characteristic auc



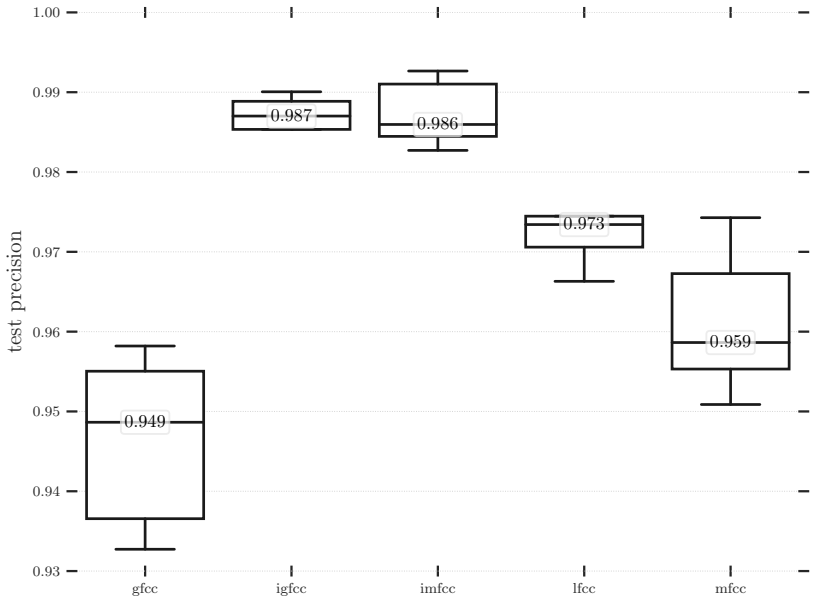
Figur 79: *test receiver operator characteristic auc* for merge ab test ab model ab ab, *test b*

C.1.9 test accuracy



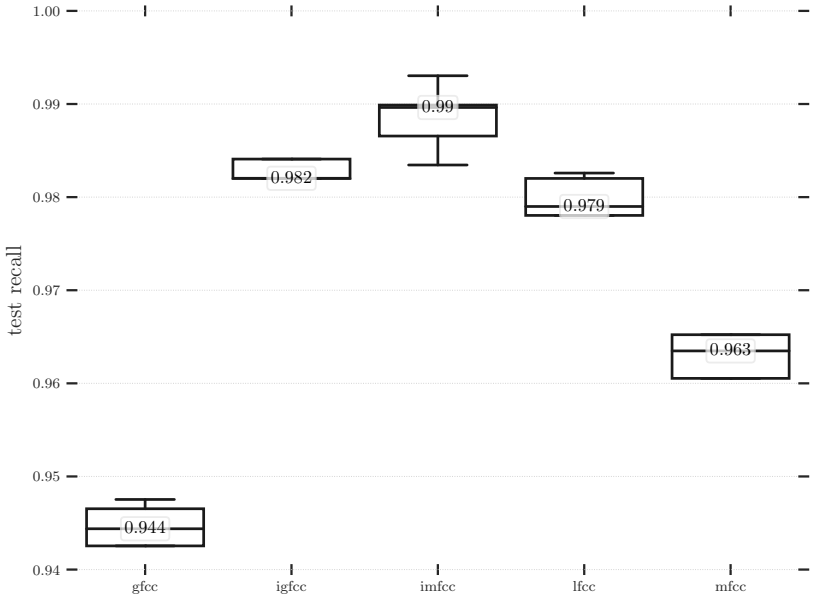
Figur 80: *test accuracy* for merge ab test ab model ab ab, *test a b*

C.1.10 test precision



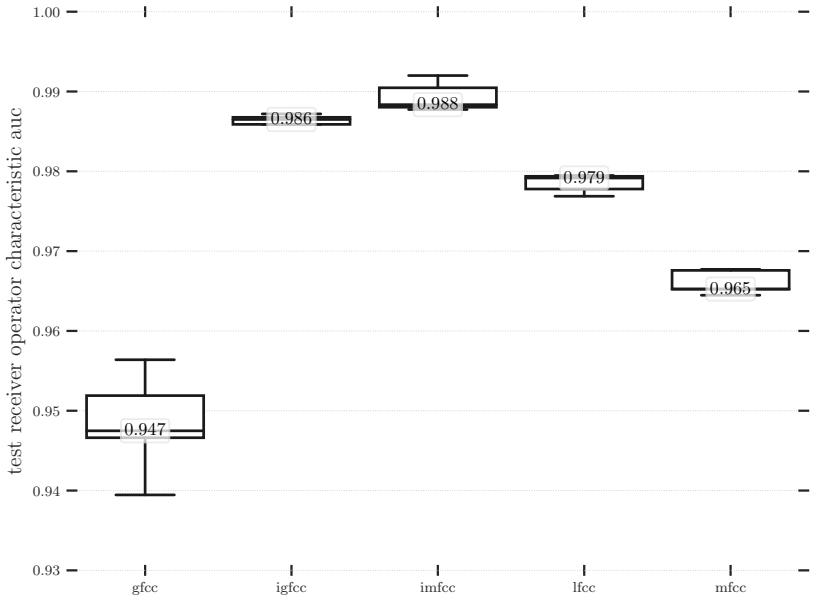
Figur 81: *test precision* for merge ab test ab model ab ab, *test a b*

C.1.11 test recall



Figur 82: *test recall* for merge ab test ab model ab ab, *test a b*

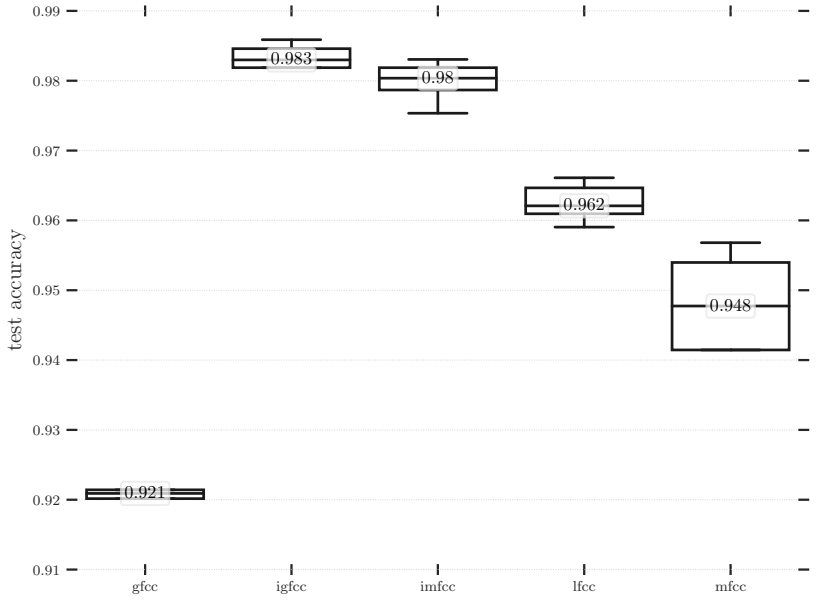
C.1.12 test receiver operator characteristic auc



Figur 83: *test receiver operator characteristic auc* for merge ab test ab model ab ab, *test a b*

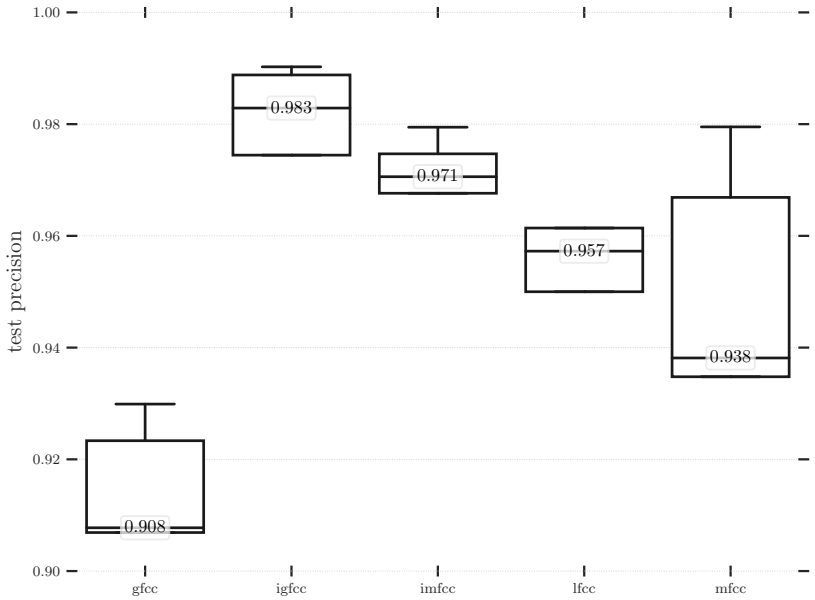
C.2 truncated a model a a

C.2.1 test accuracy



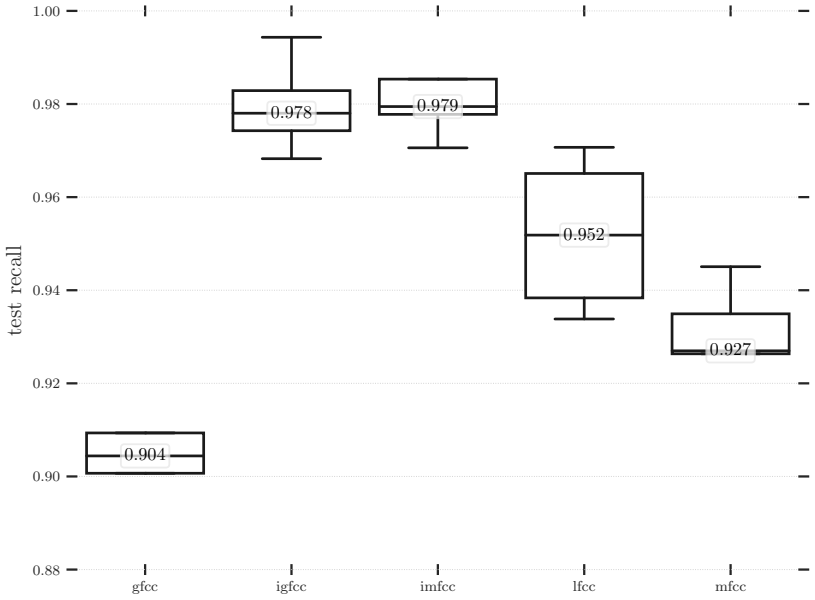
Figur 84: *test accuracy* for truncated a model a a, *test a*

C.2.2 test precision



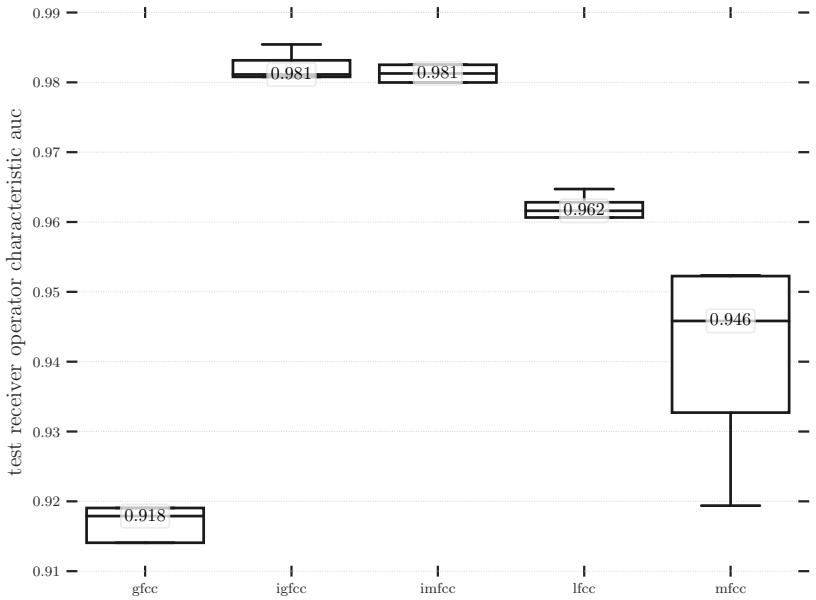
Figur 85: *test precision* for truncated a model a a, *test a*

C.2.3 test recall



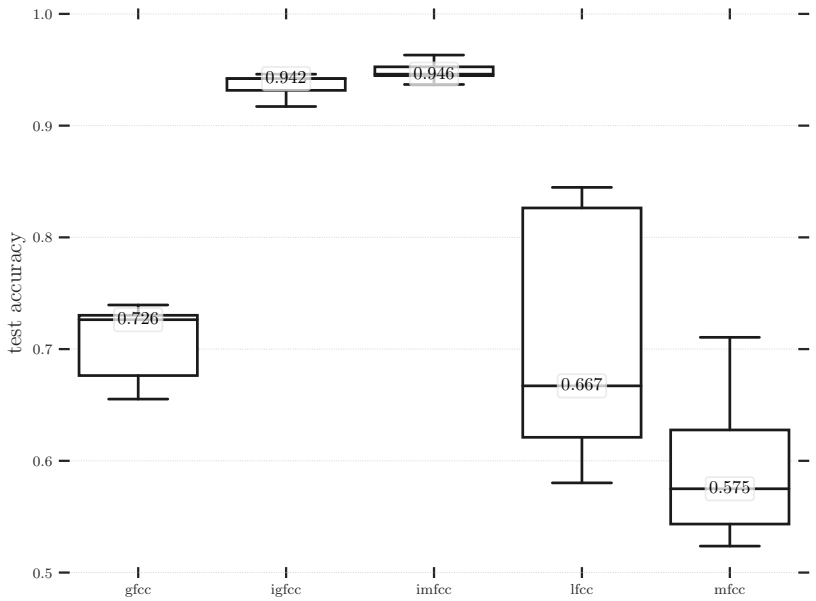
Figur 86: *test recall* for truncated a model a a, *test a*

C.2.4 test receiver operator characteristic auc



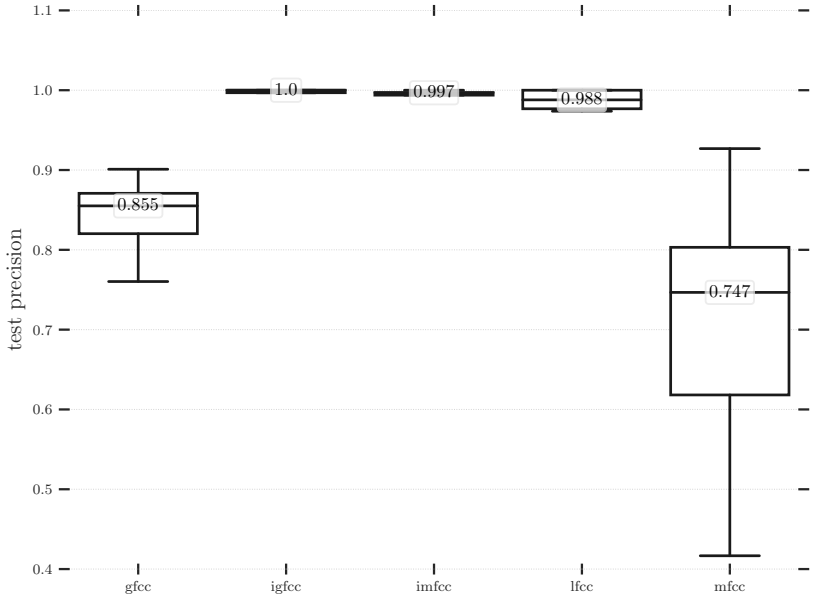
Figur 87: *test receiver operator characteristic auc* for truncated a model a a, *test a*

C.2.5 test accuracy



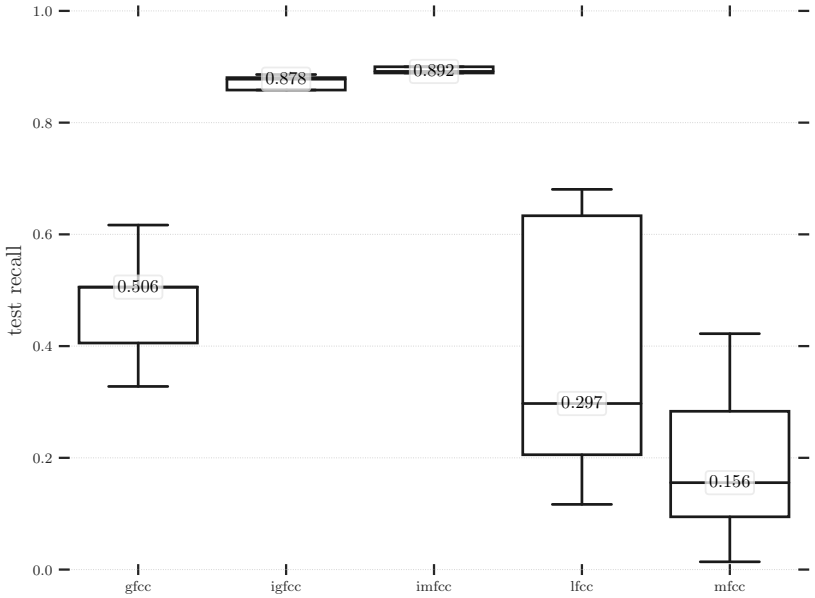
Figur 88: *test accuracy* for truncated a model a a, *test b*

C.2.6 test precision



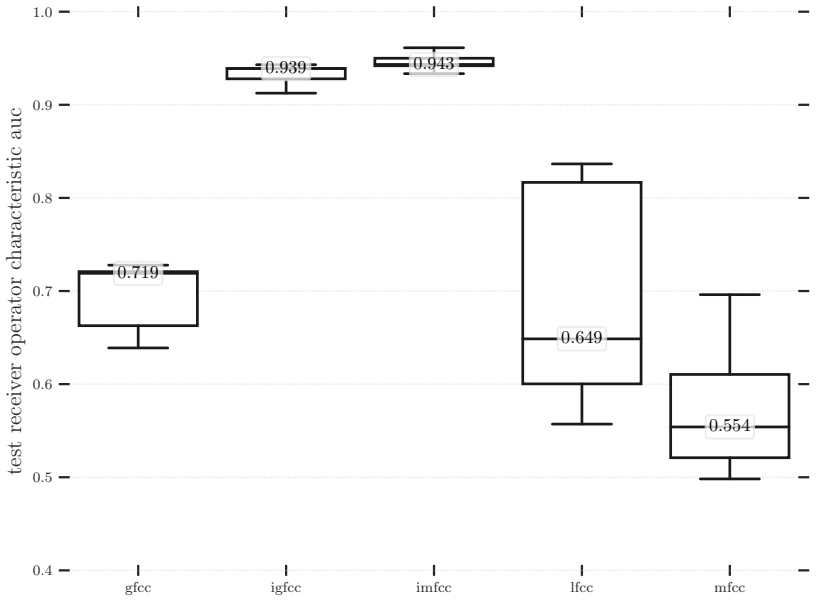
Figur 89: *test precision* for truncated a model a a, *test b*

C.2.7 test recall



Figur 90: *test recall* for truncated a model a a, *test b*

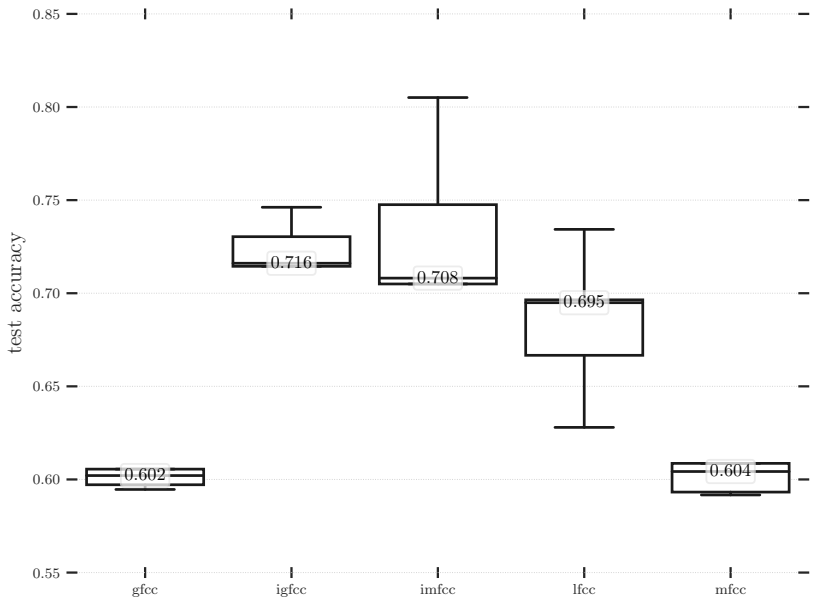
C.2.8 test receiver operator characteristic auc



Figur 91: *test receiver operator characteristic auc* for truncated a model a a, *test b*

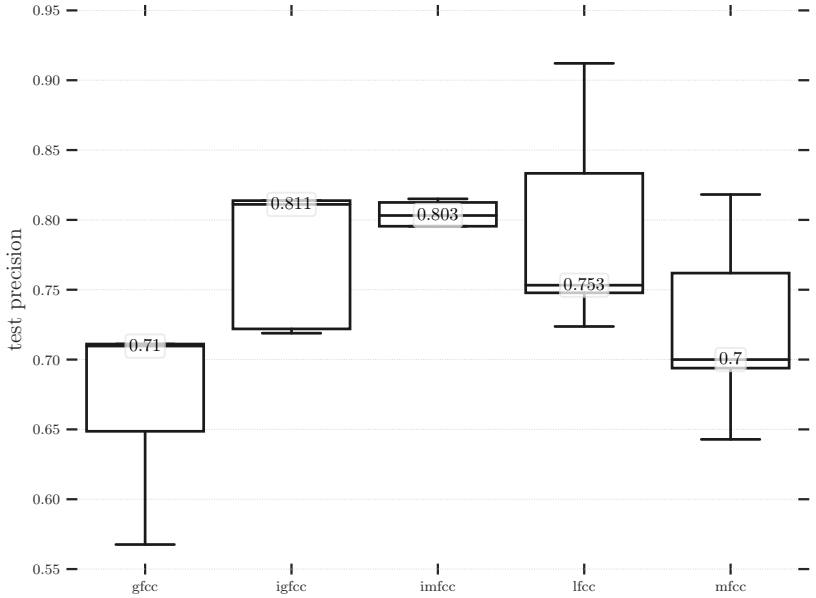
C.3 truncated b model b b

C.3.1 test accuracy



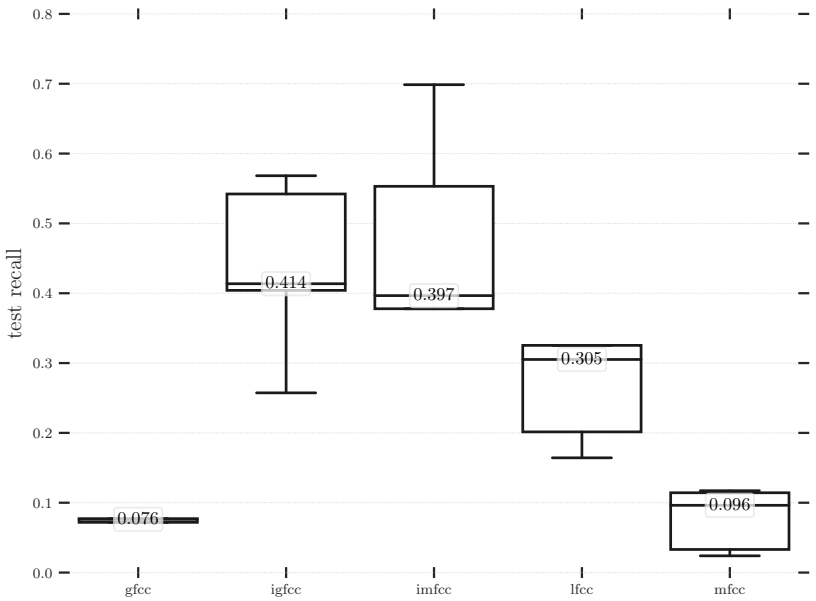
Figur 92: *test accuracy* for truncated b model b b, *test a*

C.3.2 test precision



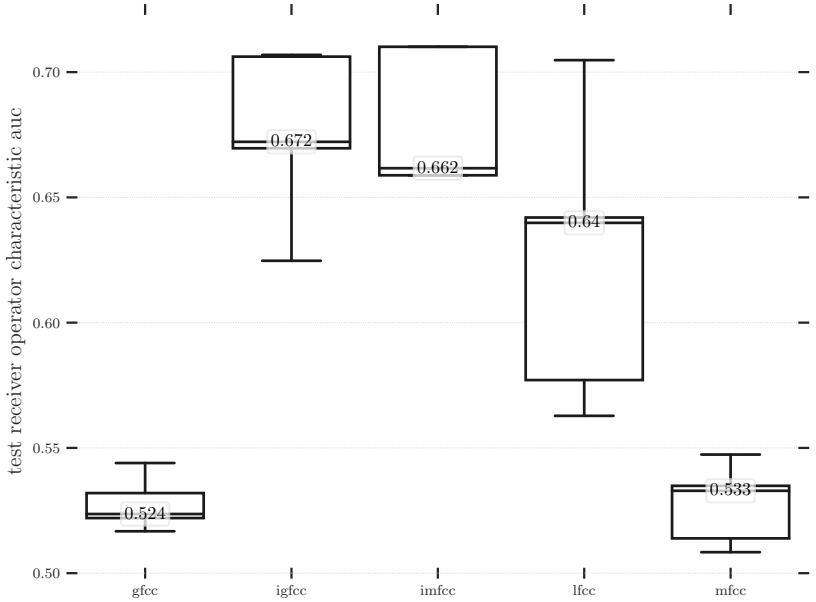
Figur 93: *test precision* for truncated b model b b, *test a*

C.3.3 test recall



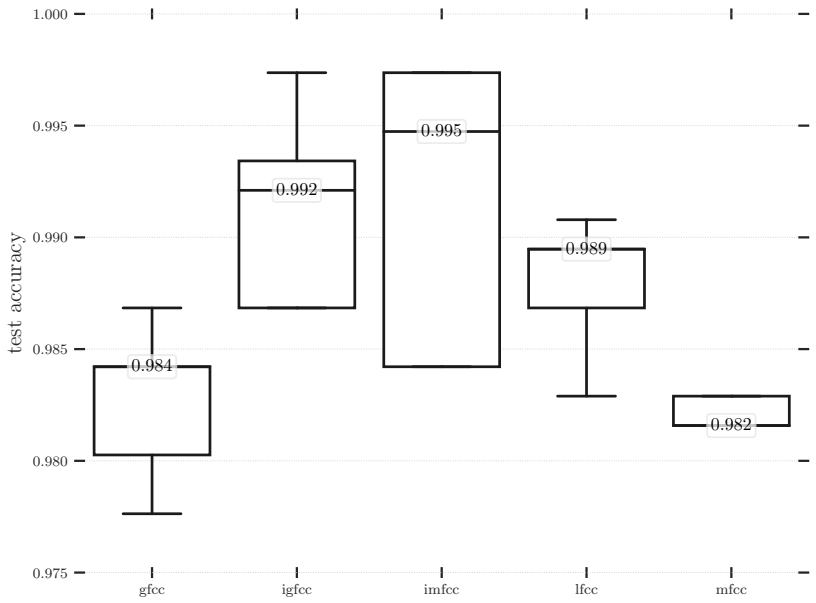
Figur 94: *test recall* for truncated b model b b, *test a*

C.3.4 test receiver operator characteristic auc



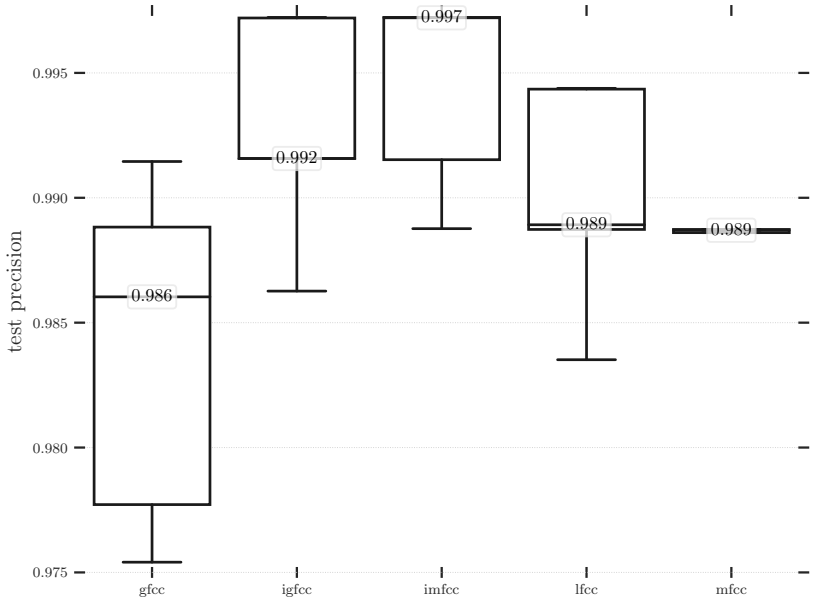
Figur 95: *test receiver operator characteristic auc* for truncated b model b b, *test a*

C.3.5 test accuracy



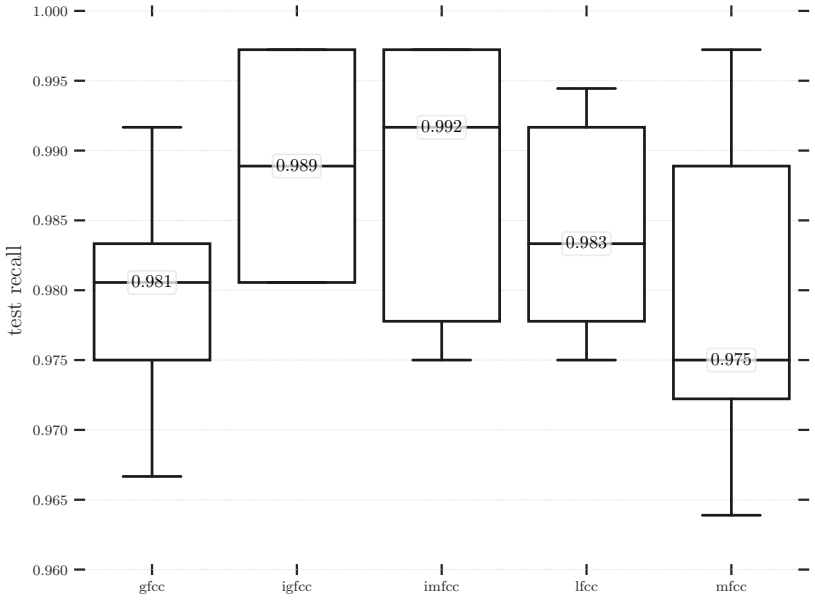
Figur 96: *test accuracy* for truncated b model b b, *test b*

C.3.6 test precision



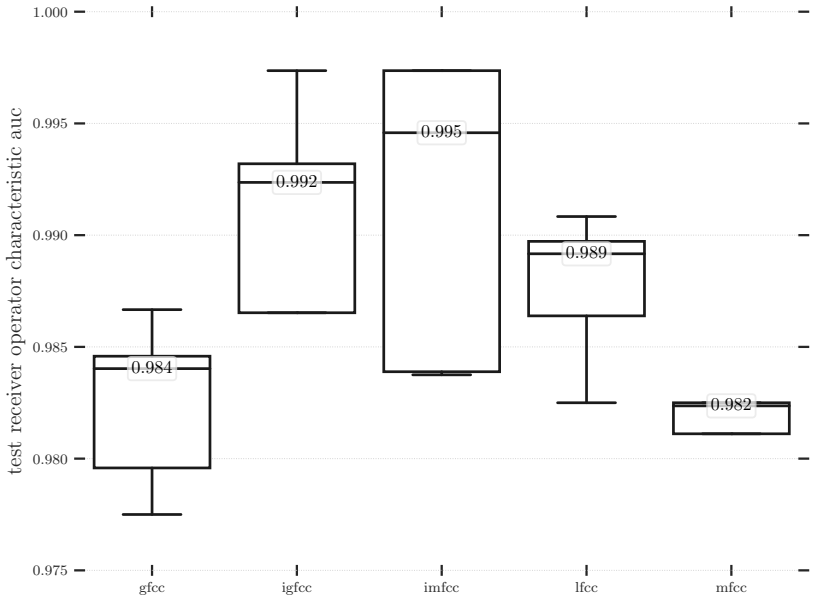
Figur 97: *test precision* for truncated b model b b, *test b*

C.3.7 test recall



Figur 98: *test recall* for truncated b model b b, *test b*

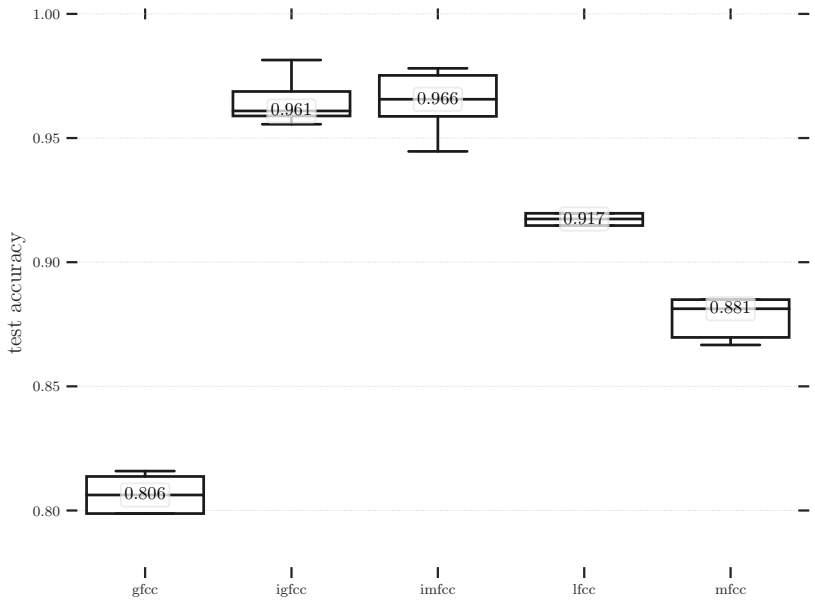
C.3.8 test receiver operator characteristic auc



Figur 99: *test receiver operator characteristic auc* for truncated b model b b, *test b*

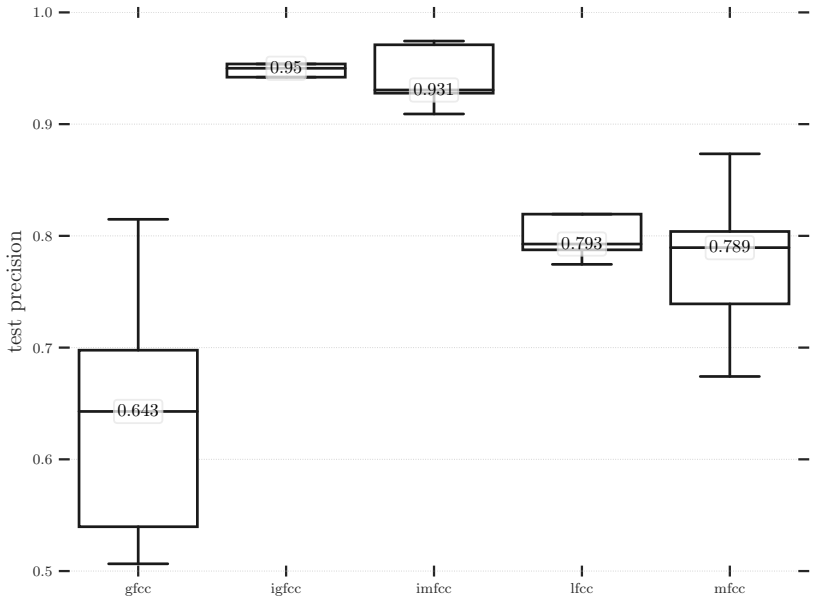
C.4 truncated a speech model speech speech

C.4.1 test accuracy



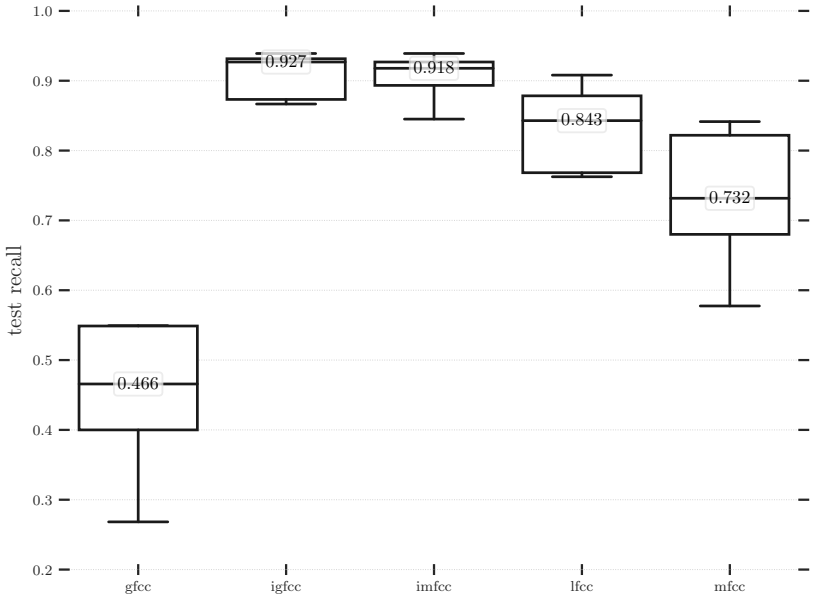
Figur 100: *test accuracy* for truncated a speech model speech speech, *test speech*

C.4.2 test precision



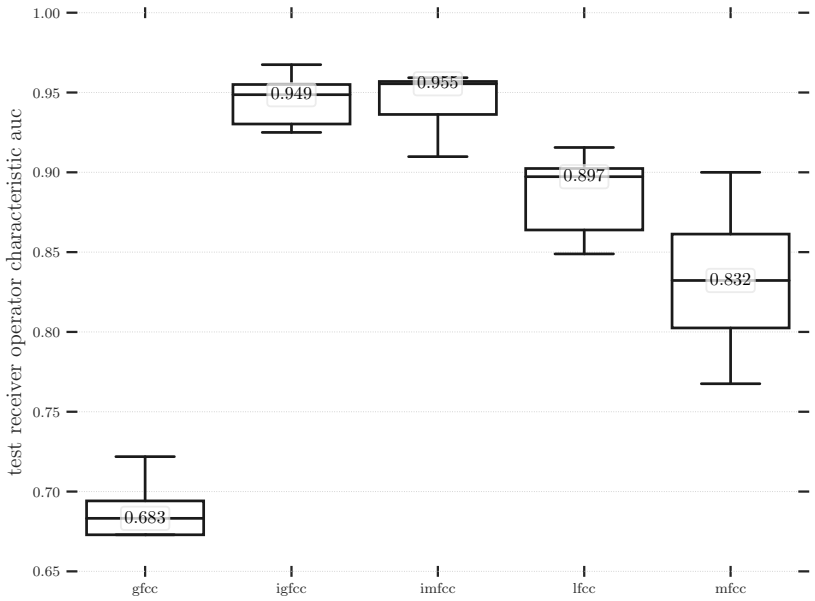
Figur 101: *test precision* for truncated a speech model speech speech, *test speech*

C.4.3 test recall



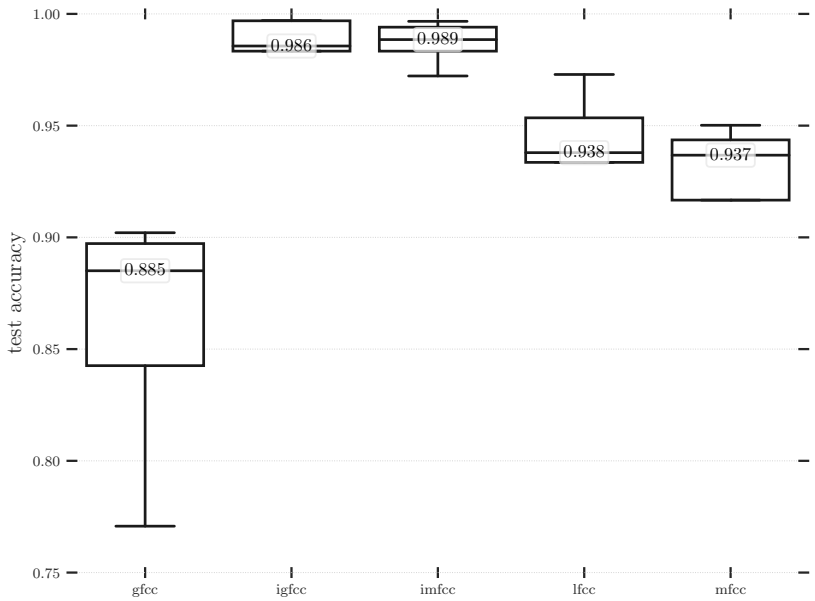
Figur 102: *test recall* for truncated a speech model speech speech, *test speech*

C.4.4 test receiver operator characteristic auc



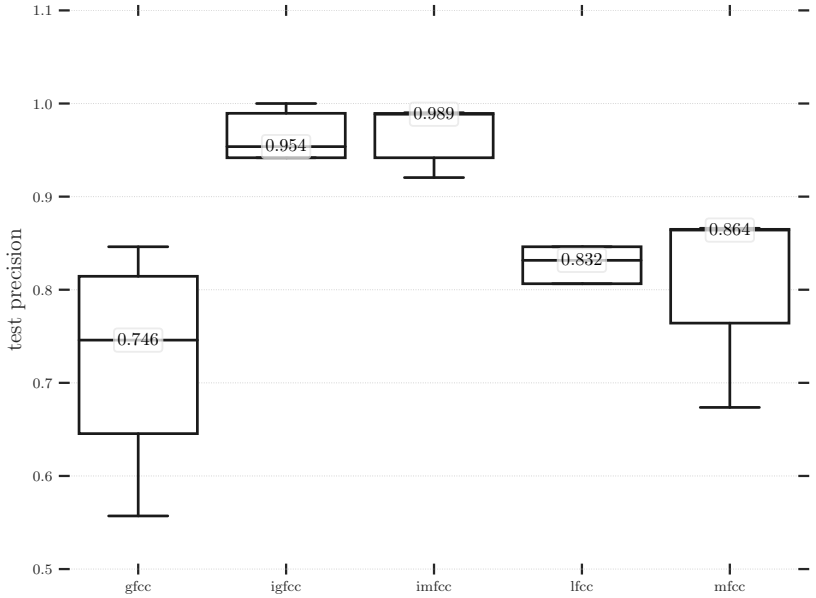
Figur 103: *test receiver operator characteristic auc* for truncated a speech model speech speech, *test speech*

C.4.5 test accuracy



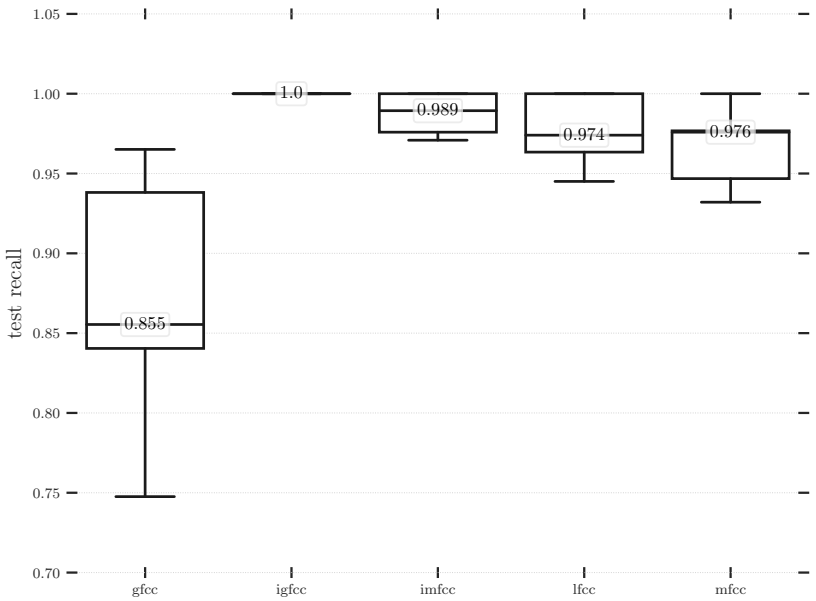
Figur 104: *test accuracy* for truncated a speech model speech speech, *test silence*

C.4.6 test precision



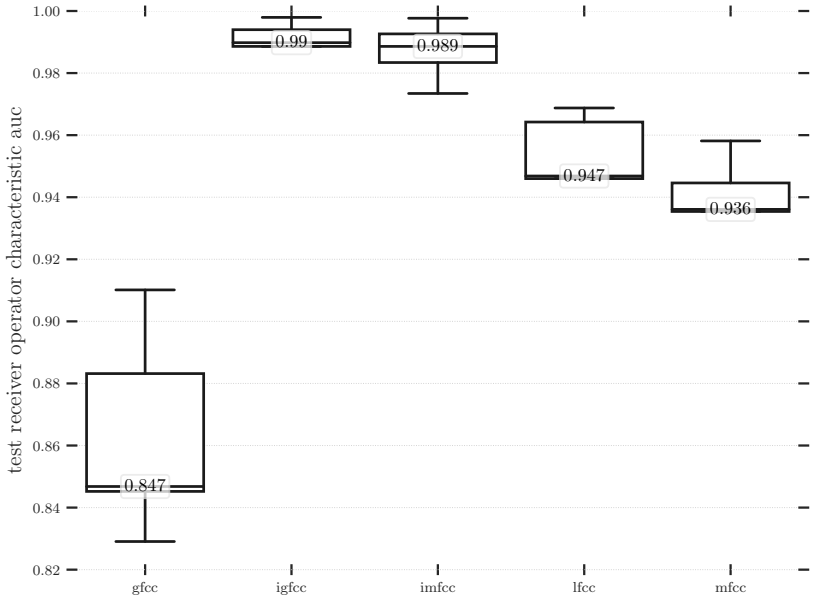
Figur 105: *test precision* for truncated a speech model speech speech, *test silence*

C.4.7 test recall



Figur 106: *test recall* for truncated a speech model speech speech, *test silence*

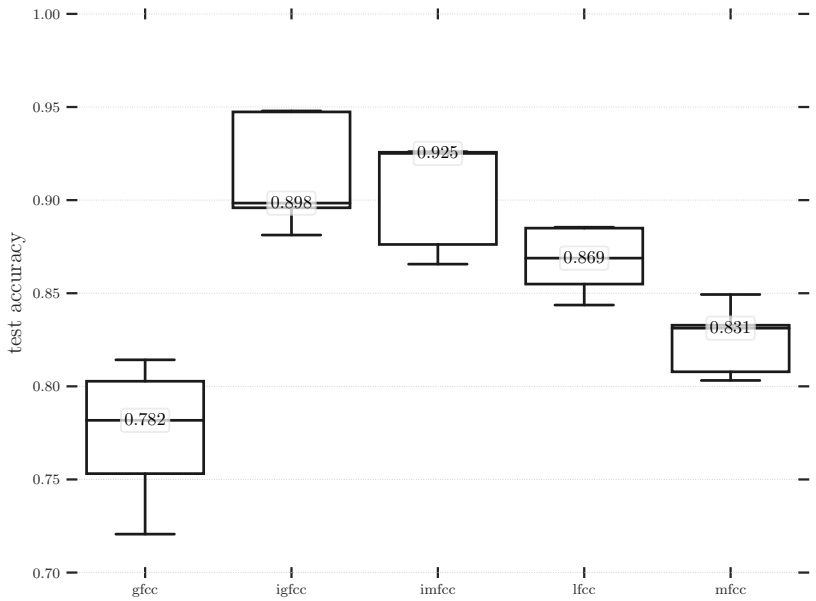
C.4.8 test receiver operator characteristic auc



