Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **492,** 1421–1431 (2020) doi:10.1093/mnras/stz3521

Deep residual detection of radio frequency interference for FAST

Zhicheng Yang,[1] Ce Yu,[1★] Jian Xiao[1★] and Bo Zhang[2,3]

[1] *College of Intelligence and Computing, Tianjin University, 135 Yaguan Road, Haihe Education Park, Tianjin, 300350, China*
[2] *National Astronomical Observatories, Chinese Academy of Sciences, 20 Datun Road, Chaoyang District, Beijing, 100012, China*
[3] *CAS Key Laboratory of FAST, National Astronomical Observatories, Chinese Academy of Sciences*

Accepted 2019 December 13. Received 2019 November 18; in original form 2019 June 18

ABSTRACT

Radio frequency interference (RFI) detection and excision are key steps in the data-processing pipeline of the Five-hundred-meter Aperture Spherical radio Telescope (FAST). Because of its high sensitivity and large data rate, FAST requires more accurate and efficient RFI flagging methods than its counterparts. In the last decades, approaches based upon artificial intelligence (AI), such as codes using convolutional neural networks (CNNs), have been proposed to identify RFI more reliably and efficiently. However, RFI flagging of FAST data with such methods has often proved to be erroneous, with further manual inspections required. In addition, network construction as well as preparation of training data sets for effective RFI flagging has imposed significant additional workloads. Therefore, rapid deployment and adjustment of AI approaches for different observations is impractical to implement with existing algorithms. To overcome such problems, we propose a model called RFI-Net. With the input of raw data without any processing, RFI-Net can detect RFI automatically, producing corresponding masks without any alteration of the original data. Experiments with RFI-Net using simulated astronomical data show that our model has outperformed existing methods in terms of both precision and recall. Besides, compared with other models, our method can obtain the same relative accuracy with fewer training data, thus reducing the effort and time required to prepare the training data set. Further, the training process of RFI-Net can be accelerated, with overfittings being minimized, compared with other CNN codes. The performance of RFI-Net has also been evaluated with observing data obtained by FAST and the Bleien Observatory. Our results demonstrate the ability of RFI-Net to accurately identify RFI with fine-grained, high-precision masks that required no further modification.

Key words: methods: data analysis – methods: observational – techniques: image processing.

1 INTRODUCTION

Any undesired signal received by radio telescopes can be referred to as radio frequency interference (RFI; see Mosiane et al. 2017). The spectral coverage of radio astronomical observations often overlaps with radio emissions originating from modern civilization. These sources of RFI negatively affect data analysis. They can be roughly classified into sporadic emissions, which may erupt occasionally with pulse-like structures, and persisting emissions (e.g. Offringa et al. 2010a; An et al. 2017). Construction equipment with electric motors, digital cameras and other similar electronic devices all generate RFI of the former type, usually contaminating multiple channels in the frequency domain (wide band). However, television, mobile phone infrastructure and radio stations usually operate in designated bands. Their emissions, along with the harmonics of their local oscillator frequencies, can give rise to persisting narrow-band RFI. All RFI can either originate from devices directly related to radio telescopes themselves (including on-site electronics, network systems, data-processing computers, etc.) or come from mobile or fixed sources outside the observatory. Besides, natural radio emitters such as the ground, lightning and the Sun can also give off extra RFI (Indermuehle et al. 2016).

As can be seen in Fig. 1, typical RFI can be much brighter than background noise or astronomical signals. Therefore, radio observations with a wide band coverage are easily affected by interference caused by human activities. Some weaker RFI may not be easily distinguished from celestial sources. Also, as occasionally seen in the observed data of FAST, RFI can also occur in the form of fluctuations in baselines, appearing randomly at any time or frequency channel, as shown in Fig. 2. It should be noted that with higher amplitudes (although much lower than typical RFI), such fluctuations should not be mistaken for standing waves (Briggs et al. 1997) with quasi-stable periodic structures in the frequency

[★] E-mail: yuce@tju.edu.cn (CY); xiaojian@tju.edu.cn (JX)

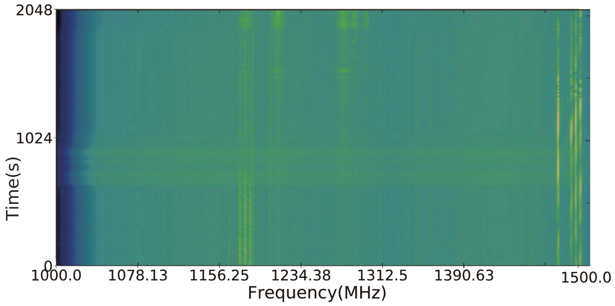


Figure 1. Data acquired by FAST shown in the time–frequency plane. Bright stripes mark the distribution of RFI.

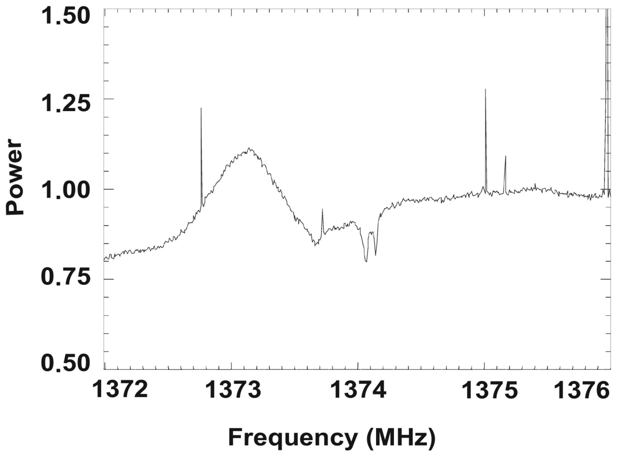


Figure 2. Example of sporadic baseline fluctuations in FAST data. It can be seen that a wide bulge exists at ∼1373.2 MHz. The amplitude of the bulge is obviously higher than the background continuum, although still weaker than common RFI. Also, an extragalactic H I absorption line with a double-horned structure can be seen at ∼1374.15 MHz. The readings along the y-axis are shown in normalized instrumental units without calibration.

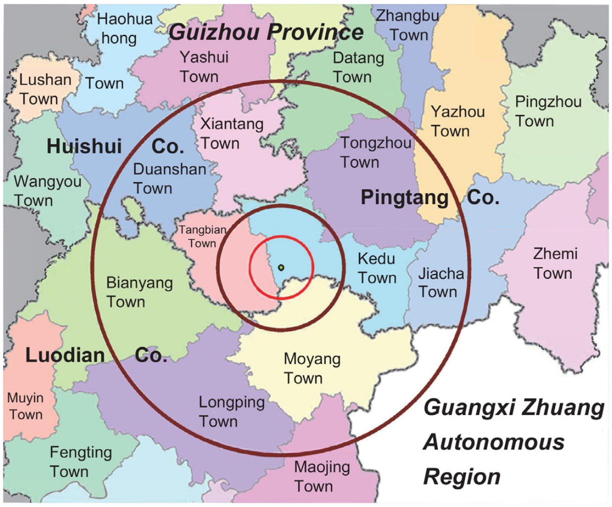


Figure 3. The RQZ established by the Guizhou Provincial People's Government. With the FAST site at its centre, the RQZ can be divided into a core area ($r \leq 5$ km), an intermediate area ($r = 5-10$ km) and a remote area ($r = 10-30$ km), according to distance r to the telescope. In the core area of this RQZ, it is forbidden to set up or use any radio stations, or to build and operate facilities that emit radio waves. In the intermediate area, it is prohibited to set up or use radio stations with working frequencies between 68 and 3000 MHz and with effective radiation power higher than 100 W. In the remote area, electromagnetic compatibility analysis should be conducted when setting up radio stations, or utilizing facilities with working frequencies between 68 and 3000 MHz and with effective radiation power higher than 100 W. (Data source: Guizhou Provincial People's Government 2019).

domain (Popping & Braun 2008). Generally speaking, random fluctuations observed by FAST usually exhibit widths of the same order of magnitude as extragalactic H I lines, although they may show broad single-peaked structures, in contrast with the double-horned structures of the H I line. Therefore, although such random fluctuations cannot be simply classified as RFI, we still need to mitigate the effects of such weak signals in radio astronomical data.