Figur 107: *test receiver operator characteristic auc* for truncated a speech model speech speech, *test silence*

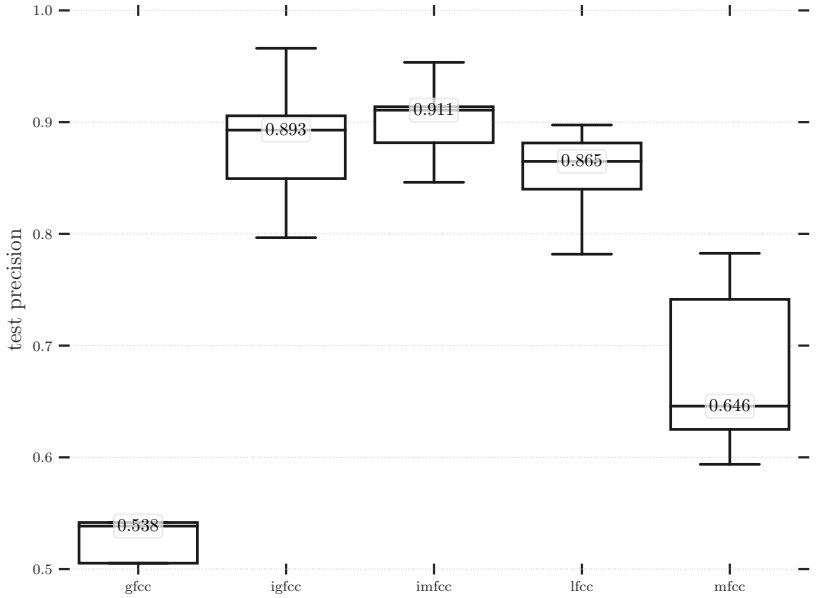
C.5 truncated a silence model silence silence

C.5.1 test accuracy



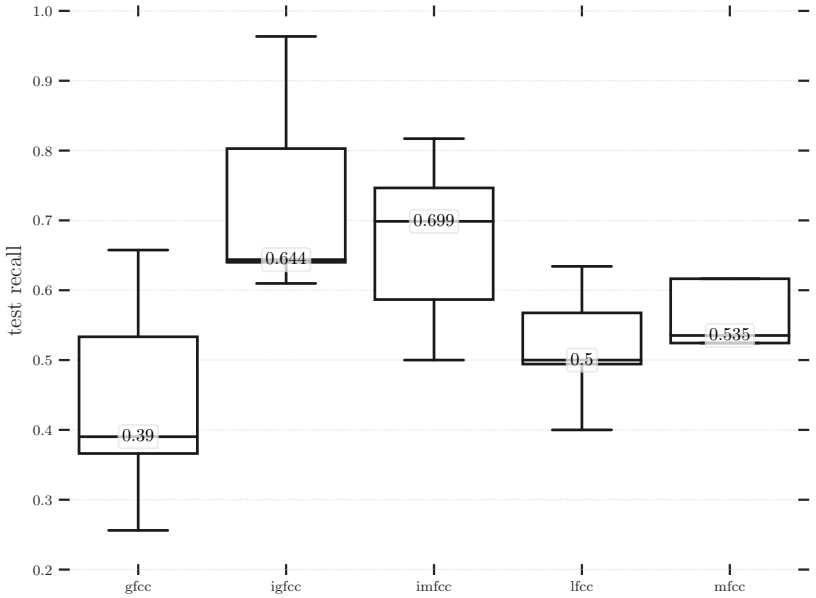
Figur 108: *test accuracy* for truncated a silence model silence silence, *test speech*

C.5.2 test precision



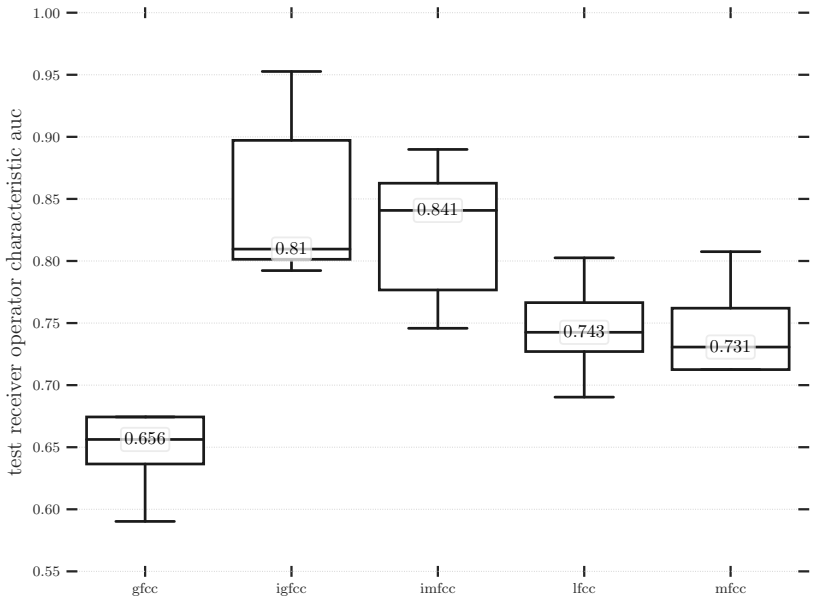
Figur 109: *test precision* for truncated a silence model silence silence, *test speech*

C.5.3 test recall



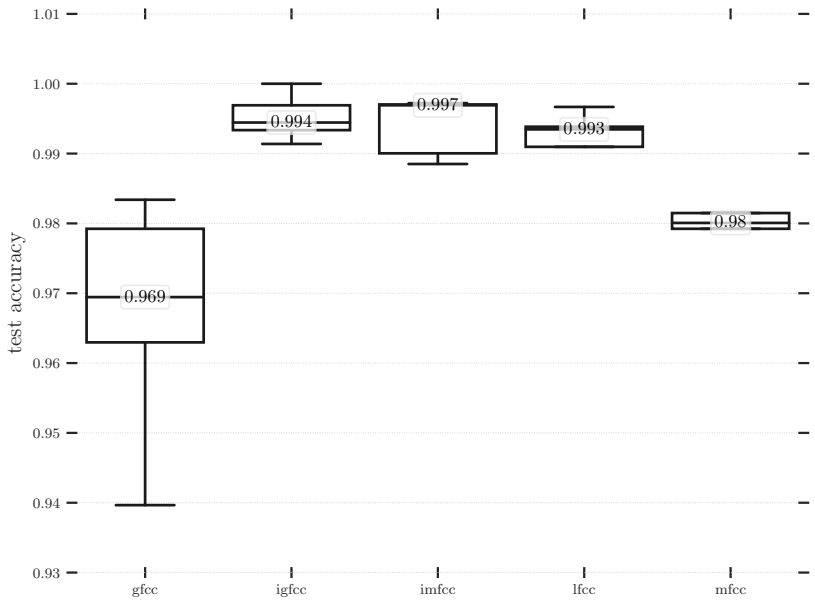
Figur 110: *test recall* for truncated a silence model silence silence, *test speech*

C.5.4 test receiver operator characteristic auc



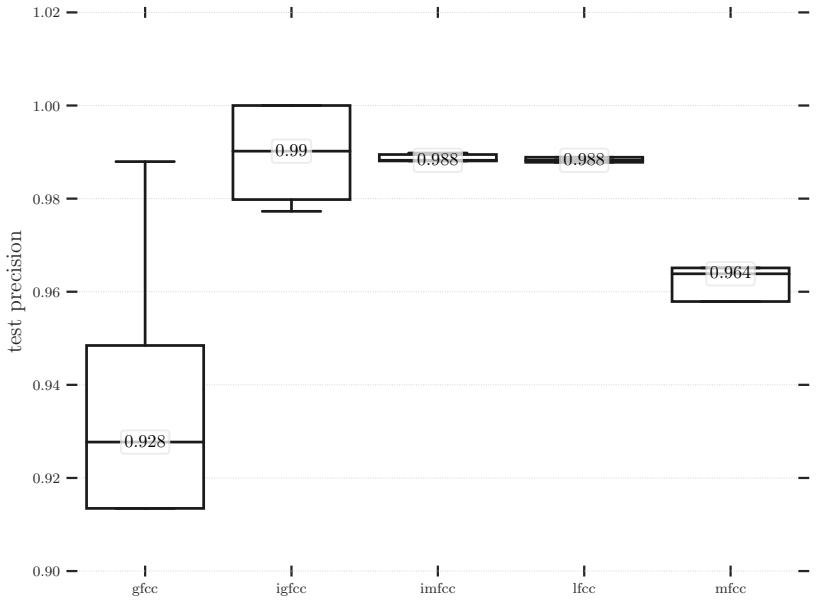
Figur 111: *test receiver operator characteristic auc* for truncated a silence model silence silence, *test speech*

C.5.5 test accuracy



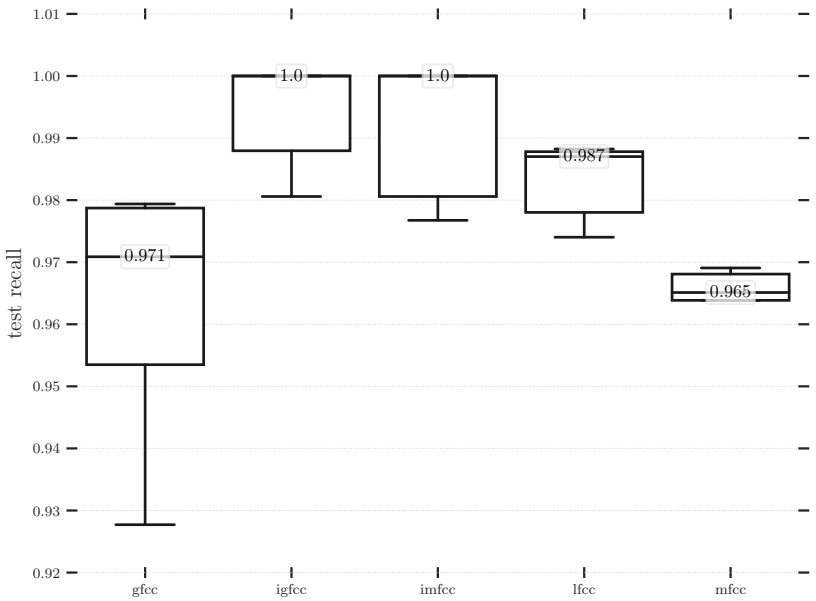
Figur 112: *test accuracy* for truncated a silence model silence silence, *test silence*

C.5.6 test precision



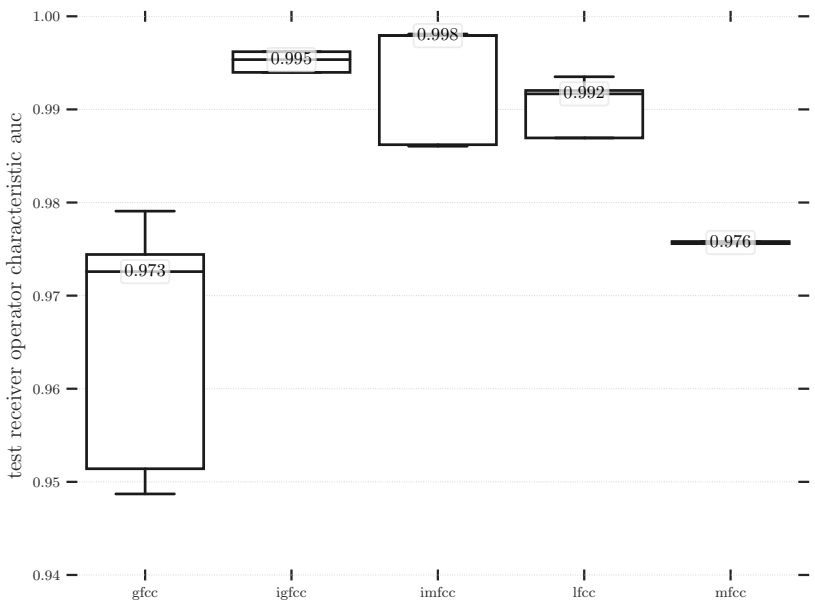
Figur 113: *test precision* for truncated a silence model silence silence, *test silence*

C.5.7 test recall



Figur 114: *test recall* for truncated a silence model silence silence, *test silence*

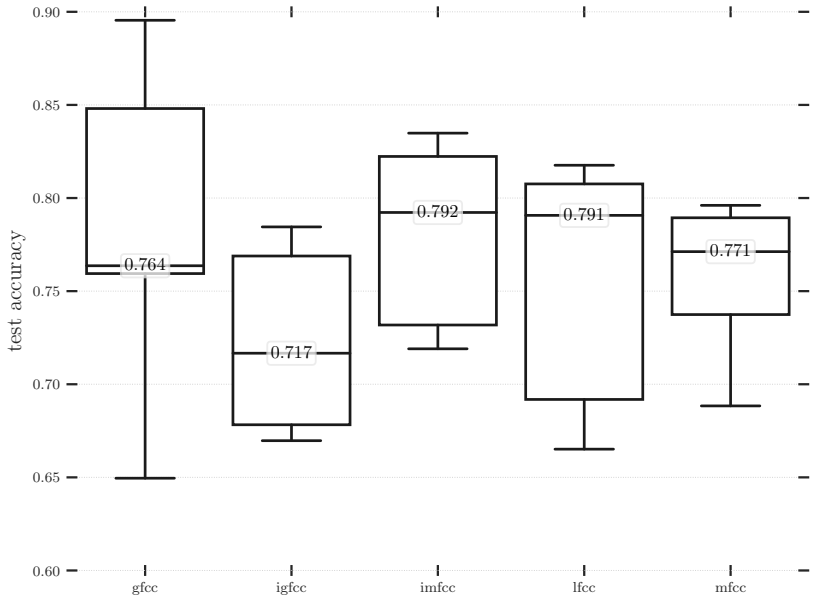
C.5.8 test receiver operator characteristic auc



Figur 115: *test receiver operator characteristic auc* for truncated a silence model silence silence, *test silence*

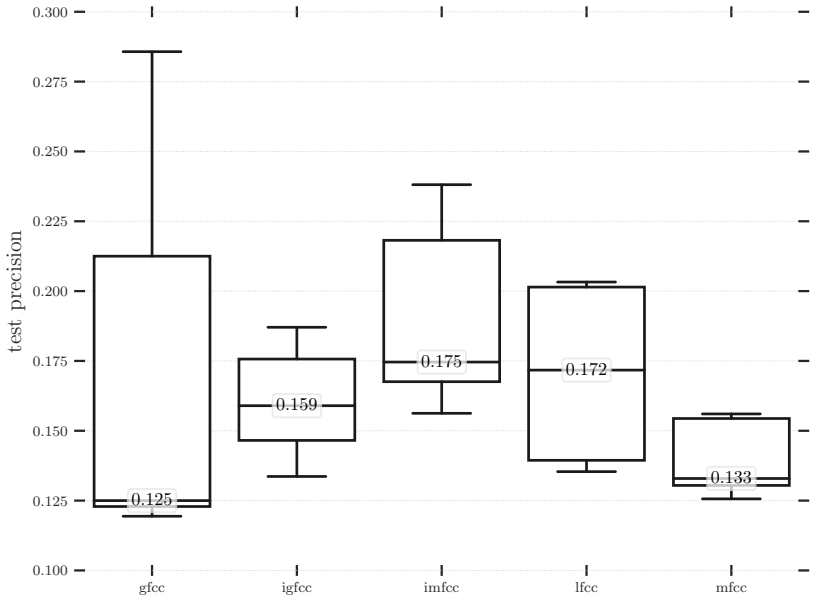
C.6 noises all snr to pcafeter model all noise snr all noise snr

C.6.1 test accuracy



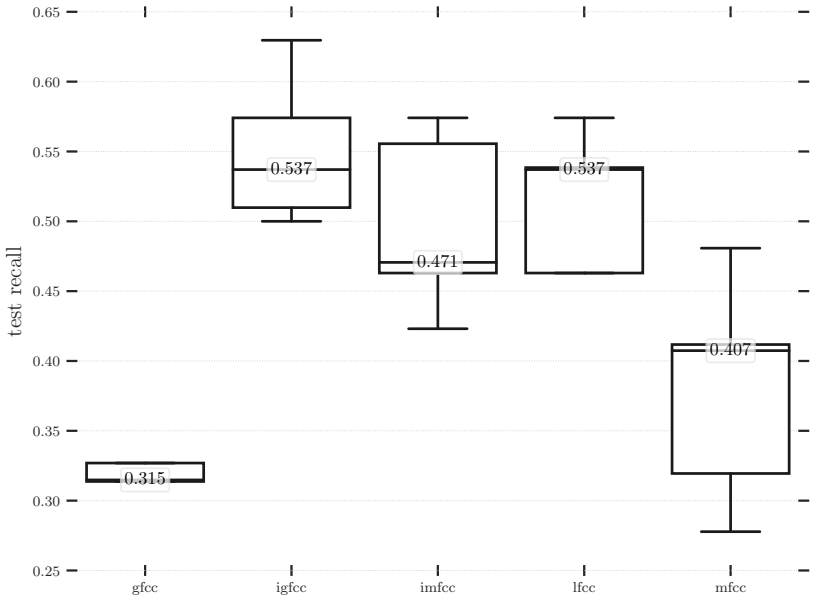
Figur 116: *test accuracy* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 0db*

C.6.2 test precision



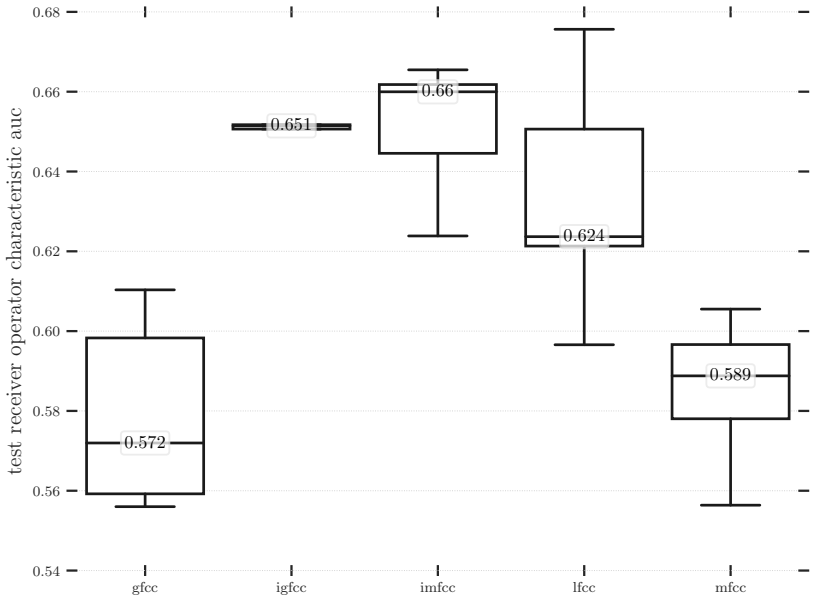
Figur 117: *test precision* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 0db*

C.6.3 test recall



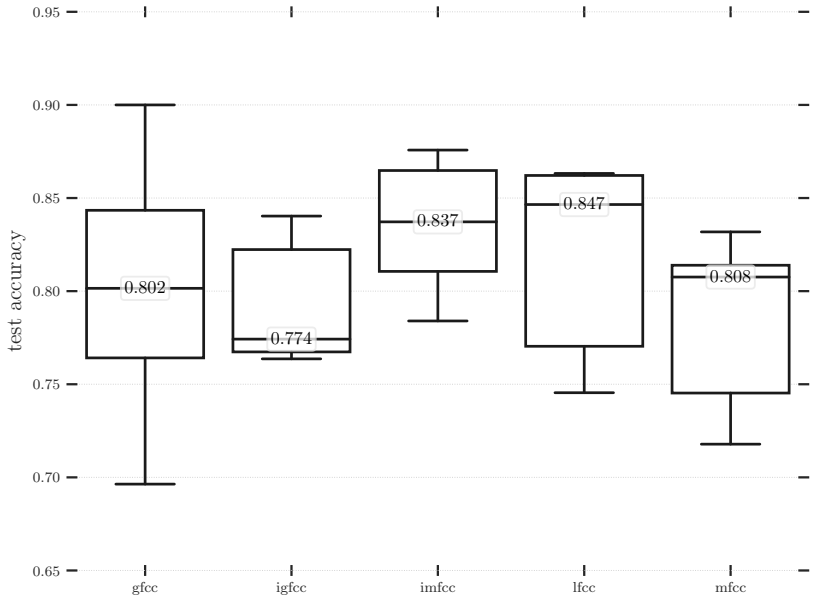
Figur 118: *test recall* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 0db*

C.6.4 test receiver operator characteristic auc



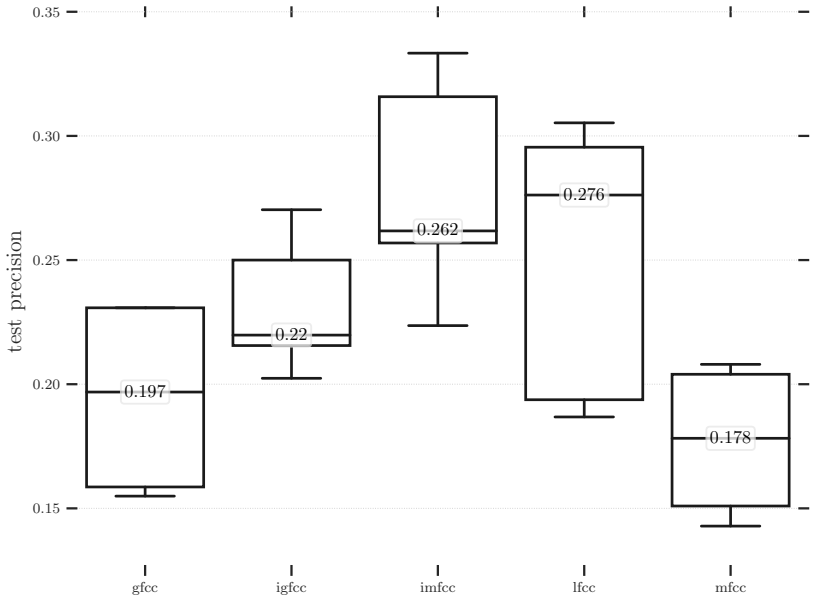
Figur 119: *test receiver operator characteristic auc* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 0db*

C.6.5 test accuracy



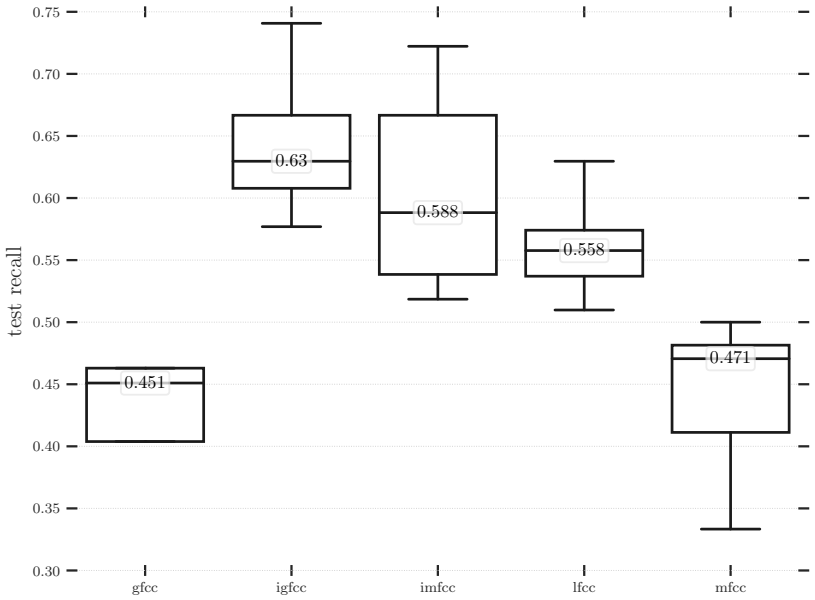
Figur 120: *test accuracy* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 5db*

C.6.6 test precision



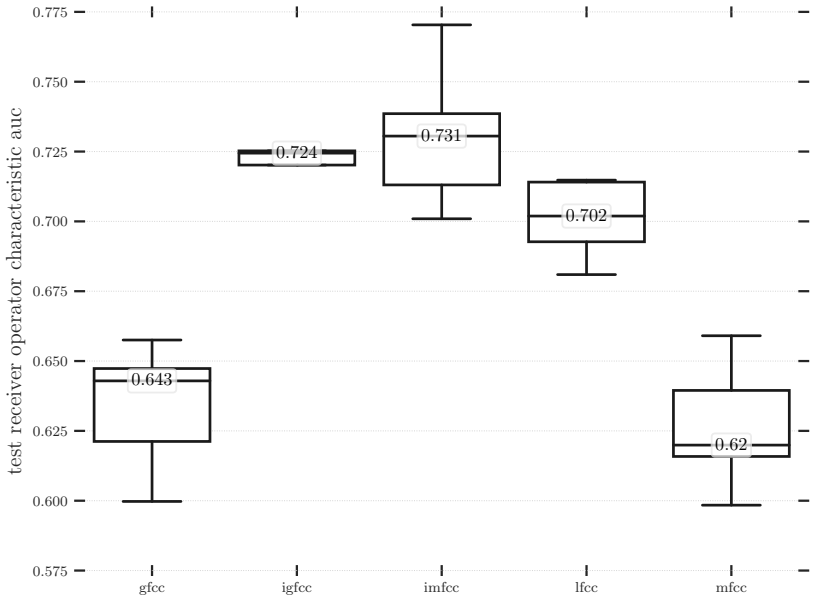
Figur 121: *test precision* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 5db*

C.6.7 test recall



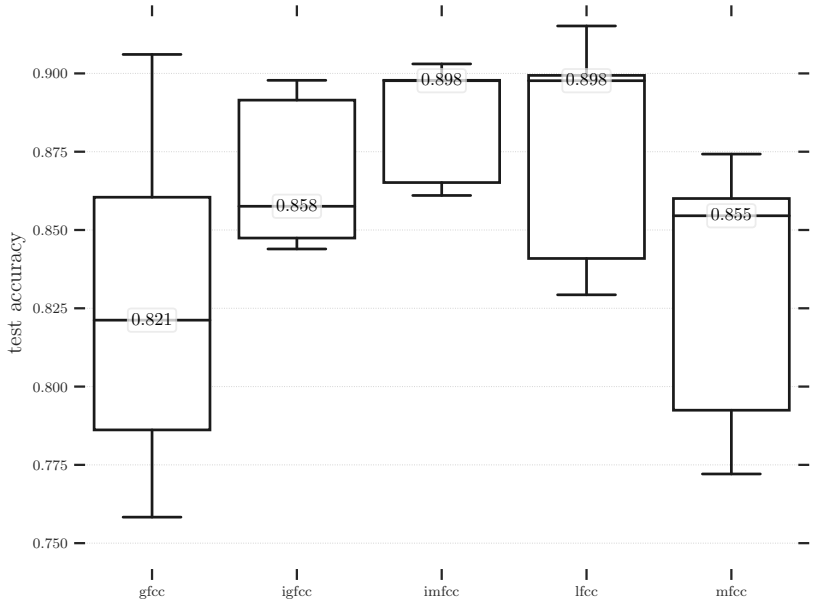
Figur 122: *test recall* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 5db*

C.6.8 test receiver operator characteristic auc



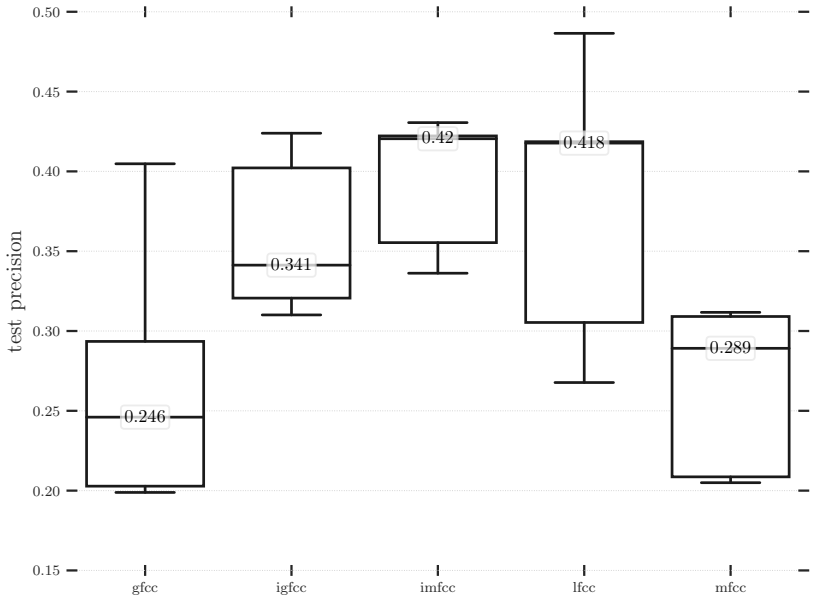
Figur 123: *test receiver operator characteristic auc* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 5db*

C.6.9 test accuracy



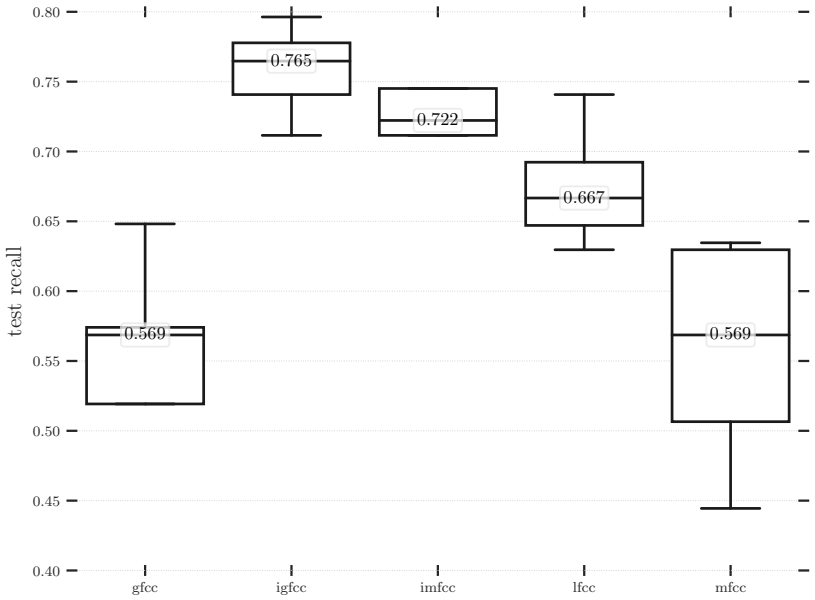
Figur 124: *test accuracy* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 10db*

C.6.10 test precision



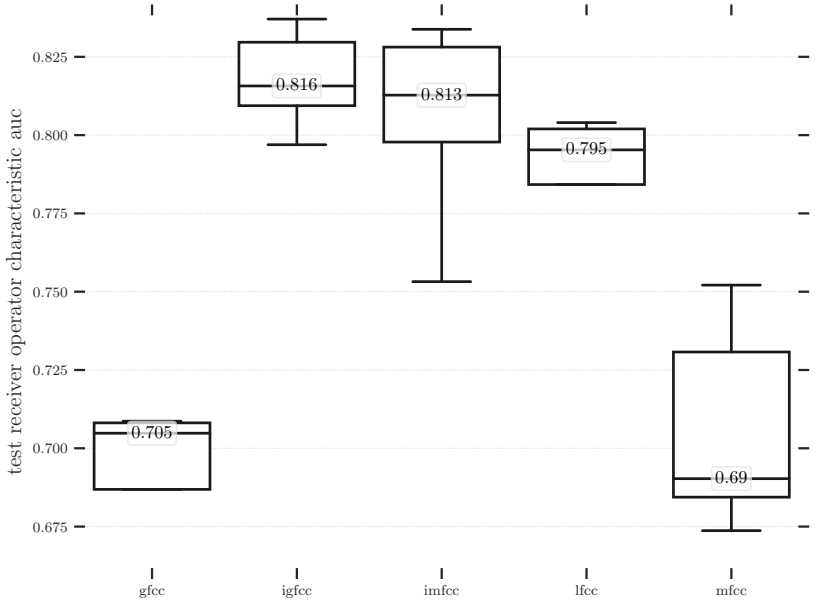
Figur 125: *test precision* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 10db*

C.6.11 test recall



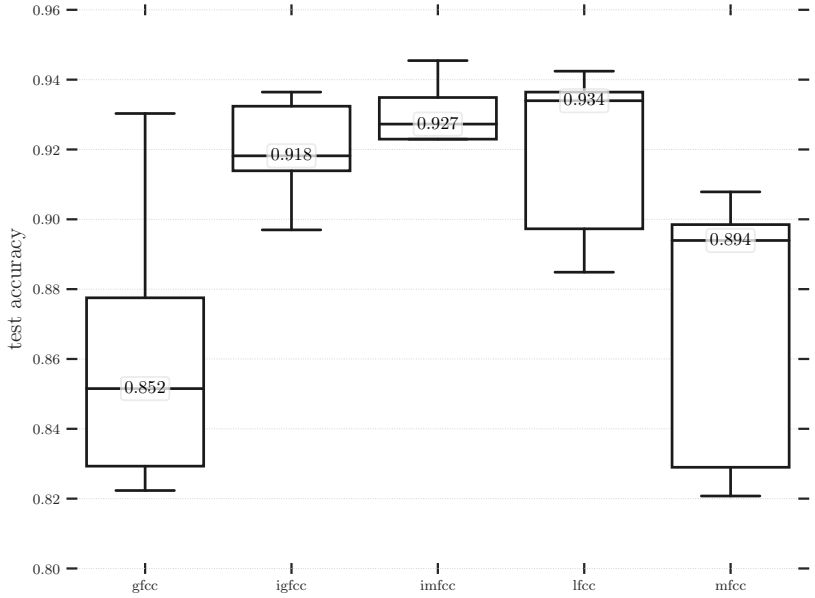
Figur 126: *test recall* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 10db*

C.6.12 test receiver operator characteristic auc



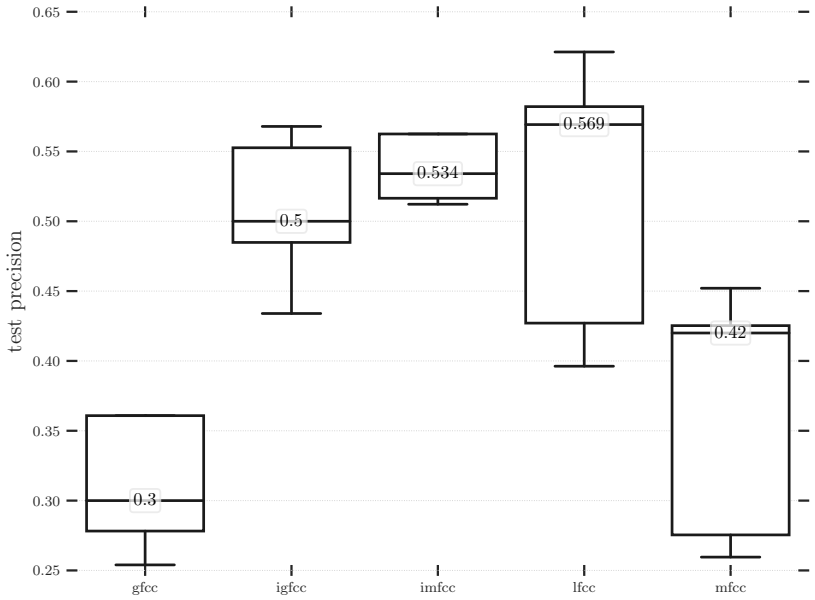
Figur 127: *test receiver operator characteristic auc* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 10db*

C.6.13 test accuracy



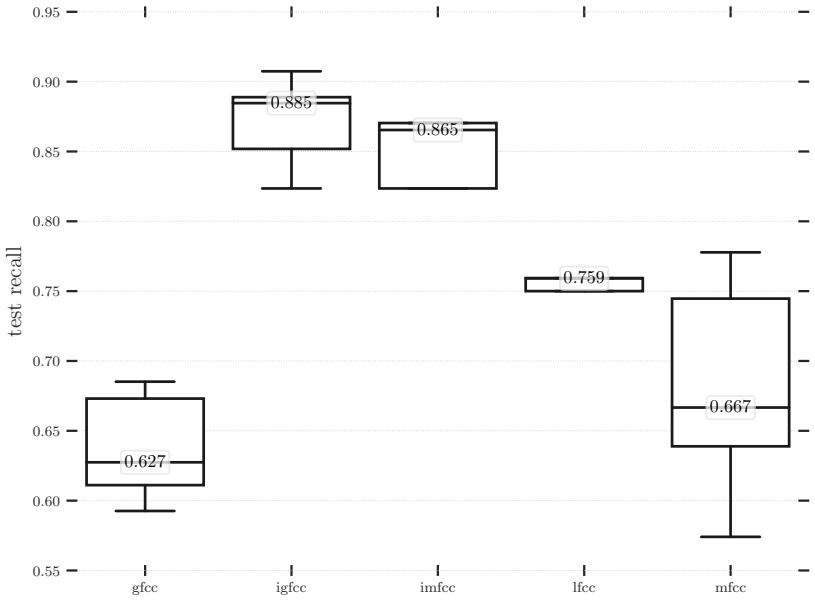
Figur 128: *test accuracy* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 15db*

C.6.14 test precision



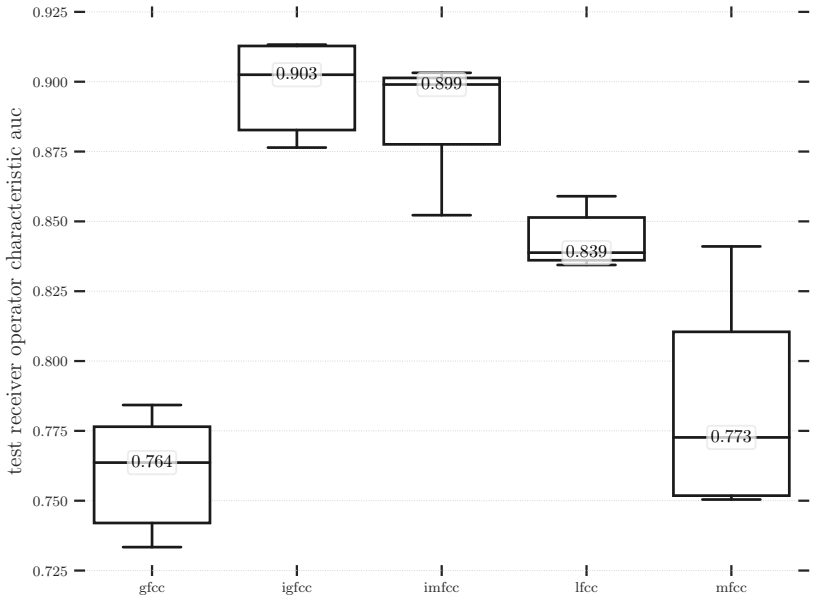
Figur 129: *test precision* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 15db*

C.6.15 test recall



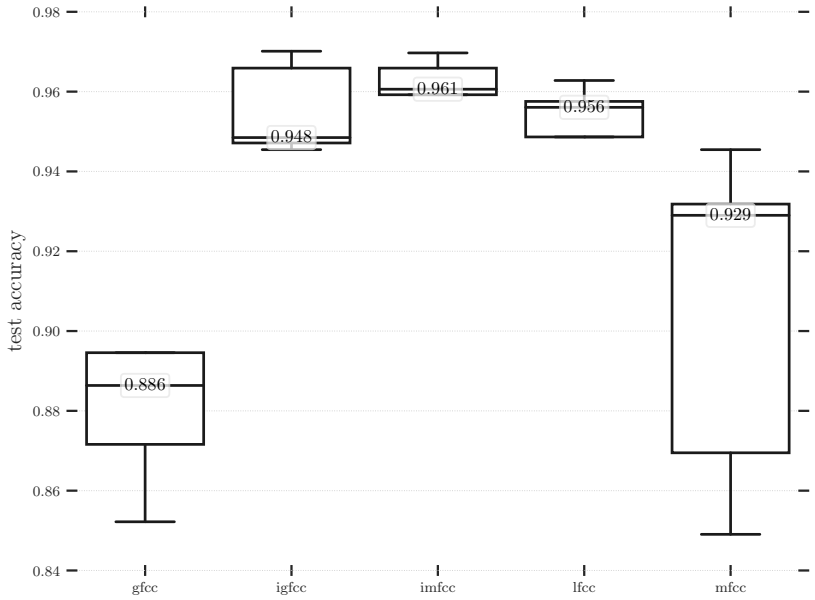
Figur 130: *test recall* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 15db*

C.6.16 test receiver operator characteristic auc



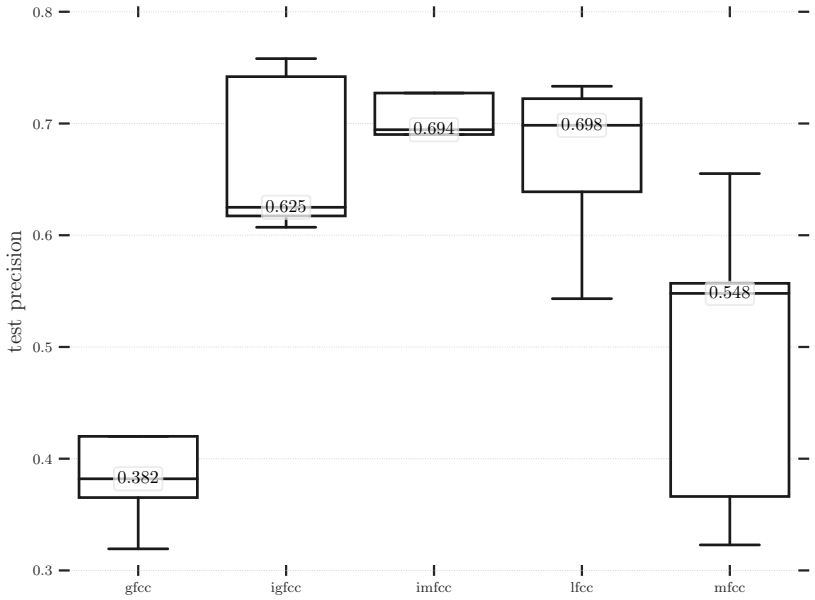
Figur 131: *test receiver operator characteristic auc* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 15db*

C.6.17 test accuracy



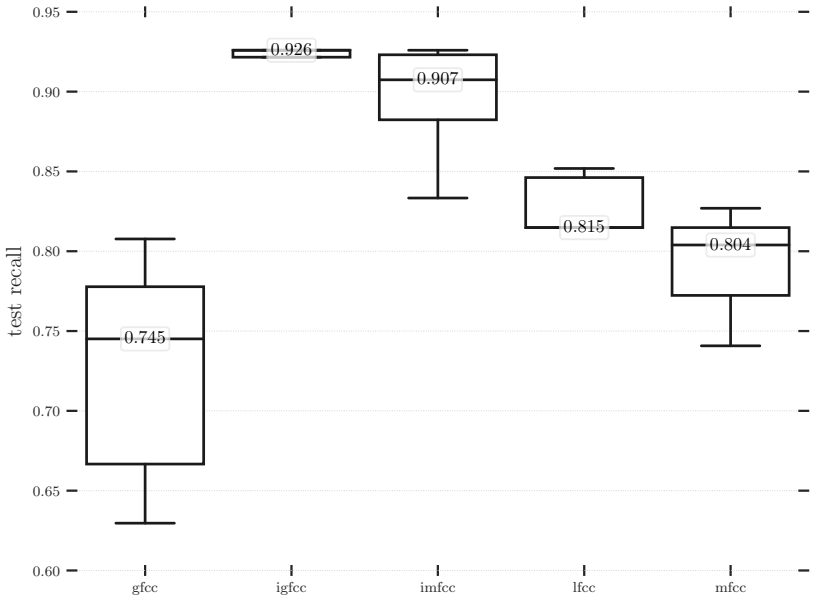
Figur 132: *test accuracy* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 20db*

C.6.18 test precision



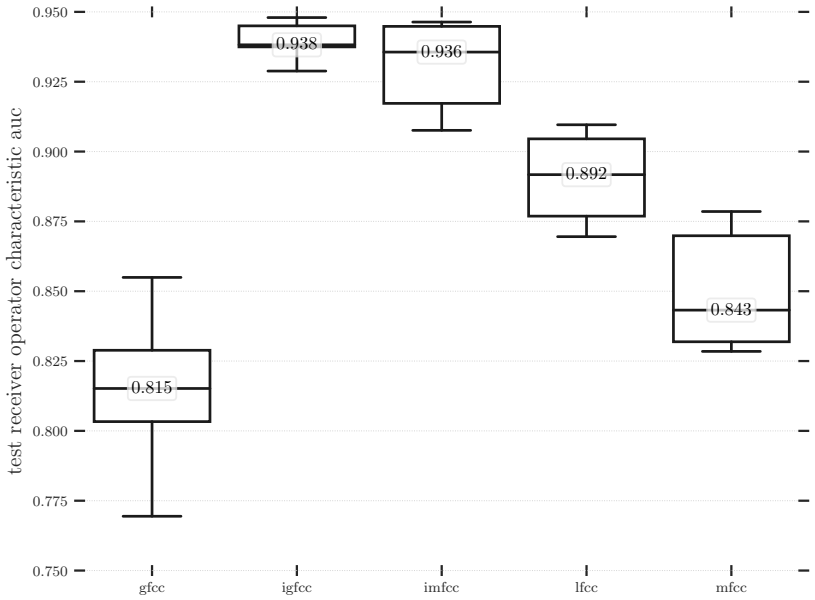
Figur 133: *test precision* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 20db*

C.6.19 test recall



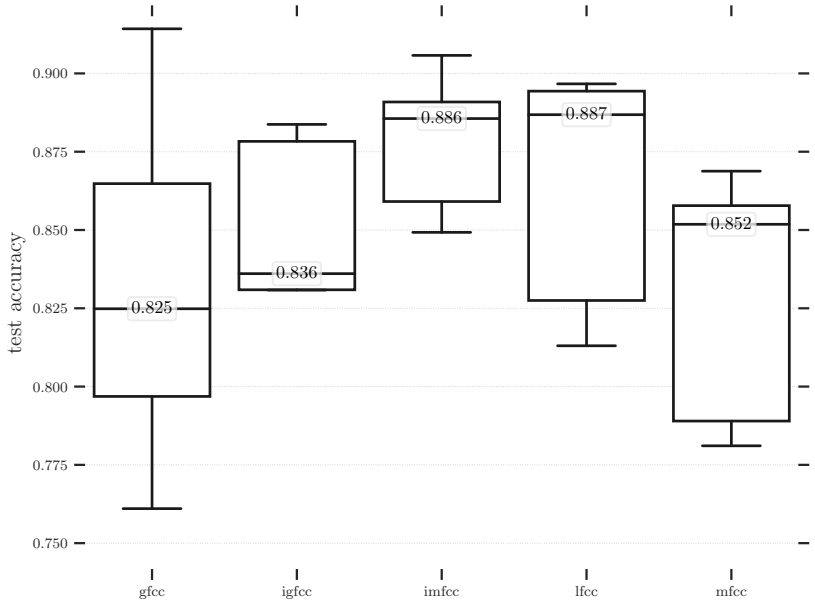
Figur 134: *test recall* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 20db*