One of the most ideal and effective approaches to minimize the effect of RFI is to mitigate it by establishing a radio quiet zone (RQZ) surrounding a telescope, in order to regulate RFI-emitting device operations within this area (An et al. 2017). For example, the Australian Communications and Media Authority (ACMA) has arranged a RQZ in Western Australia (Wilson et al. 2016) for the safe operation of radio astronomical instruments, including the low-frequency facility of the planned Square Kilometre Array (SKA). Similarly, with a government order called 'Regulations for Protection of Electromagnetic Wave Quiet Zone', Guizhou Province in China has also established a RQZ for FAST (see Zhang et al. 2013; Guizhou Provincial People's Government 2019; see also Fig. 3), protecting its observing band from 70 to 3000 MHz (Nan et al. 2011).

Nevertheless, because emissions from sources such as artificial satellites cannot be minimized by ground-based RQZs, and strong signals originating outside such areas can also be detected by radio telescopes, RFI detection still poses a challenge to radio observations, and the ability to detect RFI is an important issue for radio astronomical data reduction. The correctness and completeness of RFI flagging operations greatly affects the scientific output of each telescope. Moreover, the increase of human activities has boosted the complexity and occurrences of RFI, and complicated the task of RFI detection. Therefore, in this study, our aim is to develop a suitable model to process the data acquired with FAST. Compared with other telescopes, such as Arecibo or Effelsberg, FAST has a superior sensitivity, thus rendering itself extremely vulnerable to RFI. As a result, FAST needs a more accurate RFI detection algorithm. Considering this telescope's various scientific goals, such as neutral hydrogen sky surveys and pulsar observations, the requirements of rapid adjustment and deployment should also be considered.

Traditional RFI detection methods are mainly based on threshold algorithms (Offringa et al. 2010b; Baan, Fridman & Millenaar 2004), as well as on the physical characteristics of the RFI in the time–frequency domain (e.g. the linear detection; see Wolfaardt 2016). Related image-processing techniques are subsequently applied to improve the appearance at the edges of RFI detection. However, in actual applications, manual interventions are often required to refine or confirm the locations and ranges of interference, as well as to specify related parameters of the algorithm, thus greatly reducing the processing efficiencies.

Currently, artificial intelligence (AI) technology represented by deep learning techniques has been used to detect RFI. Burd et al. (2018) have applied recurrent neural networks (RNNs) to flag RFI, while Akeret et al. (2017b) have performed similar tasks with convolutional neural networks (CNNs). Czech, Mishra &

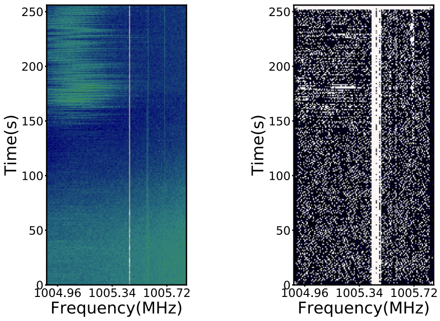


Figure 4. Detection result of U-Net (Akeret et al. 2017b) with observed data of FAST. The figure on the left shows the original data, and the figure on the right the detection result. It can be seen that the result looks like random noise. A large amount of RFI at the upper-left corner and the bottom of the figure is not correctly detected, while too many false RFI signals are detected along the time axis (shown as vertical stripes).

Inggs (2018) combined RNNs and CNNs to classify transient RFI sources. The application of AI technology has greatly reduced the workloads of astronomers, thus increasing the efficiency of data reduction. Yet, compared with traditional methods, the detection accuracies of existing deep learning algorithms still show very little advantage. Such methods usually have relatively low robustness, and are thus unable to identify complicated RFI effectively. For example, it can be clearly seen that the data shown in Fig. 4 require further processing. However, blindly improving the accuracy often leads to problems of overfitting. In addition, AI approaches could also be time-consuming, and considerable efforts are required to prepare a large amount of training data to train the neural network. Thus, efficient and fast adaptation would be impractical for such algorithms.

Inspired by these pioneering works, in this paper, we propose a new model that can improve the detection robustness without introducing artificial artefacts. Two types of residual learning (He et al. 2016b; Arsalan et al. 2019) units have been adopted for down-sampling and up-sampling processes in our CNN-based model, thus improving the accuracy without the requirements of large amounts of data. In addition, our model obviates the need for pre-treatments or further polishing, thereby increasing the efficiency and reliability of the process.

Thus, the work presented in this paper can be summarized as follows.

(i) A network architecture called RFI-Net provides higher accuracy and minimal false-positives for RFI detection.

(ii) Two types of residual units are utilized, which can lead to equally or more accurate detections with fewer training data.

(iii) A standalone method is proposed without the need for additional operations to achieve high efficiency and reliability.

In Section 2, we provide a brief overview and analysis of related work. In Section 3, we present the proposed RFI-Net with details of its features. In Section 4, we introduce the data set and framework for all experiments carried out in this study. In Section 5, we present the experiments and results. We conclude in Section 6.

2 RFI DETECTION METHODS

Generally speaking, RFI detection algorithms search for possible interference with specific signatures in the observed data, and produce RFI masks marking the positions of detected interference. Currently, traditional ways of RFI flagging include linear algorithms, such as SVD (Offringa et al. 2010b) and PCA (Wolfaardt 2016), and threshold-based methods, such as SumThreshold (Offringa et al. 2010b) and CUMSUM (Baan et al. 2004). Also, with the development of AI in image recognition (Gómez-Ríos et al. 2019) and natural language processing (Evans et al. 2019), AI-related algorithms have been invoked by various branches in astronomy, from classifications of variable stars with enhanced performance in light-curve classification benchmarks (Aguirre, Pichara & Becker 2018), to pulsar candidate identifications (Zhu et al. 2014). Also, in the radio band, efforts have been made to apply CNNs (Akeret et al. 2017b) and RNNs (Czech et al. 2018) to RFI detection.

2.1 Classical methods: SumThreshold as an example

Because the signal strength of RFI is usually much stronger than that of typical astronomical signals, classical algorithms are based on the physical characteristics of RFI. One of the notable approaches is SumThreshold, which is one of the most widely used algorithms (Akeret et al. 2017b). Introduced by Offringa et al. (2010a), the SumThreshold method has been proved to yield the highest accuracy among classical detection algorithms. SumThreshold can also be applied in combination with other algorithms, such as curvature fitting, to achieve better results (Offringa et al. 2010a).

However, it is possible that the original SumThreshold method could mistake many ‘good’ samples as RFI, if no additional rules have been applied. Taking the data set [0, 0, 5, 6, 0, 0] from Offringa et al. (2010b) as an example, the data points with values of 5 and 6 contain strong interference. As SumThreshold adopts a series of thresholds for average values of different-sized pixel combinations to identify RFI, we adopt a decreasing sequence $\chi_1 = 7$, $\chi_2 = 5$, $\chi_3 = 4$, $\cdots$, $\chi_6 = 1.8$ as thresholds of averaged values for 1, 2, 3, $\cdots$, 6 pixels, respectively, as with Offringa et al. (2010b). With an averaged value larger than $6\chi_6$, all six samples in the example data set should be marked as RFI. In order to avoid such mislabelling, it is common practice to inspect pixel combinations with increasing sizes in the data set. If a threshold χ_n for n-pixel combinations classifies a certain area as RFI, the readings of marked pixels should be replaced by χ_n, leading the final average of our example to be $\frac{2\chi_6}{6} = 0.6 < \chi_6$ (Offringa et al. 2010b). However, the exact values of each threshold also need to be finely tuned, thus increasing the need for manual interventions with this method. It is difficult to flag weak interference or unwanted baseline fluctuations with thresholding algorithms, as they can show flux levels similar to astronomical signals.