C.6.20 test receiver operator characteristic auc



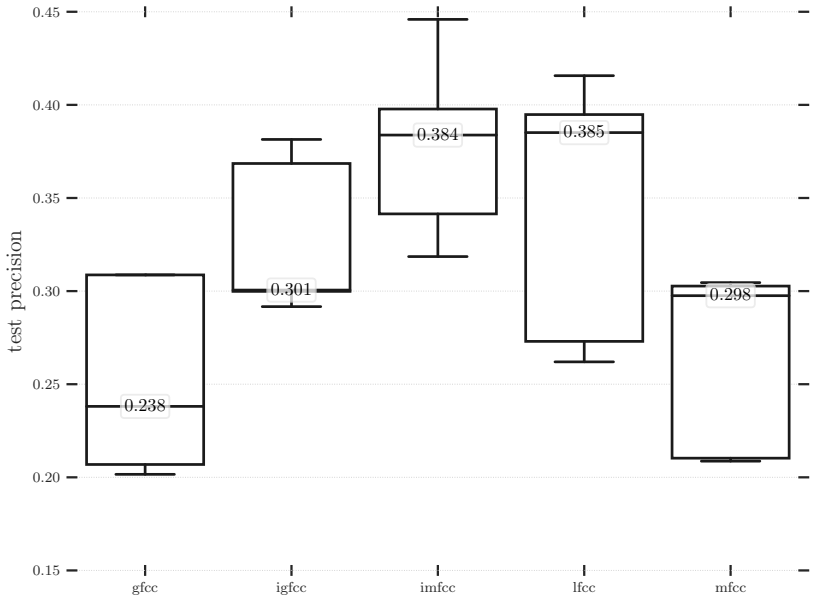
Figur 135: *test receiver operator characteristic auc* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 20db*

C.6.21 test accuracy



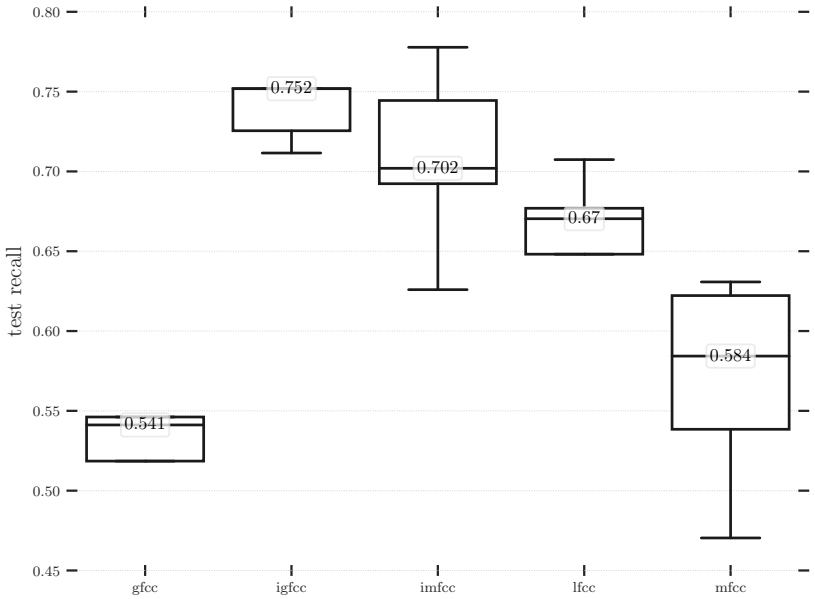
Figur 136: *test accuracy* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 0db pcafeter snr 5db pcafeter snr 10db pcafeter snr 15db pcafeter snr 20db*

C.6.22 test precision



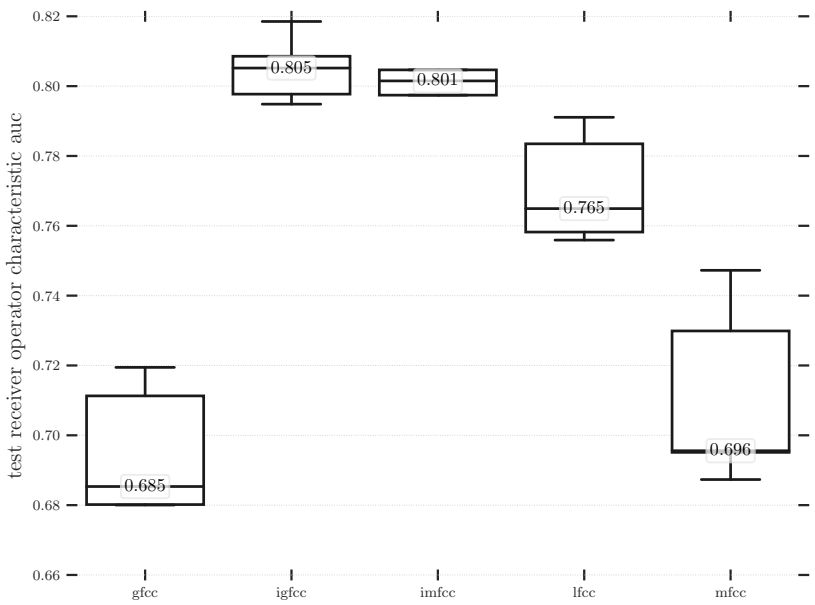
Figur 137: *test precision* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 0db pcafeter snr 5db pcafeter snr 10db pcafeter snr 15db pcafeter snr 20db*

C.6.23 test recall



Figur 138: *test recall* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 0db pcafeter snr 5db pcafeter snr 10db pcafeter snr 15db pcafeter snr 20db*

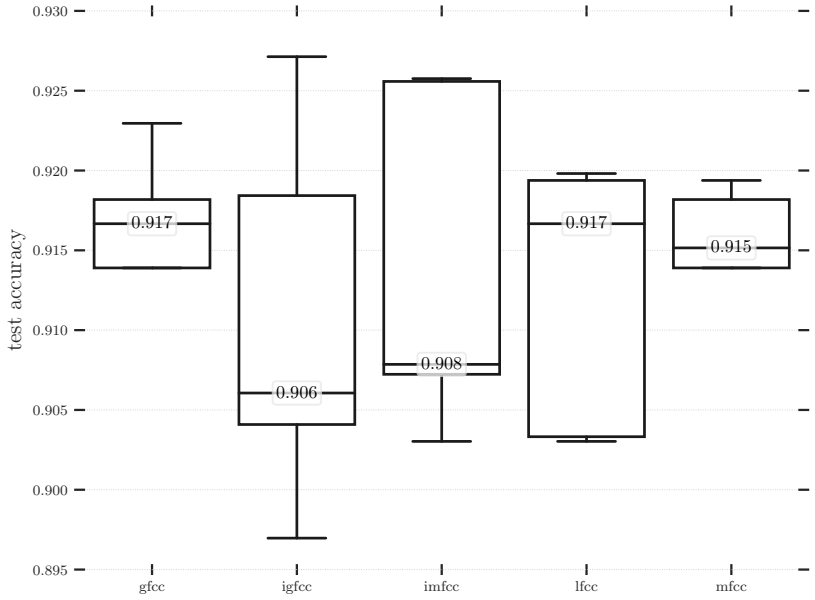
C.6.24 test receiver operator characteristic auc



Figur 139: *test receiver operator characteristic auc* for noises all snr to pcafeter model all noise snr all noise snr, *test pcafeter snr 0db pcafeter snr 5db pcafeter snr 10db pcafeter snr 15db pcafeter snr 20db*

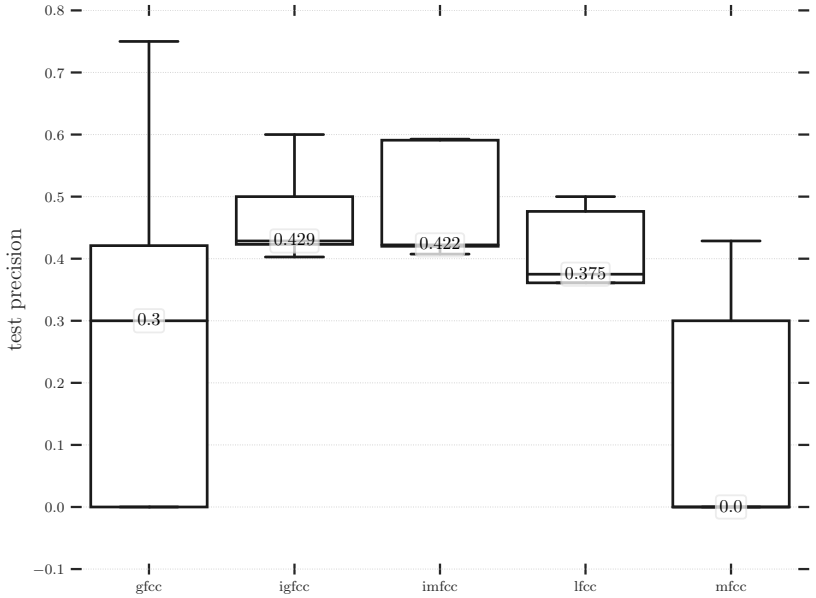
C.7 bbl morten

C.7.1 test accuracy



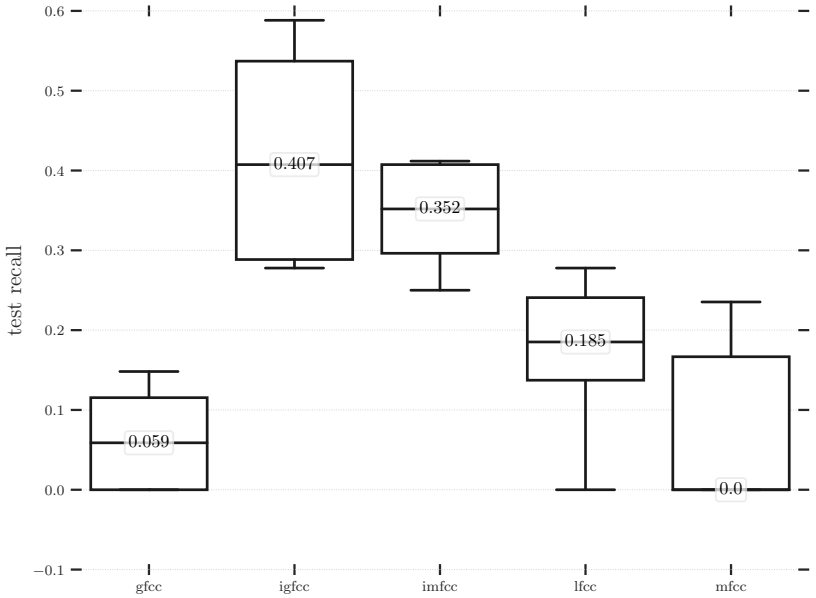
Figur 140: *test accuracy* for bbl morten snr 0db

C.7.2 test precision



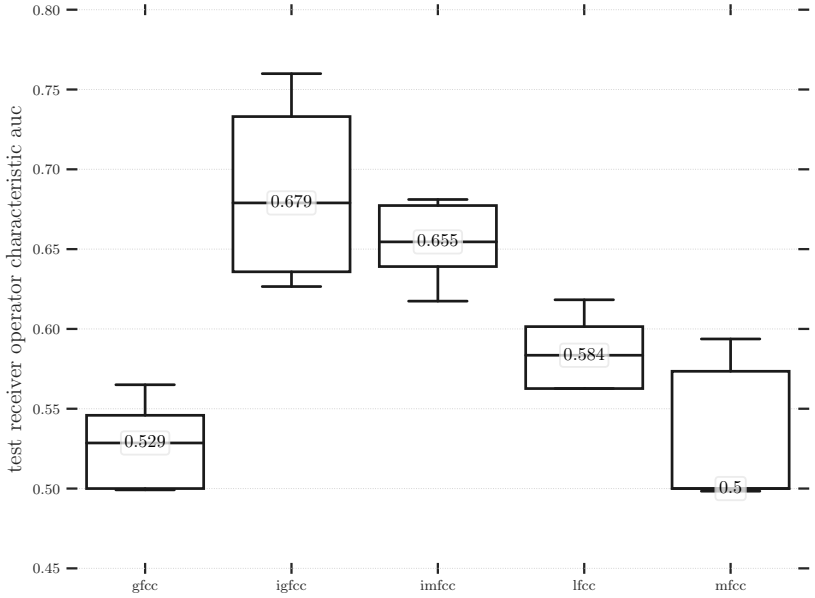
Figur 141: *test precision* for bbl morten snr 0db

C.7.3 test recall



Figur 142: *test recall* for bbl morten snr 0db

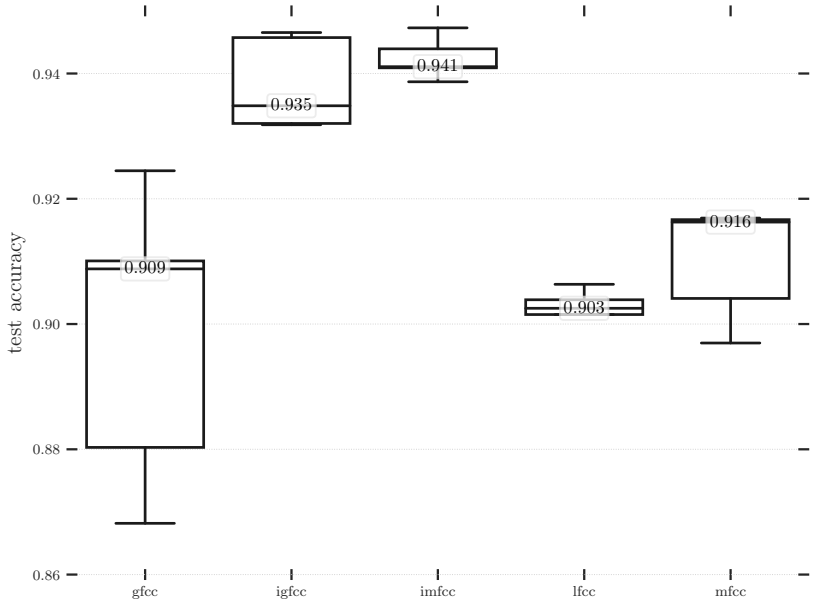
C.7.4 test receiver operator characteristic auc



Figur 143: *test receiver operator characteristic auc* for bbl morten snr 0db

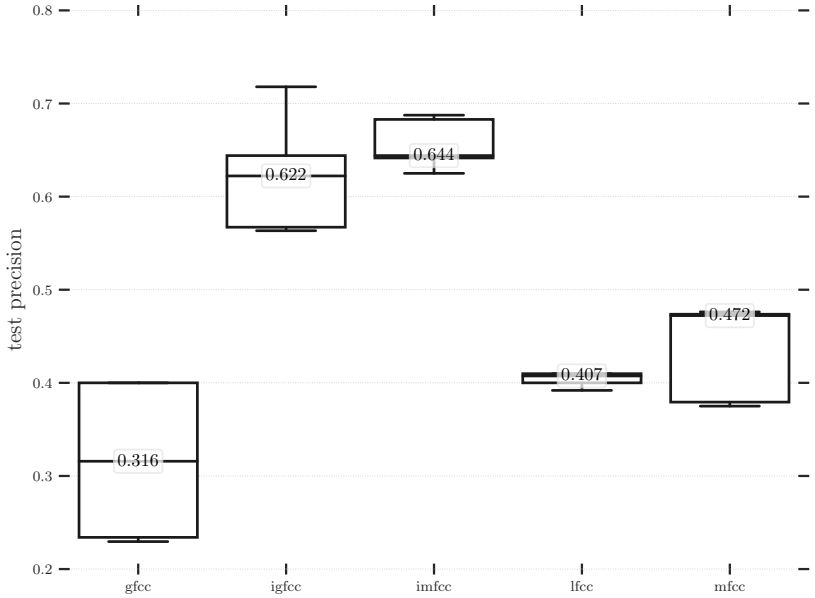
C.8 bbl morten

C.8.1 test accuracy



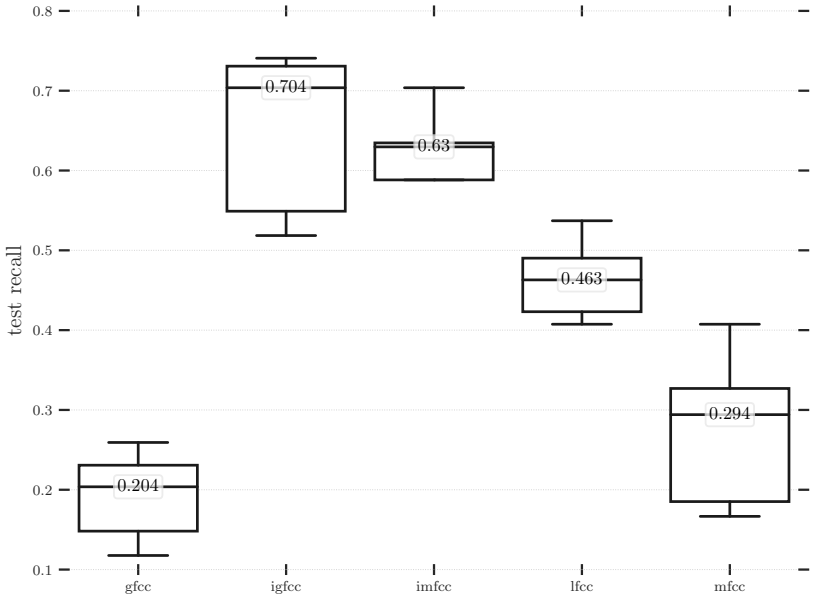
Figur 144: *test accuracy* for bbl morten snr 5db

C.8.2 test precision



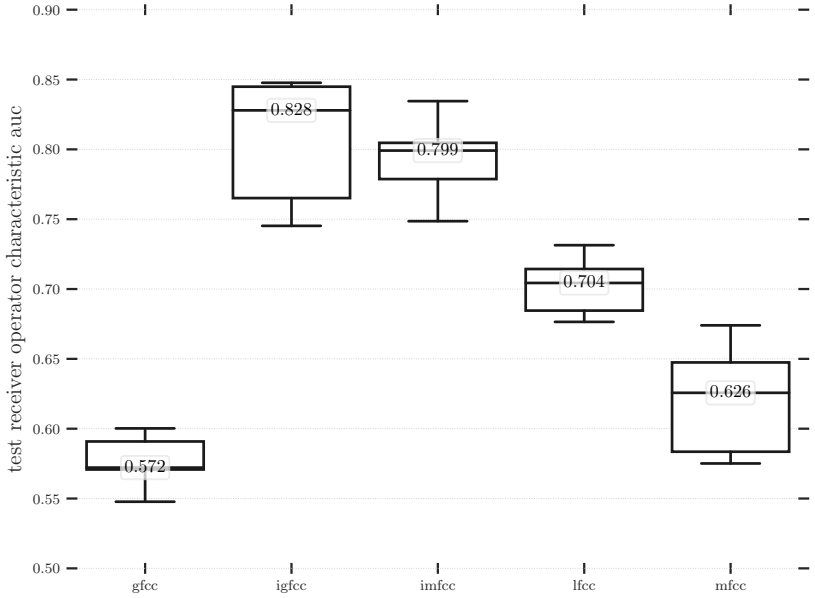
Figur 145: *test precision* for bbl morten snr 5db

C.8.3 test recall



Figur 146: *test recall* for bbl morten snr 5db

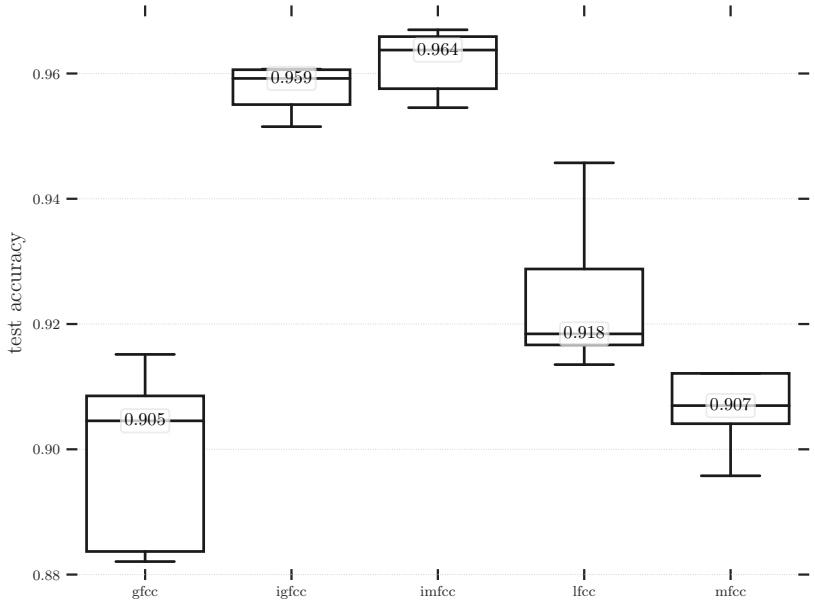
C.8.4 test receiver operator characteristic auc



Figur 147: *test receiver operator characteristic auc* for bbl morten snr 5db

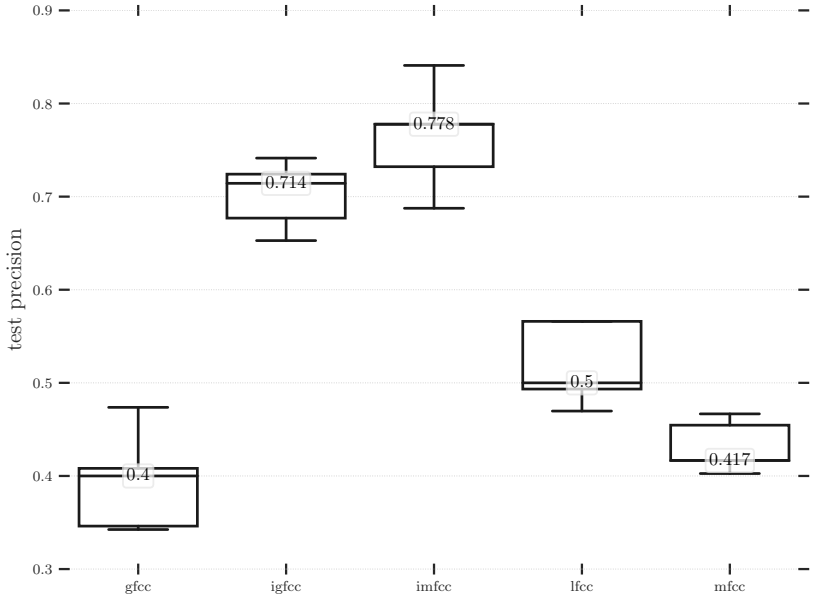
C.9 bbl morten

C.9.1 test accuracy



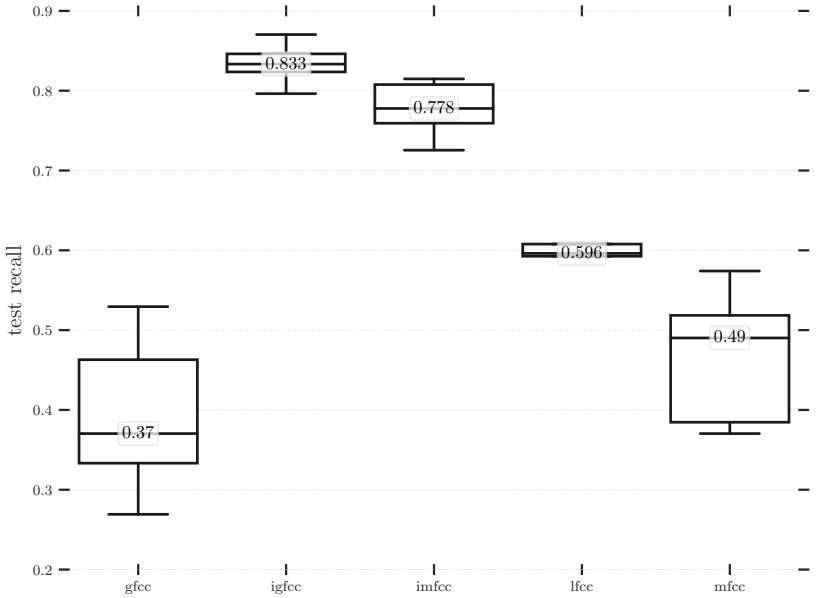
Figur 148: *test accuracy* for bbl morten snr 10db

C.9.2 test precision



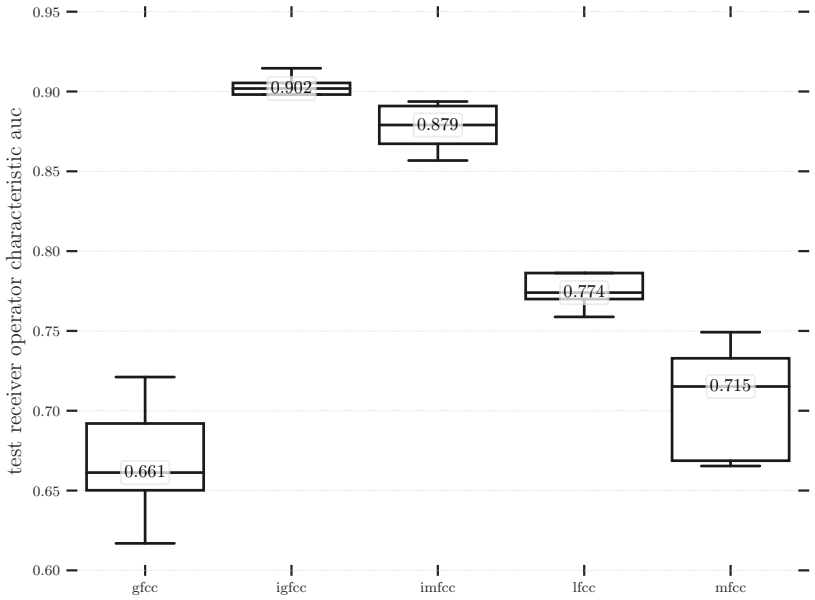
Figur 149: *test precision* for bbl morten snr 10db

C.9.3 test recall



Figur 150: *test recall* for bbl morten snr 10db

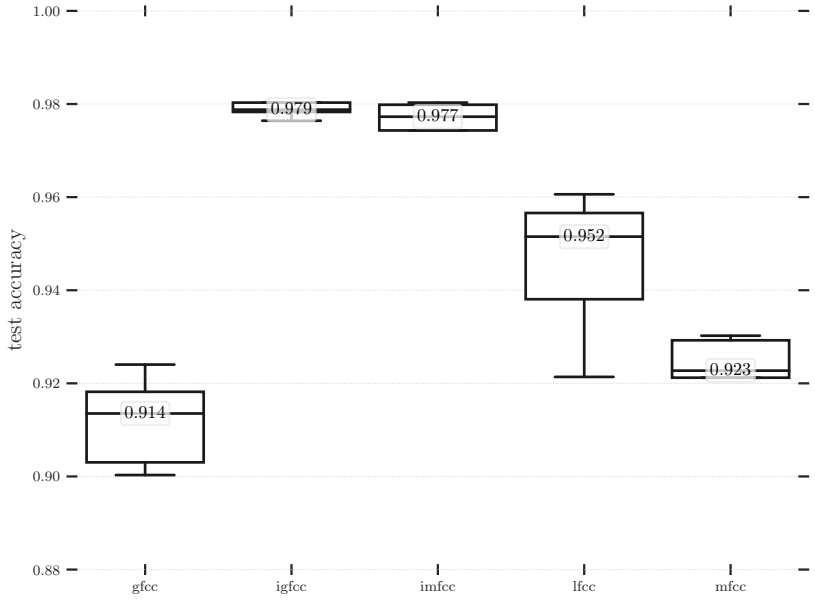
C.9.4 test receiver operator characteristic auc



Figur 151: *test receiver operator characteristic auc* for bbl morten snr 10db

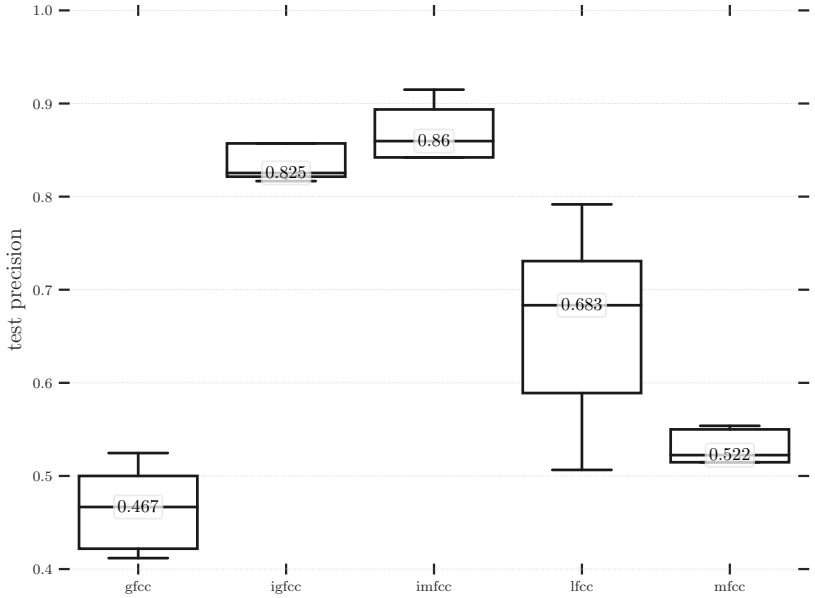
C.10 bbl morten

C.10.1 test accuracy



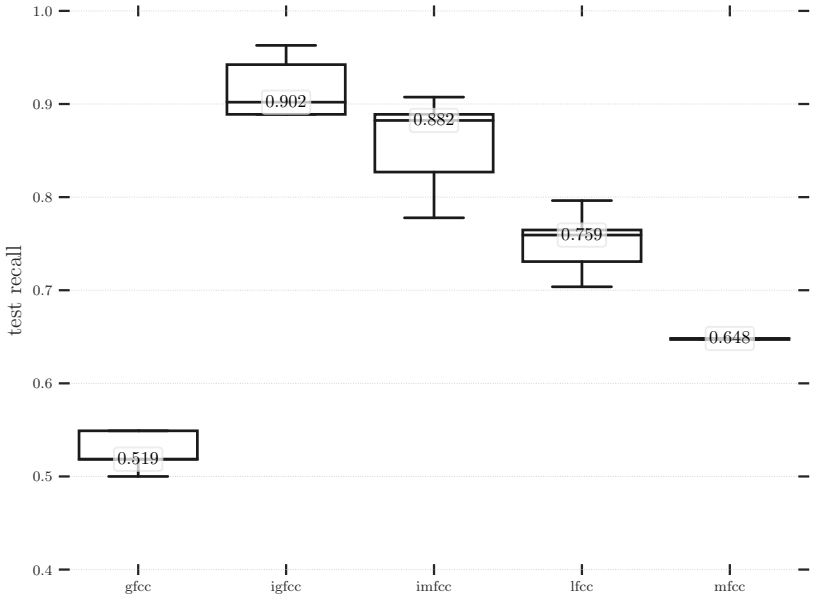
Figur 152: *test accuracy* for bbl morten snr 15db

C.10.2 test precision



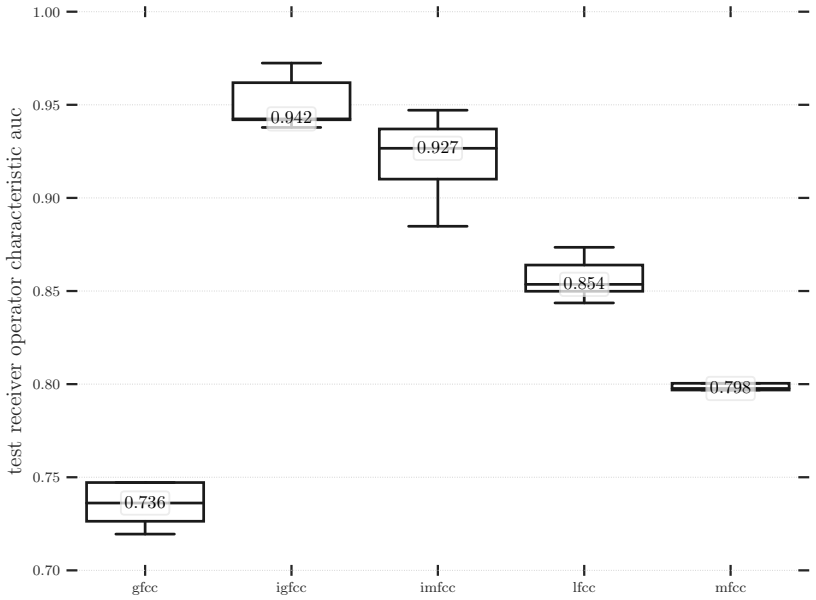
Figur 153: *test precision* for bbl morten snr 15db

C.10.3 test recall



Figur 154: *test recall* for bbl morten snr 15db

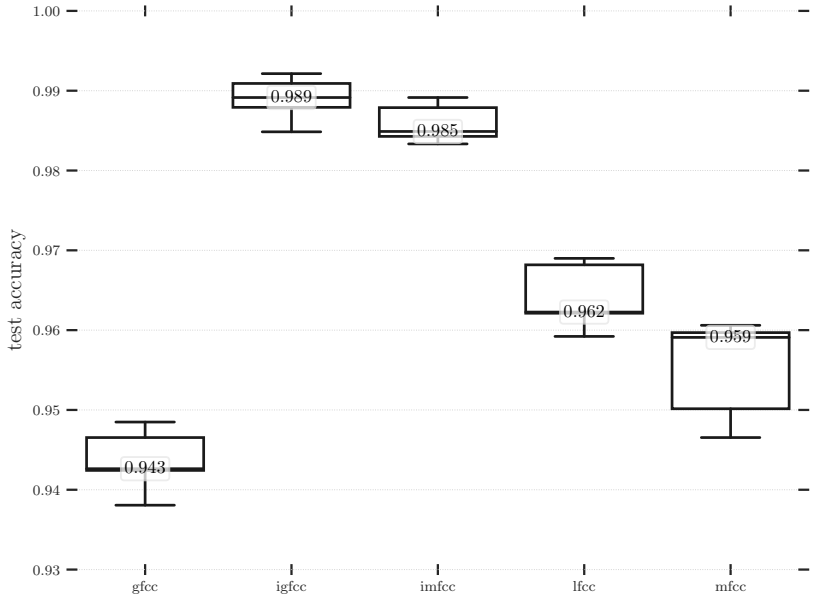
C.10.4 test receiver operator characteristic auc



Figur 155: *test receiver operator characteristic auc* for bbl morten snr 15db

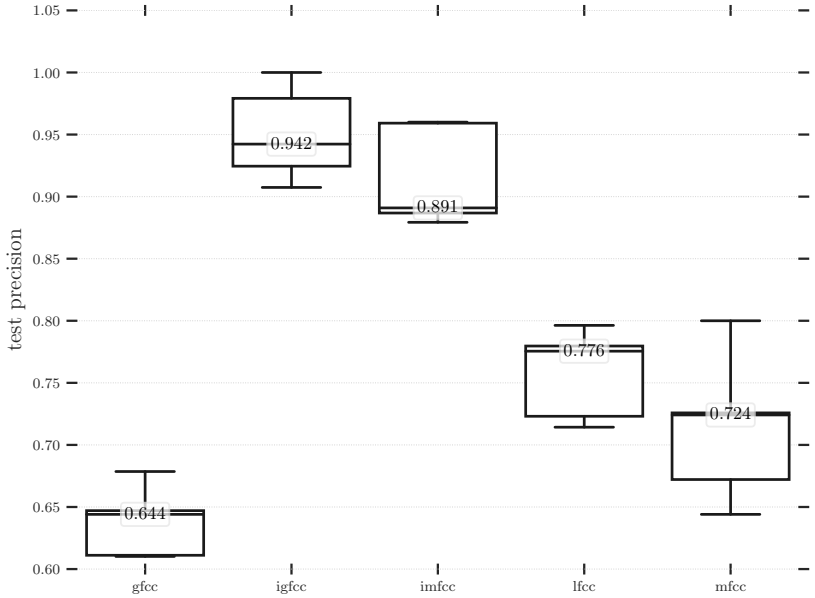
C.11 bbl morten

C.11.1 test accuracy



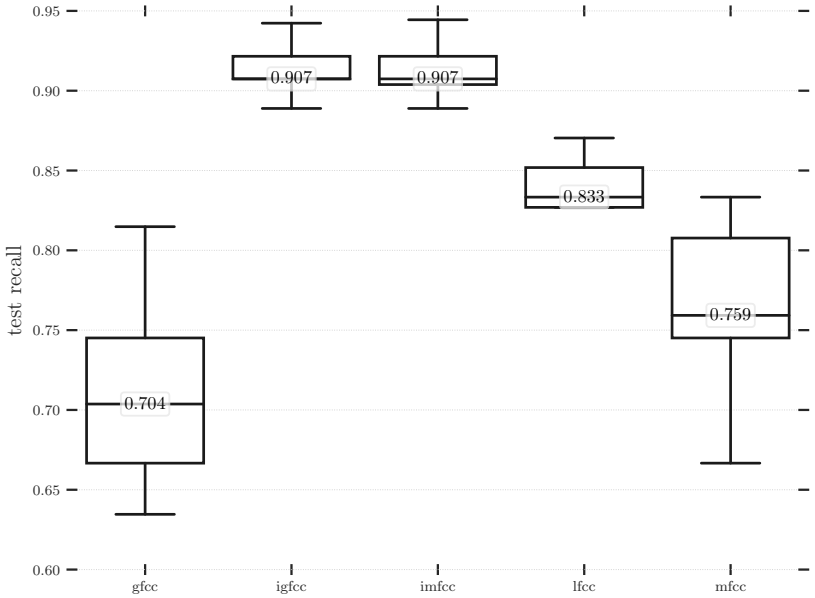
Figur 156: *test accuracy* for bbl morten snr 20db

C.11.2 test precision



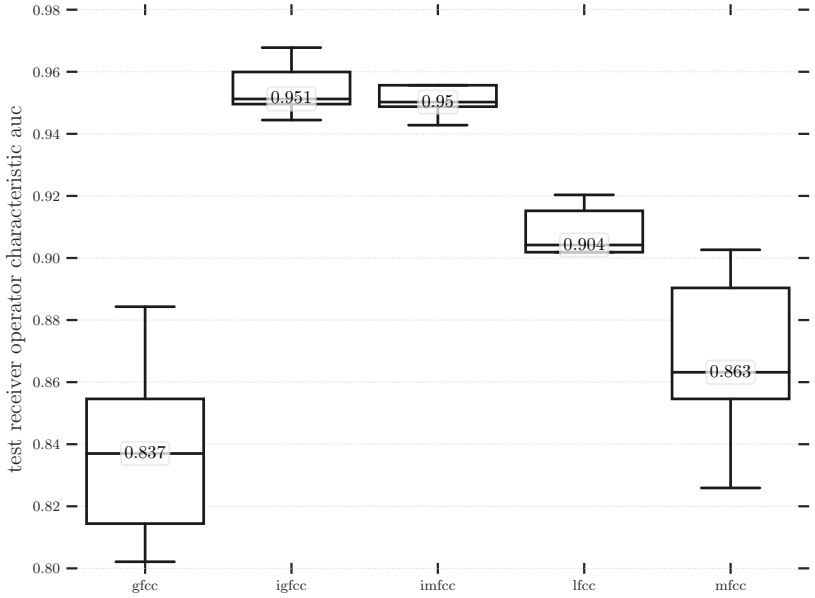
Figur 157: *test precision* for bbl morten snr 20db

C.11.3 test recall



Figur 158: *test recall* for bbl morten snr 20db

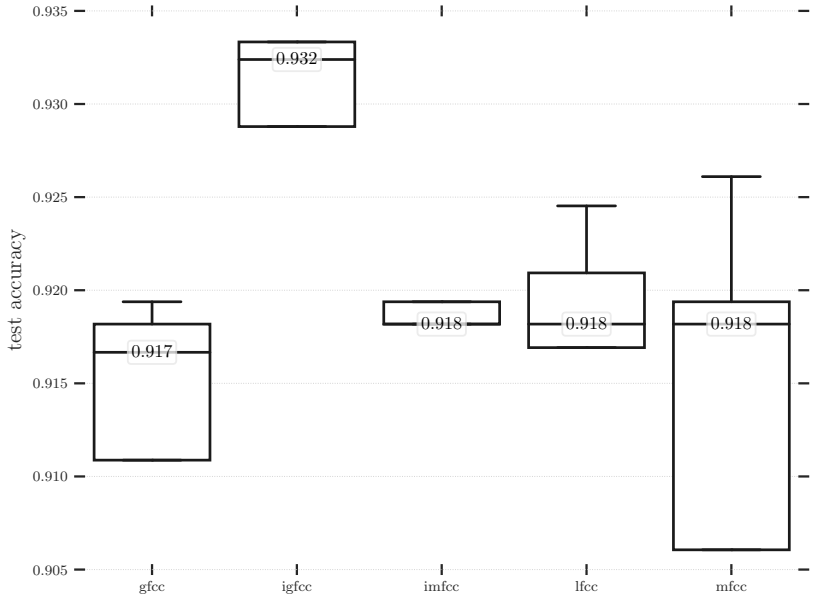
C.11.4 test receiver operator characteristic auc



Figur 159: *test receiver operator characteristic auc* for bbl morten snr 20db

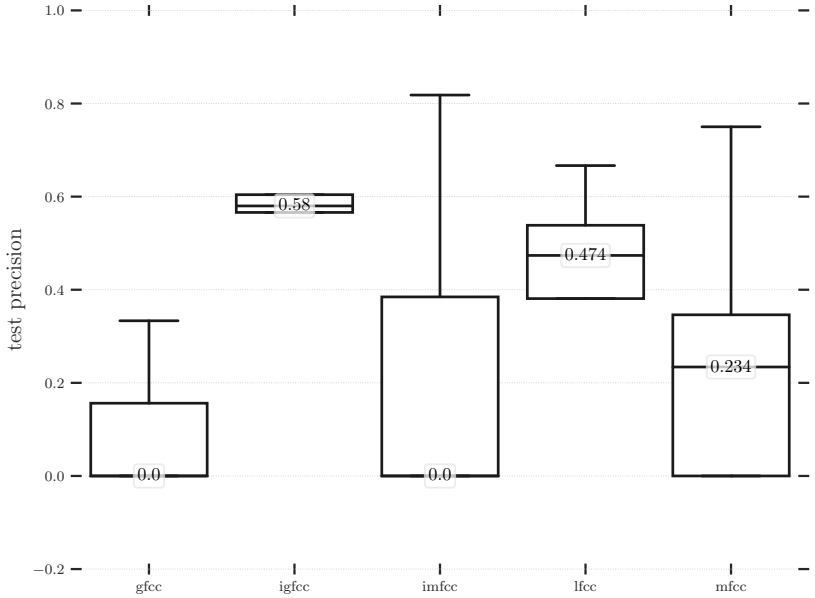
C.12 ssn morten

C.12.1 test accuracy



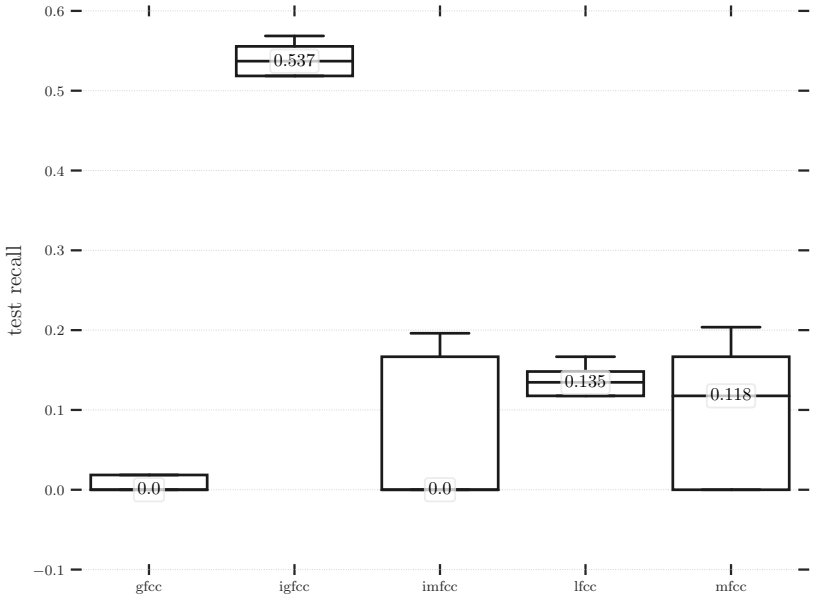
Figur 160: *test accuracy* for ssn morten snr 0db

C.12.2 test precision



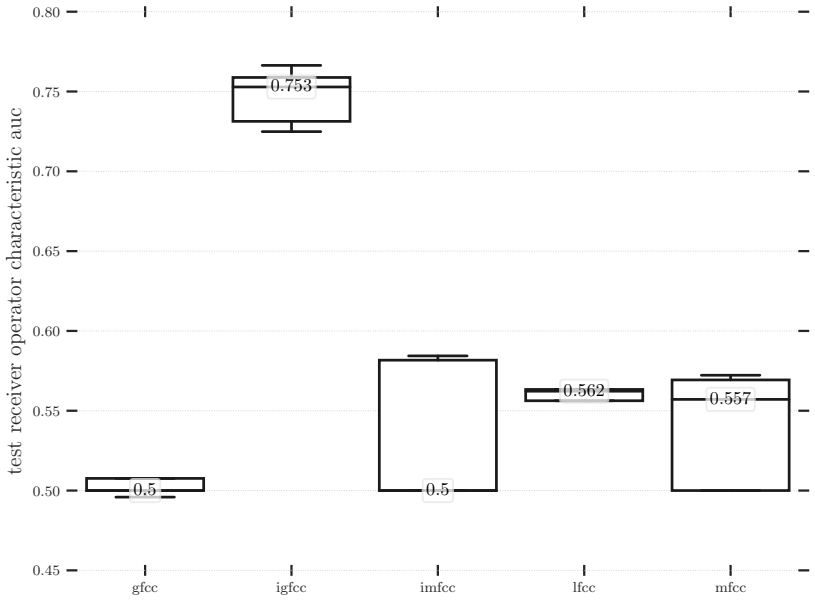
Figur 161: *test precision* for ssn morten snr 0db

C.12.3 test recall



Figur 162: *test recall* for ssn morten snr 0db

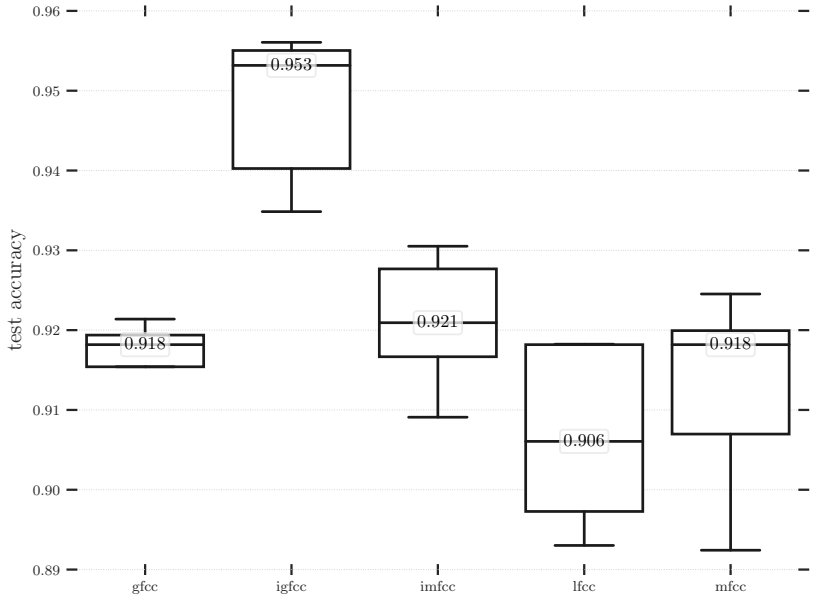
C.12.4 test receiver operator characteristic auc



Figur 163: *test receiver operator characteristic auc* for ssn morten snr 0db

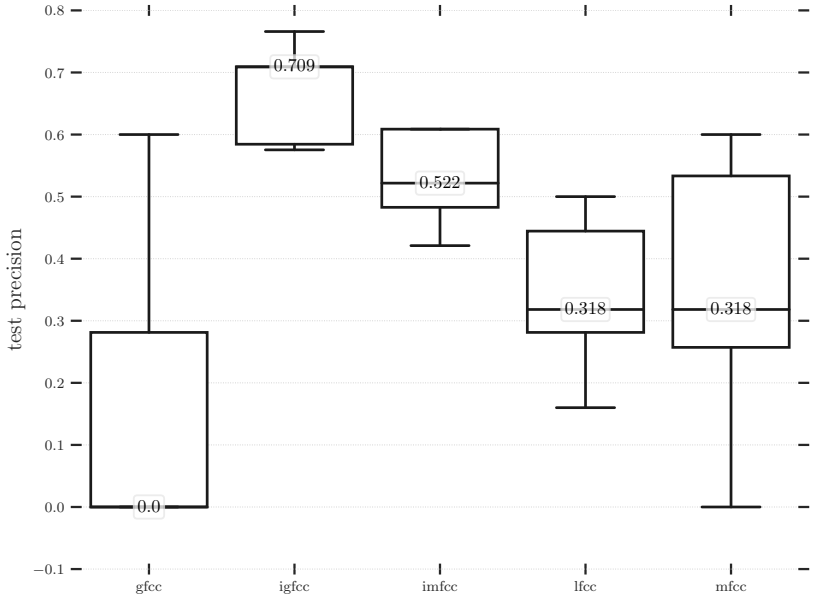
C.13 ssn morten

C.13.1 test accuracy



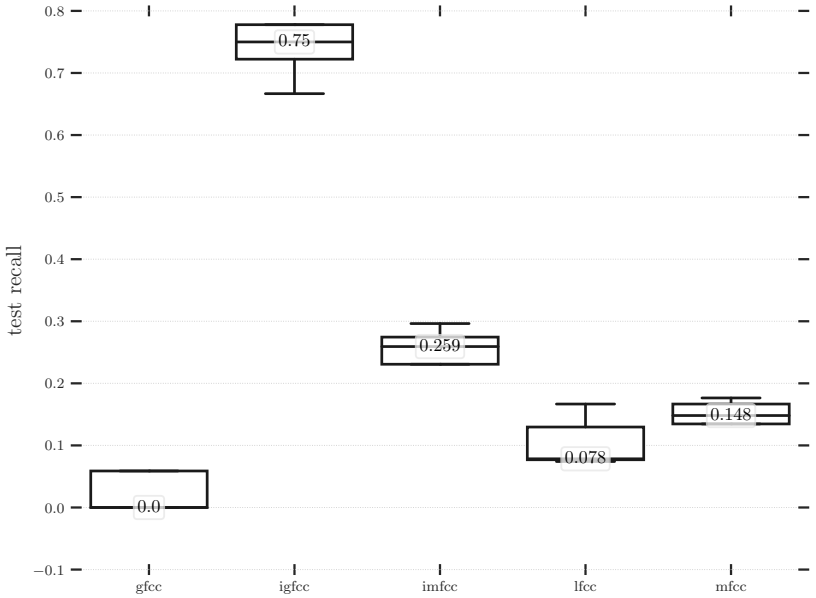
Figur 164: *test accuracy* for ssn morten snr 5db

C.13.2 test precision



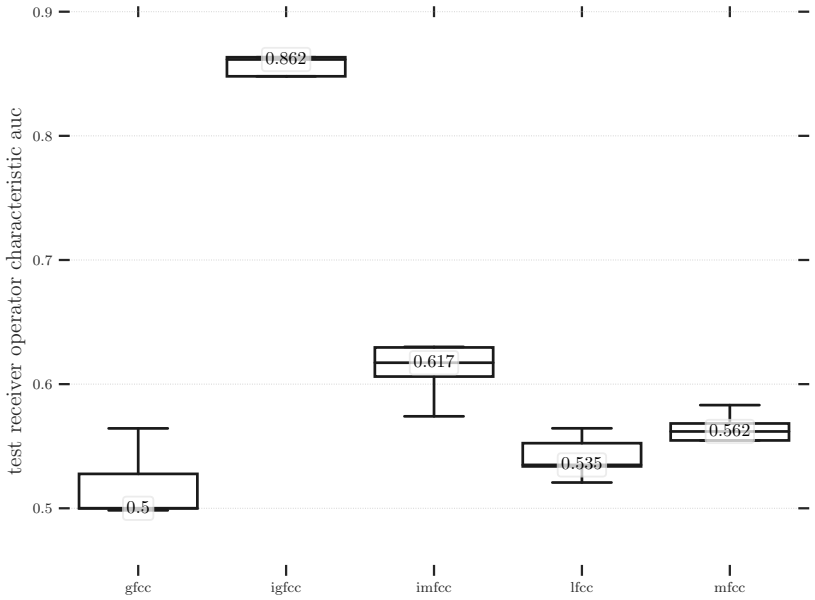
Figur 165: *test precision* for ssn morten snr 5db

C.13.3 test recall



Figur 166: *test recall* for ssn morten snr 5db

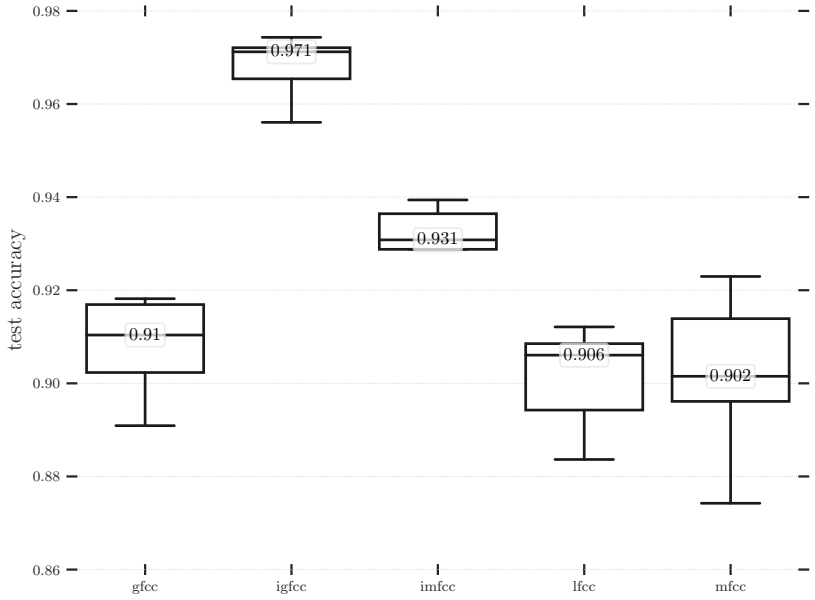
C.13.4 test receiver operator characteristic auc



Figur 167: *test receiver operator characteristic auc* for ssn morten snr 5db

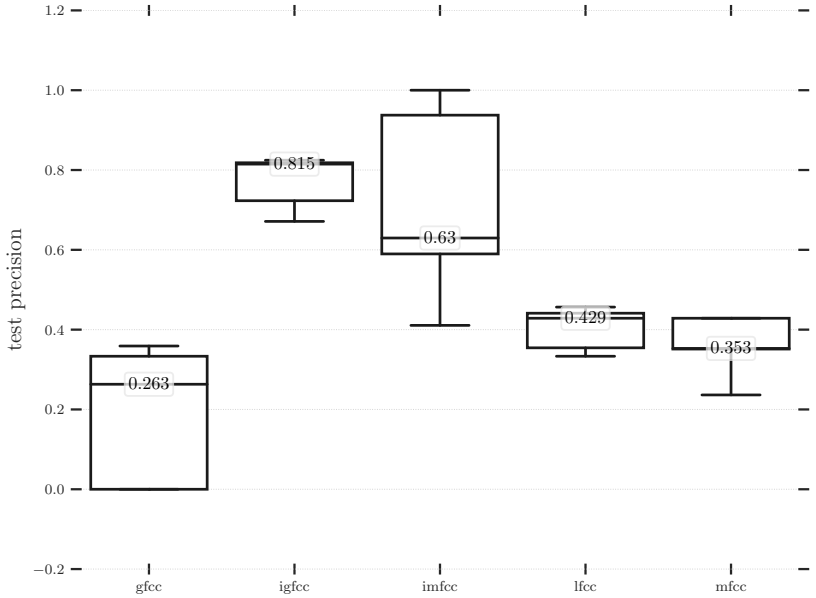
C.14 ssn morten

C.14.1 test accuracy



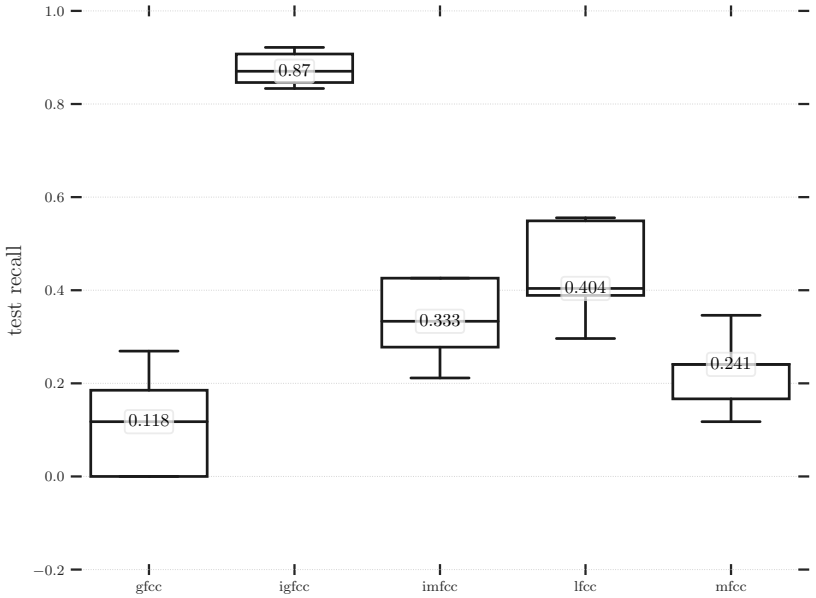
Figur 168: *test accuracy* for ssn morten snr 10db

C.14.2 test precision



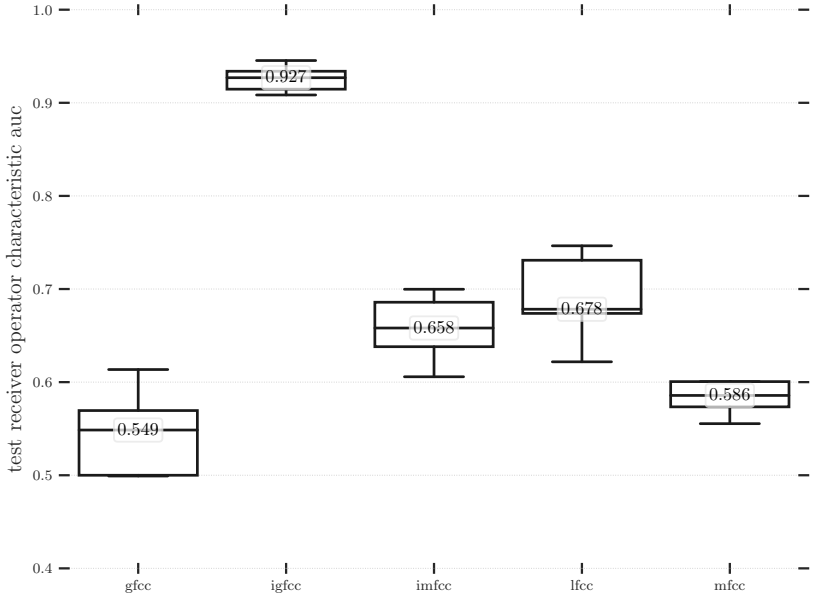
Figur 169: *test precision* for ssn morten snr 10db

C.14.3 test recall



Figur 170: *test recall* for ssn morten snr 10db

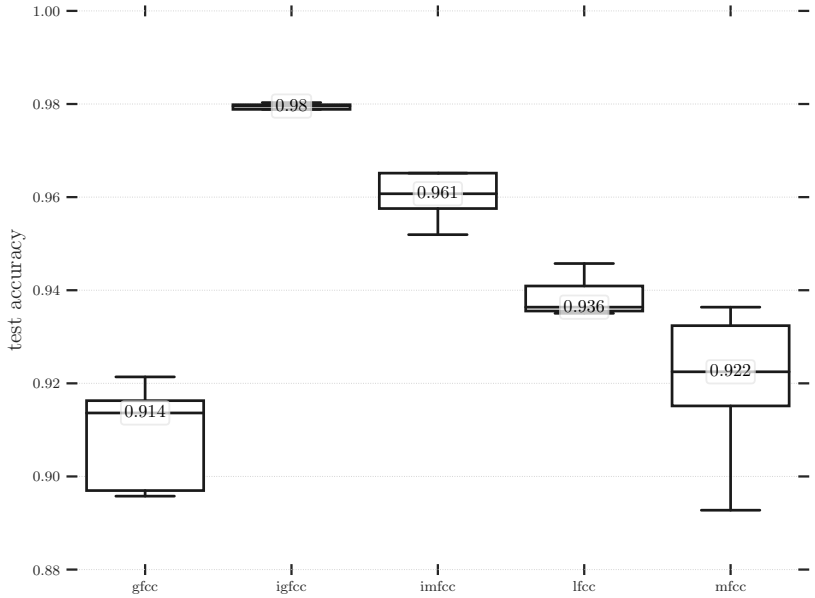
C.14.4 test receiver operator characteristic auc



Figur 171: *test receiver operator characteristic auc* for ssn morten snr 10db

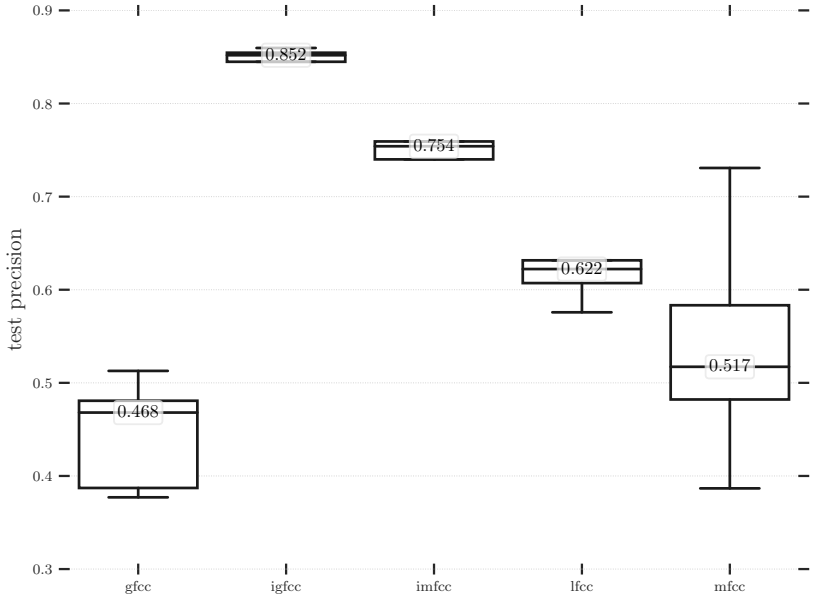
C.15 ssn morten

C.15.1 test accuracy



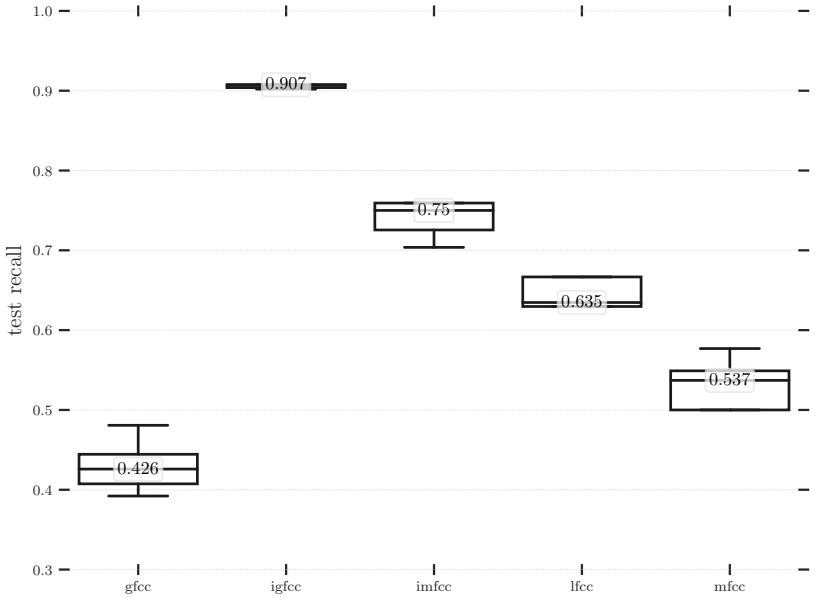
Figur 172: *test accuracy* for ssn morten snr 15db

C.15.2 test precision



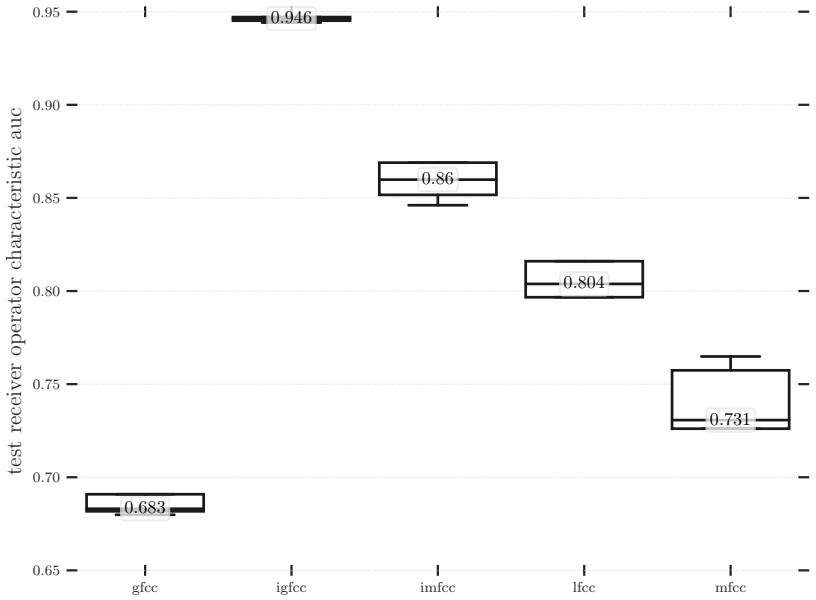
Figur 173: *test precision* for ssn morten snr 15db

C.15.3 test recall



Figur 174: *test recall* for ssn morten snr 15db

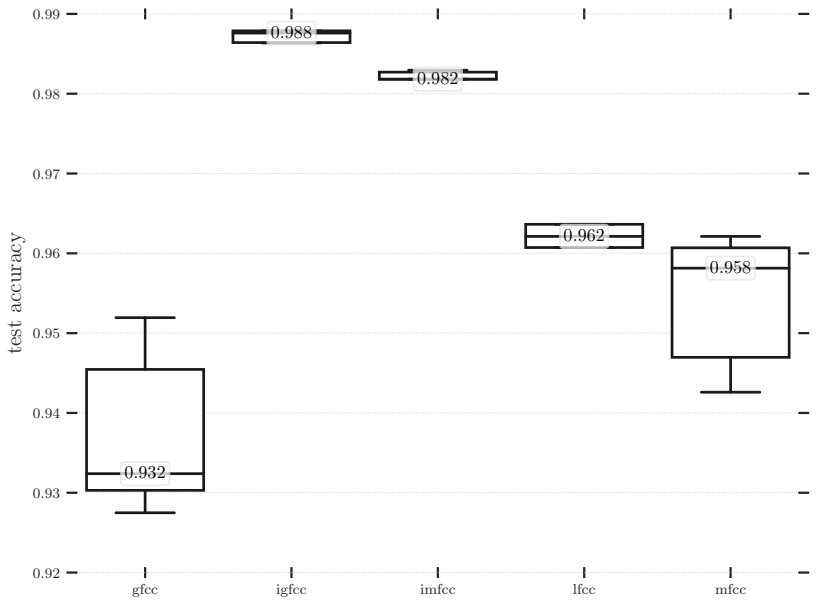
C.15.4 test receiver operator characteristic auc



Figur 175: *test receiver operator characteristic auc* for ssn morten snr 15db

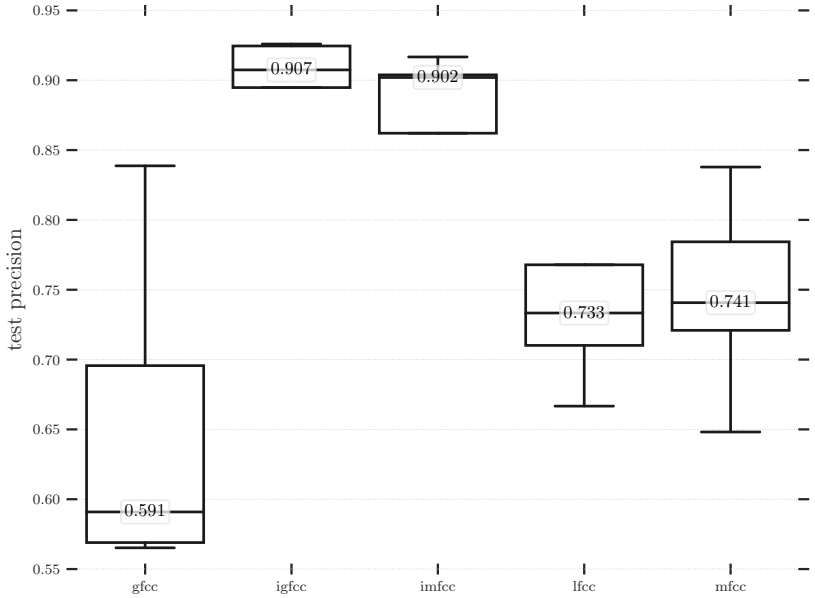
C.16 ssn morten

C.16.1 test accuracy



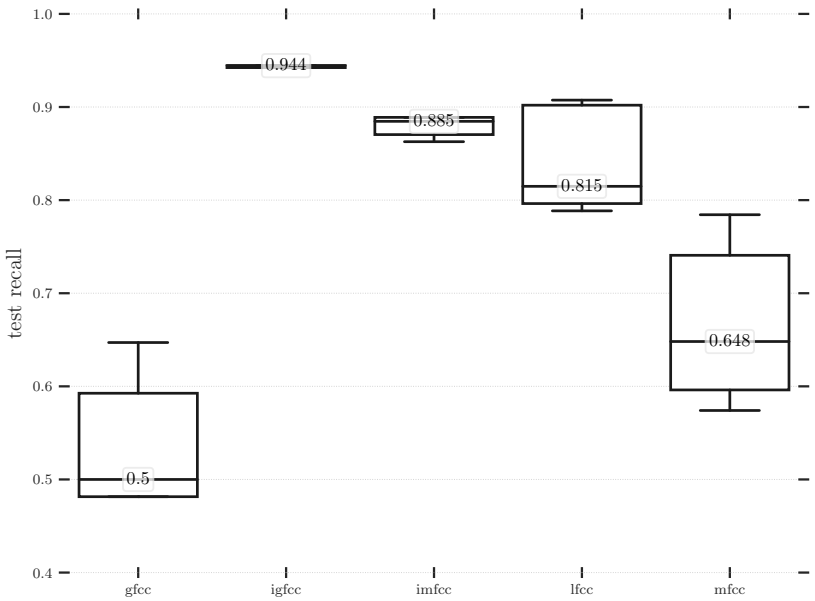
Figur 176: *test accuracy* for ssn morten snr 20db

C.16.2 test precision



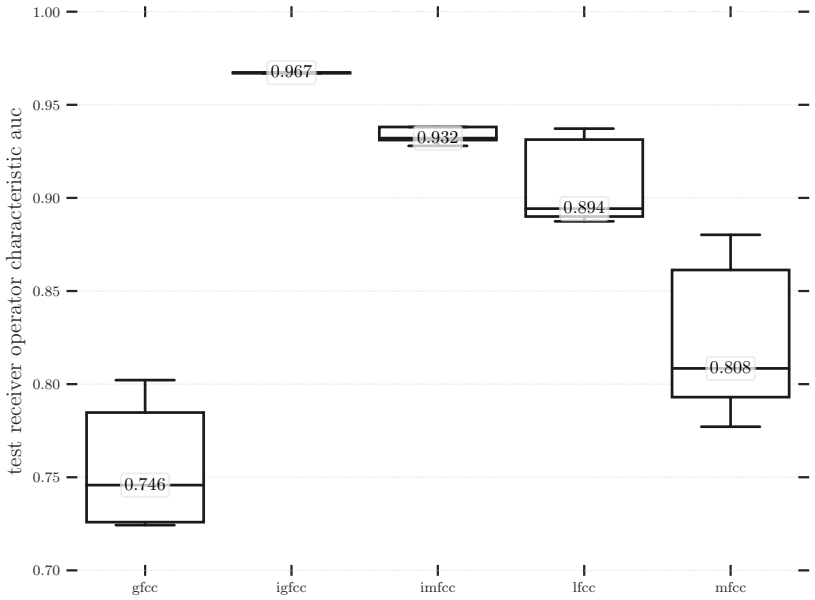
Figur 177: *test precision* for ssn morten snr 20db

C.16.3 test recall



Figur 178: *test recall* for ssn morten snr 20db

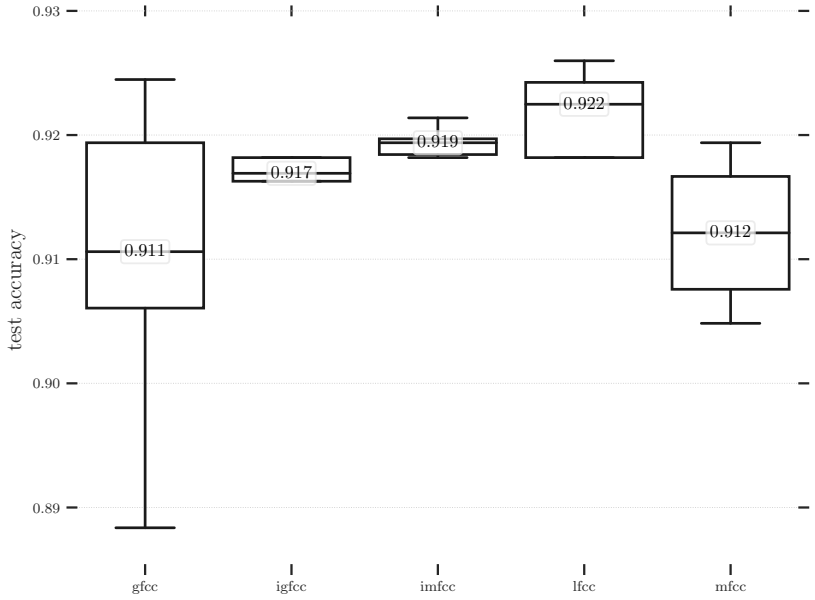
C.16.4 test receiver operator characteristic auc



Figur 179: *test receiver operator characteristic auc* for ssn morten snr 20db

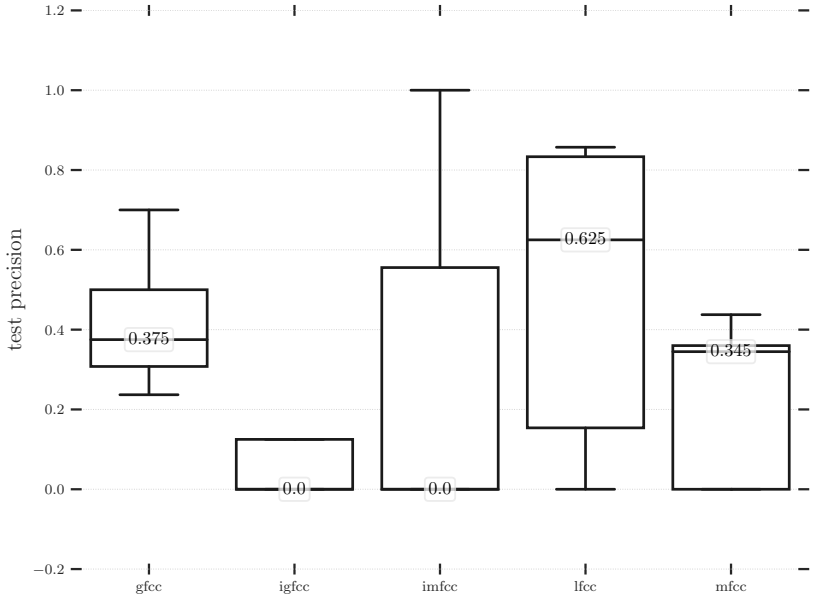
C.17 dkitchen

C.17.1 test accuracy



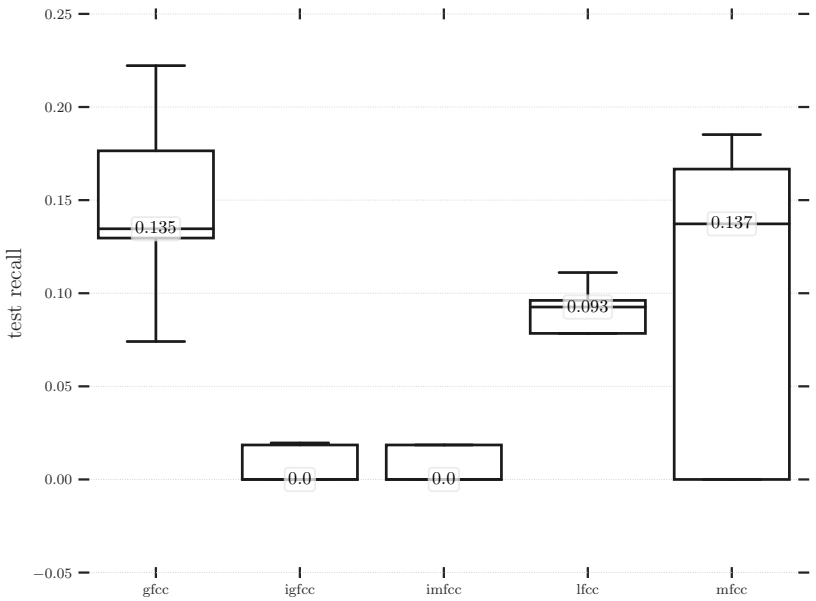
Figur 180: *test accuracy* for dkitchen snr 0db

C.17.2 test precision



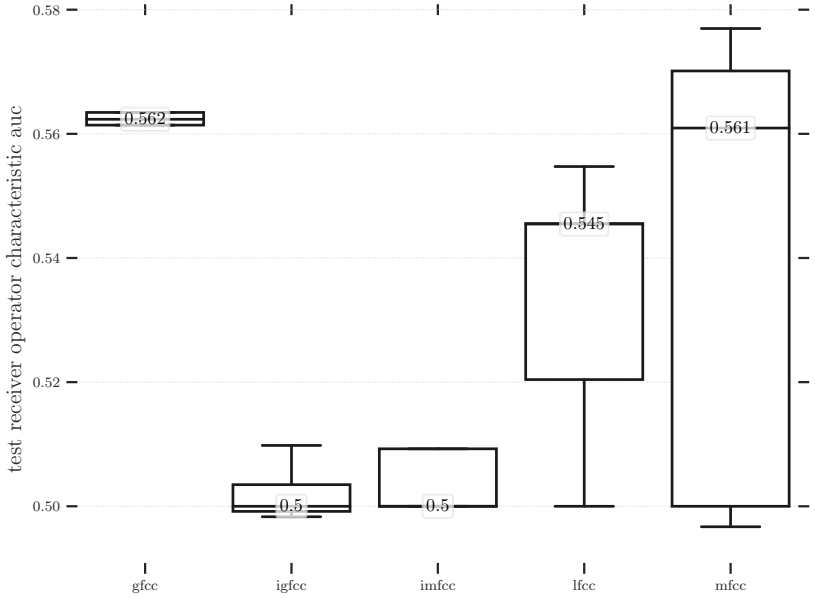
Figur 181: *test precision* for dkitchen snr 0db

C.17.3 test recall



Figur 182: *test recall* for dkitchen snr 0db

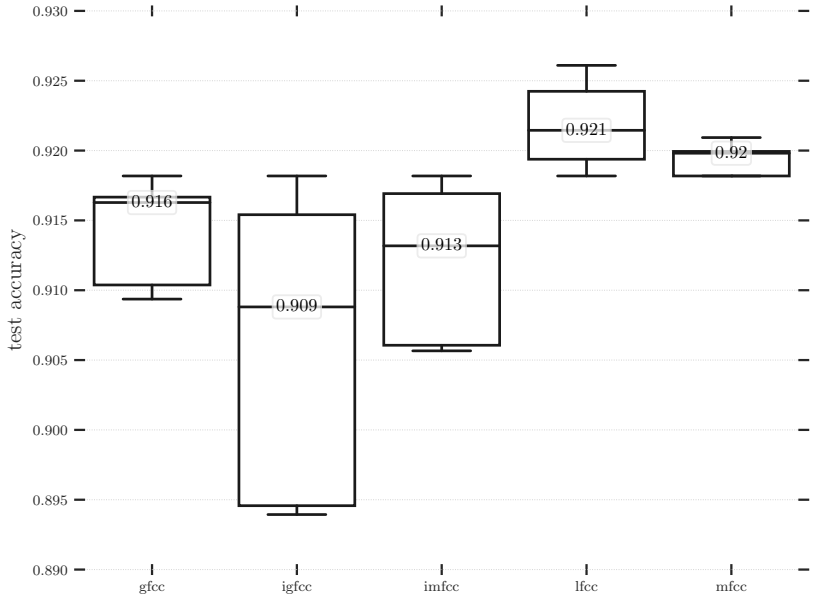
C.17.4 test receiver operator characteristic auc



Figur 183: *test receiver operator characteristic auc* for dkitchen snr 0db

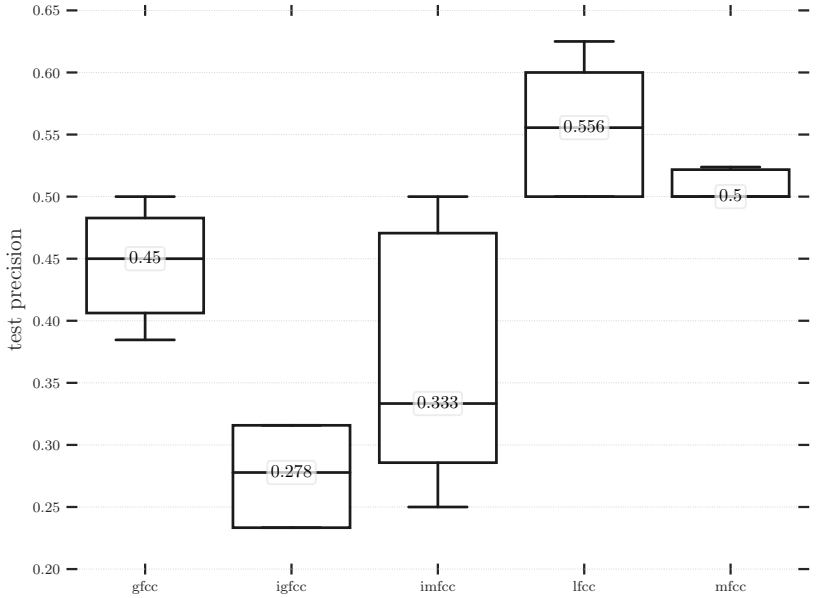
C.18 dkitchen

C.18.1 test accuracy



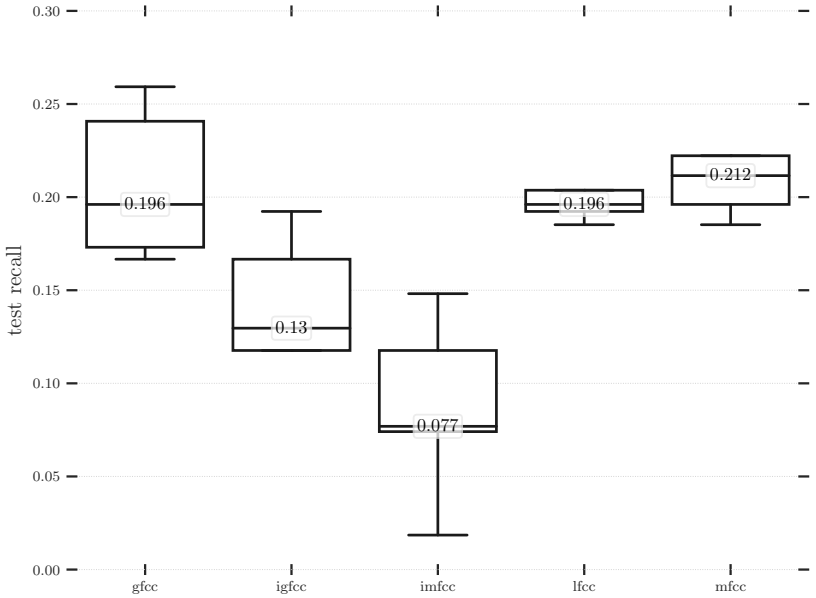
Figur 184: *test accuracy* for dkitchen snr 5db

C.18.2 test precision



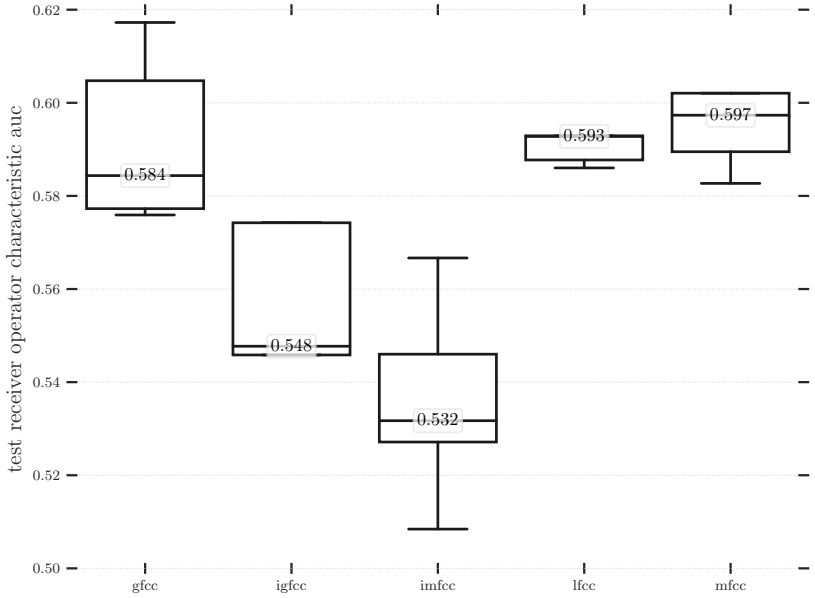
Figur 185: *test precision* for dkitchen snr 5db

C.18.3 test recall



Figur 186: *test recall* for dkitchen snr 5db

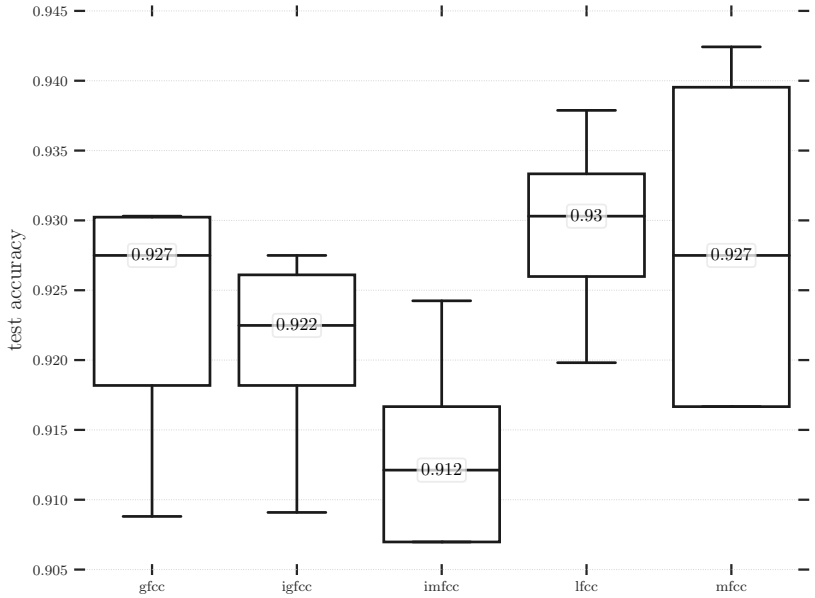
C.18.4 test receiver operator characteristic auc



Figur 187: *test receiver operator characteristic auc* for dkitchen snr 5db

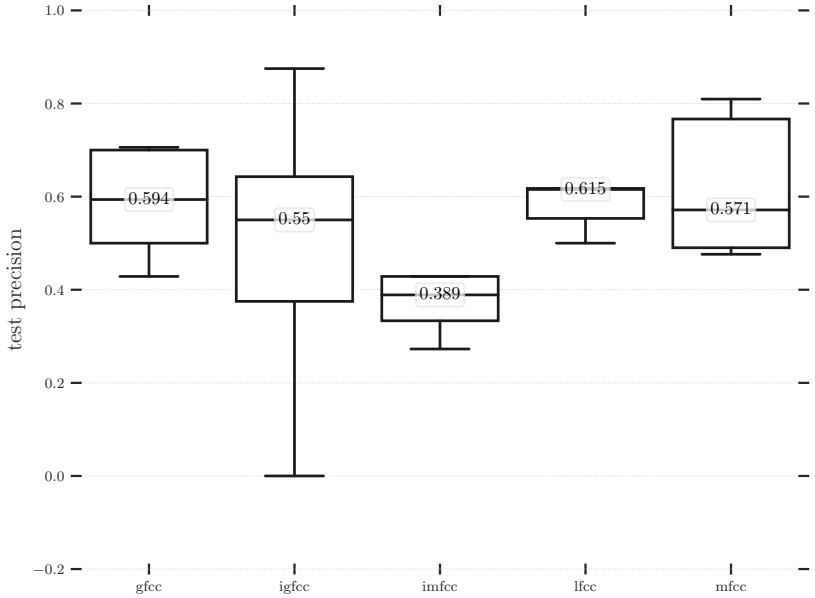
C.19 dkitchen

C.19.1 test accuracy



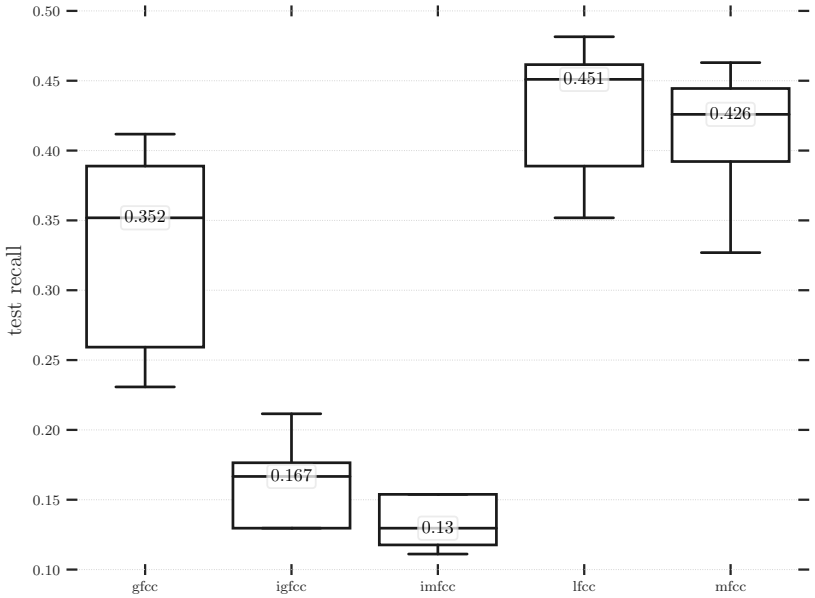
Figur 188: *test accuracy* for dkitchen snr 10db

C.19.2 test precision



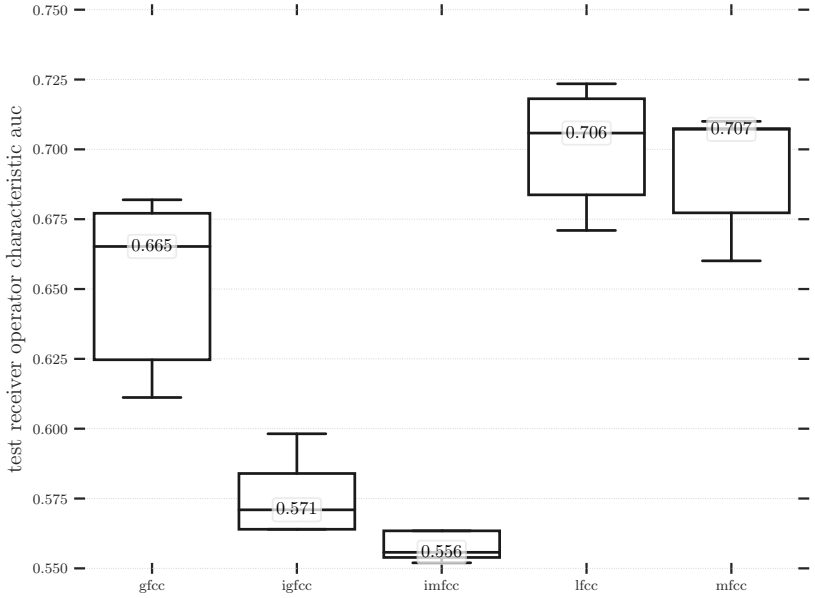
Figur 189: *test precision* for dkitchen snr 10db

C.19.3 test recall



Figur 190: *test recall* for dkitchen snr 10db

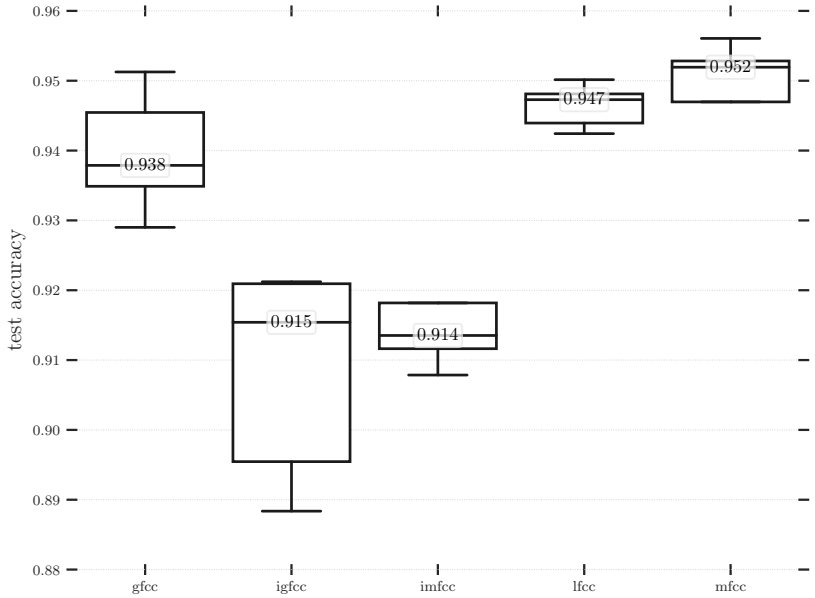
C.19.4 test receiver operator characteristic auc