2.2 Detecting RFI with AI

RFI flagging using CNN-based models has been studied in recent years (e.g. Akeret et al. 2017b). In terms of accuracy, currently, the best variation of the CNN model should be the U-Net model described in Akeret et al. (2017b). Here, we present an overview of the CNN model, along with descriptions of the U-Net model.

Computer vision adopts the CNN as its main network scheme. The basic operation of the CNN is convolution, in which the summed value of the product from the convolutional kernel multiplies all sample values within an area covered by the kernel. Shallow convolutional operations extract textural information of images, while deep ones can integrate features obtained with shallow networks to obtain image semantics (Ronneberger, Fischer & Brox 2015). Thus, each layer of the CNN performs convolution on one or more planes, extracting information from them, and it applies pooling to reduce the volume of information. In this way, with multiple convolutions and pooling, specific information about certain areas of the image can be obtained. CNNs are suitable to make identifications in structural data, that is, data associated with spatially adjacent counterparts (e.g. images).

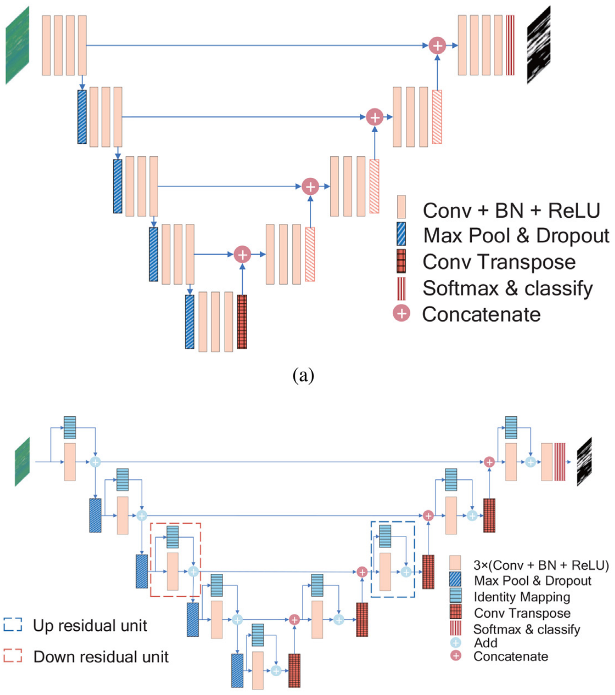
The U-Net model was originally proposed to meet the challenge as part of a workshop held prior to the IEEE International Symposium on Biomedical Imaging (ISBI) 2012 with notable results (Ronneberger et al. 2015). It can be considered as CNNs with extended architectures, which have been adjusted for RFI detection (Akeret et al. 2017b). It utilizes down- and up-sampling operations to extract required information (such as RFI) from original data, enabling the network to actually 'learn' about the extracted characteristics. In this model, shallow-layered convolutions make identifications of fine features in data (e.g. RFI intersections), while deeper networks are used to splice such features into more abstract forms (Akeret et al. 2017b). Meanwhile, features extracted by pooling operations during down-sampling are passed as copies to the right side after several steps of up-sampling, thus completing a U-shaped structure. With intensive tests, a structure with three layers and 64 feature graphs has achieved a good balance between flagging accuracy and computational cost. A structure like this can be seen in fig. 1 of Akeret et al. (2017b).

Akeret et al. (2017b) have also tested the U-Net model with data acquired by the Bleien Observatory (Cosmology Research Group from ETH Zurich 2016). The visual inspections have been compared with results from U-Net, demonstrating the advantages of the latter technique with graphs as well as index scores. It has been proven that the U-Net model has an advantage over traditional RFI flagging algorithms.

2.3 Analysis of previous methods

RFI with repetitive temporal or spectral behaviours, such as radar or radio beacon emissions, can be best identified using linear detection-based algorithms, although such methods are not suitable to detect stochastic RFI, including pulse-like signals, and RFI with frequency drifts (Akeret et al. 2017b). In contrast, thresholding algorithms are more effective if the observed background is relatively stable and the RFI is distributed discretely. Thresholding has the advantages of relatively fast execution speed, easy implementation and high efficiency and robustness (with properly adjusted parameters), so it is best to flag strong RFI. Also, algorithms incorporating the SumThreshold method are especially popular in radio astronomy (Akeret et al. 2017b). However, in the presence of weak interference or baseline fluctuations with fluxes comparable with celestial sources, or broad-band signals/extremely large amounts of RFI (which means that most channels are RFI-contaminated), thresholding methods would become less effective, if not useless.

Methods invoking machine learning/deep learning techniques could greatly reduce manual involvements, thus enhancing the automation level of RFI detections. However, in terms of practical applications, these algorithms have not achieved satisfactory detection accuracy. As shown in Fig. 4, the U-Net model proposed by Akeret et al. (2017b) has identified too many time domain structures as RFI; however, interference in the frequency domain and point-like RFI have been largely ignored. Furthermore, a lot of noise (also described as false-positive) exists in the flagging results, especially at the edges of regions identified as RFI. Apparently, such performance needs to be further refined, which may decrease the efficiency of the algorithm as a result. Moreover, training neural networks like this often requires large amounts of pre-labelled data. While RFI flagging with raw data is a time-consuming and tedious task, considering the requirements for subsequent training, it would not be easy to make adjustments to existing approaches to improve their suitability for RFI detection in different observing configurations (Aguirre et al. 2018).



(a)

(b)

Figure 5. Structure of RFI-Net. Here, Conv denotes convolutional computation, BN is batch normalization and ReLU denotes the Rectified Linear Unit, which is a non-linear activation function (Krizhevsky, Sutskever & Hinton 2012). Max Pool refers to a sample-based discretization process for down-sampling of input representation, taking the maximum of the input, reducing its dimensionality and making assumptions about features of the input data (Boureau, Ponce & LeCun 2010). Dropout is used to prevent overfitting, as well as to reduce the time spent on training (Srivastava et al. 2014). Identify Mapping makes it easier to train the network (He et al. 2016a).

3 RFI-NET FOR RFI DETECTIONS AT FAST

To overcome the shortcomings of algorithms discussed so far, we propose a new model combining U-Net with a residual network. In this model, the corresponding layers are connected with short cuts, which are shown in Fig. 5 as horizontal lines connecting the left and right sides. Two types of residual units have been designed and constructed. In addition, two other hyperparameters have been introduced to further enhance the performance.[1]

3.1 Architecture of RFI-Net

We have constructed our model with two steps. For the first step, we add more layers to U-Net, as shown in Fig. 5(a). This is because, for deep learning algorithms, the depth of a certain neural network

[1] The codes related to this work are available at GitHub https://github.com/DuFanXin/RFI-Net.

Table 1. RFI-Net structure in detail. BN means batch normalization for more stable data distributions, while transpose is sometimes called ‘deconvolution’ after deconvolutional networks, but it is in fact the transpose of the convolution. Most of the layers take ReLUs as activation functions to minimize the problem of gradient vanishing or explosion. In the last layer, SoftMax, the bracket means that if the network is in the process of training, it needs SoftMax only. Otherwise, ArgMax should also be processed to obtain the RFI flagging result.

Layer	Operation	Input size	Filter size	Output size
conv 1	Conv+BN+ReLU	256 × 128 × 1	3 × 3 × 1 × 32	256 × 128 × 32
conv 2	Conv+BN+ReLU	256 × 128 × 32	3 × 3 × 1 × 32	256 × 128 × 32
down residual unit 1	Conv+BN+ReLU Conv+BN Conv+BN+ReLU Conv+BN	256 × 128 × 32		256 × 128 × 64
max pool 1	Max pooling+Dropout	256 × 128 × 64	1 × 2 × 2 × 1	128 × 64 × 64
down residual unit 2	Conv+BN+ReLU Conv+BN Conv+BN+ReLU Conv+BN	128 × 64 × 64	–	128 × 64 × 128
max pool 2	Max pooling+Dropout	128 × 64 × 128	1 × 2 × 2 × 1	64 × 32 × 128
down residual unit 3	Conv+BN+ReLU Conv+BN Conv+BN+ReLU Conv+BN	64 × 32 × 128	–	64 × 32 × 256
max pool 3	Max pooling+Dropout	64 × 32 × 256	1 × 2 × 2 × 1	32 × 16 × 256
down residual unit 4	Conv+BN+ReLU Conv+BN Conv+BN+ReLU Conv+BN	32 × 16 × 256	–	32 × 16 × 512
max pool 4	Max pooling+Dropout	32 × 16 × 512	1 × 2 × 2 × 1	16 × 8 × 512
down residual unit 5	Conv+BN+ReLU Conv+BN Conv+BN+ReLU Conv+BN	16 × 8 × 512	–	16 × 8 × 1024
up sample 1	Transpose+BN+ReLU	16 × 8 × 1024	1 × 2 × 2 × 1	32 × 16 × 512
concat 1	Concatenate	2 × (32 × 16 × 512)	–	32 × 16 × 1024
up residual unit 1	Conv+BN+ReLU Conv+BN Conv+BN+ReLU Conv+BN	32 × 16 × 1024	–	32 × 16 × 512
up sample 2	Transpose+BN+ReLU	32 × 16 × 512	1 × 2 × 2 × 1	32 × 16 × 256
concat 2	Concatenate	2 × (64 × 32 × 256)	–	64 × 32 × 512
up residual unit 2	Conv+BN+ReLU Conv+BN Conv+BN+ReLU Conv+BN	64 × 32 × 512	–	64 × 32 × 256
up sample 3	Transpose+BN+ReLU	64 × 32 × 256	1 × 2 × 2 × 1	128 × 64 × 128
concat 3	Concatenate	2 × (128 × 64 × 128)	–	128 × 64 × 256
up residual unit 3	Conv+BN+ReLU Conv+BN Conv+BN+ReLU Conv+BN	128 × 64 × 256	–	128 × 64 × 128
up sample 4	Transpose+BN+ReLU	128 × 64 × 128	1 × 2 × 2 × 1	256 × 128 × 64
concat 4	Concatenate	2 × (256 × 128 × 64)	–	256 × 128 × 128
up residual unit 4	Conv+BN+ReLU Conv+BN Conv+BN+ReLU Conv+BN	256 × 128 × 128	–	256 × 128 × 64
conv 2	Conv+BN	256 × 128 × 64	3 × 3 × 64 × 2	256 × 128 × 2
softmax	SoftMax(+ ArgMax)	256 × 128 × 2	–	256 × 128 × 1

can be of great importance. Regardless of filter sizes or chosen widths, a deeper network can usually achieve more optimal results, compared with shallower networks with roughly the same temporal complexity (He & Sun 2015), because an increase in depth can lead to more information being extracted. Furthermore, compared with a shallower network, a deeper network can significantly enhance the capability to perform tasks that are more computationally demanding. For the second step of our model construction, the residual units have been built into the network, as shown in Fig. 5(b). Identity mappings serve as short cuts to connect three convolutional layers. Two types of residual units have been designed in this work. The unit marked with orange dotted lines in Fig. 5(b) is used for down-sampling, thereby doubling the number of channels, whereas the unit indicated by blue dotted lines halves the number of channels for up-sampling. To support this structure, the network adopts batch normalization to normalize the input data, and it is equipped with a fast and stable optimizer. Detailed descriptions of the network structure are provided in Table 1.

3.2 Residual units

The residual units have been added to prevent network degeneration. When a deep network no longer diverges despite implementations of various optimizations, it will begin to degrade: with an increasing network depth, the detection accuracy would finally meet a ‘bottleneck’, with rapidly growing training errors afterwards. It should be noted that such errors introduced by network degradation are

Table 2. Detailed hyperparameters of the residual unit for down-sampling. Here, H represents the height of the data, W is the data width and C is the number of channels. The input data of 'short cut' and 'conv 1' layers are identical. The layer 'add' has two sources of inputs: layers 'conv 1' and 'short cut'.

Layer	Operation	Input size	Filter size	Output size
conv 1	Conv+BN+ReLU	$H \times W \times C$	$3 \times 3 \times C \times 2C$	$H \times W \times 2C$
conv 2	Conv+BN+ReLU	$H \times W \times 2C$	$3 \times 3 \times 2C \times 2C$	$H \times W \times 2C$
conv 3	Conv+BN	$H \times W \times 2C$	$3 \times 3 \times 2C \times 2C$	$H \times W \times 2C$
short cut	Conv+BN	$H \times W \times C$ (same as conv 1 input)	$1 \times 1 \times C \times 2C$	$H \times W \times 2C$
add	Add+BN+ReLU	$H \times W \times 2C$ (from conv1 result) $+H \times W \times 2C$ (from short cut result)	–	$H \times W \times 2C$

Table 3. Detailed hyperparameters of the residual unit for up-sampling. Here, H represents the height of the data, W is the data width and C is the number of channels. This unit halves the channel number of the data.

Layer	Operation	Input size	Filter size	Output size
conv 1	Conv+BN+ReLU	$H \times W \times C$	$3 \times 3 \times C \times C/2$	$H \times W \times C/2$
conv 2	Conv+BN+ReLU	$H \times W \times C/2$	$3 \times 3 \times C/2 \times C/2$	$H \times W \times C/2$
conv 3	Conv+BN	$H \times W \times C/2$	$3 \times 3 \times C/2 \times C/2$	$H \times W \times C/2$
short cut	Conv+BN	$H \times W \times C$ (same as conv 1 input)	$1 \times 1 \times C \times C/2$	$H \times W \times C/2$
add	Add+BN+ReLU	$H \times W \times C/2$ (from conv1 result) $+H \times W \times C/2$ (from short cut result)	–	$H \times W \times C/2$

a result of more biased calculations caused by a larger number of layers, rather than overfitting (He et al. 2016b).

Therefore, inspired by the residual network (He et al. 2016b), we introduce short cuts to our model (represented by the dotted boxes in Fig. 5),

$$z = F(x) \tag{1}$$

where x and z denote the input and output, respectively. Identity mapping in most residual networks can be represented as follows:

$$z = F(x) + x. \tag{2}$$

Equation (2) expresses that the input should be directly added to the result of the convolutional operations. However, since identity mapping does not have the flexibility to resize, especially with respect to channel numbers, RFI-Net chooses to perform convolutional operations with a kernel size of 1 × 1. With the assistance of batch normalization, the short cut can be expressed as

$$z = F(x) + H(x), \tag{3}$$

where $H(x)$ denotes the combination of 1 × 1 convolution, as well as batch normalization. Our network design adopts two units corresponding to the processes of down- and up-sampling to make adjustment of channel numbers possible. Details of the unit hyperparameters are presented in Tables 2 and 3.

By connecting three layers as a unit, the short cuts can stabilize the update process, and slow down the gradient disappearance resulting from the inability of the model's middle layers to update the parameters effectively. Moreover, it is expected that models with short cuts should not only see improvements of their performance, but also achieve a faster convergence speed (Drozdzal et al. 2016). In addition, the long connecting structures used in the U-Net model could give rise to a slower learning rate with unstable parameter updates (Drozdzal et al. 2016). In contrast, a network equipped with short cuts makes a larger initial learning rate possible, thus accomplishing a faster convergence.