Figur 191: *test receiver operator characteristic auc* for dkitchen snr 10db

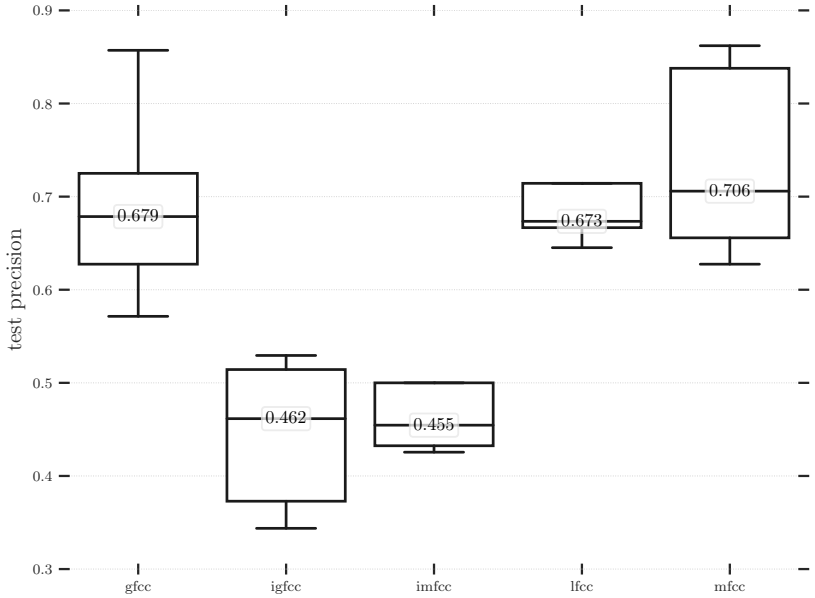
C.20 dkitchen

C.20.1 test accuracy



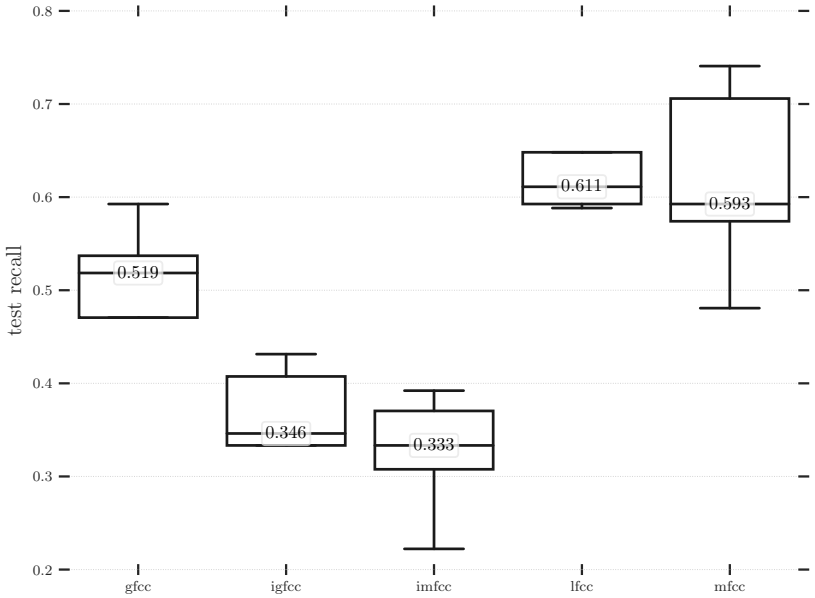
Figur 192: *test accuracy* for dkitchen snr 15db

C.20.2 test precision



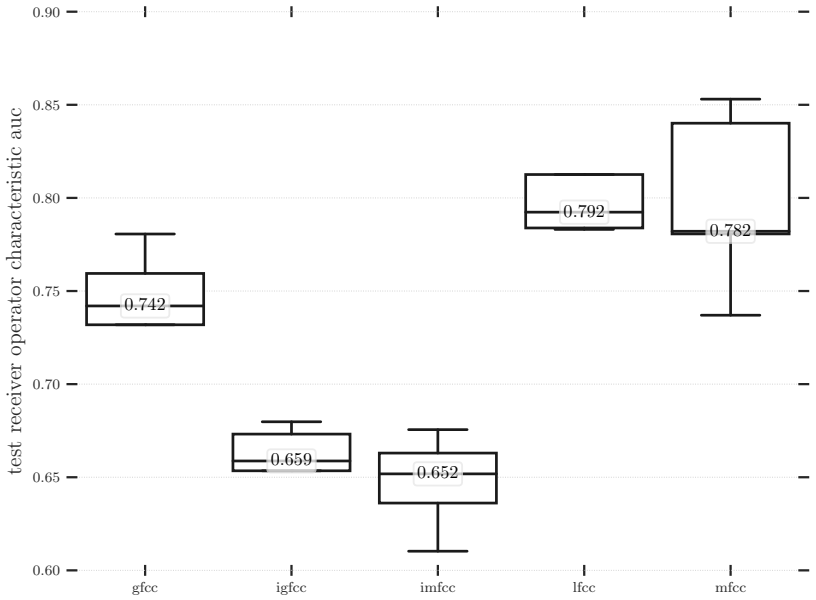
Figur 193: *test precision* for dkitchen snr 15db

C.20.3 test recall



Figur 194: *test recall* for dkitchen snr 15db

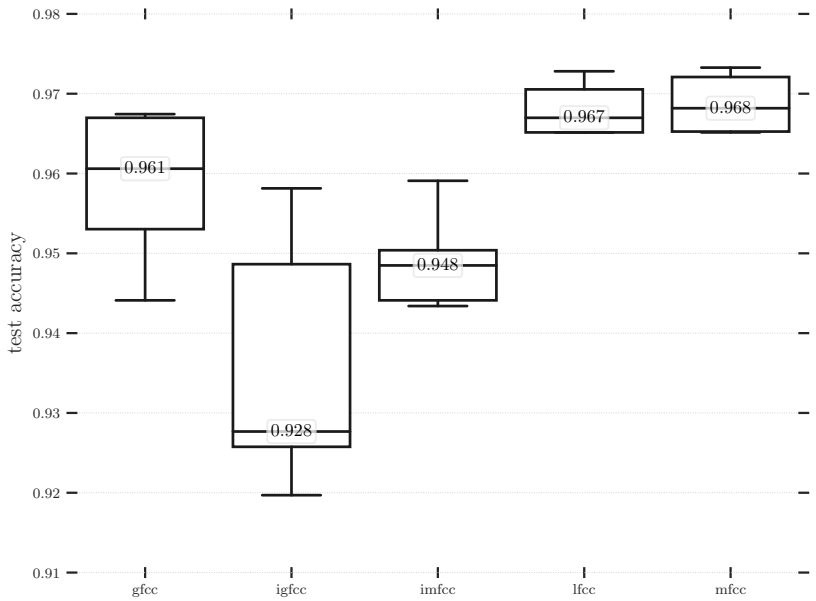
C.20.4 test receiver operator characteristic auc



Figur 195: *test receiver operator characteristic auc* for dkitchen snr 15db

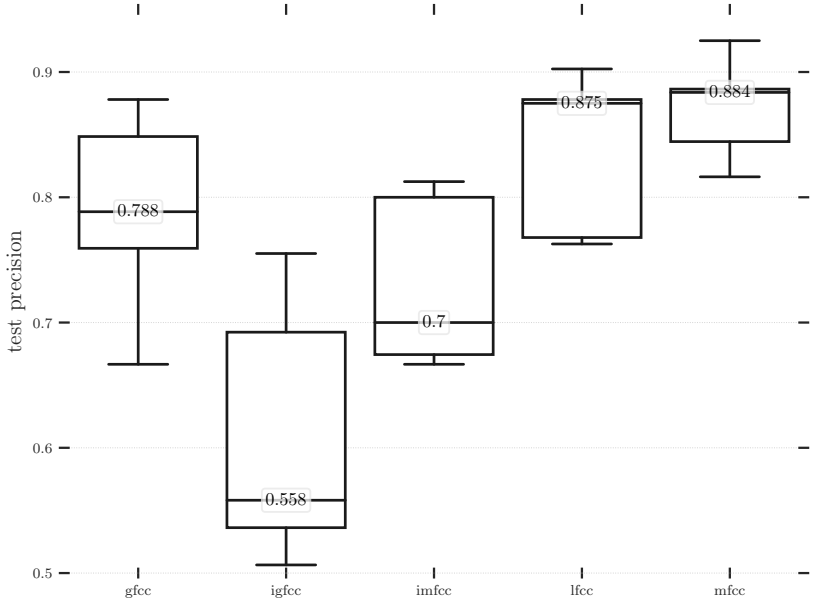
C.21 dkitchen

C.21.1 test accuracy



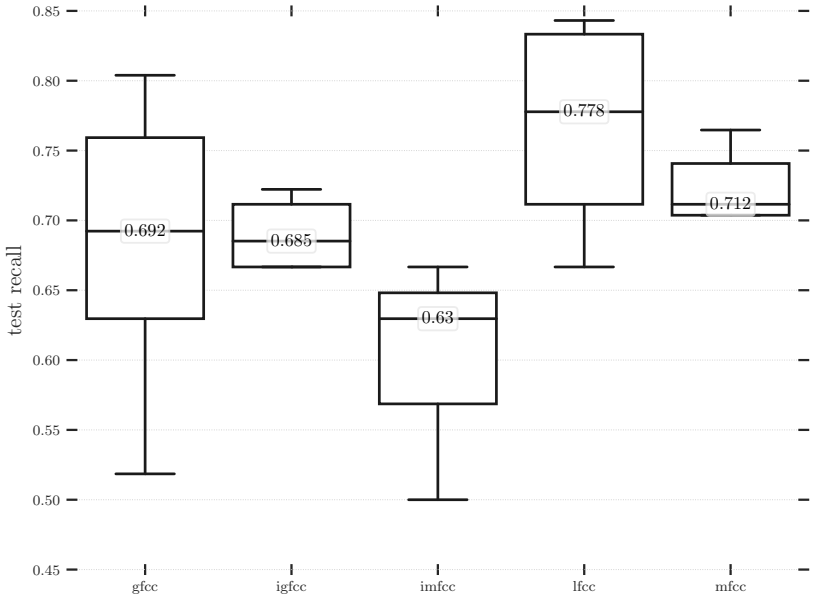
Figur 196: *test accuracy* for dkitchen snr 20db

C.21.2 test precision



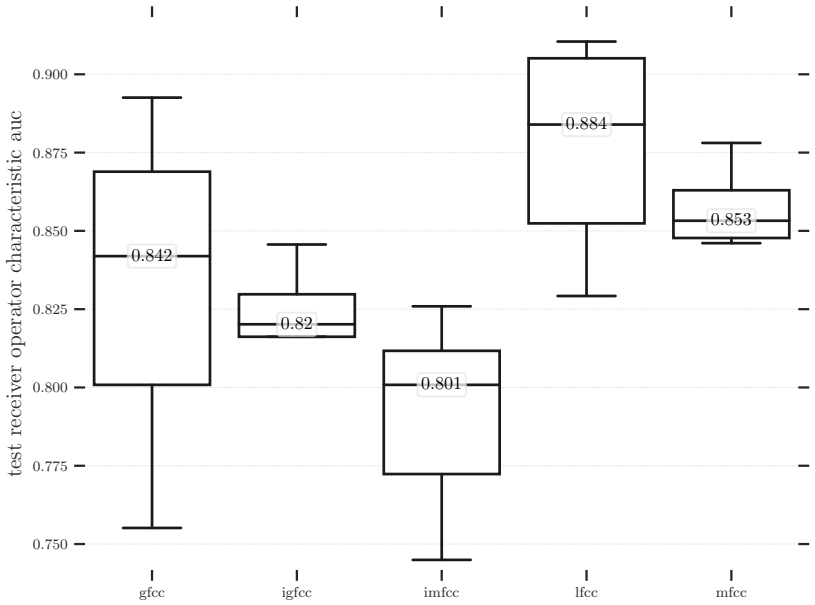
Figur 197: *test precision* for dkitchen snr 20db

C.21.3 test recall



Figur 198: *test recall* for dkitchen snr 20db

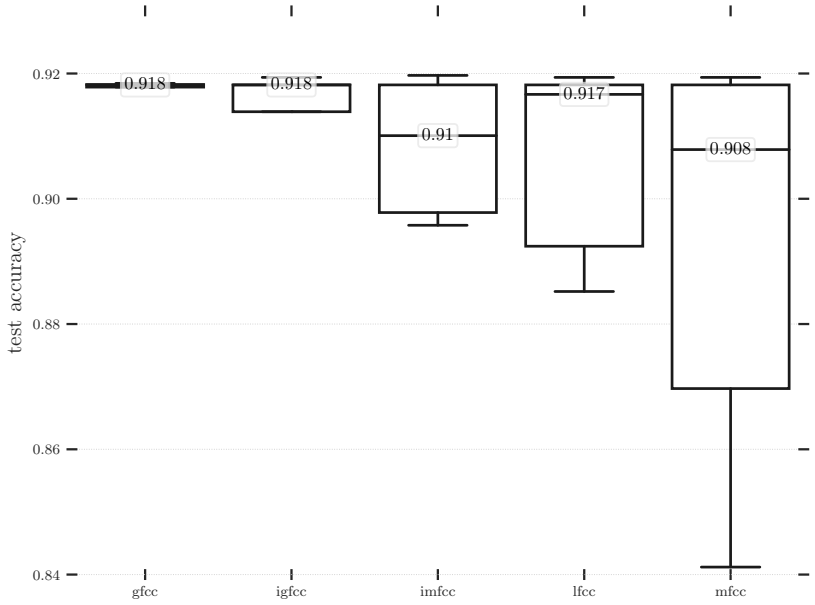
C.21.4 test receiver operator characteristic auc



Figur 199: *test receiver operator characteristic auc* for dkitchen snr 20db

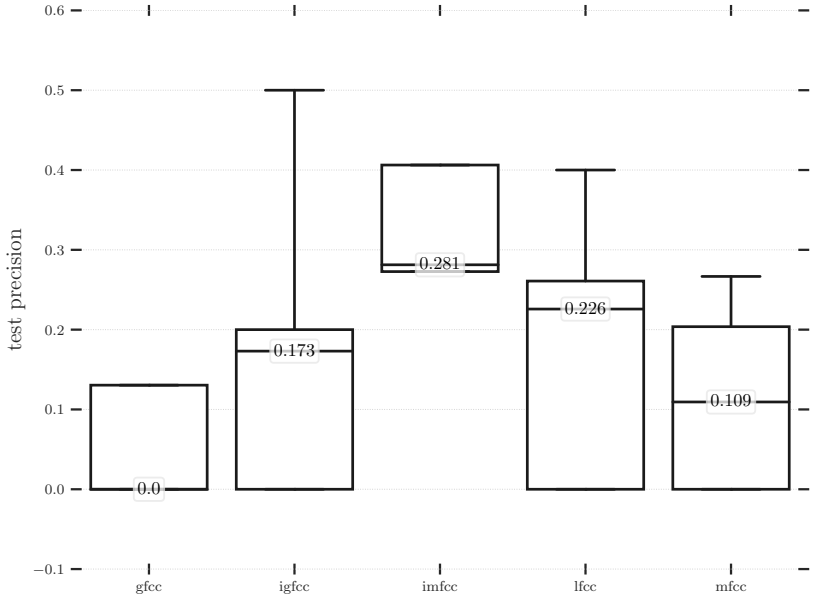
C.22 pcafeter

C.22.1 test accuracy



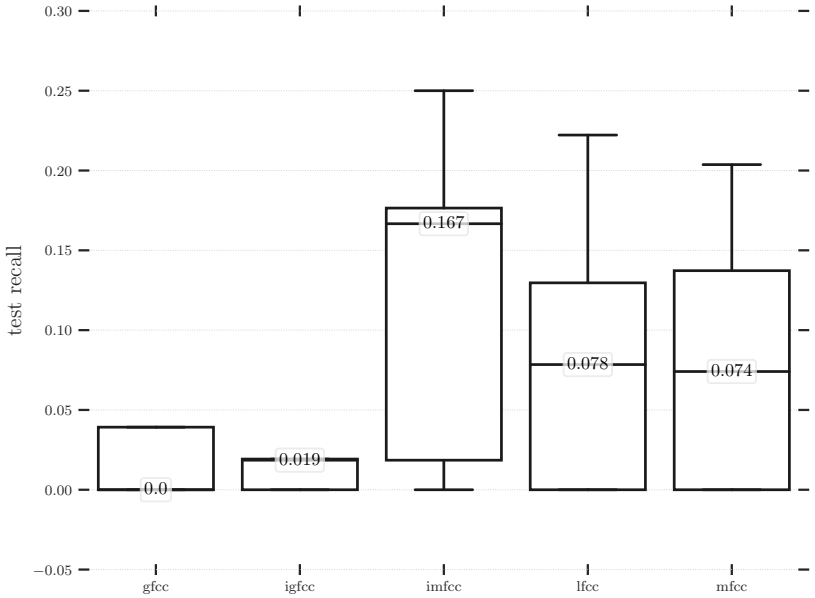
Figur 200: *test accuracy* for pcafeter snr 0db

C.22.2 test precision



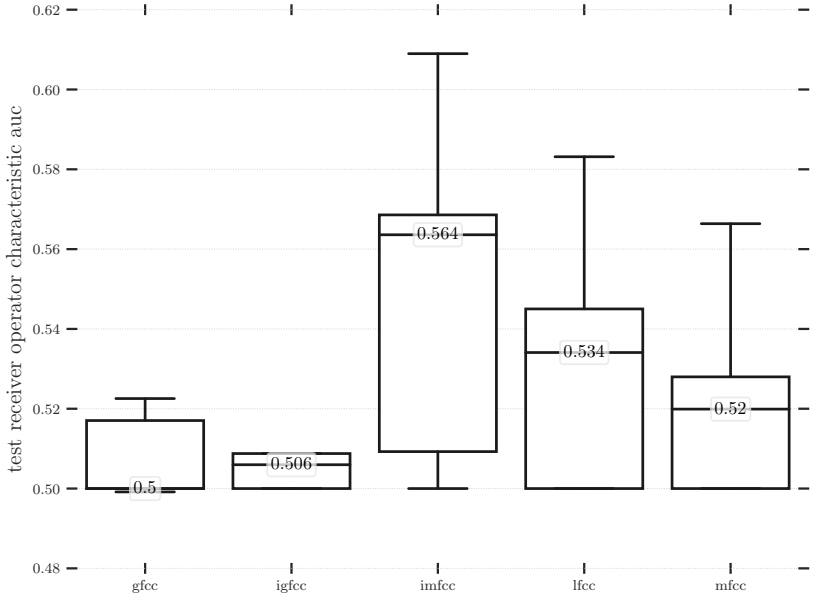
Figur 201: *test precision* for pcafeter snr 0db

C.22.3 test recall



Figur 202: *test recall* for pcafeter snr 0db

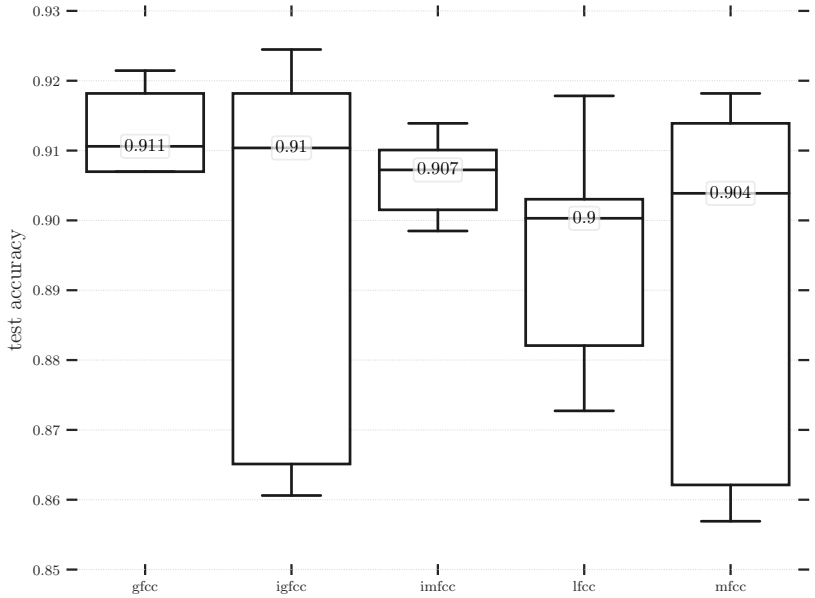
C.22.4 test receiver operator characteristic auc



Figur 203: *test receiver operator characteristic auc* for pcafeter snr 0db

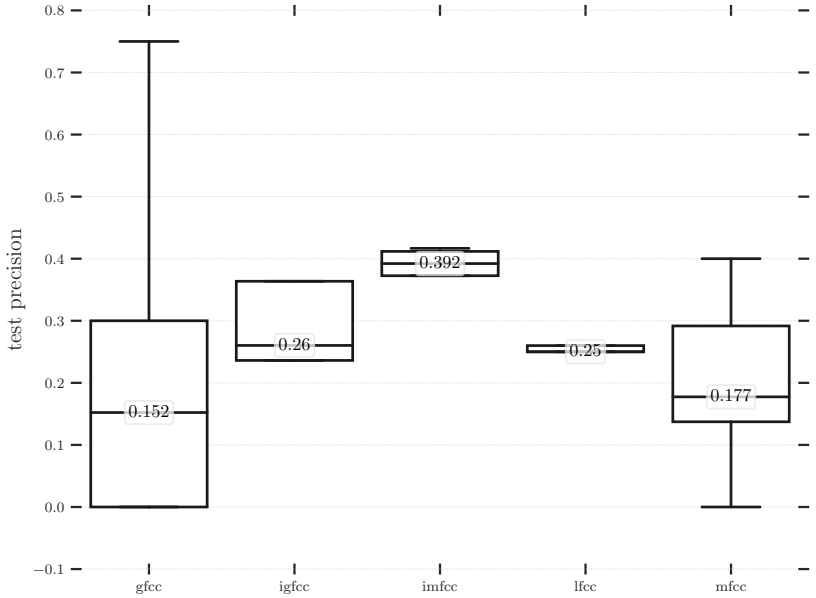
C.23 pcafeter

C.23.1 test accuracy



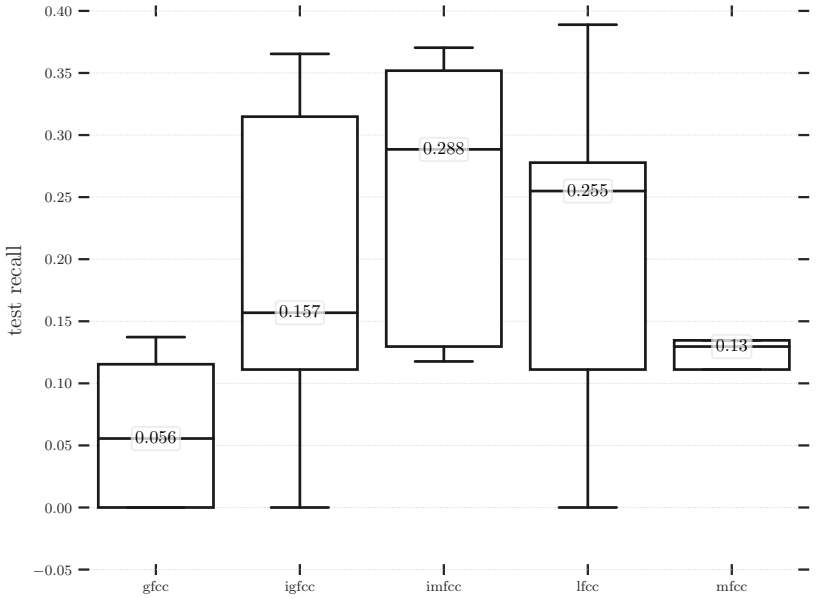
Figur 204: *test accuracy* for pcafeter snr 5db

C.23.2 test precision



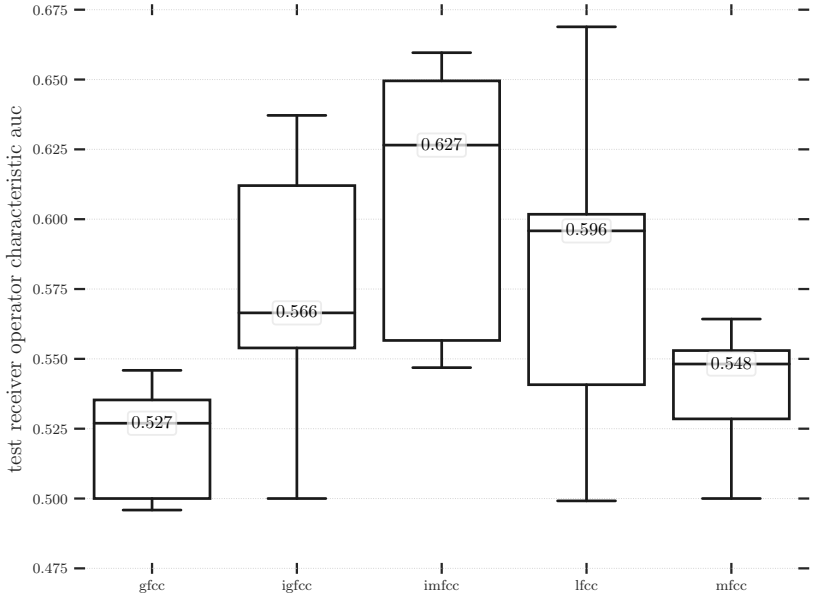
Figur 205: *test precision* for pcafeter snr 5db

C.23.3 test recall



Figur 206: *test recall* for pcafeter snr 5db

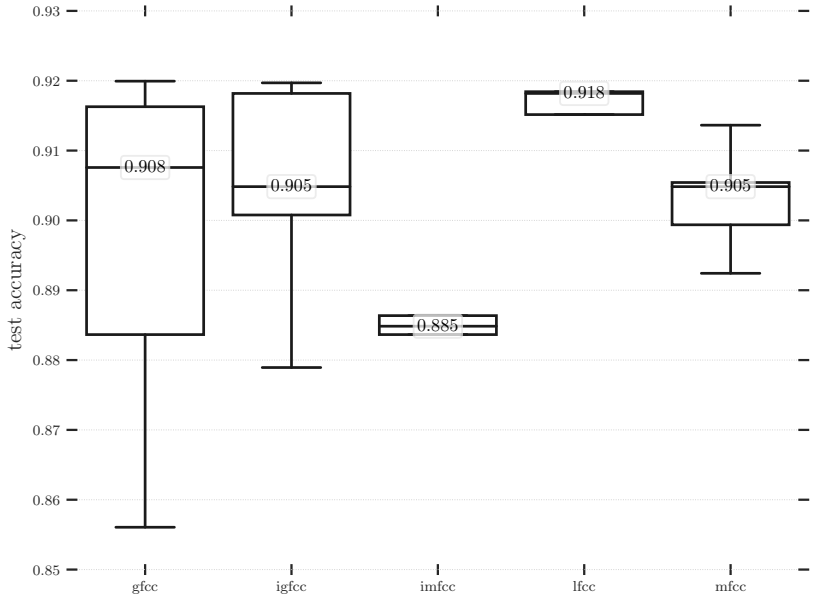
C.23.4 test receiver operator characteristic auc



Figur 207: *test receiver operator characteristic auc* for pcafeter snr 5db

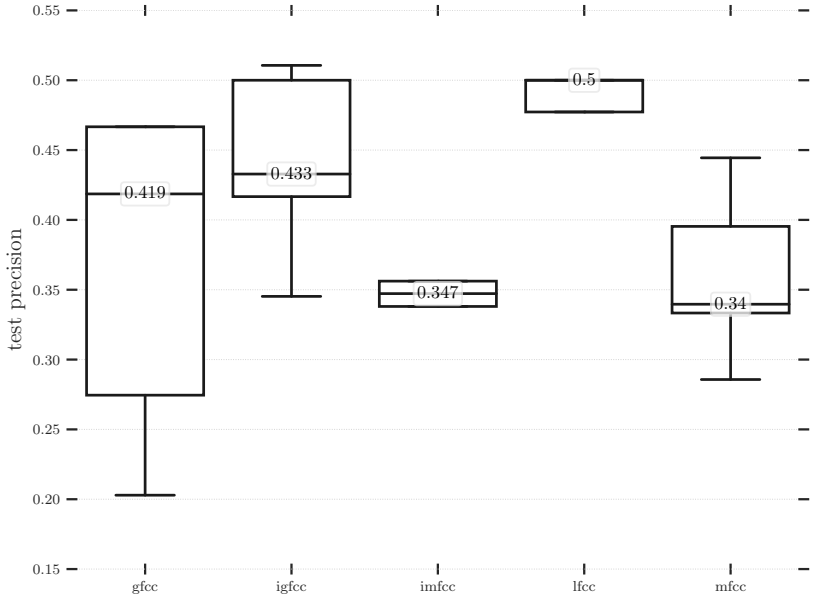
C.24 pcafeter

C.24.1 test accuracy



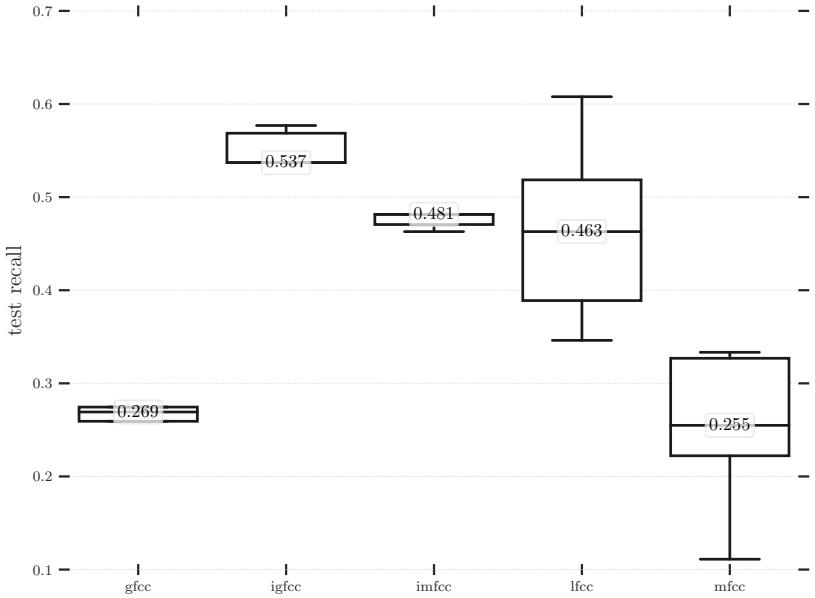
Figur 208: *test accuracy* for pcafeter snr 10db

C.24.2 test precision



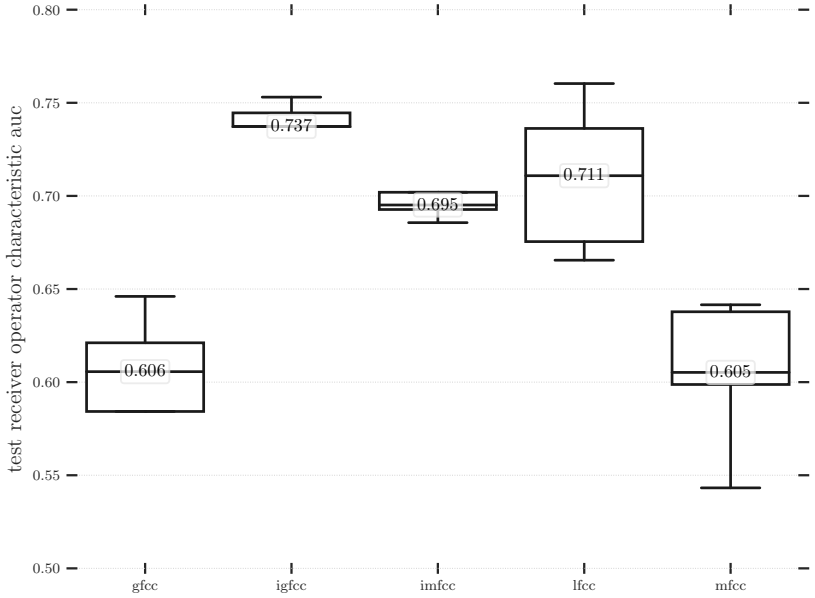
Figur 209: *test precision* for pcafeter snr 10db

C.24.3 test recall



Figur 210: *test recall* for pcafeter snr 10db

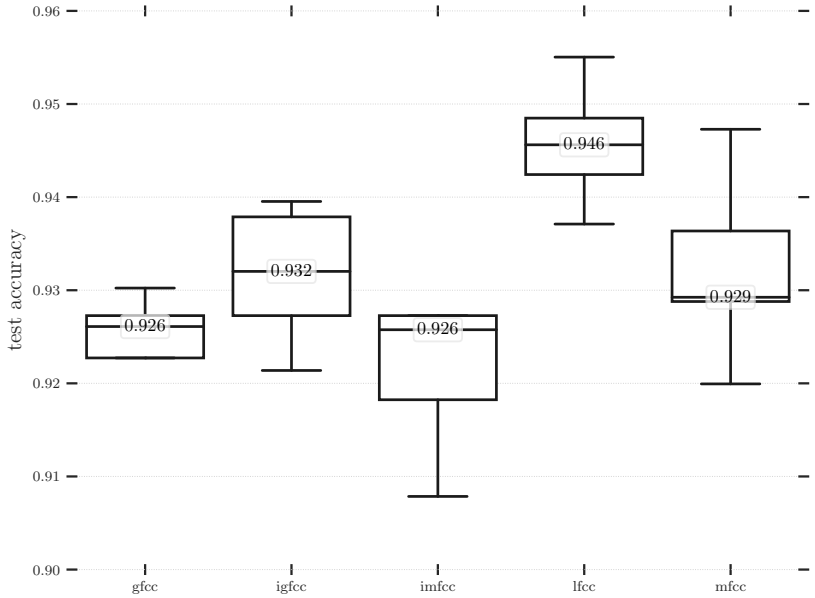
C.24.4 test receiver operator characteristic auc



Figur 211: *test receiver operator characteristic auc* for pcafeter snr 10db

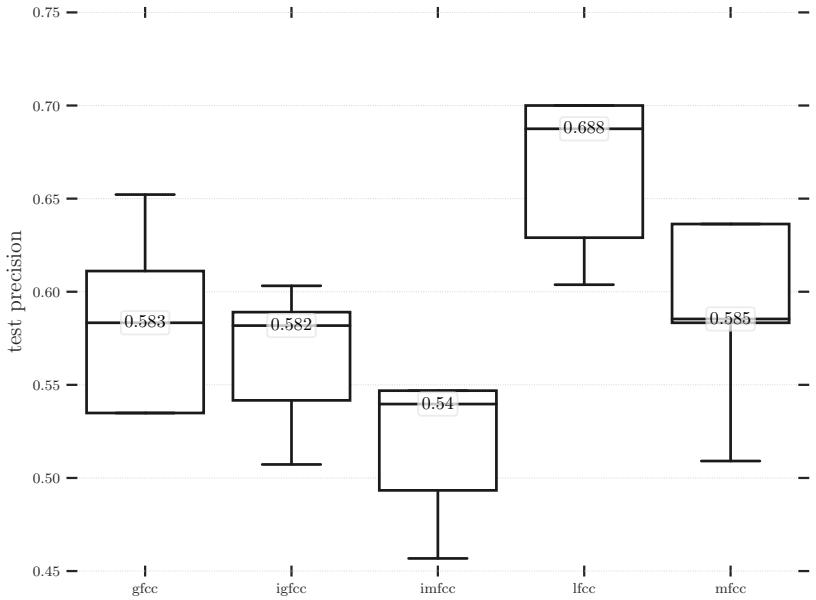
C.25 pcafeter

C.25.1 test accuracy



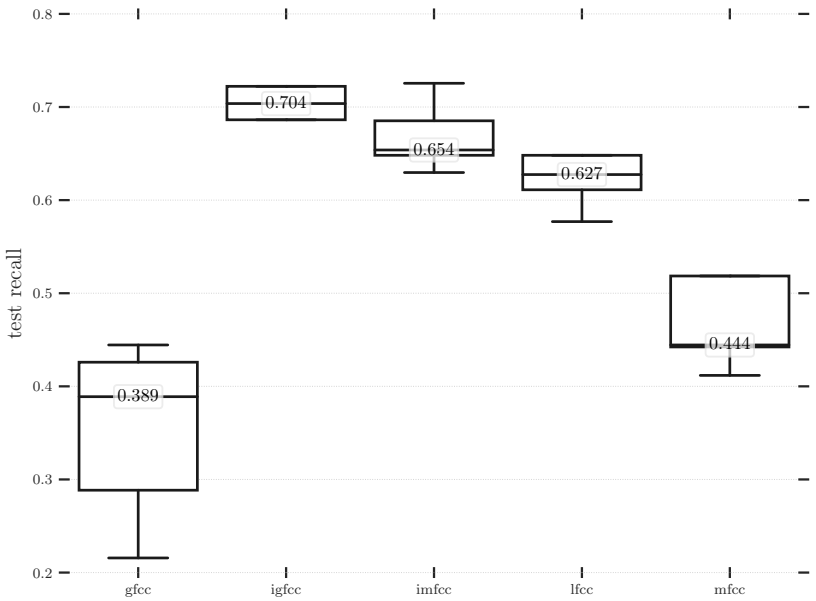
Figur 212: *test accuracy* for pcafeter snr 15db

C.25.2 test precision



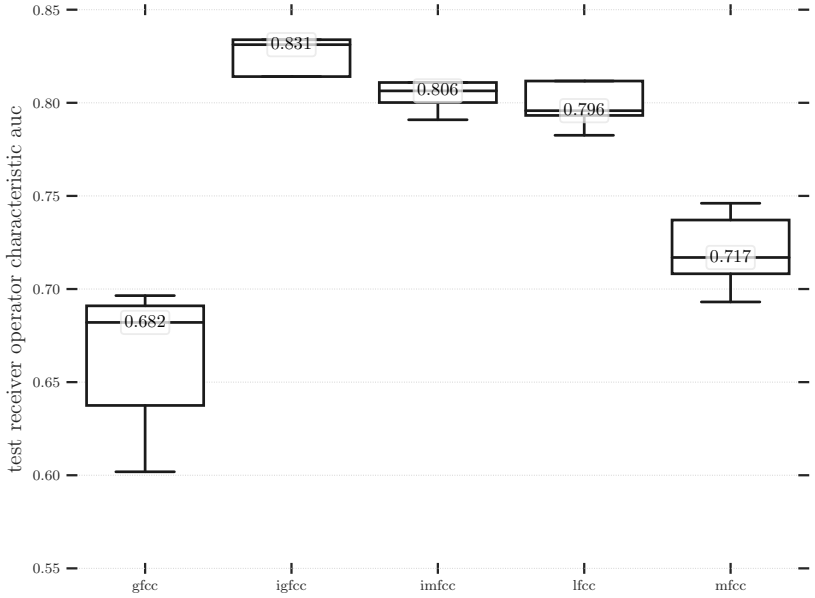
Figur 213: *test precision* for pcafeter snr 15db

C.25.3 test recall



Figur 214: *test recall* for pcafeter snr 15db

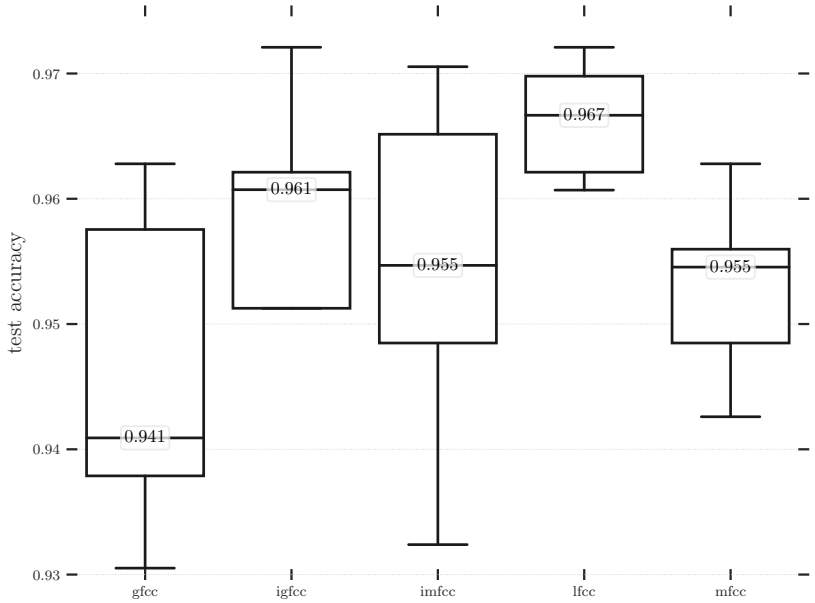
C.25.4 test receiver operator characteristic auc



Figur 215: *test receiver operator characteristic auc* for pcafeter snr 15db

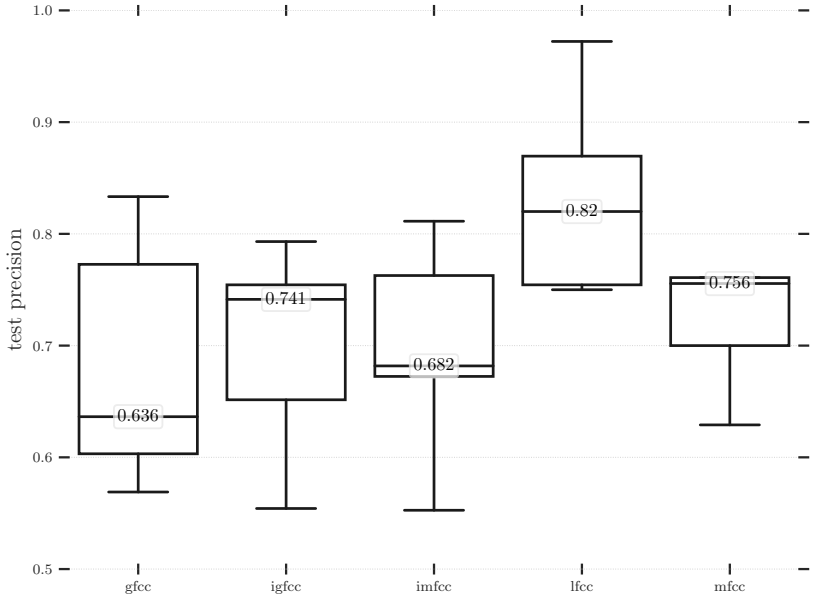
C.26 pcafeter

C.26.1 test accuracy



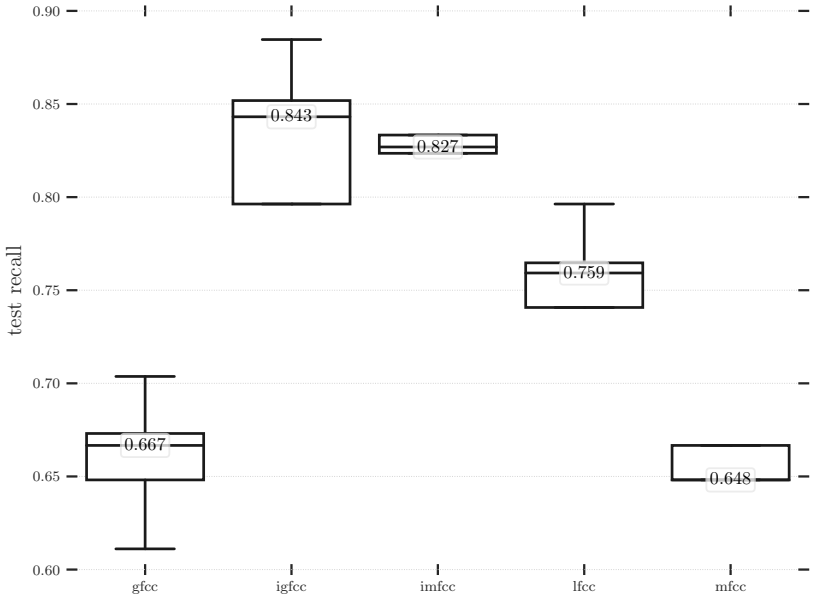
Figur 216: *test accuracy* for pcafeter snr 20db

C.26.2 test precision



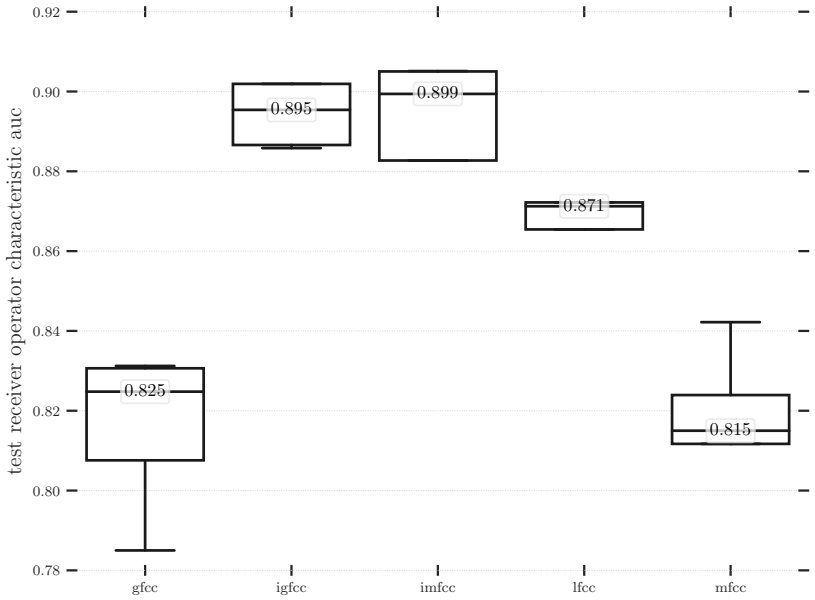
Figur 217: *test precision* for pcafeter snr 20db

C.26.3 test recall



Figur 218: *test recall* for pcafeter snr 20db

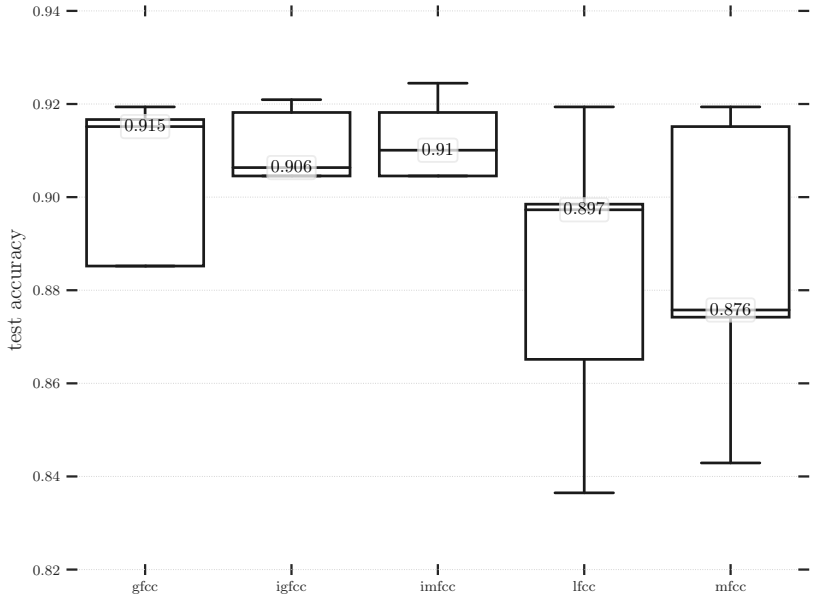
C.26.4 test receiver operator characteristic auc