3.3 Additional hyperparameters

Batch normalization, which is generally referred to as batch-norm, is achieved by using an algorithm to normalize batches of data with adjustments of averaged data to 0, and the variance to 1. Let m be the size of input sample $X = (x_1, \cdots, x_m)$. The normalized data should be calculated as

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i, \tag{4}$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m} x_i^2 \longrightarrow \sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2 \tag{5}$$

and

$$\overline{x_i} = \frac{x_i - \mu}{\sqrt{\mu^2 + \varepsilon}} \longrightarrow \overline{x_i} = \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}}. \tag{6}$$

Equation (4) calculates the mean value of the sample, with equation (5) providing the sample variance. Equation (6) subtracts the mean value from each data point in the sample, dividing the results with the variance, thus completing the normalization process. The bias ε is introduced in equation (6) to prevent the occurrence of zero in the divisor. Without batch normalization, the neural network needs to constantly update the parameters with continuous back-propagation to obtain results from each layer. As the distribution of such results could change continually (Ioffe & Szegedy 2015), time-to-time re-adjustments of parameters from the previous layer are required to accommodate new distributions, thus preventing the model from achieving a faster learning rate during training. The application of batch normalization ensures that the resulting distribution of each layer can be nearly constant with time, thus avoiding the need to re-adjust the previous layer. However, because each layer is normalized, the absolute values of the results are small, with results from the front layers differing insignificantly from those of the back layers, which can lead the network model to 'learn' features ineffectively (Ioffe & Szegedy 2015). Thus, a controllable adjustment method to appropriately scale the normalized results is proposed:

$$y_i = \gamma \overline{x_i} + \beta. \tag{7}$$

In equation (7), γ has the task of scaling sizes of the normalized results, whereas β shifts the result. Both of these parameters can make self-learning possible. Scaling is used to restore the result to its original value at least (Ioffe & Szegedy 2015). This approach based on batch normalization enables the model's learning rate to be accelerated without the need to impose strict parameter initialization, thus promoting faster convergence.

The optimizer strategy we adopted here to set the mini-batch size in the hyperparameter is called the small batch gradient descent, which is used in combination with the Adam algorithm (Kingma & Ba 2015) as the computing strategy for gradient descent. This strategy can speed up the gradient descent, and minimize oscillation of the gradient.

The small batch gradient descent is based on the stochastic gradient descent method (Ruder 2016), which adopts a sample size between that of the batch and the random gradient drop. This method selects samples from data to calculate the gradient, and then performs the gradient drop. This approach not only enables the gradient to decrease at a faster rate, but also ensures that the gradient remains relatively stable during gradient descent.

The Adam algorithm (Kingma & Ba 2015), which combines Momentum (Lv, Jiang & Li 2017) with RMSprop (Ruder 2016), uses both first- and second-order moments, and performs a similar standardized operation on the gradient. The main advantage of the Adam algorithm is that, after offset correction, the learning rate of each iterative cycle can fall within a certain range, thus stabilizing the parameters. This algorithm can also calculate different adaptive learning rates for different parameters. Experiences have shown that the Adam algorithm can perform well in practice, and is more computationally efficient compared with other adaptive learning algorithms (Ruder 2016).

4 BASIC FRAMEWORK OF DATA AND EXPERIMENTS

We now describe our experiments. In Section 4.1, we discuss the data used for training and evaluation, and we descibe the experimental framework in Section 4.2. We compare the models employed in the experiments, discuss the software and hardware used in the experiments and present an evaluation of the model performance.

4.1 Data used in the experiments

We adopted an astronomical simulator to simulate data captured by a radio telescope. FAST can be used to conduct observations of neutral hydrogen (H I) and the 21-cm line from the hyperfine transition of neutral hydrogen emitted at a rest frequency of ∼1420.4 MHz. Many software packages are available for H I simulation and data processing. The simulator we used for this study is the HIDE (the H I data emulator), which was also used in the studies of Akeret et al. (2017b) to simulate the training data set.

The simulated data are comprised of astronomical data and RFI. Because HIDE can produce both 'pure' RFI and simulated astronomical data (which already contain RFI), as shown in the top-left panel of Fig. 6, it is possible to label all the interference precisely as fundamental references (i.e. the ground_truth) for our experiments. By comparing RFI detected by various algorithms with the ground_truth, the accuracy of each method can be evaluated. The simulated RFI is displayed in the top-centre panel of Fig. 6, with a corresponding RFI mask shown in the top-right panel.

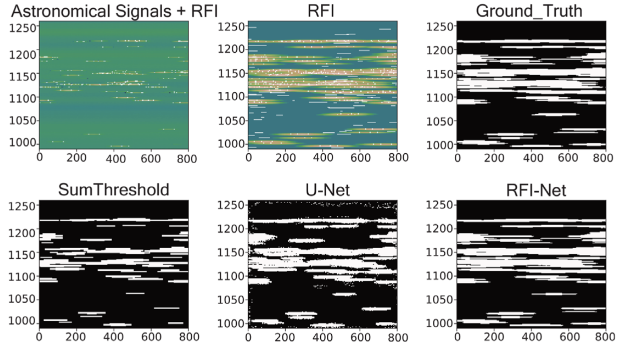


Figure 6. Simulated data with identified RFI masks obtained using various algorithms. From left to right, the top images present the simulated astronomical data, which contain signals and RFI, RFI and the corresponding RFI mask served as ground_truth, in which white denotes RFI and black is for 'good' data, as generated by HIDE (Akeret et al. 2017a). The bottom images show the results from different methods. It can be seen that the one obtained with our method resembles most closely the ground_truth. The result of U-Net is not so clean, while that of SumThreshold fails to detect some of the RFI.

Experiments have also been conducted with manually labelled observed data sets captured by FAST in 2018 September, as well as open access data (Cosmology Research Group from ETH Zurich 2016) acquired by the Bleien Observatory on 2016 March 21 (Akeret et al. 2017a).

4.2 Experimental framework

For the purpose of comparison, our data set has been processed by several existing RFI flagging methods, including U-Net, KNN (K-nearest neighbour; Guo et al. 2003), one of the classification algorithms used in machine learning, as well as SumThreshold. Similar to Akeret et al. (2017b), we have found that U-Net with three layers and 64 characters can achieve a good balance between accuracy and speed, after several cycles of tests. Thus, in this study, we have adopted the same structure and hyperparameters for RFI-Net as Akeret et al. (2017b). In addition, we applied Scikit-learn (Pedregosa et al. 2011), the machine-learning library for Python, to assist with the implementation of the KNN algorithm.

Specifically, the following experiments have been conducted.

(i) We compared the results obtained from RFI-Net for simulated data with those from previous methods.

(ii) A detailed analysis of algorithm accuracies was carried out.

(iii) The results obtained from RFI-Net for observed data were studied to determine whether additional operations were required.

(iv) We used RFI-Net to process fewer training data (i.e. 25, 50 and 75 per cent of the complete data) to validate its ability to achieve a comparable performance on smaller data sets.

(v) The ability of RFI-Net to overcome the overfitting problem was demonstrated.

(vi) The training speed was investigated

All these methods have been tested under Ubuntu 16.04 on a Dell PowerEdge T630 with 32 GB RAM. The deep learning models have been executed on a NVIDIA Tesla K40c with 12 GB RAM.

The indicators we adopted for experimental evaluations are precision, recall and the F1 score (see Akeret et al. 2017b; Davis & Goadrich 2006).

Precision, which is the fraction of genuine RFI among all flagged instances, can be considered as the accuracy metric of the following algorithm:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}. \quad (8)$$

Recall indicates the fraction of RFI that has been identified among all RFI, showing the comprehensiveness of the detection:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}. \quad (9)$$

F1, the reciprocal mean of precision and recall, can be considered as the overall model performance :

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (10)$$

It should be noted that, as our observed data were labelled manually, with no guarantee of completeness of RFI flags, the related performances cannot be evaluated exactly with the indicators listed above. They rely on visual inspections, and can only provide a general approximation of algorithm characteristics.

5 RESULTS OF EXPERIMENTS

Here, we compare the visual results obtained by our model with those of previous methods in Section 5.1. This is followed by Section 5.2, which proves the advantages of using detailed indicators. The test results with real observing data are presented in Section 5.3 and the experimental results obtained with smaller training data sets are listed in Section 5.4. Section 5.5 demonstrates the effects of the overfitting problem, while the ability of RFI-Net to accelerate the training process is presented in Section 5.6.

5.1 Comparison of RFI-Net with previous methods

The results of our RFI-Net model, the U-Net model and SumThreshold are shown in Fig. 6, with the identified RFI of each method listed as RFI masks. It can be clearly seen that the result of the model proposed in this paper shows the greatest resemblance to the ground_truth of our simulated data. The RFI-Net model correctly labelled most RFI, with only a small amount of data misidentified. A few flaws in our result only show up in certain details (such as at the edges of some RFI regions). By comparison, although the detection result of the SumThreshold algorithm seems to be relatively clean, with fewer cases of false RFI detections, some of the RFI is not correctly detected when compared with the ground_truth, resulting in a lower recall rate. For the U-Net model, too many false-positives have been identified, despite a large amount of RFI remaining undetected. In addition, U-Net erroneously labels a lot of 'good' pixels as RFI, which means its detection result is less 'clean' than our model.

5.2 Comparison of accuracy

As illustrated in Section 4.2, the performances of various algorithms can be measured with scores based on parameter precision, recall and F1 score. Table 4 lists the results of our RFI-Net model, the U-Net model, the KNN and SumThreshold with simulated data, indicating that RFI-Net has outperformed two other methods, with the ability to detect RFI more accurately and comprehensively. Although the result obtained with SumThreshold is as accurate as ours in terms of precision, it failed to detect RFI consistently. Although the performance of the CNN-based U-Net method shows a more balanced performance for all parameters, the detection quality of its results cannot be compared with those of RFI-Net. However, KNN shows the worst performance in terms of all three parameters. In general, these results show that the application of AI for RFI detection tasks is promising, and our model has a better performance than previous algorithms.

Table 4. Detailed scores of algorithm performances obtained with simulated data set. The precision represents accuracy, recall shows detecting comprehensiveness, while the F1 score is the harmonic mean of these two indicators. The highest scores for each indicator are shown in bold font.

	Precision	Recall	F1 score
RFI-Net	**97.93**	**95.43**	**96.67**
SumThreshold	97.20	58.64	73.15
KNN	76.83	68.77	72.57
U-Net	80.87	81.30	81.04

Table 5. Detailed scores of performances tested with FAST's observations. The best scores of each indicator are shown in bold font.

	Precision	Recall	F1 score
RFI-Net	**97.34**	89.12	**93.05**
SumThreshold	91.20	59.63	72.11
KNN	87.69	**94.05**	90.75
U-Net	95.11	88.19	91.52

Table 6. Detailed performances obtained using data from the Bleien Observatory. The highest scores are shown in bold font.

	Precision	Recall	F1 score
RFI-Net	**95.61**	91.07	**93.28**
SumThreshold	87.26	58.64	70.14
KNN	70.58	**97.60**	81.36
U-Net	94.08	89.14	91.54

In Table 5, RFI-Net demonstrates the ability to detect RFI more accurately in real data. Although here the RFI is more complex than in the simulated data, RFI-Net has still done the best job.

As shown in Table 6, RFI-Net has achieved the highest precision and F1 score when tested with data from the Bleien Observatory. KNN scored the highest accuracy, 97.60, along with a much lower comprehensiveness. In contrast, SumThreshold failed to achieve good results, and ranked the last with worst recall and F1 score. U-Net has performed well, with all three indicators scoring about 90. Nevertheless, these scores imply that there is still room to further improve the performance of the RFI-Net model.

5.3 Dependence on extra processing

The suitability of RFI-Net for practical applications has been evaluated with experiments on real data from FAST, as well as open access data from the Bleien Observatory. Fig. 7 shows the results obtained with the FAST data set. Most of the RFI has been correctly identified with fine detection granularities, although very few instances have been missed. Results like this can be used as RFI masks without further processing. In addition, in Fig. 8,

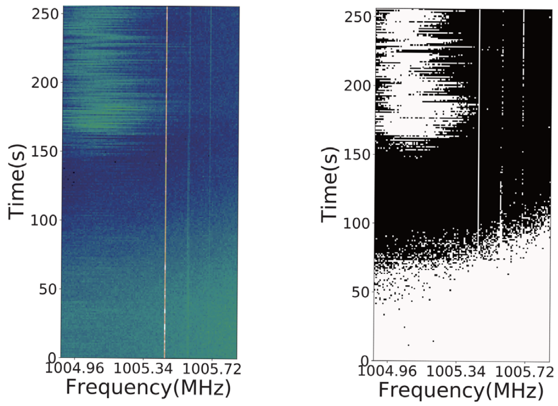


Figure 7. Left: result obtained with FAST observations, which is the same as Fig. 4. Most RFI in frequency and time domains, as well as point-like disturbances, can be detected. However, some of the indistinct RFI has been missed. Right: RFI flagging mask of FAST observations. Conspicuous RFI has been successfully identified, whereas some indistinct RFI remains undetected.

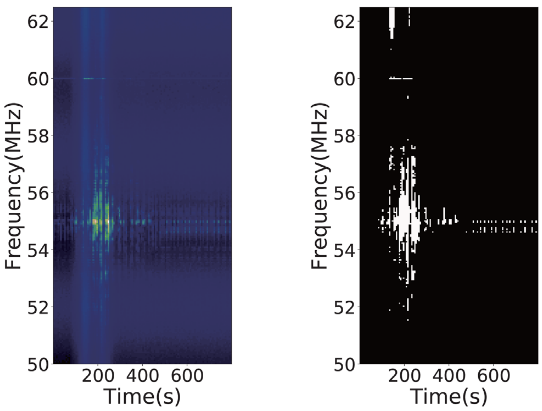


Figure 8. Detection result with data from the Bleien Observatory. Most of the RFI has been identified, while there are false-positives and missing marks on the edge of the detected areas.

RFI-Net also shows a competitive performance. All evident RFI has been successfully flagged, although some insignificant RFI with flux levels similar to the background has missed detection.

Compared with observed data, our simulated data are simpler and have been more accurately marked, so the accuracy of detection obtained with simulated data is higher than with real data. Thus, labelling observations more precisely would lead to further improvements in our results.

5.4 Accuracy with a smaller training data set

For a smaller training data set, Figs 9, 10 and 11 compare the results of our model with the original U-Net model using all three performance metrics. These results confirm the outstanding performance of RFI-Net. It can be seen from Fig. 9 that despite the volume of training data, the recalls of RFI-Net are higher than those of U-Net. When using all of the training data, the recall of RFI-Net is 97.5 per cent, compared to 85 per cent for U-Net. While for 25 per cent of the data set, although the difference between two models is smaller, the recall of RFI-Net (89 per cent) is still higher than that of U-Net (82 per cent). This indicates that RFI-Net can make more comprehensive detections, with very few instances of RFI remaining unidentified. From the perspective of data volumes, a smaller amount of training data does affect the performance of the model in terms of completeness. The recall of RFI-Net has shown a significant decrease with a 75–100 per cent training data set, in contrast to a more gradual decline of the original U-Net. However, with less than 75 per cent of the original data, the amount of data has less effect on the recall of our model.

Fig. 10 shows the precision of the detections with different training data volumes. It can be seen that the amount of training data had a more significant effect on the original U-Net than RFI-Net this time. The accuracy of U-Net experienced a decline when the data volume was reduced to 75 per cent, but then it was restored

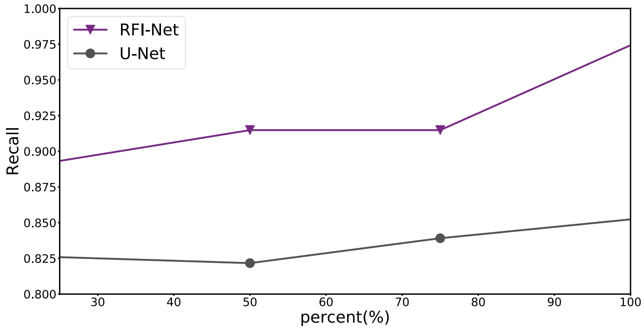


Figure 9. Recall of two models trained with different portions of the complete training data set. It can be seen that with a different sized data set, the scores of RFI-Net are always higher than those of U-Net, yielding more stable results.