Figur 219: *test receiver operator characteristic auc* for pcafeter snr 20db

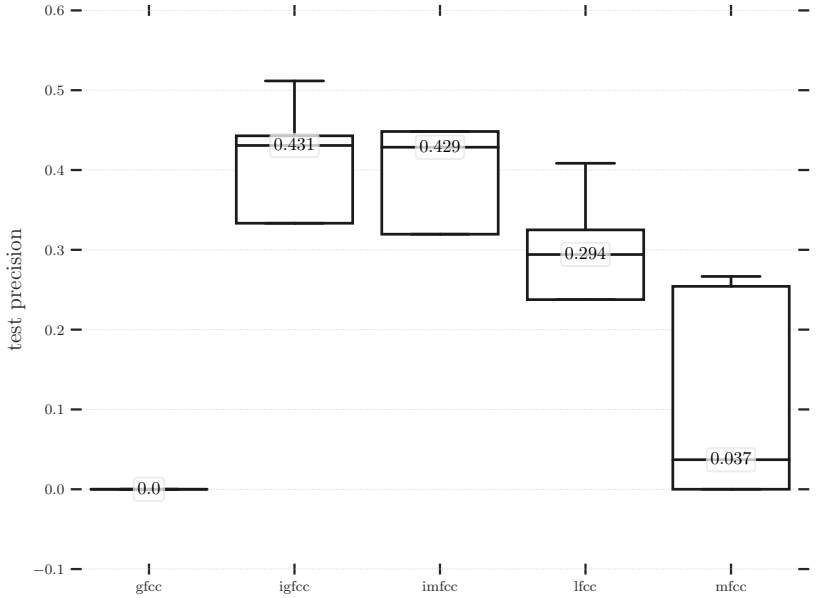
C.27 spsquare

C.27.1 test accuracy



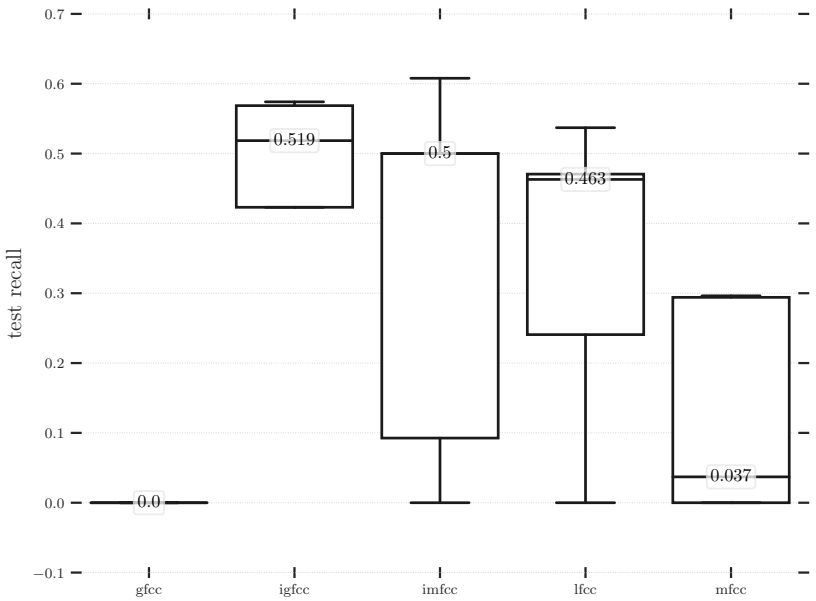
Figur 220: *test accuracy* for spsquare snr 0db

C.27.2 test precision



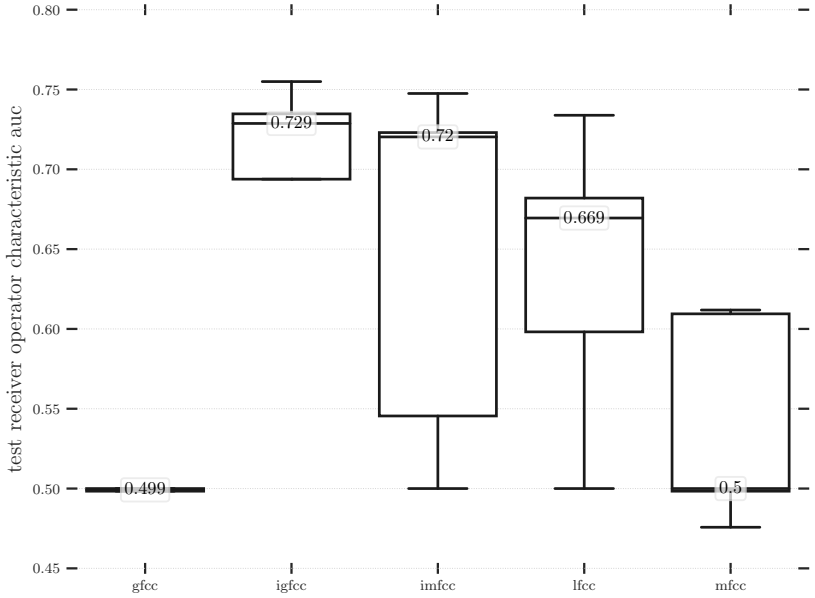
Figur 221: *test precision* for spsquare snr 0db

C.27.3 test recall



Figur 222: *test recall* for spsquare snr 0db

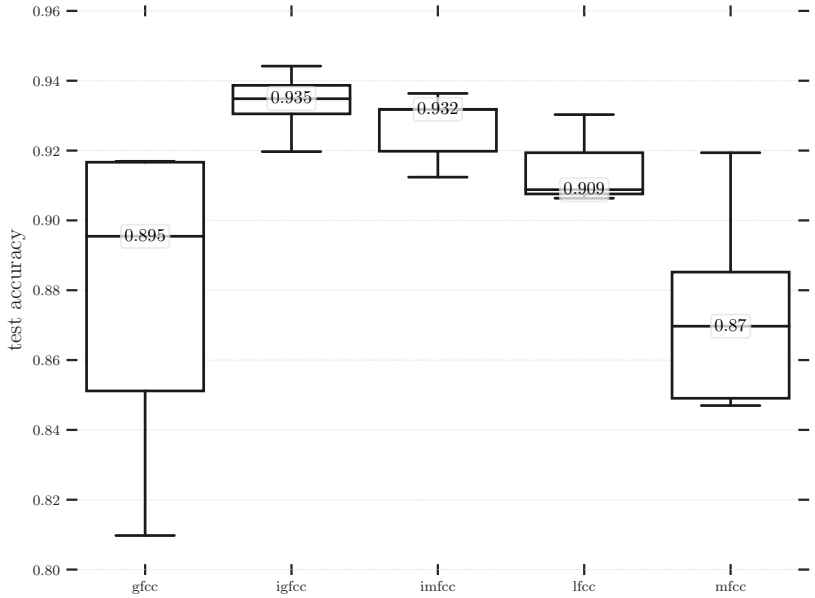
C.27.4 test receiver operator characteristic auc



Figur 223: *test receiver operator characteristic auc* for spsquare snr 0db

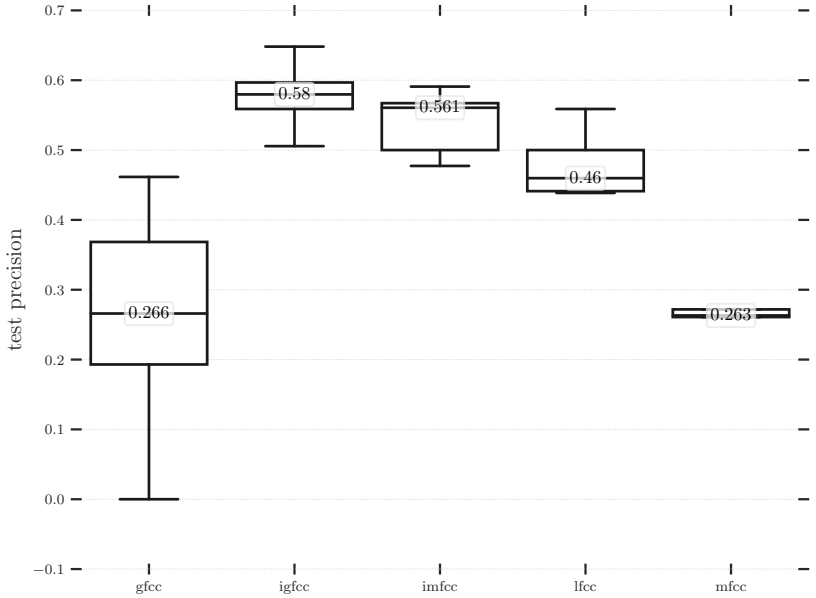
C.28 spsquare

C.28.1 test accuracy



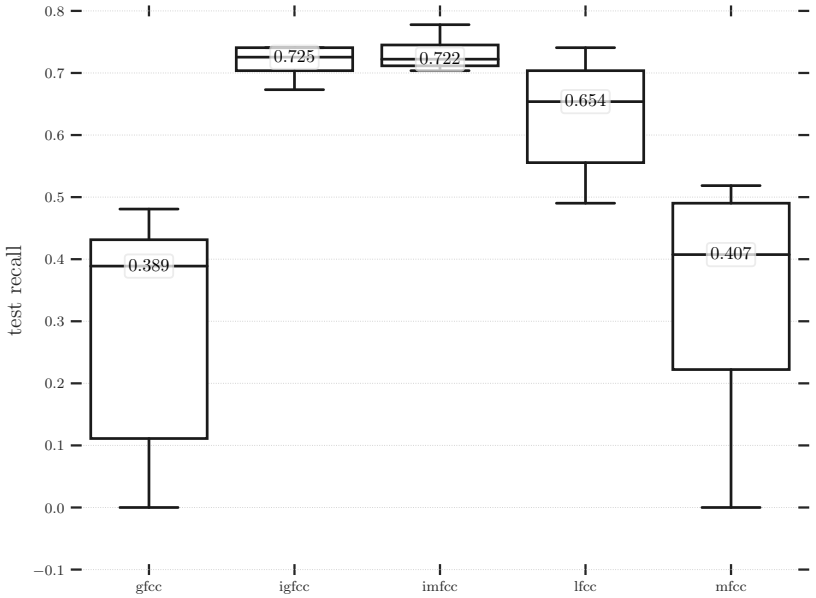
Figur 224: *test accuracy* for spsquare snr 5db

C.28.2 test precision



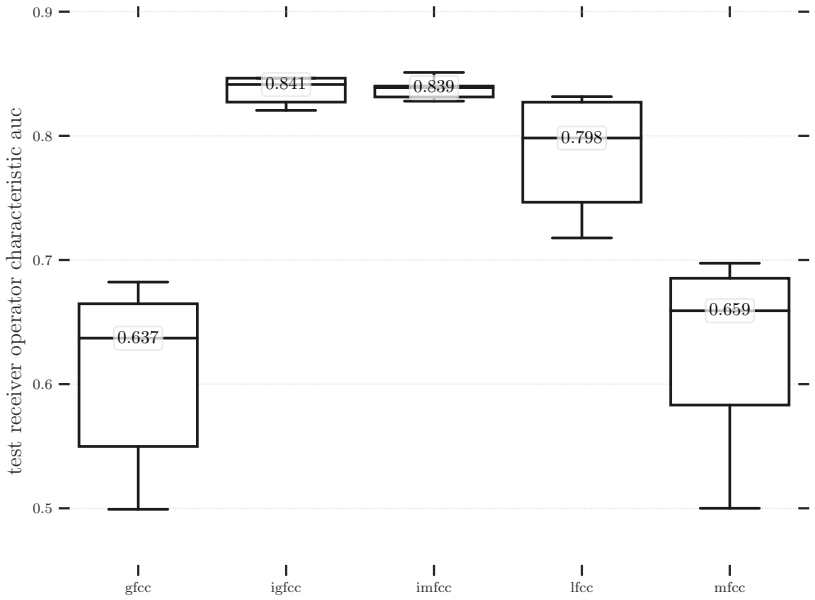
Figur 225: *test precision* for spsquare snr 5db

C.28.3 test recall



Figur 226: *test recall* for spsquare snr 5db

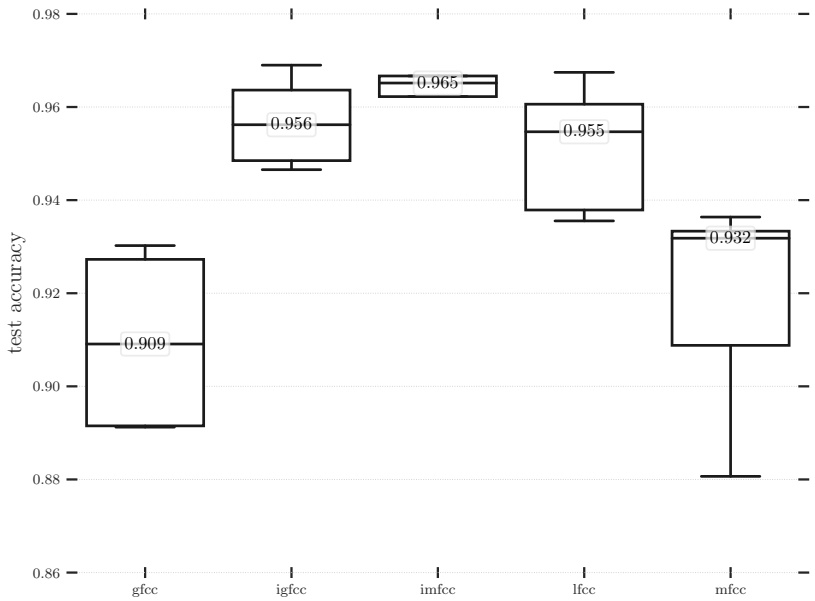
C.28.4 test receiver operator characteristic auc



Figur 227: *test receiver operator characteristic auc* for spsquare snr 5db

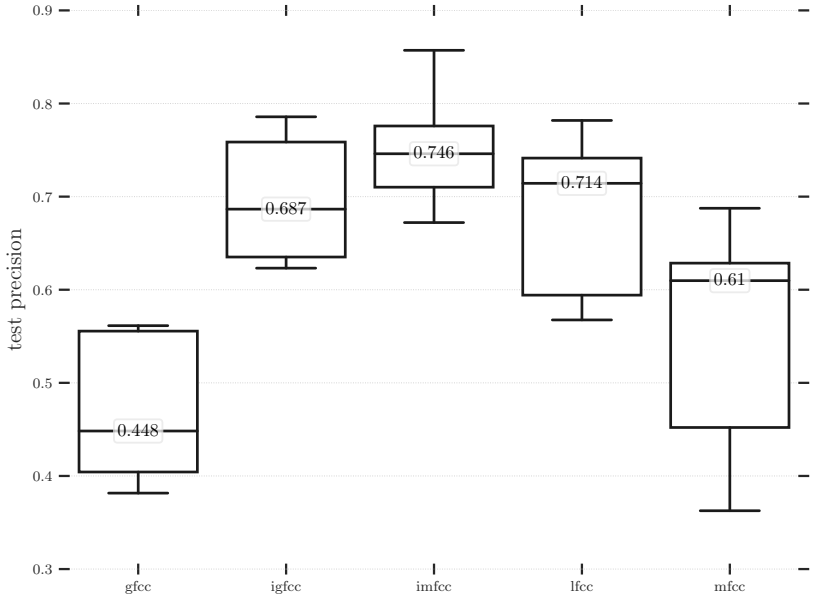
C.29 spsquare

C.29.1 test accuracy



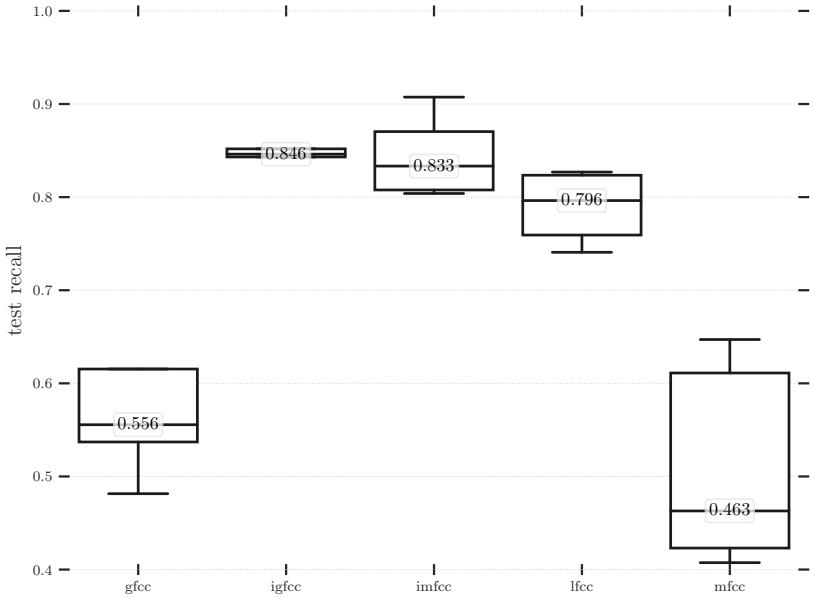
Figur 228: *test accuracy* for spsquare snr 10db

C.29.2 test precision



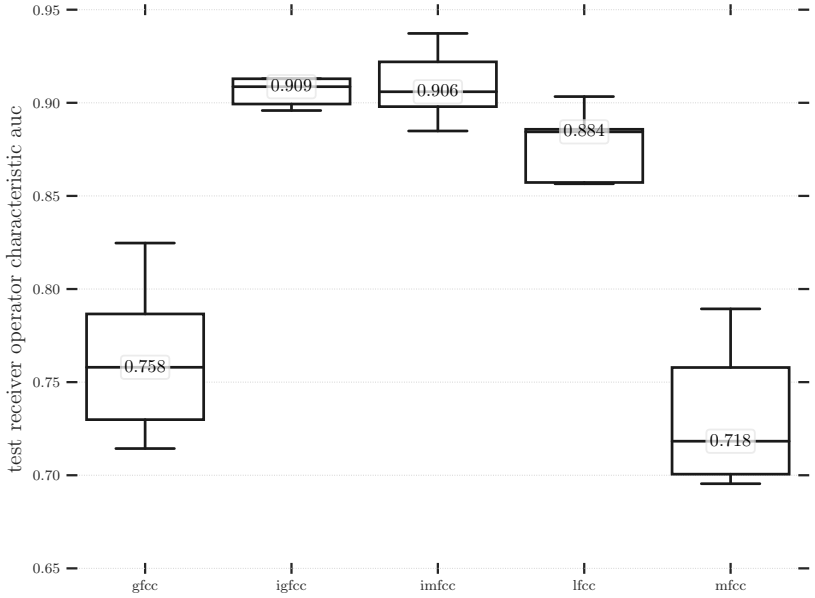
Figur 229: *test precision* for spsquare snr 10db

C.29.3 test recall



Figur 230: *test recall* for spsquare snr 10db

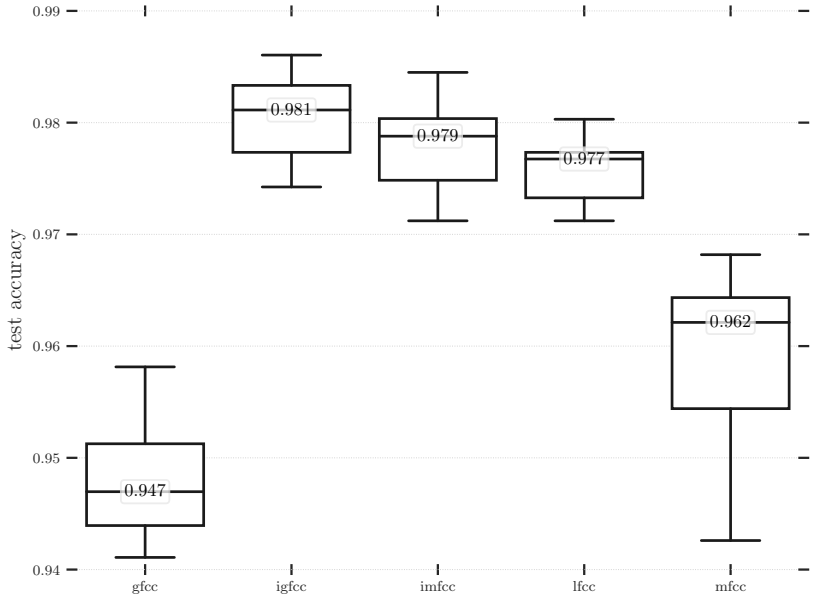
C.29.4 test receiver operator characteristic auc



Figur 231: *test receiver operator characteristic auc* for spsquare snr 10db

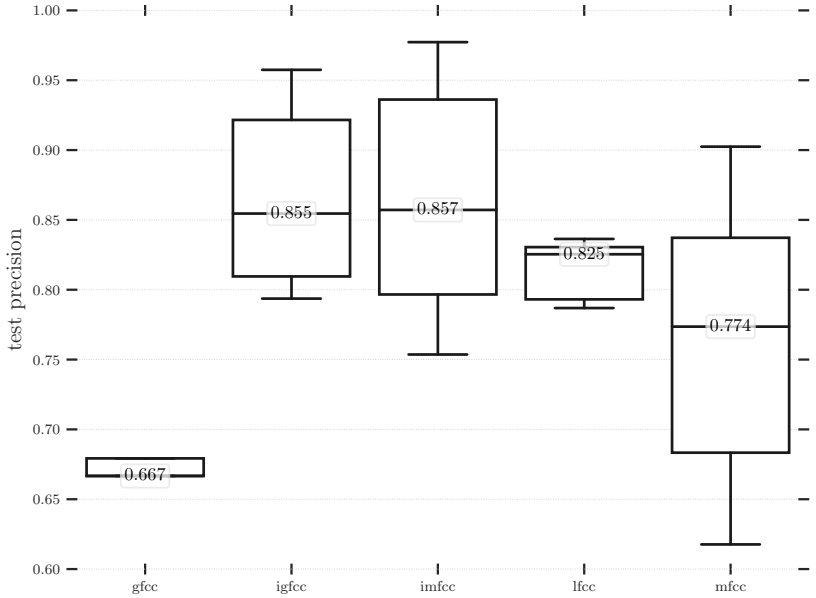
C.30 spsquare

C.30.1 test accuracy



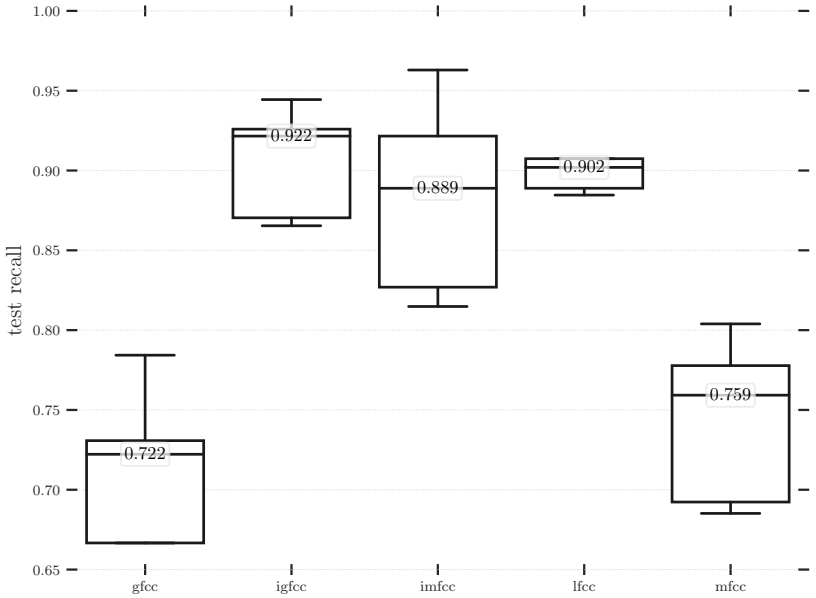
Figur 232: *test accuracy* for spsquare snr 15db

C.30.2 test precision



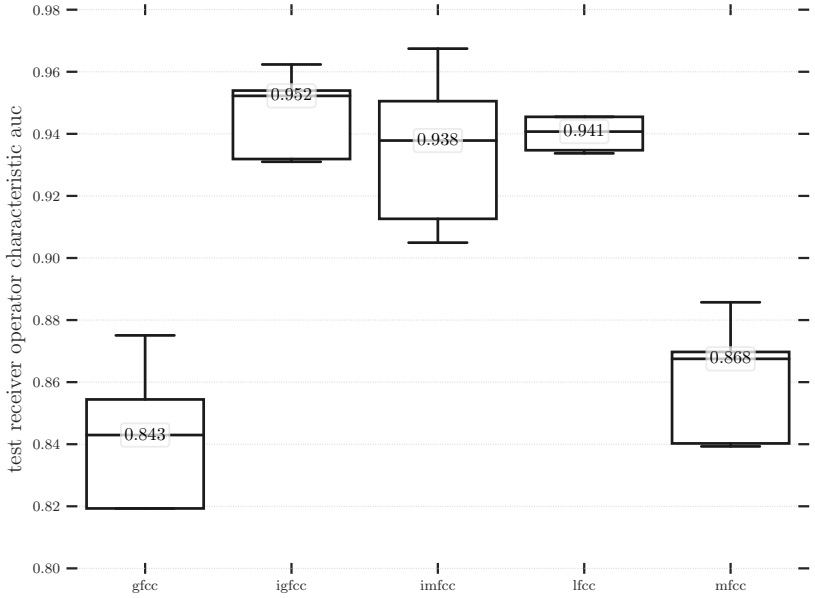
Figur 233: *test precision* for spsquare snr 15db

C.30.3 test recall



Figur 234: *test recall* for spsquare snr 15db

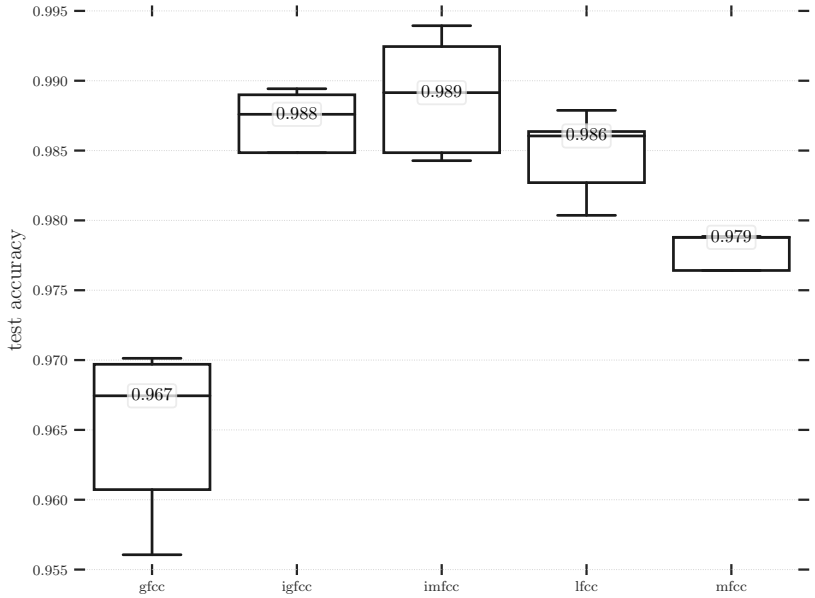
C.30.4 test receiver operator characteristic auc



Figur 235: *test receiver operator characteristic auc* for spsquare snr 15db

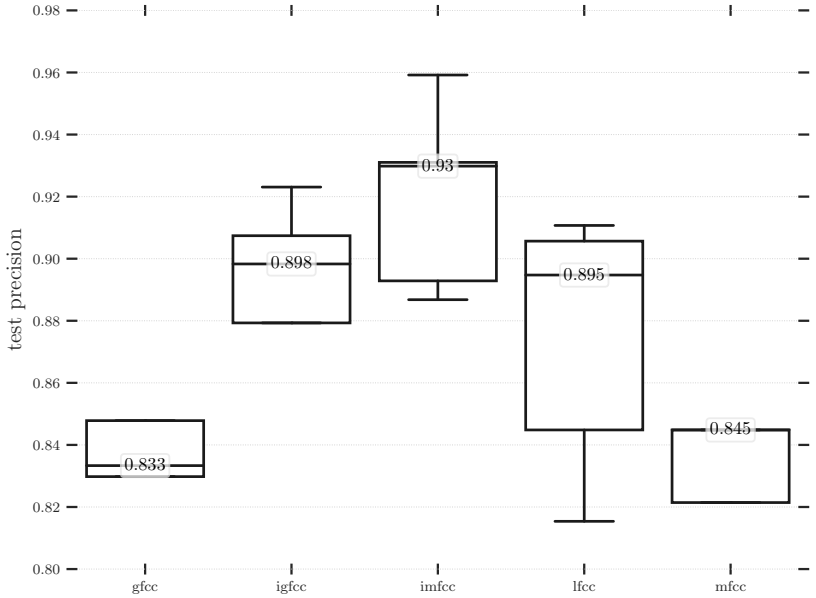
C.31 spsquare

C.31.1 test accuracy



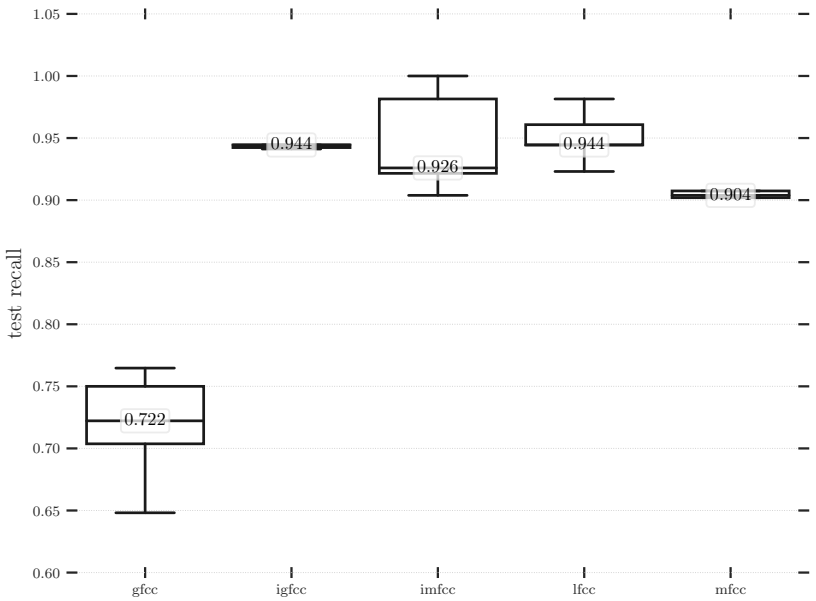
Figur 236: *test accuracy* for spsquare snr 20db

C.31.2 test precision



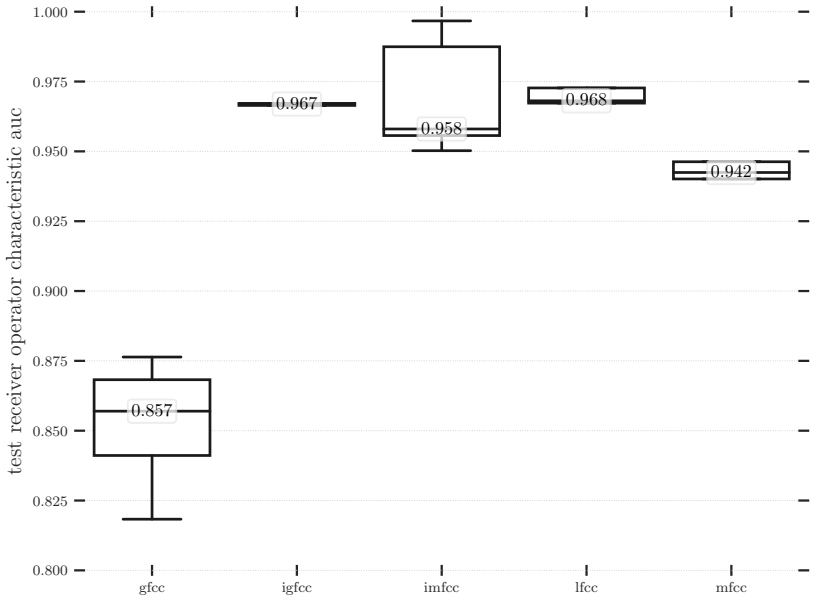
Figur 237: *test precision* for spsquare snr 20db

C.31.3 test recall



Figur 238: *test recall* for spsquare snr 20db

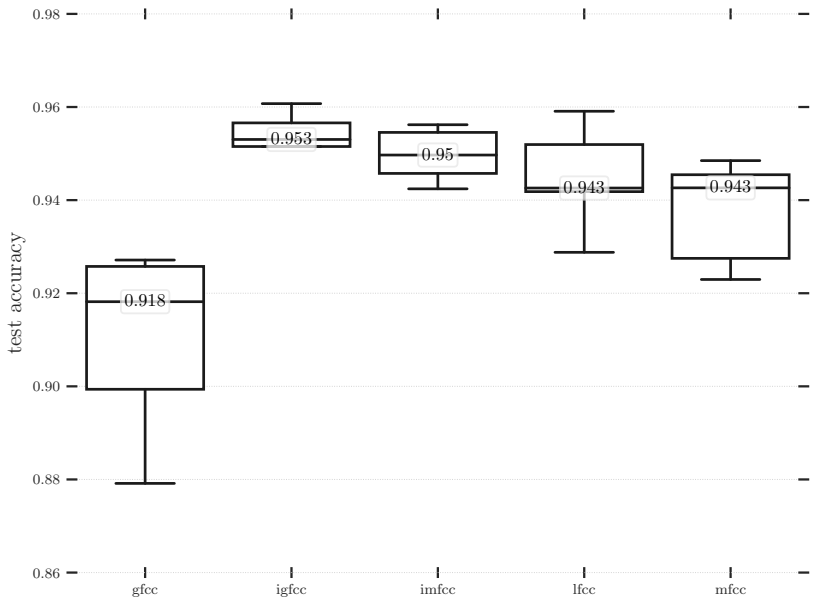
C.31.4 test receiver operator characteristic auc



Figur 239: *test receiver operator characteristic auc* for spsquare snr 20db

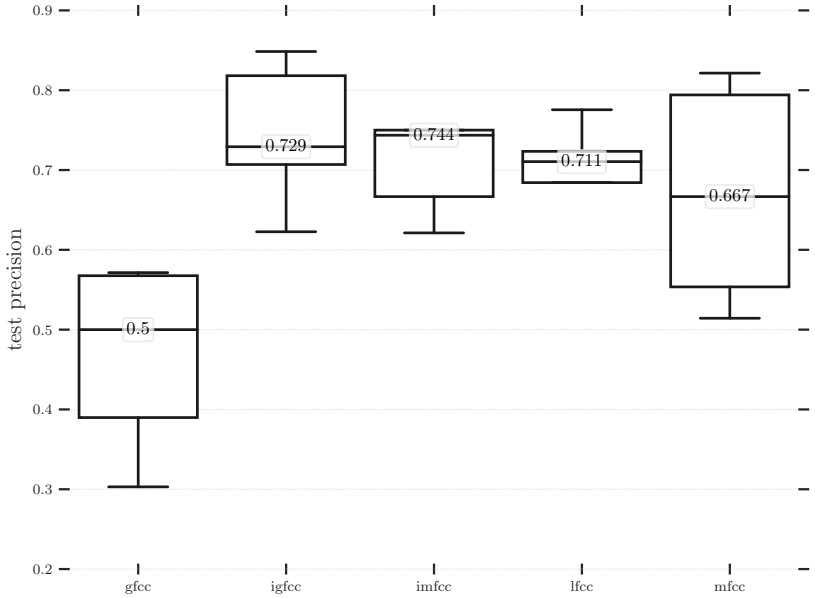
C.32 tbus

C.32.1 test accuracy



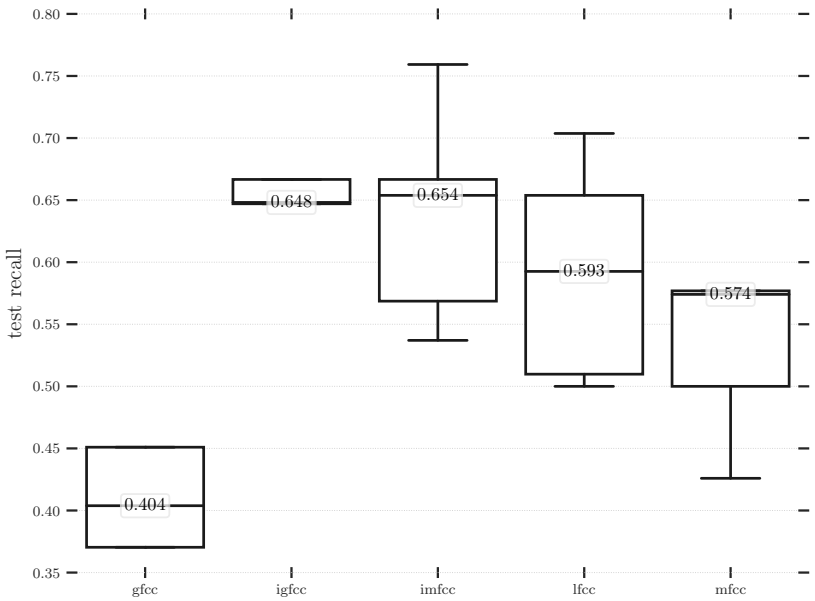
Figur 240: *test accuracy* for tbus snr 0db

C.32.2 test precision



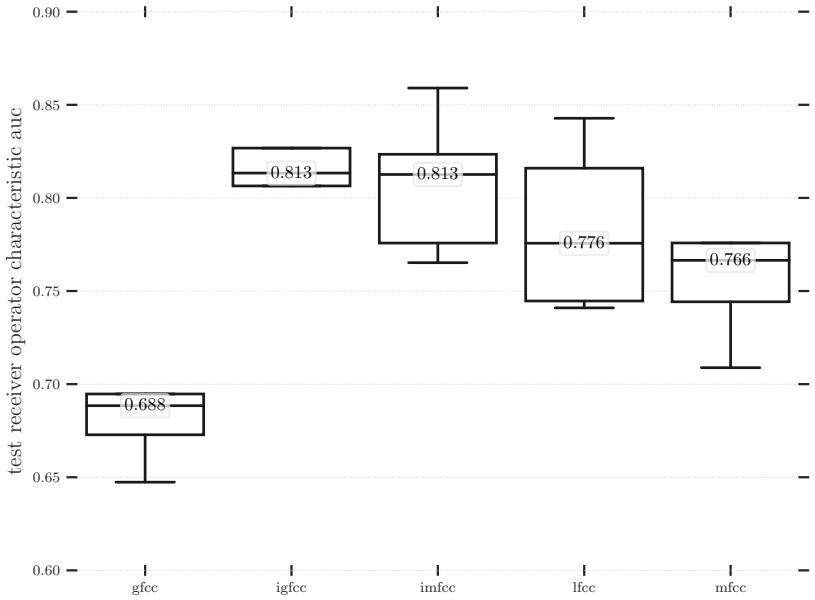
Figur 241: *test precision* for tbus snr 0db

C.32.3 test recall



Figur 242: *test recall* for tbus snr 0db

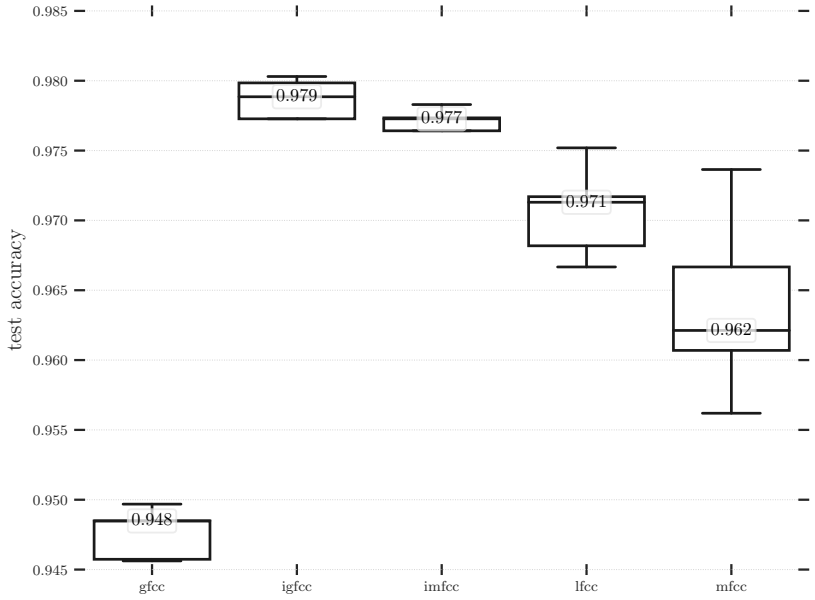
C.32.4 test receiver operator characteristic auc



Figur 243: *test receiver operator characteristic auc* for tbus snr 0db

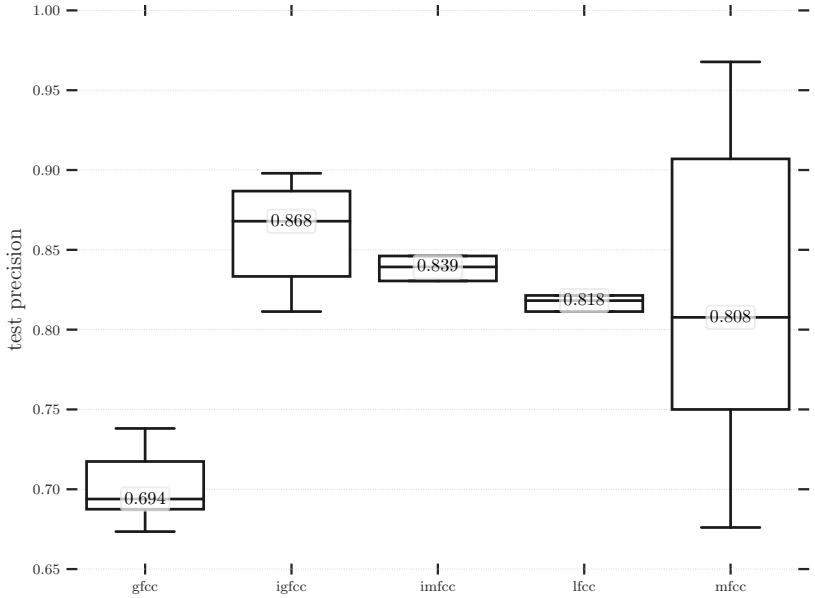
C.33 tbus

C.33.1 test accuracy



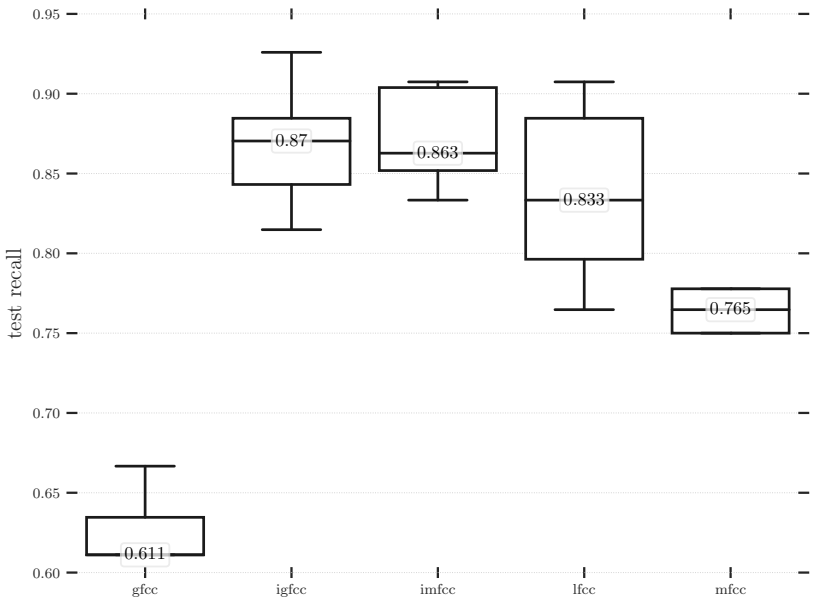
Figur 244: *test accuracy* for tbus snr 5db

C.33.2 test precision



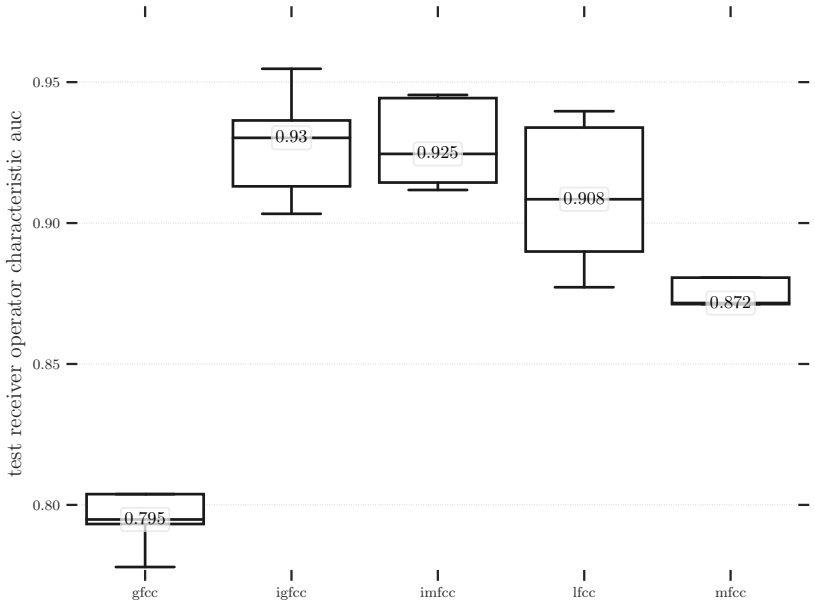
Figur 245: *test precision* for tbus snr 5db

C.33.3 test recall



Figur 246: *test recall* for tbus snr 5db

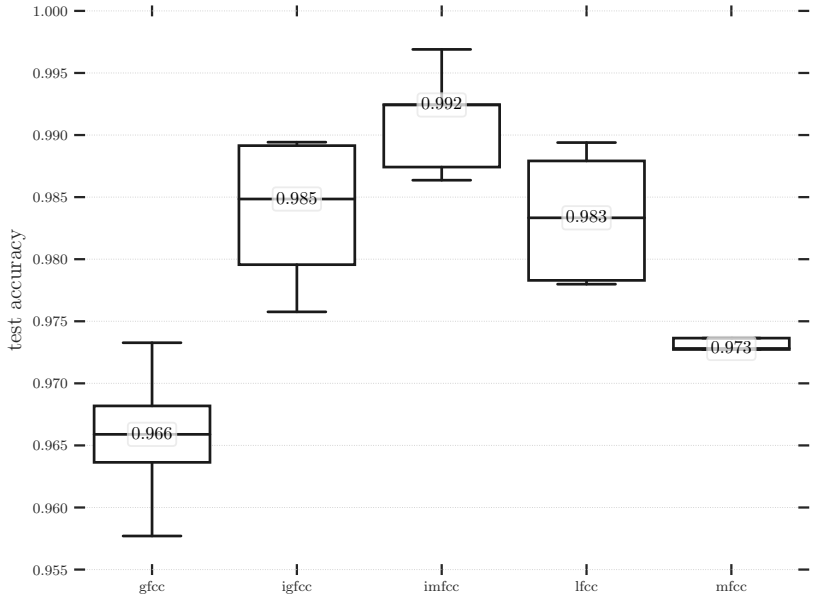
C.33.4 test receiver operator characteristic auc