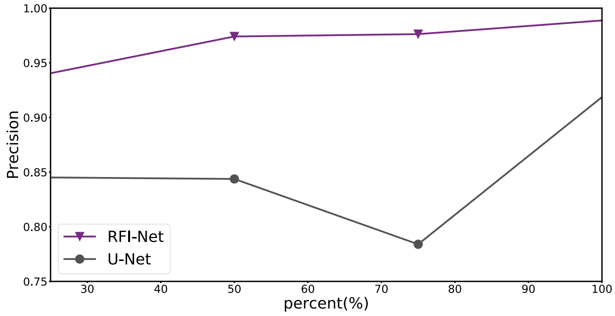


Figure 10. Precision scores of the two models with different portions of the complete training data set. The performance of U-Net declines as the amount of training data is reduced from 100 per cent to 75 per cent, while the RFI-Net model remains more stable.

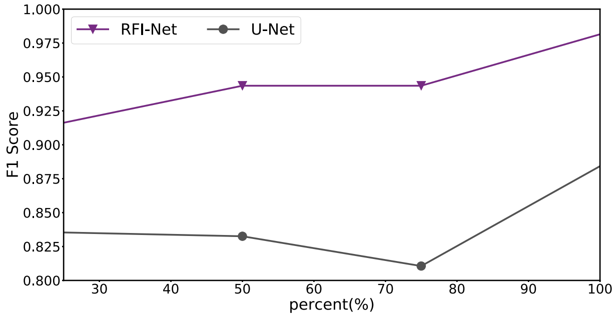


Figure 11. The overall performance of the two models, trained on different ratios of the whole training data, is judged by the F1 score. The result is similar to the precision. U-Net shows larger fluctuations, while RFI-Net is more stable, yielding higher scores.

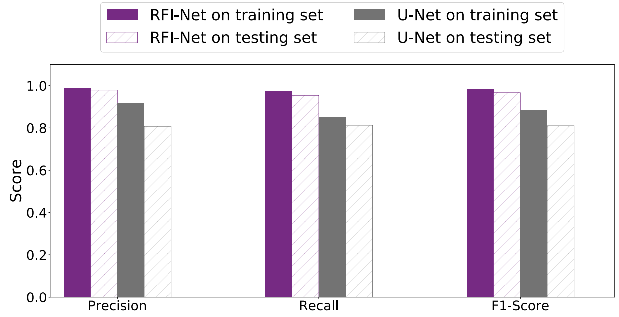


Figure 12. Results of RFI-Net regarding the overfitting problem. The difference between training and testing data sets with RFI-Net is much smaller than that of U-Net, which means overfitting can be further reduced by RFI-Net.

Table 7. Detailed scores on the overfitting problem with three performance metrics. It can be seen that our RFI-Net has achieved a more consistent performance.

	Precision		Recall		F1 score	
	RFI-Net	U-Net	RFI-Net	U-Net	RFI-Net	U-Net
Training set	98.87	91.86	97.42	85.23	98.14	88.43
Testing set	97.93	80.78	95.43	81.30	96.67	81.04

from 77 per cent to ∼ 85 per cent as the data volume continued to be reduced. The performance of RFI-Net, however, is much more stable even without enough data for adequate training. The precision of our model shows a decrease with less than 50 per cent training data, which can lead to less accurate RFI detections.

The F1 score, which is the reciprocal average (harmonic mean) of precision and recall, can be used to demonstrate the overall performance of each model, as shown in Fig. 11. Still, the volume of training data can influence the performances of both models. However, RFI-Net delivered a much more stable curve, with a gradual decrease in the F1 score. In addition, RFI-Net maintains a higher score regardless of data volume. In contrast, U-Net performed less optimally, with a more consistent decline with the size of the data set.

5.5 Effects of overfitting

The relatively lower accuracy acquired during practical applications compared with the model training process is called the overfitting problem. Fig. 12 compares the performance of RFI-Net and U-Net with our training and testing data sets, respectively. Here, the testing data come from the same source as our training data, with a volume ∼10 per cent of the latter. It can be seen that RFI-Net exhibits less degradation in the gradient when transferred from training set to testing set, whereas U-Net has degraded considerably.

The detailed test results are listed in Table 7. It is clearly shown that RFI-Net scored similarly on all indicators with both training and testing data sets. By comparison, U-Net only obtained much lower scores with our test data. However, it should be noted that such results are based upon simulated data. Although RFI-Net exhibits a better consistency with different data sets, it is possible that such consistency may degrade when dealing with more complicated observed data, and further investigations on this issue have been planned.

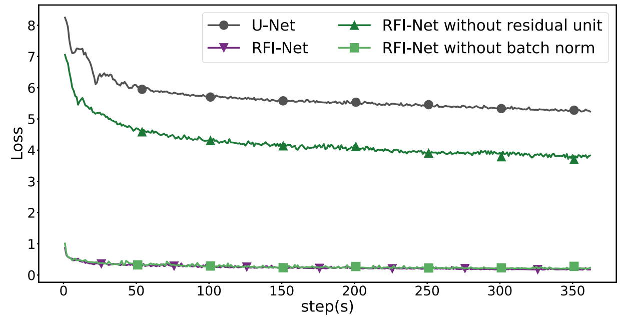


Figure 13. Loss curves produced by four configurations. The loss decreases as the number of steps increases. RFI-Net with and without batch normalization (blue and brown, respectively) are both located in the lower part of the figure, as they converge early into straight lines. The losses of U-Net and RFI-Net without residual units (green and magenta) are so large, that a log scale has been applied, in order to show the corresponding curves with residual units in the same graph.

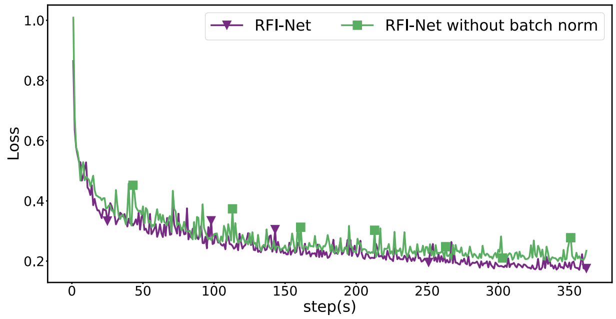


Figure 14. Zoomed-in view of the lower part of Fig. 13, showing loss curves of RFI-Net with and without batch normalization. It can be seen that the introduction of batch normalization can help to smooth and accelerate the process on a lesser (and insignificant) scale.

5.6 Training speed of RFI-Net

Fig. 13 shows the gradient descents of four configurations during training. Because the RFI-Net with residual units converges relatively fast, such models both with or without batch normalization appear at the bottom of the graph. Compared with the original U-Net, as well as RFI-Net without residual units, it can be seen that the residual units can reduce the loss significantly. In addition, such effects take place earlier than for the original U-Net, because of earlier stabilization. Thus, with their presence, the residual units can accelerate the gradient descent effectively.

Fig. 14 provides a zoomed-in version of the lower section of Fig. 13, showing the losses of RFI-Net with and without batch normalization. The application of batch normalization further improves the acceleration by three orders of magnitude. Thus, although the application of batch normalization does not introduce many differences, even with the help of residual units, it still can facilitate the training process with faster acceleration.

Fig. 15 presents the training rate in terms of accuracy, and shows that RFI-Net can stabilize itself faster and more smoothly. RFI-Net with or without batch normalization shows a similar performance, although the latter model exhibits minor fluctuations and a slight decline in accuracy. In contrast, the case of RFI-Net without residual units shows more variabilities, as well as much lower accuracy. Also, the performance of U-Net is similarly poor, with continued oscillations even when the accuracy has been stabilized.

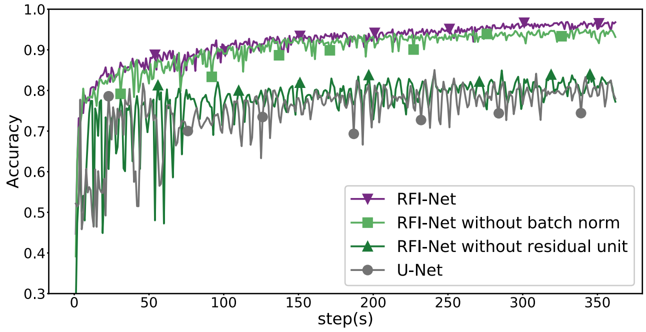


Figure 15. Accuracy of the training process. With the help of residual units, RFI-Net can reach a stable state faster with fewer disturbances. The batch normalization has a smaller contribution to model stability and accuracy.

6 CONCLUSION

Flagging RFI robustly is a challenging task. The high sensitivity and large data rate of FAST mean that it is a challenge to flag RFI and to avoid false-positives. Although several studies have attempted to solve this problem, they either exhibit low-precision detections or require a large amount of manual interventions, leading to a less-effective data reduction. Thus, they are unsuitable for FAST.

We have proposed the RFI-Net model in this paper, and have demonstrated that, compared with the U-Net, KNN and SumThreshold methods, the performance of RFI-Net is superior in terms of RFI identification, making more comprehensive detections possible with higher accuracies, fewer errors, finer edges and lower false-positives. Our test results show no need for further manual inspections, thereby improving the efficiency of the data reduction pipeline. Our evaluations with a smaller training set have demonstrated the ability of RFI-Net to maintain its performance, with time and effort required for data preparation greatly reduced, overfitting minimized and the training process accelerated.

ACKNOWLEDGEMENTS

This work is supported by the Joint Research Fund in Astronomy (U1731125, U1731243) under a cooperative agreement between the National Natural Science Foundation of China (NSFC) and the Chinese Academy of Sciences. BZ is also supported by NSFC grants 11903056 and U1531246, the Cultivation Project for FAST Scientific Payoff and Research Achievement of CAMS-CAS, as well as the Open Project Program of the Key Laboratory of FAST, NAOC, Chinese Academy of Sciences.

The authors declare no conflicts of interest.

REFERENCES

Aguirre C., Pichara K., Becker I., 2018, MNRAS, 482, 5078
Akeret J., Seehars S., Chang C., Monstein C., Amara A., Refregier A., 2017a, Astronomy and Computing, 18, 8
Akeret J., Chang C., Lucchi A., Refregier A., 2017b, Astronomy and Computing, 18, 35
An T., Chen X. MOHAN P., Lao B., 2017, Acta Astronomica Sinica (in Chinese), 58, 18
Arsalan M., Kim D. S., Lee M. B., Owais M., Park K. R., 2019, Expert Systems with Applications, 122, 217
Baan W. A., Fridman P. A., Millenaar R. P., 2004, AJ, 128, 933
Boureau Y.-L., Ponce J., LeCun Y., 2010, Proc. 27th Int. Conf. on Machine Learning (ICML'10). Omnipress, Madison, WI, p. 111
Briggs F., Sorar E., Kraan-Korteweg R., van Driel W., 1997, PASA, 14, 37
Burd P., Mannheim K., März T., Ringholz J., Kappes A., Kadler M., 2018, Astron. Nachr., 339, 358
Czech D., Mishra A., Inggs M., 2018, Astronomy and Computing, 25, 52
Davis J., Goadrich M., 2006, in Cohen W., Moore A.(eds), Proc. 23rd Int. Conf. on Machine Learning, Omnipress, Madison, WI, p. 233
Drozdzal M., Vorontsov E., Chartrand G., Kadoury S., Pal C. J., 2016, Deep Learning and Data Labeling for Medical Applications, First International Workshop, LABELS 2016, and Second International Workshop, DLMIA 2016, held in Conjunction with MICCAI 2016, Lecture Notes in Computer Science Vol. 10008. Springer, Berlin, p. 179
Cosmology Research Group from ETH Zurich, 2016, Raw data from the Bleien Observatory . Available online at: https://people.phys.ethz.ch/ ipa/cosmo/bgs_example_data/
Evans L., Owda M., Crockett K., Vilas A., 2019, Expert Systems with Applications, 127, 353
Gómez-Ríos A., Tabik S., Luengo J., Shihavuddin A. S. M., Krawczyk B., Herrera F., 2019, Expert Systems with Applications, 118, 315
Guizhou Provincial People's Government, 2019, Regulations for protection of electromagnetic wave quiet zone of 500 meter aperture spherical radio telescope in Guizhou province. Available at: http://www.guizhou.gov.cn/zwgk/zcfg/szfwj_8191/szfl_8192/201901/t20190121_2222872.html
Guo G., Wang H., Bell D., Bi Y., Greer K., 2003, in Meersman R., Tari Z., Schmidt D. C., eds, On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. Springer , Berlin, p. 986
He K., Sun J., 2015, Proc. IEEE Conf. Computer Vision and Pattern Recognition. Curran Associates, Red Hook, NY, p. 5353
He K., Zhang X., Ren S., Sun J., 2016a, European Conference on Computer Vision, Lecture Notes in Computer Science Vol. 9908. Springer, Berlin, p. 630
He K., Zhang X., Ren S., Sun J., 2016b, Proc. IEEE Conf. Computer Vision and Pattern Recognition. IEEE, Piscataway, NJ, p. 770
Indermuehle B. T., Harvey-Smith L., Wilson C., Chow K., 2016, 2016 Radio Frequency Interference (RFI). Curran Associates, Red Hook, NY, p. 43
Ioffe S., Szegedy C., 2015, preprint (arXiv:1502.03167)
Kingma D. P., Ba J., 2015, International Conference on Learning Representations, preprint (arXiv:1412.6980)
Krizhevsky A., Sutskever I., Hinton G. E., 2012, Proc. 25th Int. Conf. Neural Information Processing Systems, Volume 1 (NIPS'12). Curran Associates, Red Hook, NY, p. 1097
Lv K., Jiang S., Li J., 2017, preprint (arXiv:1703.03633)
Mosiane O., Oozeer N., Aniyan A., Bassett B. A., 2017, IEEE Radio and Antenna Days of the Indian Ocean, Materials Science and Engineering Conference Series Vol. 198. IOP Publishing, Bristol, p. 012012
Nan R. et al., 2011, International Journal of Modern Physics D, 20, 989
Offringa A. R., de Bruyn A. G., Zaroubi S., Biehl M., 2010a, Proc. Sci., A LOFAR RFI Detection Pipeline and its First Results. SISSA, Trieste, PoS(RFI2010)036
Offringa A. R., de Bruyn A. G., Biehl M., Zaroubi S., Bernardi G., Pandey V. N., 2010b, MNRAS, 405, 155
Pedregosa F. et al., 2011, Journal of Machine Learning Research, 12, 2825
Popping A., Braun R., 2008, A&A, 479, 903
Ronneberger O., Fischer P., Brox T., 2015, in Navab N., Hornegger J., Wells W. M., Frangi A. F., eds, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Springer, Berlin, p. 234
Ruder S., 2016, preprint (arXiv:1609.04747)
Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, Journal of Machine Learning Research, 15, 1929
Wilson C., Chow K., Harvey-Smith L., Indermuehle B., Sokolowski M., Wayth R., 2016, 2016 International Conference on Electromagnetics in Advanced Applications (ICEAA). IEEE, Piscataway, NJ, p. 922
Wolfaardt C. J., 2016, PhD thesis, Stellenbosch University
Zhang H., Nan R., Peng B., Xia Y., Jin C., Li J., Zheng X., Gao L., 2013, 2013 Asia-Pacific Symposium on Electromagnetic Compatibility (APEMC). IEEE, Piscataway, NJ, p. 1
Zhu W. W. et al., 2014, ApJ, 781, 117

This paper has been typeset from a TEX/LATEX file prepared by the author.