Figur 247: *test receiver operator characteristic auc* for tbus snr 5db

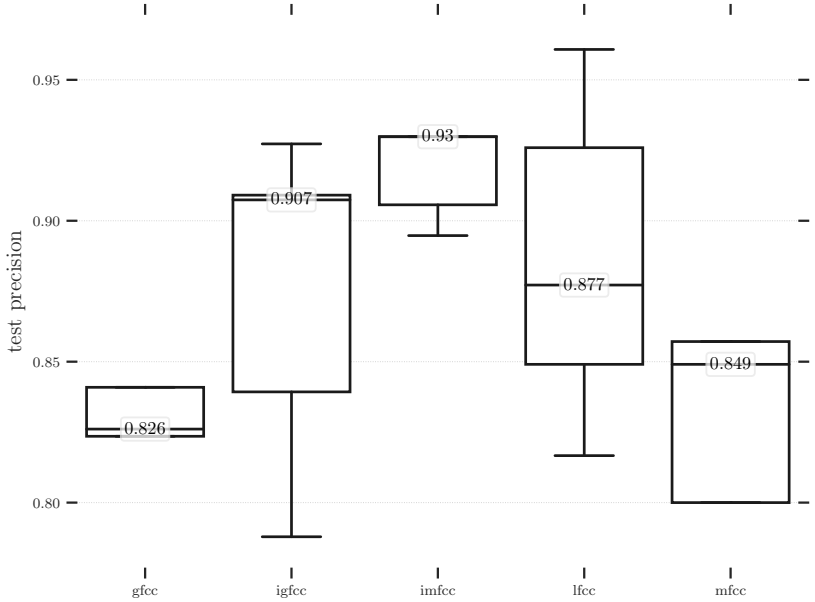
C.34 tbus

C.34.1 test accuracy



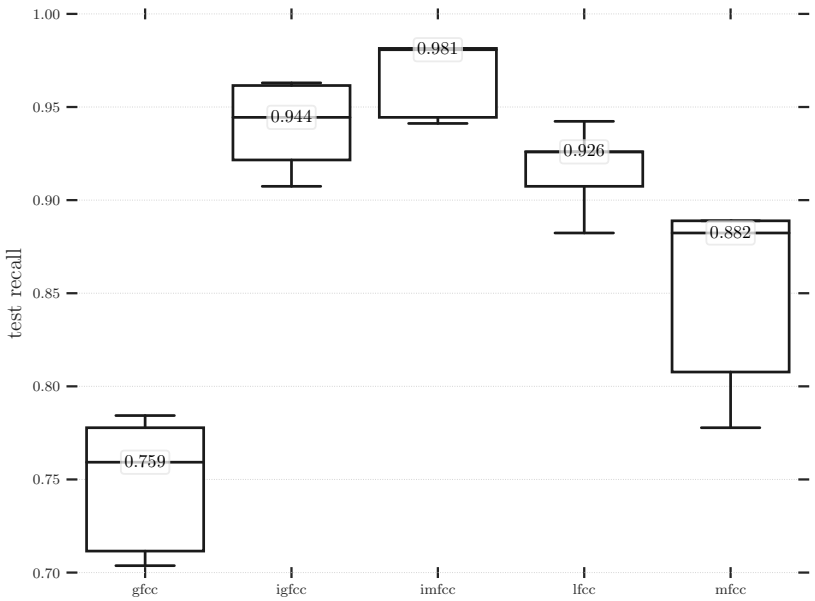
Figur 248: *test accuracy* for tbus snr 10db

C.34.2 test precision



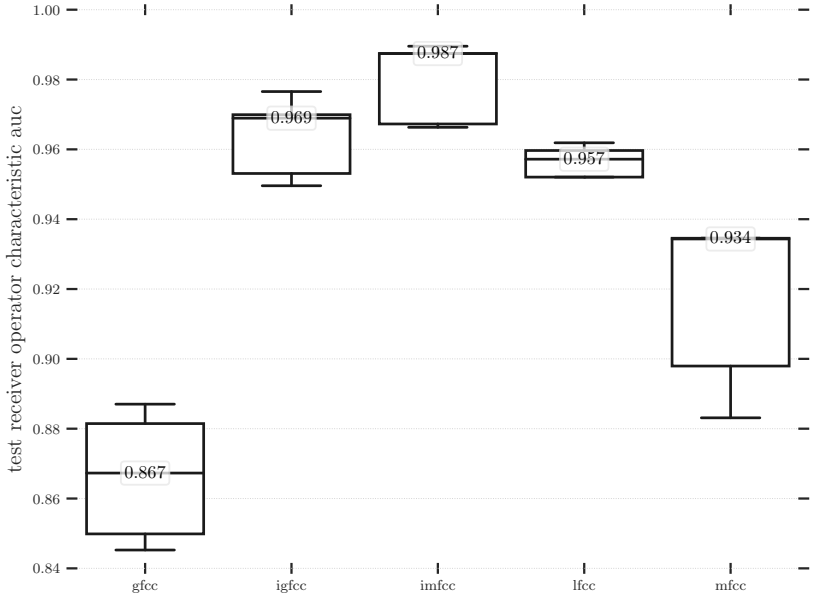
Figur 249: *test precision* for tbus snr 10db

C.34.3 test recall



Figur 250: *test recall* for tbus snr 10db

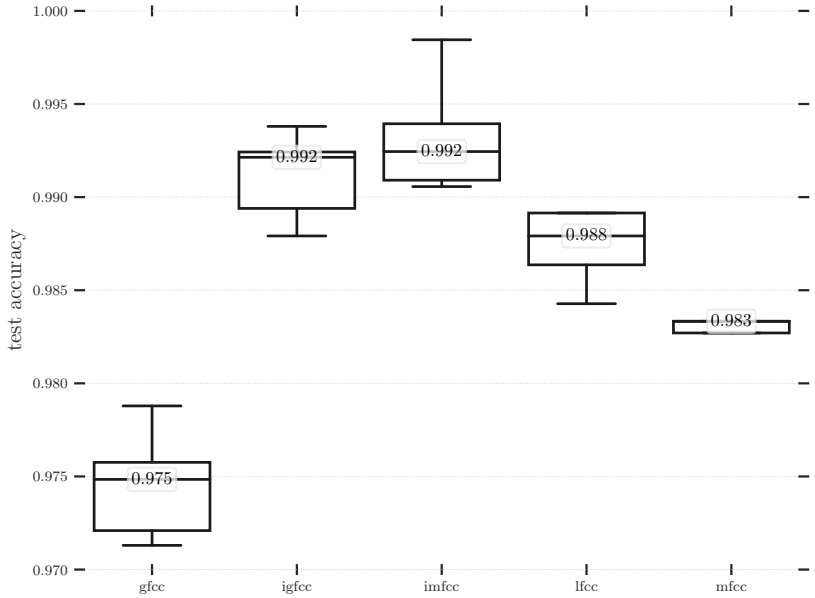
C.34.4 test receiver operator characteristic auc



Figur 251: *test receiver operator characteristic auc* for tbus snr 10db

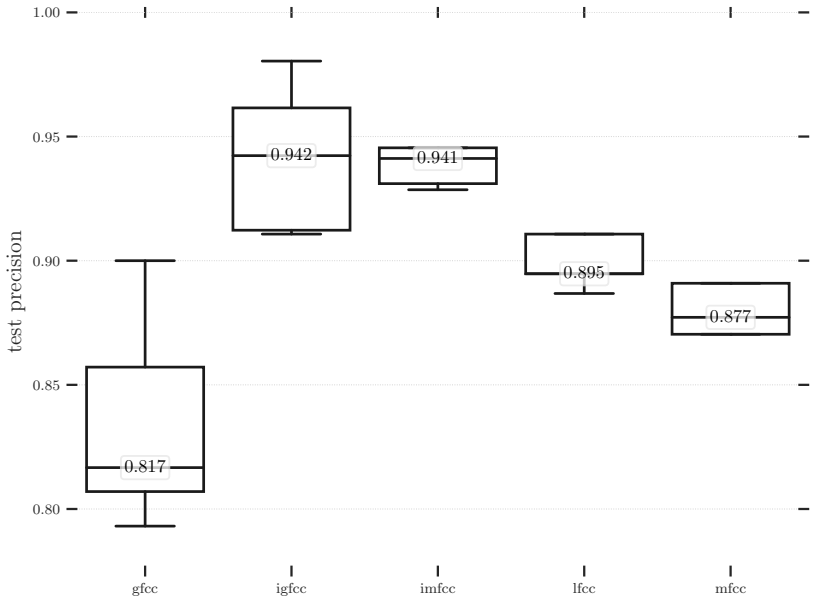
C.35 tbus

C.35.1 test accuracy



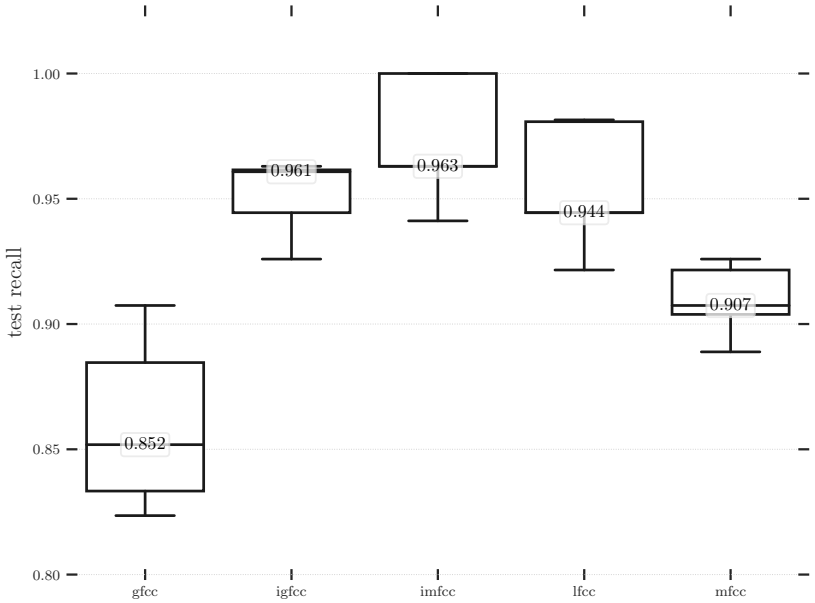
Figur 252: *test accuracy* for tbus snr 15db

C.35.2 test precision



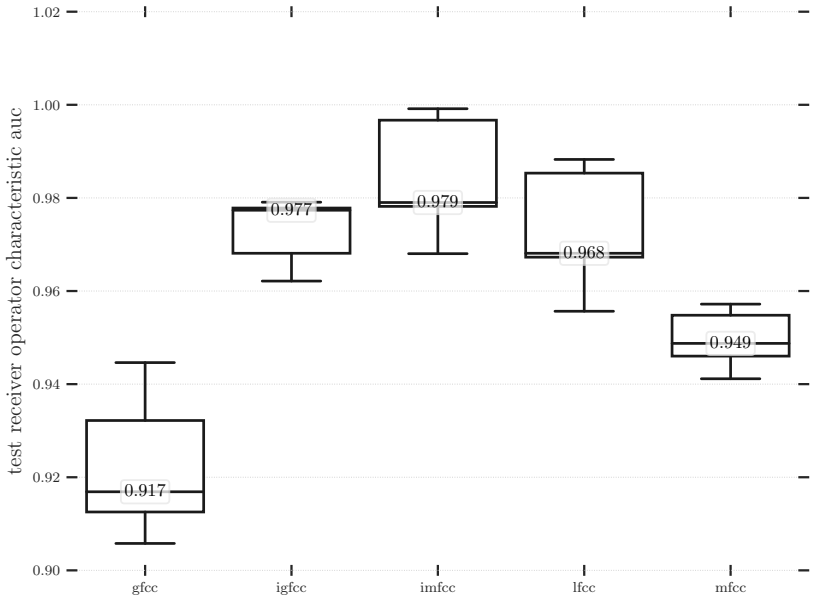
Figur 253: *test precision* for tbus snr 15db

C.35.3 test recall



Figur 254: *test recall* for tbus snr 15db

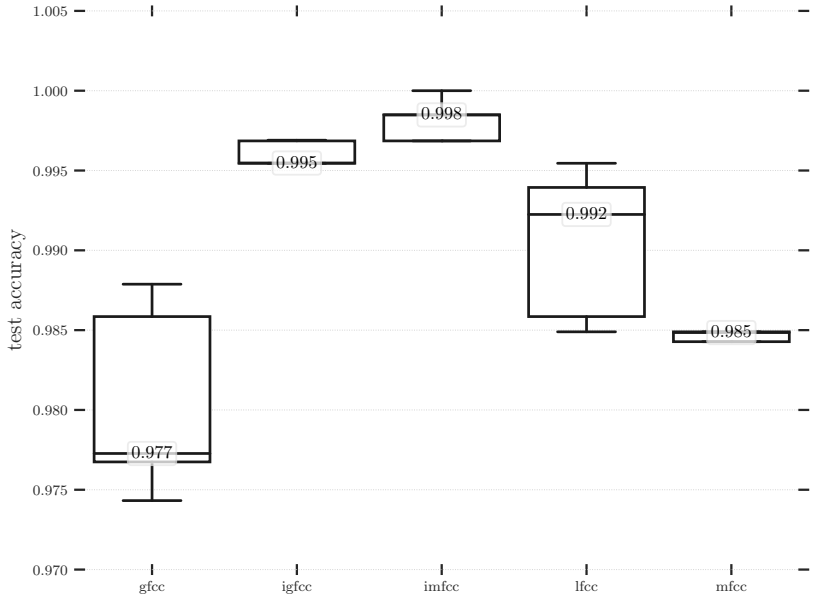
C.35.4 test receiver operator characteristic auc



Figur 255: *test receiver operator characteristic auc* for tbus snr 15db

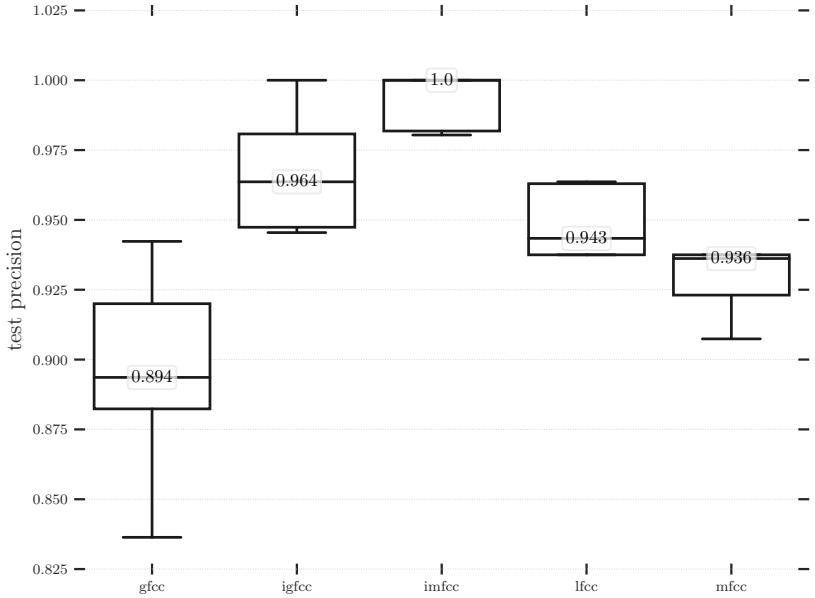
C.36 tbus

C.36.1 test accuracy



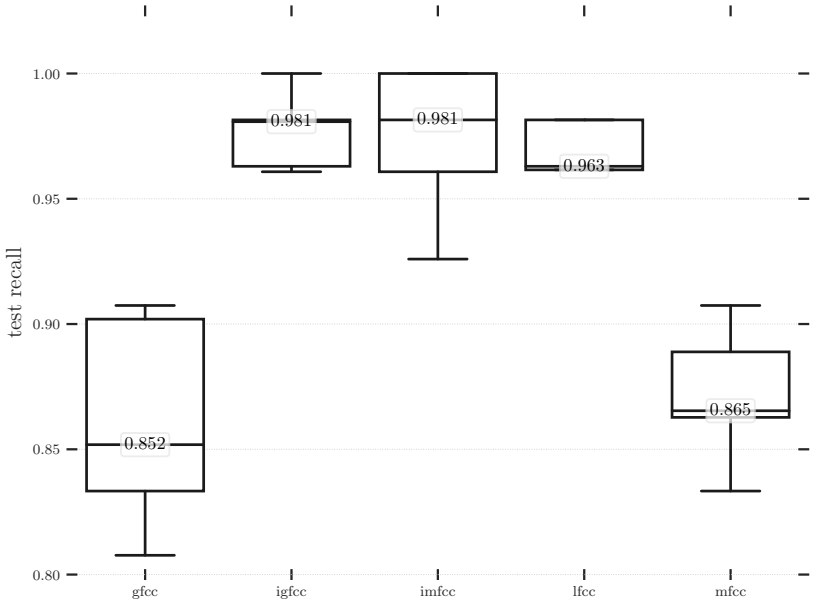
Figur 256: *test accuracy* for tbus snr 20db

C.36.2 test precision



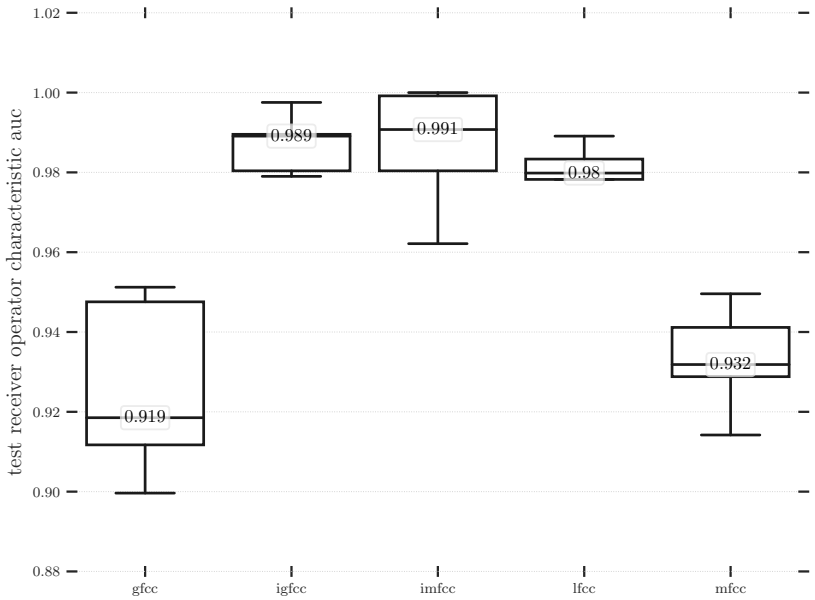
Figur 257: *test precision* for tbus snr 20db

C.36.3 test recall



Figur 258: *test recall* for tbus snr 20db

C.36.4 test receiver operator characteristic auc



Figur 259: *test receiver operator characteristic auc* for tbus snr 20db

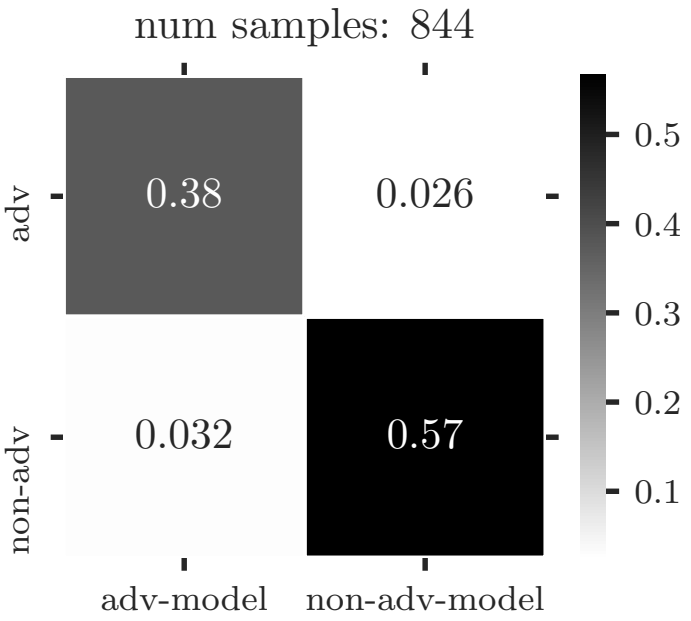
D Model Results CF

Due to time limit it was not possible to include all the confusion matrices of the cepstral space for the majority of the following experiments, it is only the MFCC results that are displayed in this version of the document. Only the noise experimentD.6 has all of the five transformations confusion matrices presented.

Note that confusion matrices are normalised across all entries, as this gives an indication of the categorical distribution of the truth labels in the test sets. Which

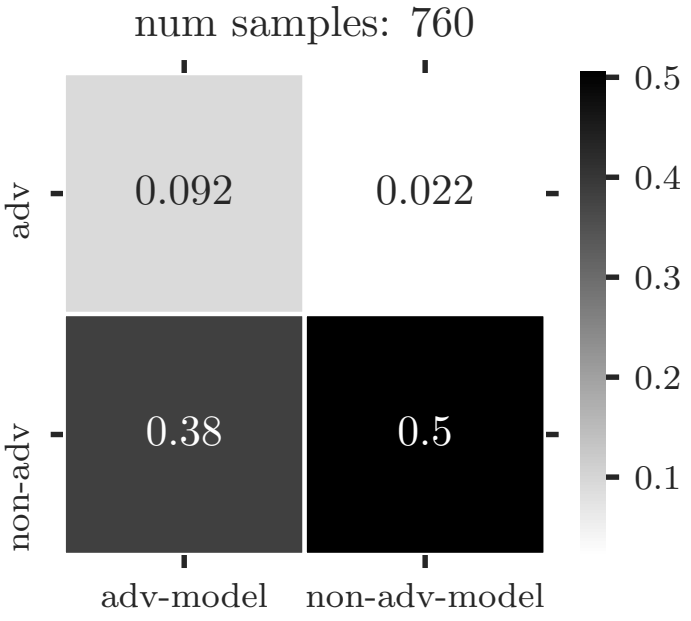
D.1 truncated a model a a

D.1.1 test a



Figur 260: MFCC, Confusion Matrix for truncated a model a a, test a

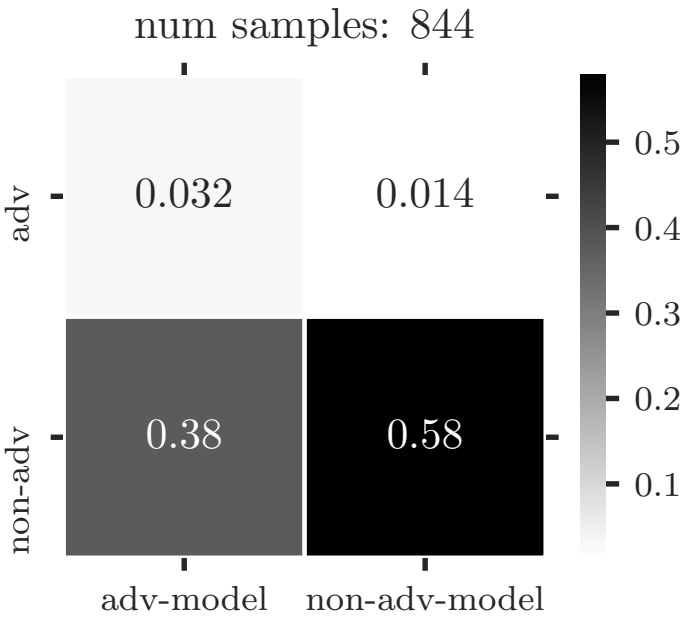
D.1.2 test b



Figur 261: MFCC, Confusion Matrix for truncated a model a a, test b

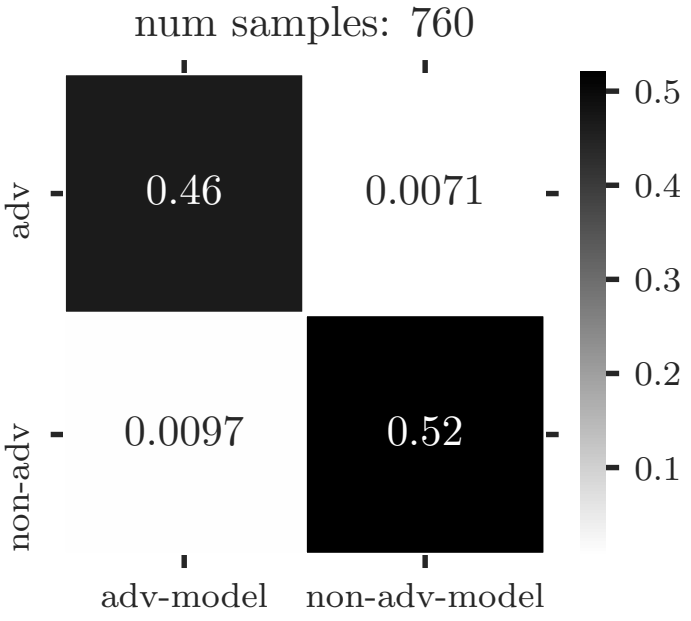
D.2 truncated b model b b

D.2.1 test a



Figur 262: MFCC, Confusion Matrix for truncated b model b b, test a

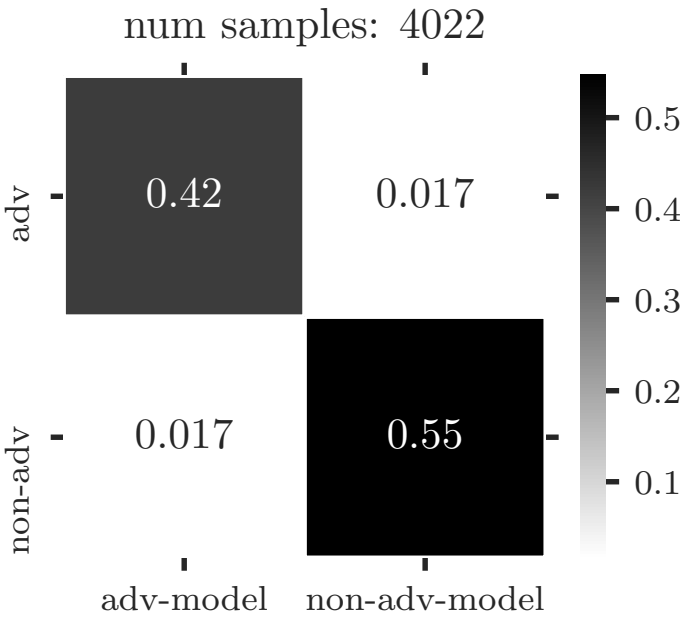
D.2.2 test b



Figur 263: MFCC, Confusion Matrix for truncated b model b b, test b

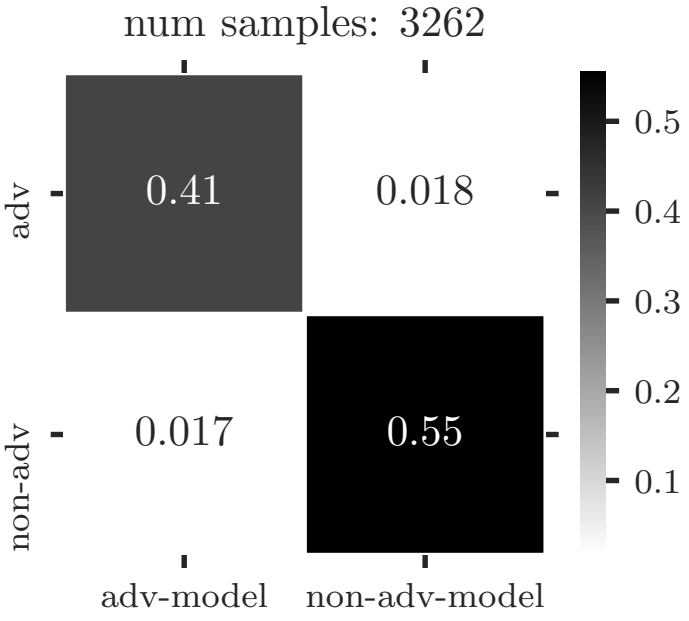
D.3 merge ab test ab model ab ab

D.3.1 test a b



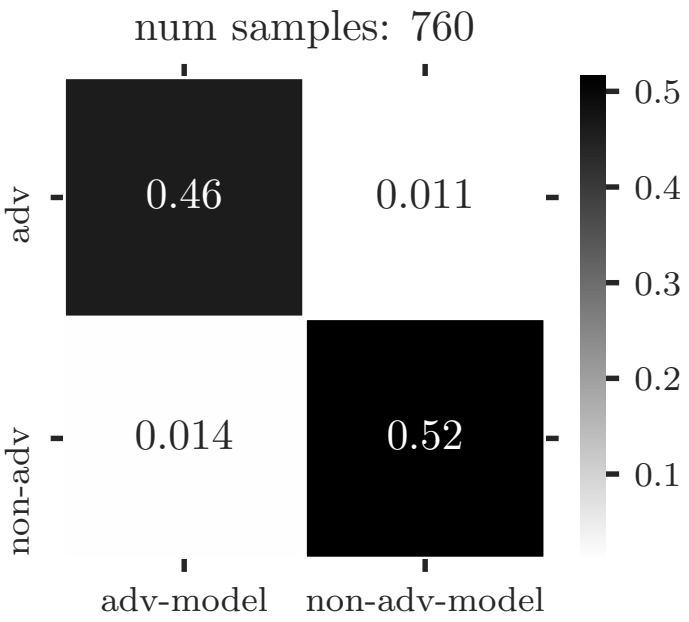
Figur 264: MFCC, Confusion Matrix for merge ab test ab model ab ab, test a b

D.3.2 test a



Figur 265: MFCC, Confusion Matrix for merge ab test ab model ab ab, test a

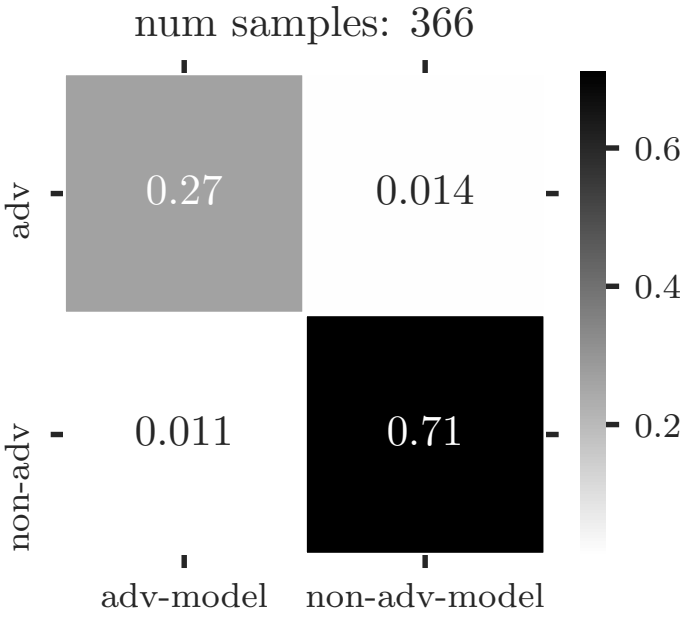
D.3.3 test b



Figur 266: MFCC, Confusion Matrix for merge ab test ab model ab ab, test b

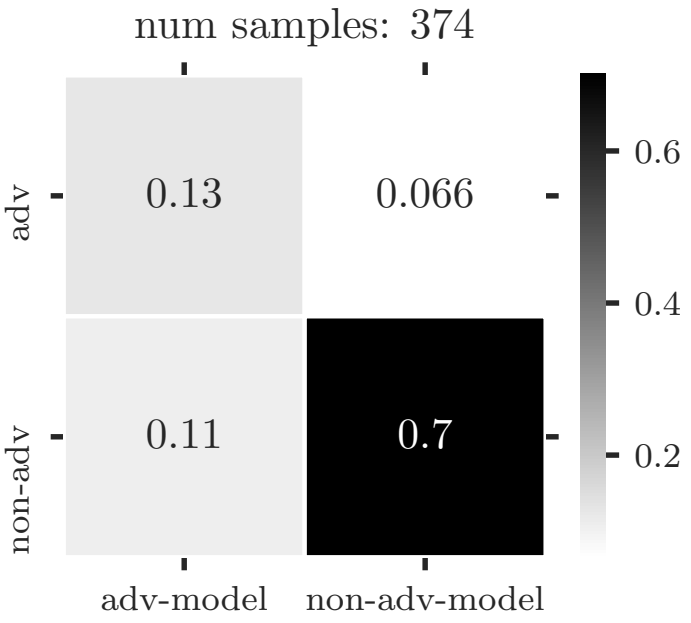
D.4 truncated a silence model silence silence

D.4.1 test silence



Figur 267: MFCC, Confusion Matrix for truncated a silence model silence silence, test silence

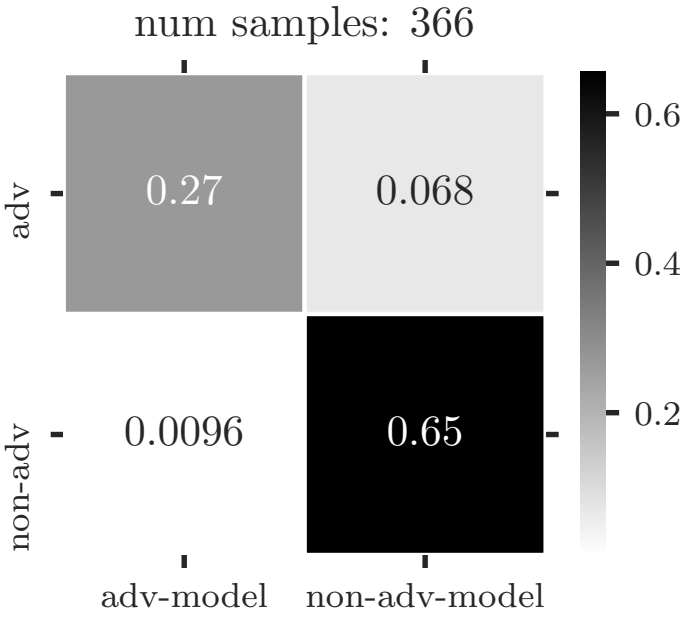
D.4.2 test speech



Figur 268: MFCC, Confusion Matrix for truncated a silence model silence silence, test speech

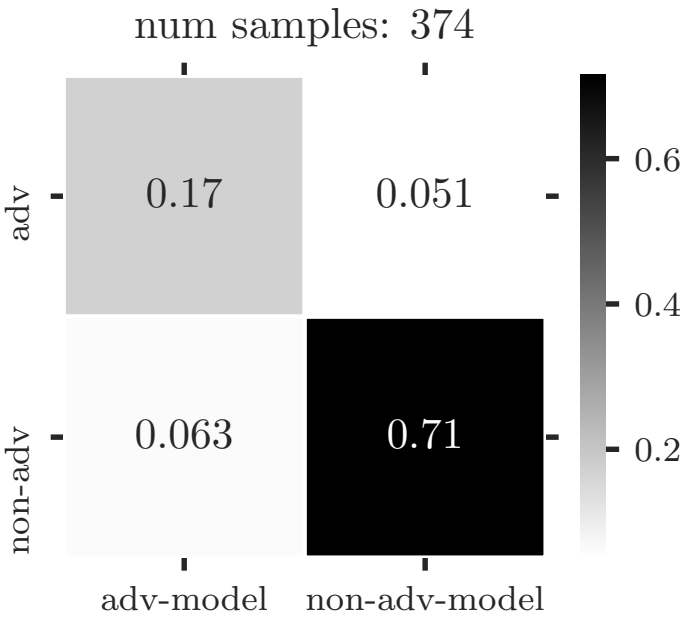
D.5 truncated a speech model speech speech

D.5.1 test silence



Figur 269: MFCC, Confusion Matrix for truncated a speech model speech speech, test silence

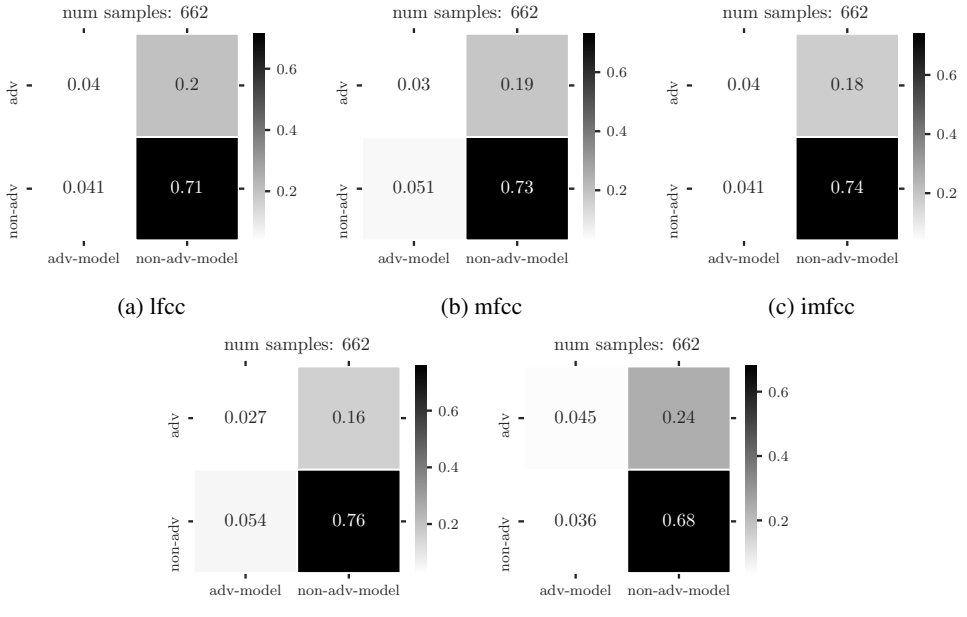
D.5.2 test speech



Figur 270: MFCC, Confusion Matrix for truncated a speech model speech speech, test speech

D.6 noises all snr to pcafeter model all noise snr all noise snr

D.6.1 test pcafeter snr 0db

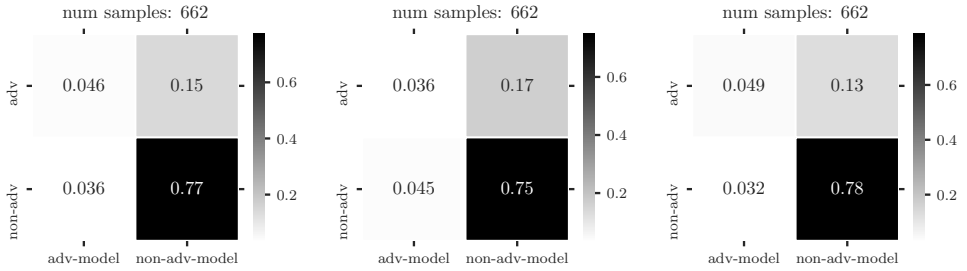


(a) lfcc (b) mfcc (c) imfcc

(d) gfcc (e) igfcc

Figur 271: Confusion Matrix for noises all snr to pcafeter model all noise snr all noise snr, test pcafeter snr 0db

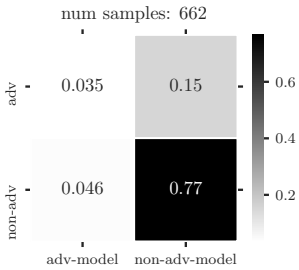
D.6.2 test pcafeter snr 5db



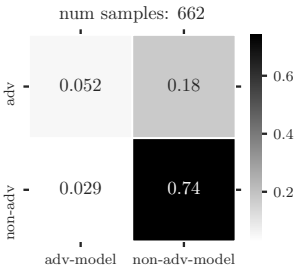
(a) lfcc

(b) mfcc

(c) imfcc



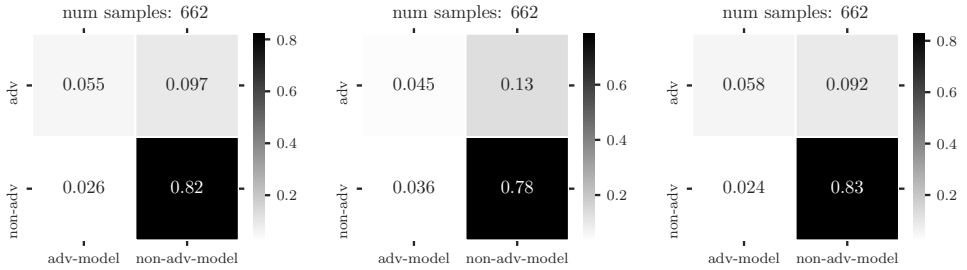
(d) gfcc



(e) igfcc

Figur 272: Confusion Matrix for noises all snr to pcafeter model all noise snr all noise snr, test pcafeter snr 5db

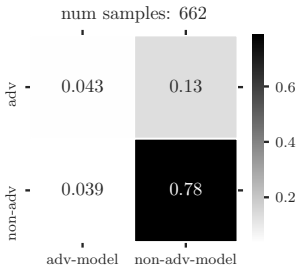
D.6.3 test pcafeter snr 10db



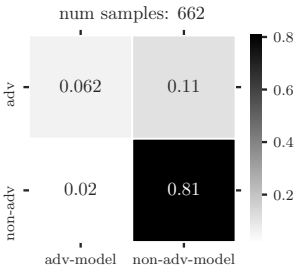
(a) lfcc

(b) mfcc

(c) imfcc



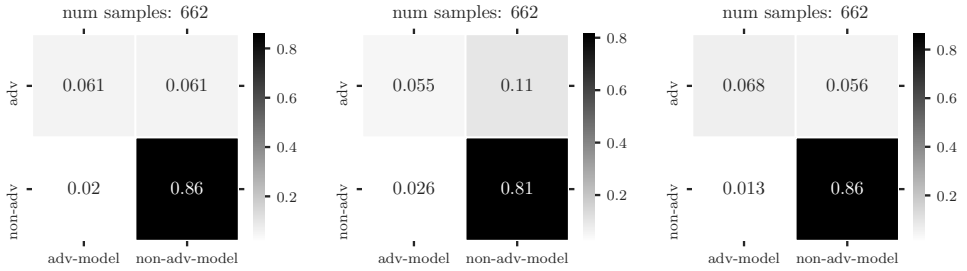
(d) gfcc



(e) igfcc

Figur 273: Confusion Matrix for noises all snr to pcafeter model all noise snr all noise snr, test pcafeter snr 10db

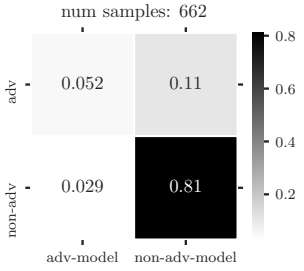
D.6.4 test pcafeter snr 15db



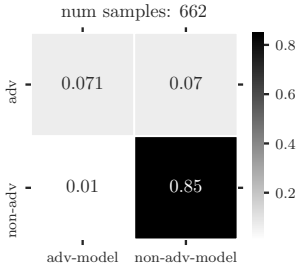
(a) lfcc

(b) mfcc

(c) imfcc



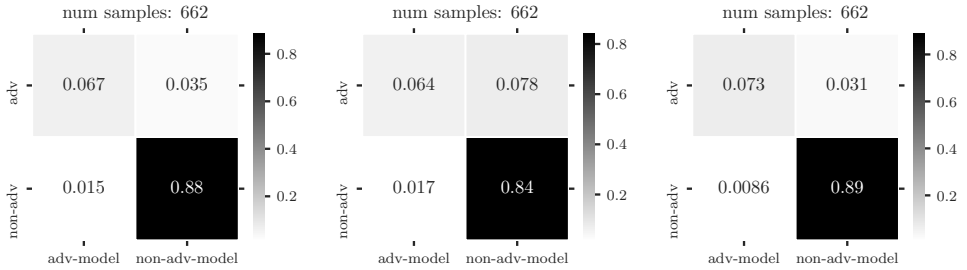
(d) gfcc



(e) igfcc

Figur 274: Confusion Matrix for noises all snr to pcafeter model all noise snr all noise snr, test pcafeter snr 15db

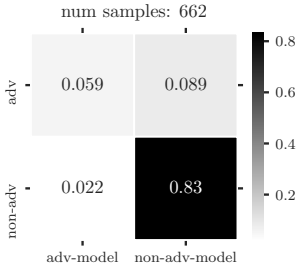
D.6.5 test pcafeter snr 20db



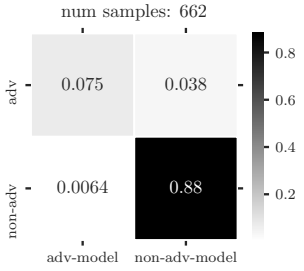
(a) lfcc

(b) mfcc

(c) imfcc



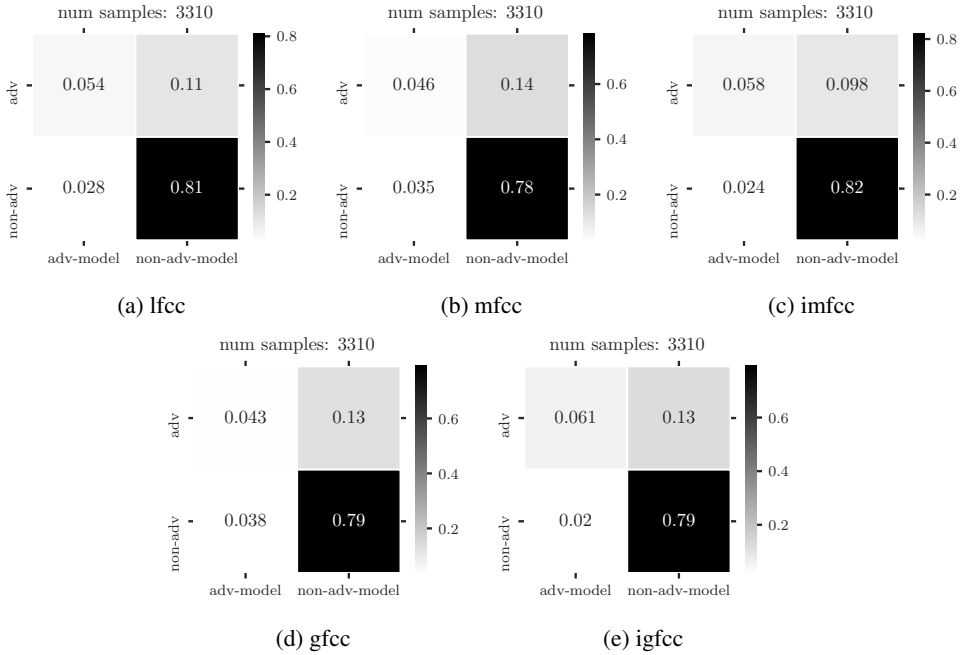
(d) gfcc



(e) igfcc

Figur 275: Confusion Matrix for noises all snr to pcafeter model all noise snr all noise snr, test pcafeter snr 20db

D.6.6 test pcafeter snr 0db pcafeter snr 5db pcafeter snr 10db pcafeter snr 15db pcafeter snr 20db

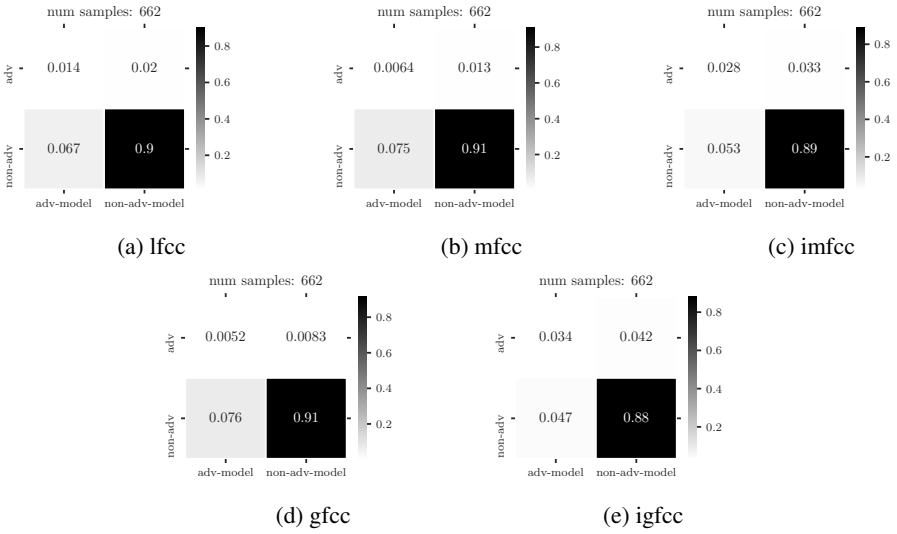


(a) lfcc (b) mfcc (c) imfcc

(d) gfcc (e) igfcc

Figur 276: Confusion Matrix for noises all snr to pcafeter model all noise snr all noise snr, test pcafeter snr 0db pcafeter snr 5db pcafeter snr 10db pcafeter snr 15db pcafeter snr 20db

D.7 bbl morten

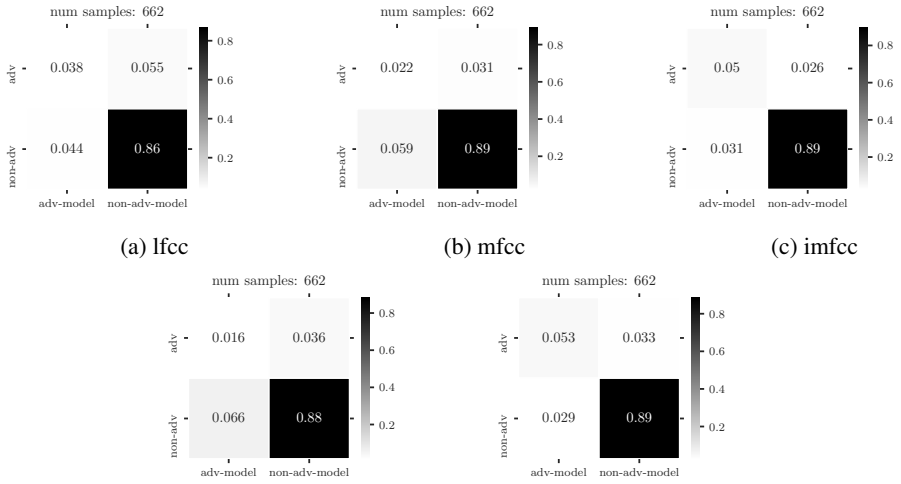


(a) lfcc (b) mfcc (c) imfcc

(d) gfcc (e) igfcc

Figur 277: Confusion Matrix for bbl morten snr 0db

D.8 bbl morten

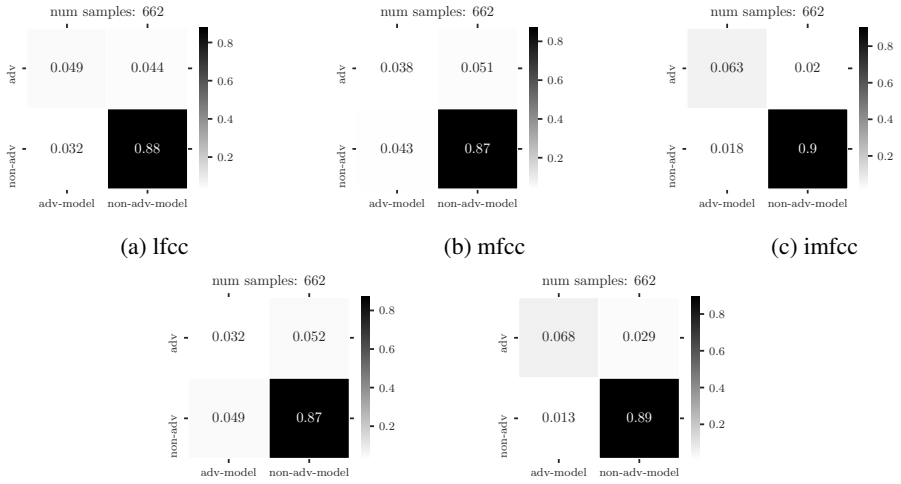


(a) lfcc (b) mfcc (c) imfcc

(d) gfcc (e) igfcc

Figur 278: Confusion Matrix for bbl morten snr 5db

D.9 bbl morten

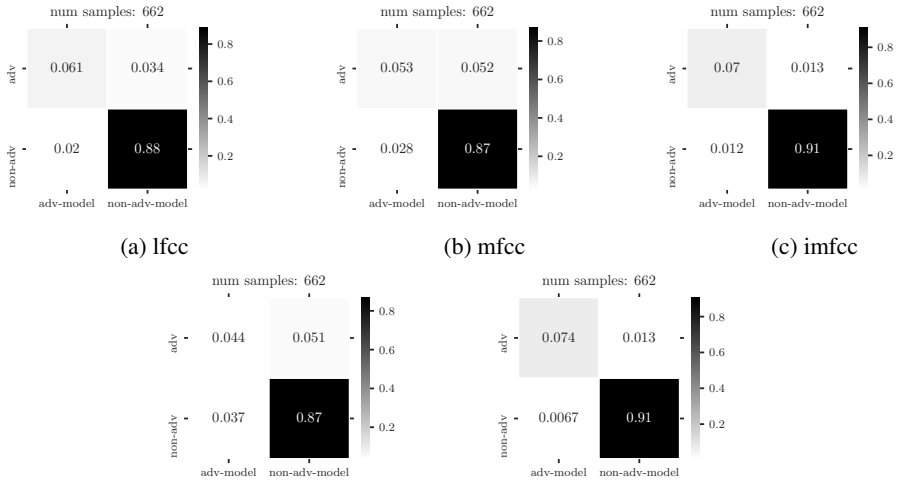


(a) lfcc (b) mfcc (c) imfcc

(d) gfcc (e) igfcc

Figur 279: Confusion Matrix for bbl morten snr 10db

D.10 bbl morten

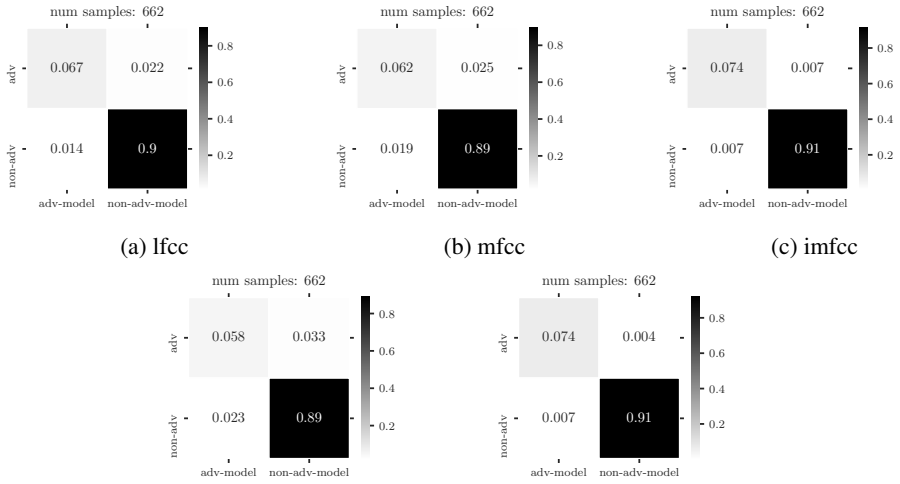


(a) lfcc (b) mfcc (c) imfcc

(d) gfcc (e) igfcc

Figur 280: Confusion Matrix for bbl morten snr 15db

D.11 bbl morten

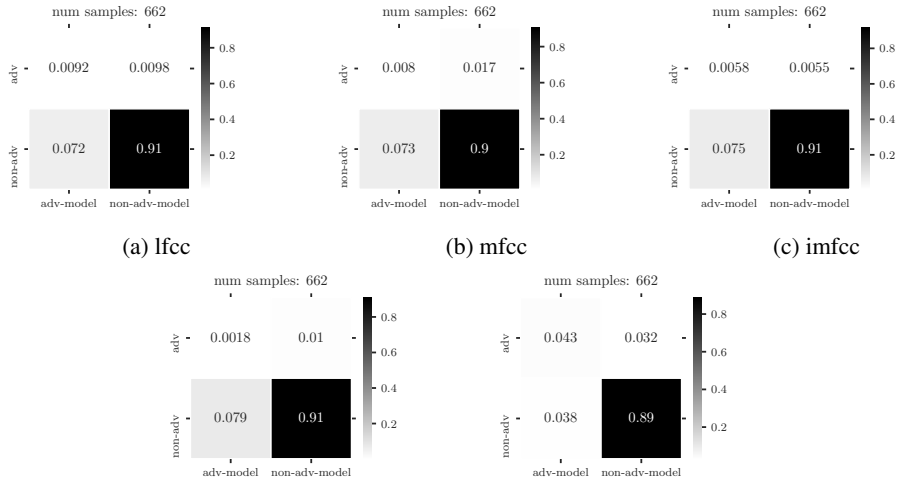


(a) lfcc (b) mfcc (c) imfcc

(d) gfcc (e) igfcc

Figur 281: Confusion Matrix for bbl morten snr 20db

D.12 ssn morten



(a) lfcc

(b) mfcc

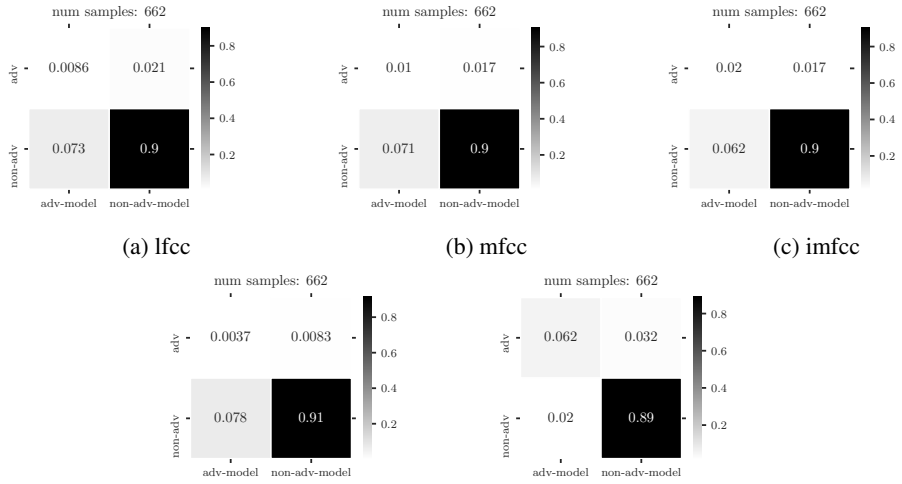
(c) imfcc

(d) gfcc

(e) igfcc

Figur 282: Confusion Matrix for ssn morten snr 0db

D.13 ssn morten



(a) lfcc

(b) mfcc

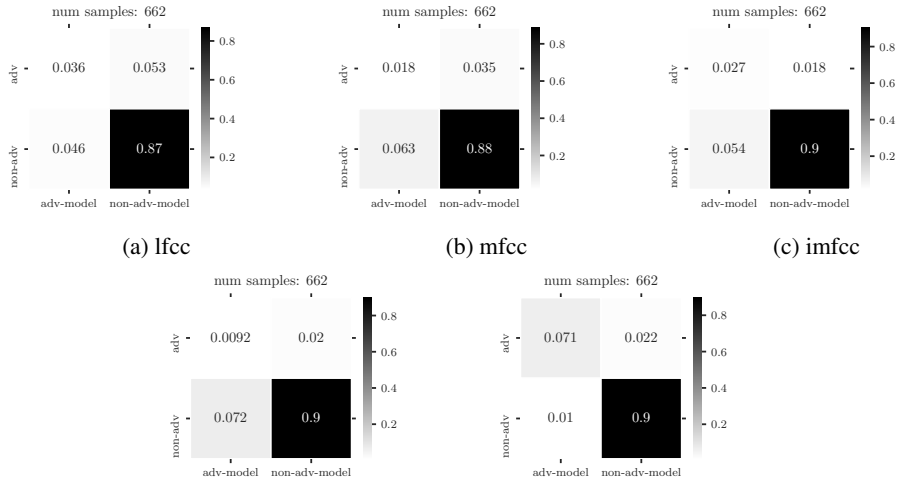
(c) imfcc

(d) gfcc

(e) igfcc

Figur 283: Confusion Matrix for ssn morten snr 5db

D.14 ssn morten



(a) lfcc

(b) mfcc

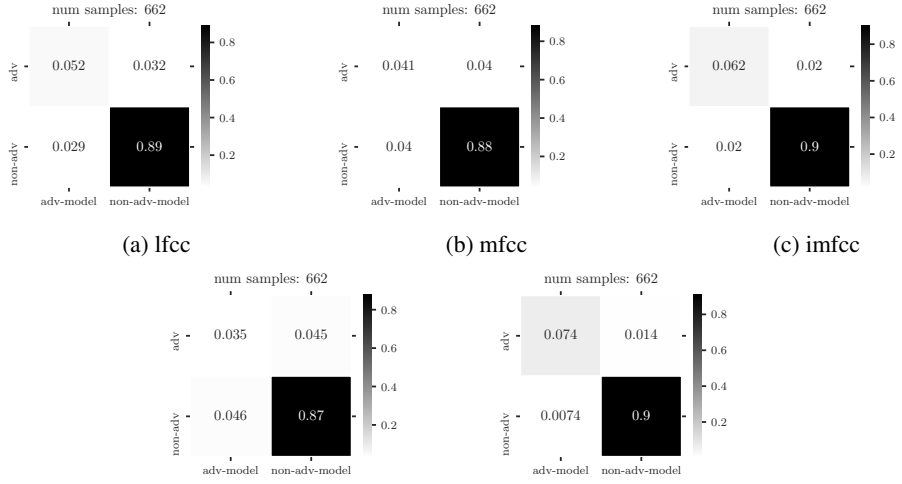
(c) imfcc

(d) gfcc

(e) igfcc

Figur 284: Confusion Matrix for ssn morten snr 10db

D.15 ssn morten



(a) lfcc

(b) mfcc

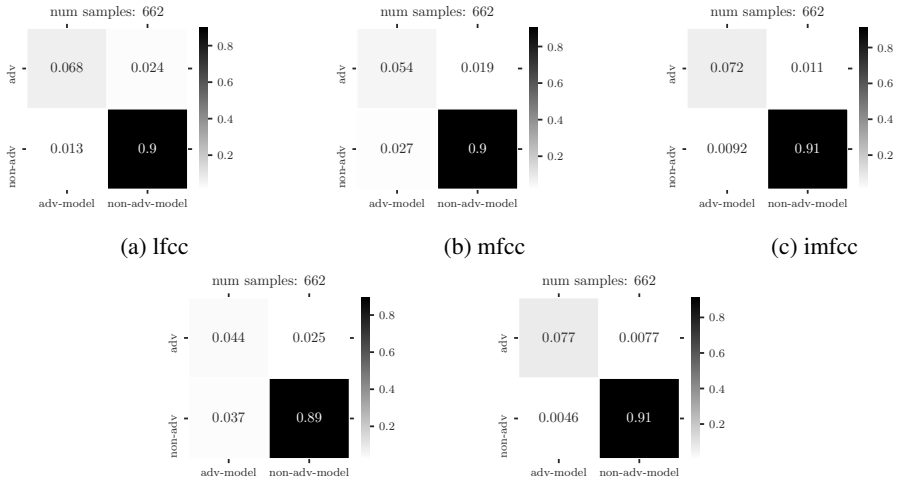
(c) imfcc

(d) gfcc

(e) igfcc

Figur 285: Confusion Matrix for ssn morten snr 15db

D.16 ssn morten



(a) lfcc

(b) mfcc

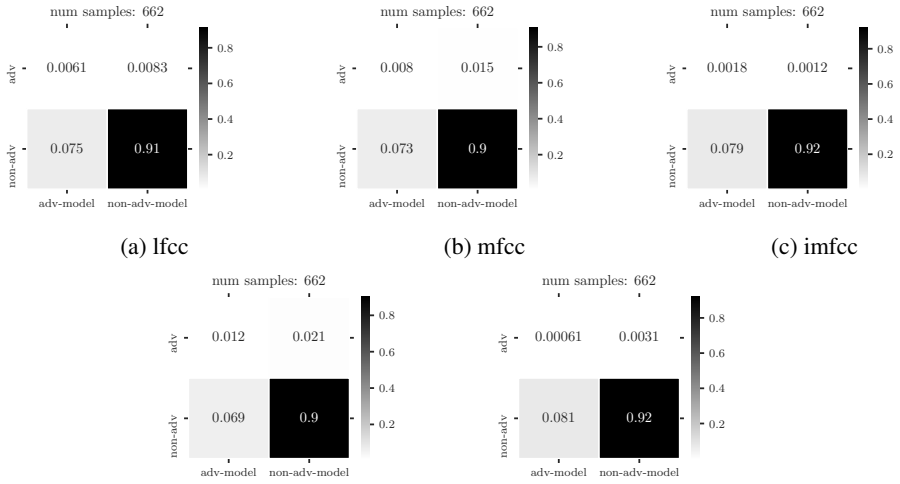
(c) imfcc

(d) gfcc

(e) igfcc

Figur 286: Confusion Matrix for ssn morten snr 20db

D.17 dkitchen



(a) lfcc

(b) mfcc

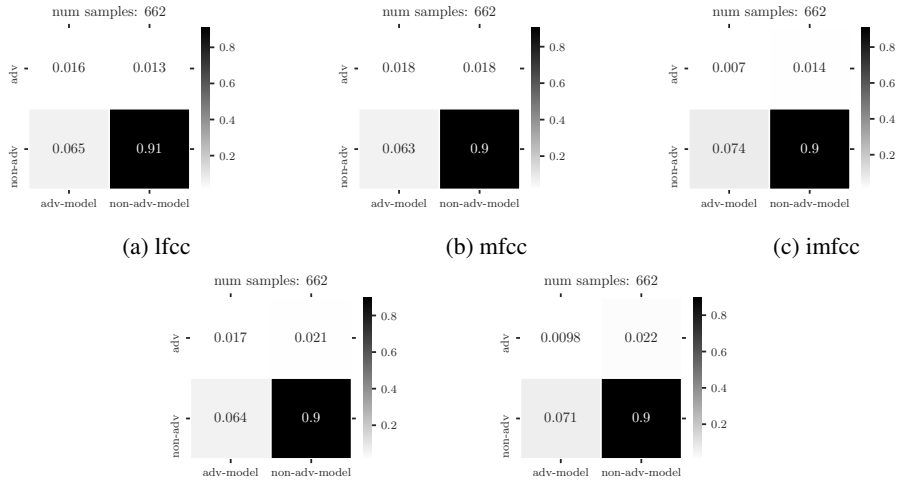
(c) imfcc

(d) gfcc

(e) igfcc

Figur 287: Confusion Matrix for dkitchen snr 0db

D.18 dkitchen



(a) lfcc

(b) mfcc

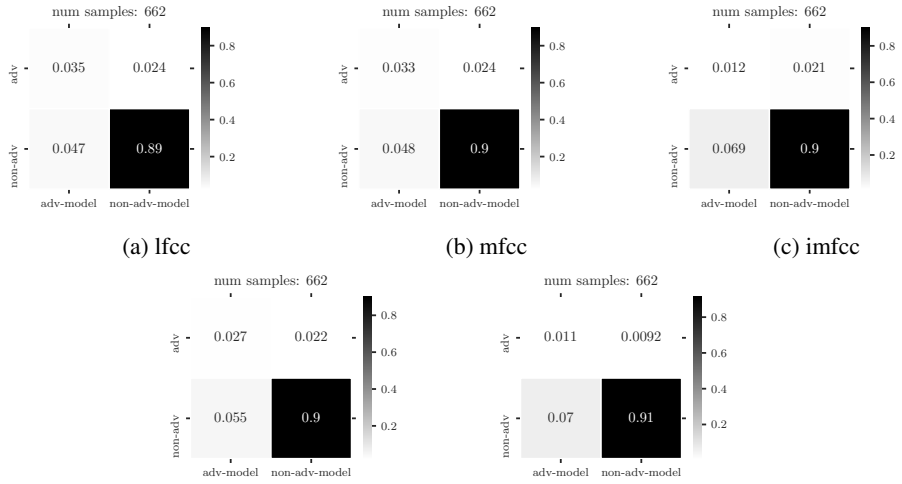
(c) imfcc

(d) gfcc

(e) igfcc

Figur 288: Confusion Matrix for dkitchen snr 5db

D.19 dkitchen



(a) lfcc

(b) mfcc

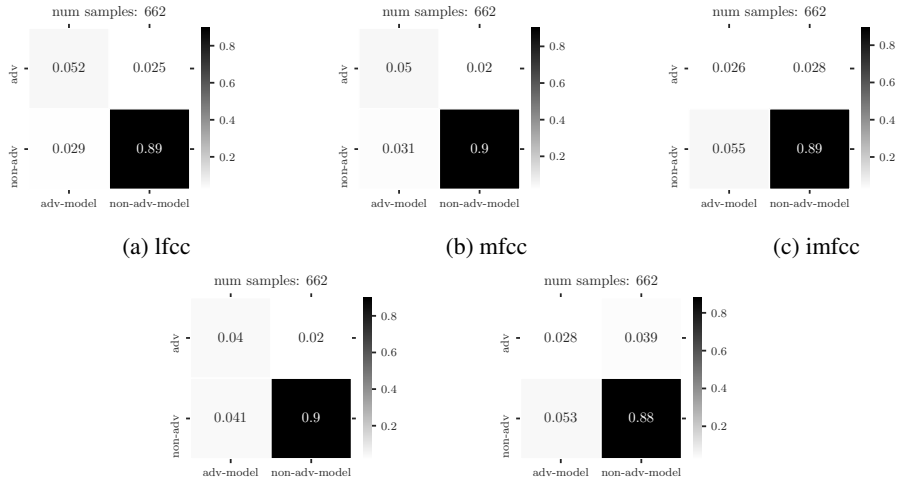
(c) imfcc

(d) gfcc

(e) igfcc

Figur 289: Confusion Matrix for dkitchen snr 10db

D.20 dkitchen



(a) lfcc

(b) mfcc

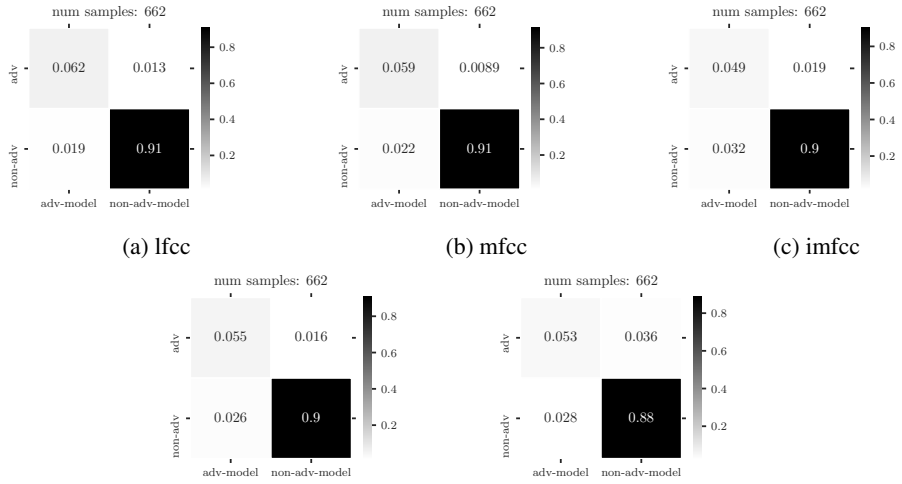
(c) imfcc

(d) gfcc

(e) igfcc

Figur 290: Confusion Matrix for dkitchen snr 15db

D.21 dkitchen



(a) lfcc

(b) mfcc

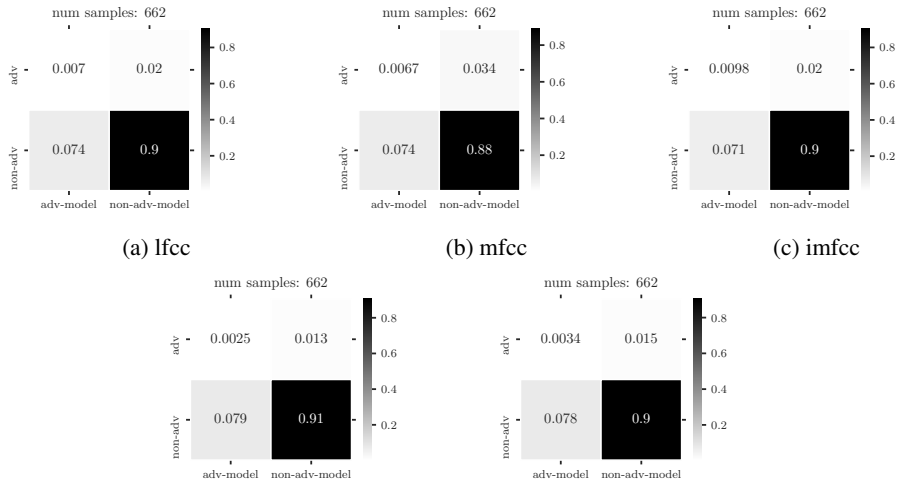
(c) imfcc

(d) gfcc

(e) igfcc

Figur 291: Confusion Matrix for dkitchen snr 20db

D.22 pcafeter



(a) lfcc

(b) mfcc

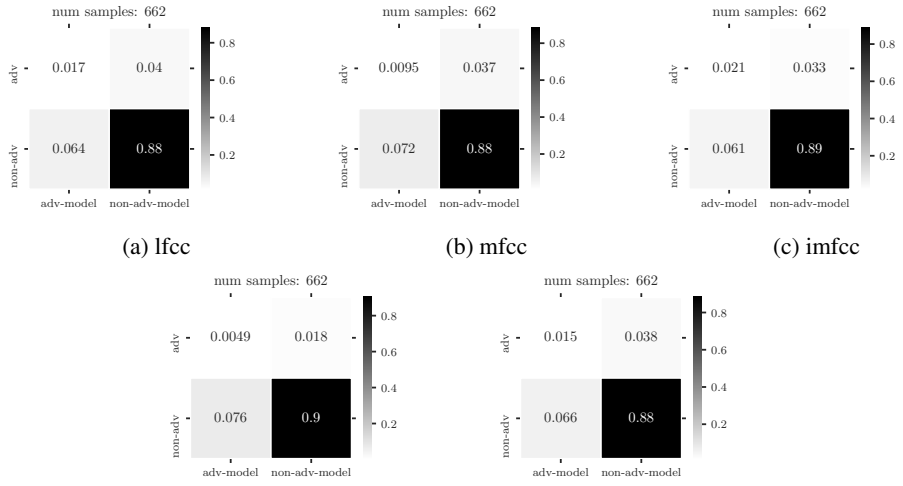
(c) imfcc

(d) gfcc

(e) igfcc

Figur 292: Confusion Matrix for pcafeter snr 0db

D.23 pcafeter



(a) lfcc

(b) mfcc

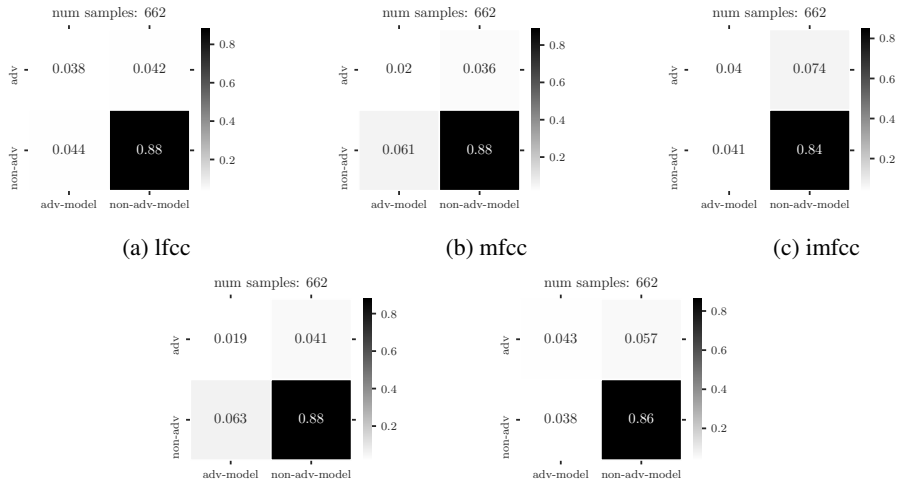
(c) imfcc

(d) gfcc

(e) igfcc

Figur 293: Confusion Matrix for pcafeter snr 5db

D.24 pcafeter



(a) lfcc

(b) mfcc

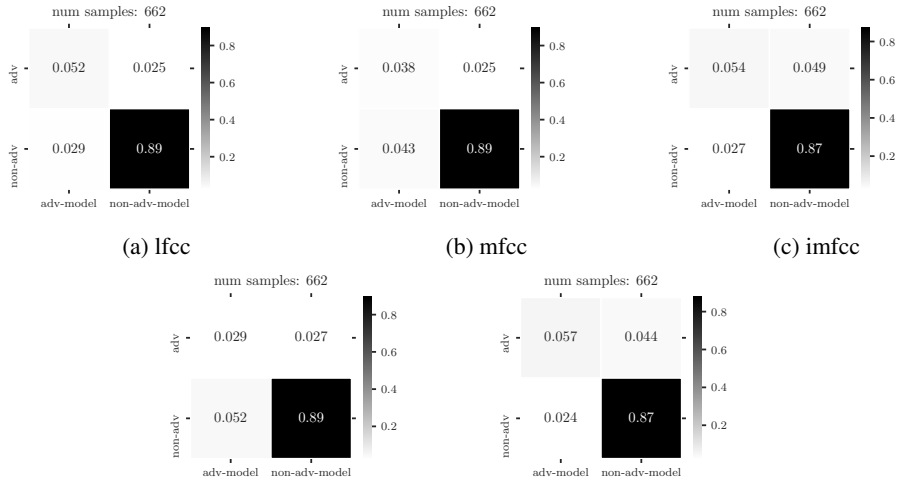
(c) imfcc

(d) gfcc

(e) igfcc

Figur 294: Confusion Matrix for pcafeter snr 10db

D.25 pcafeter



(a) lfcc

(b) mfcc

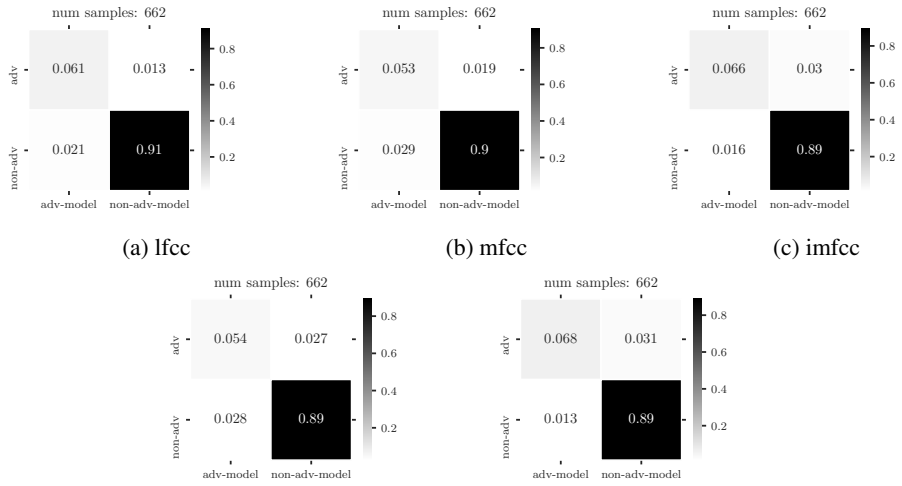
(c) imfcc

(d) gfcc

(e) igfcc

Figur 295: Confusion Matrix for pcafeter snr 15db

D.26 pcafeter



(a) lfcc

(b) mfcc

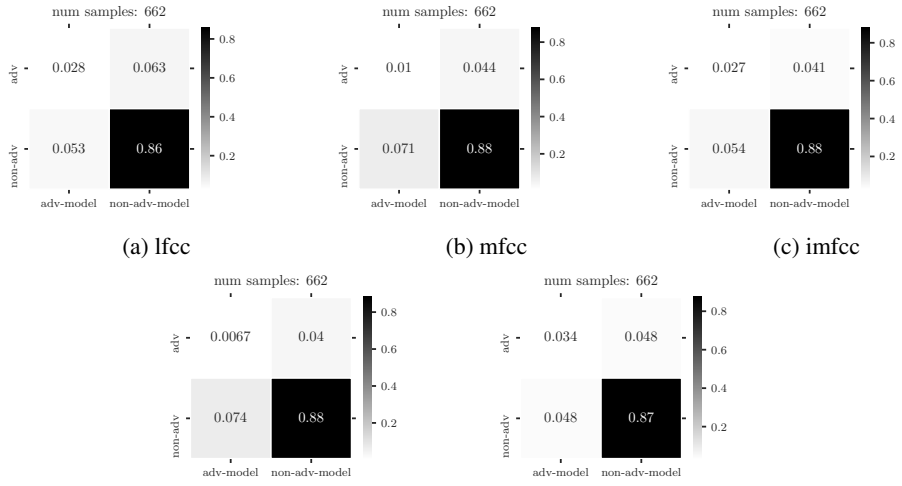
(c) imfcc

(d) gfcc

(e) igfcc

Figur 296: Confusion Matrix for pcafeter snr 20db

D.27 spsquare



(a) lfcc

(b) mfcc

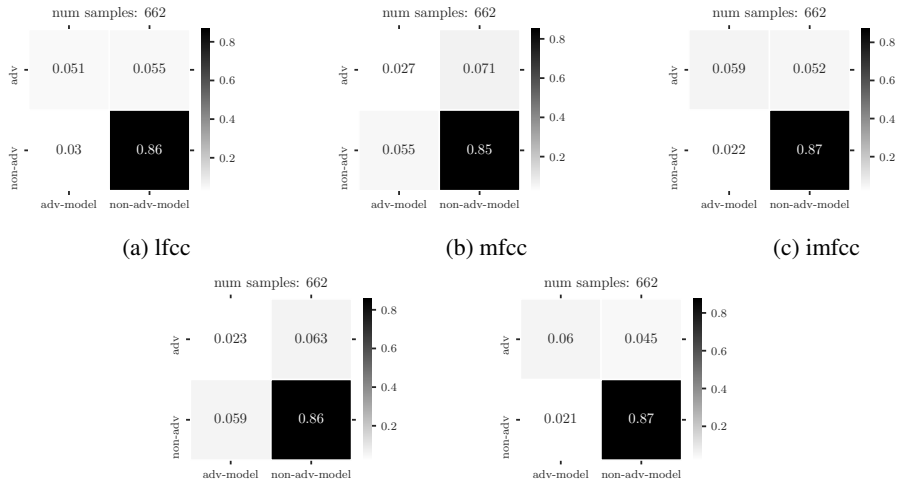
(c) imfcc

(d) gfcc

(e) igfcc

Figur 297: Confusion Matrix for spsquare snr 0db

D.28 spsquare



(a) lfcc

(b) mfcc

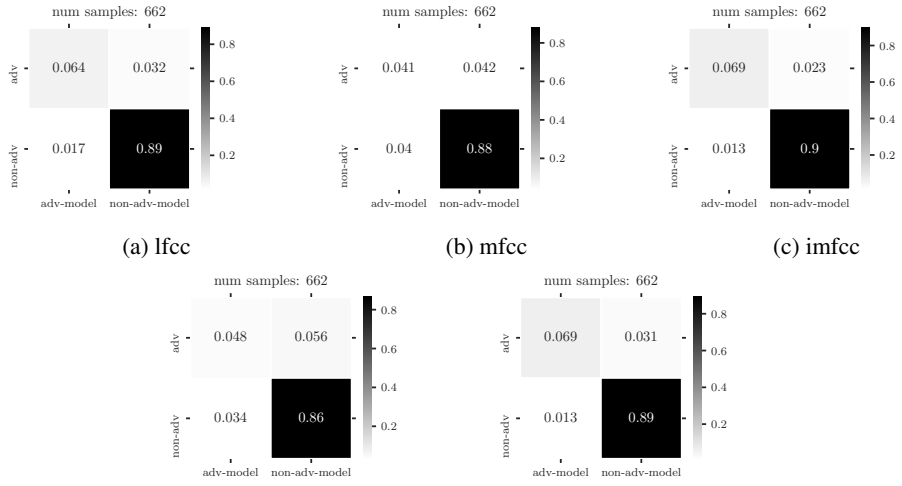
(c) imfcc

(d) gfcc

(e) igfcc

Figur 298: Confusion Matrix for spsquare snr 5db

D.29 spsquare



(a) lfcc

(b) mfcc

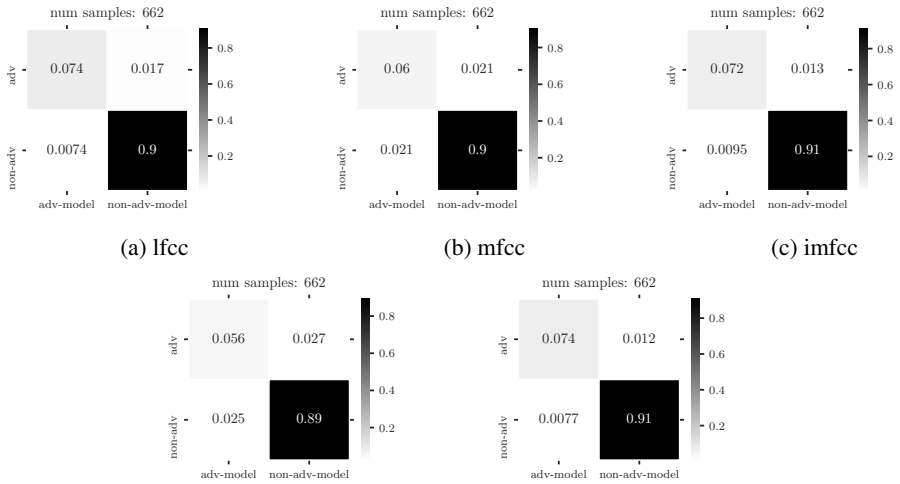
(c) imfcc

(d) gfcc

(e) igfcc

Figur 299: Confusion Matrix for spsquare snr 10db

D.30 spsquare

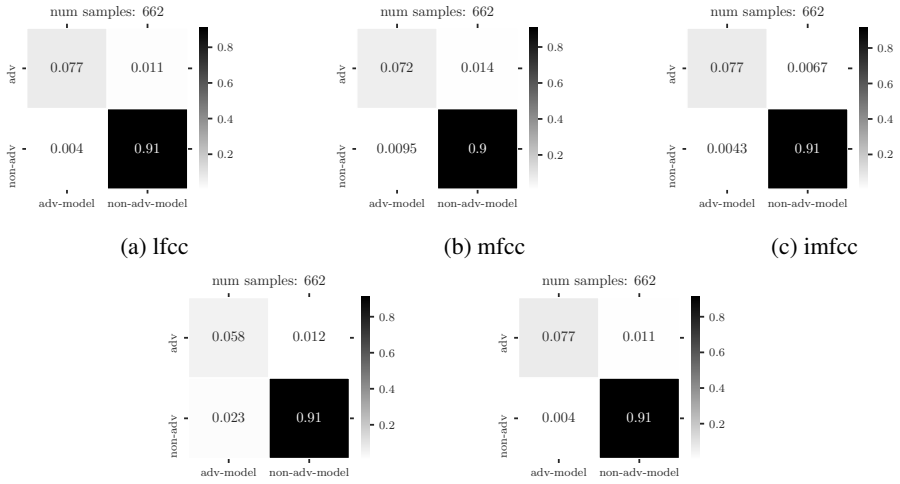


(a) lfcc (b) mfcc (c) imfcc

(d) gfcc (e) igfcc

Figur 300: Confusion Matrix for spsquare snr 15db

D.31 spsquare

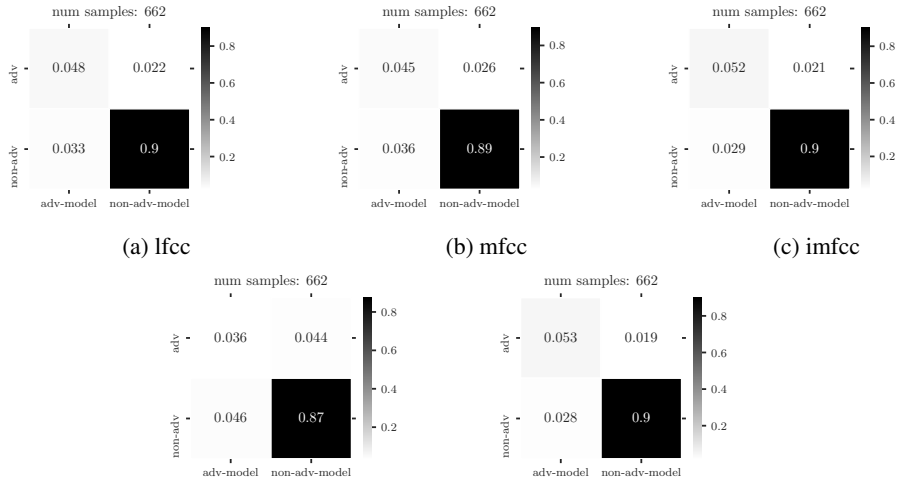


(a) lfcc (b) mfcc (c) imfcc

(d) gfcc (e) igfcc

Figur 301: Confusion Matrix for spsquare snr 20db

D.32 tbus



(a) lfcc

(b) mfcc

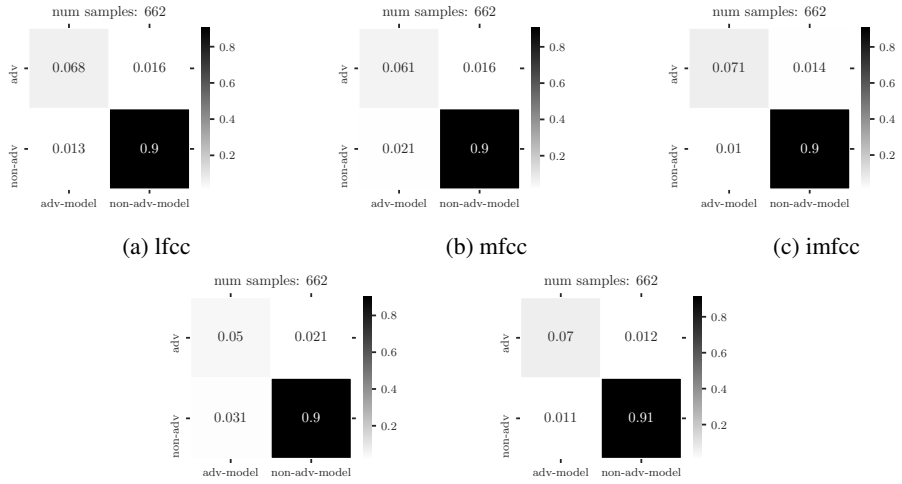
(c) imfcc

(d) gfcc

(e) igfcc

Figur 302: Confusion Matrix for tbus snr 0db

D.33 tbus



(a) lfcc

(b) mfcc

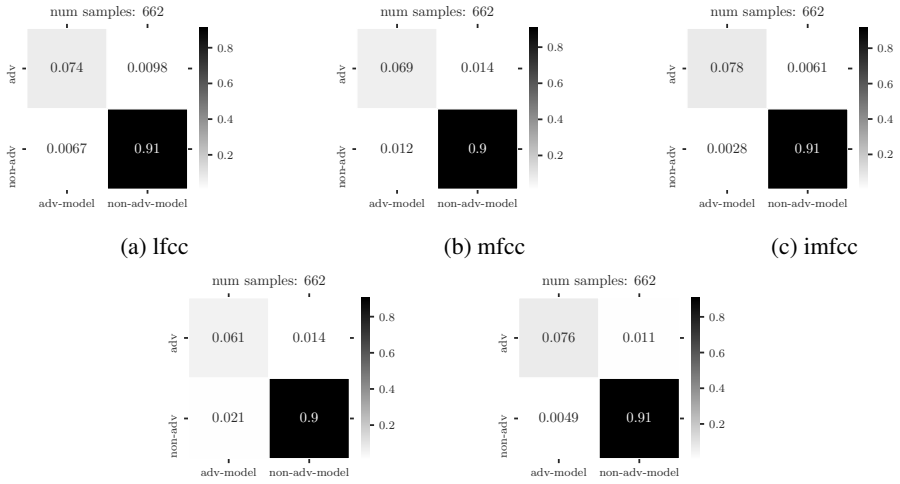
(c) imfcc

(d) gfcc

(e) igfcc

Figur 303: Confusion Matrix for tbus snr 5db

D.34 tbus



(a) lfcc

(b) mfcc

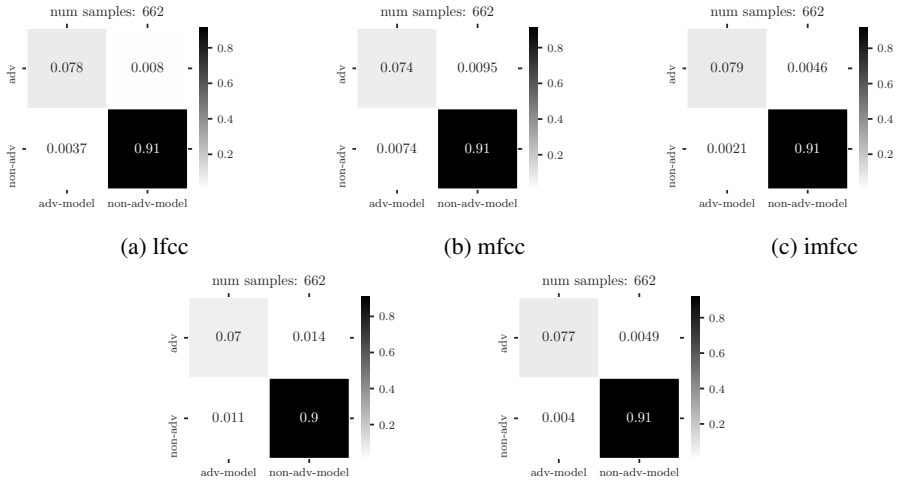
(c) imfcc

(d) gfcc

(e) igfcc

Figur 304: Confusion Matrix for tbus snr 10db

D.35 tbus



(a) lfcc

(b) mfcc

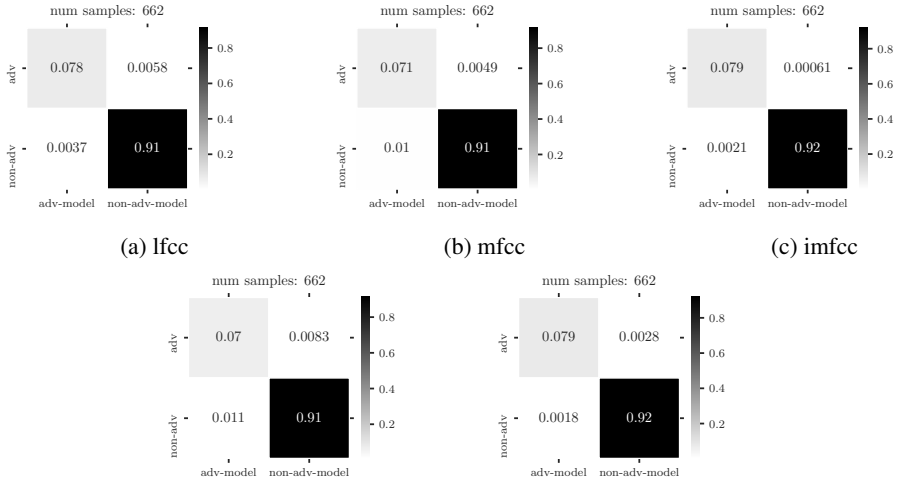
(c) imfcc

(d) gfcc

(e) igfcc

Figur 305: Confusion Matrix for tbus snr 15db

D.36 tbus



(a) lfcc

(b) mfcc

(c) imfcc

(d) gfcc

(e) igfcc

Figur 306: Confusion Matrix for tbus snr 20db