An Evaluation of Neural Network Performance Using Complex-Valued Input Data

Kushal Thapa*, Stan McClellan†
Ingram School of Engineering
Texas State University
San Marcos, TX, USA
email: *k_t260@txstate.edu, †stan.mcclellan@txstate.edu

***Abstract*—Complex-valued data is ubiquitous in many scientific fields. However, machine learning for complex-valued input is still in the developmental stage. Alternatively, complex data can be transformed to real data in a few different ways to fit the traditional machine learning framework. In this research, we compare the performance of two such ways - combining real and imaginary components or stacking them - on a simple neural network. To compare these two methods, we create magnitude (combined) and rectangular (stacked) spectrograms from artificial time-series data. Then, we feed the raw 1D time-series dataset, 2D magnitude spectrogram dataset, and 3D rectangular spectrogram dataset to a neural network for training and validation. As a measure of performance, we track the accuracy of each dataset model. From our experimentation, we found out that the rectangular dataset outperforms the magnitude spectrogram in most cases.**

Keywords-complex-valued data; machine learning; neural network; real spectrogram; imaginary spectrogram.

I. INTRODUCTION

Machine Learning (ML) is a computational technique of building models for complex systems using experiential data. Data is at the heart of machine learning. Therefore, the quality, quantity, format, and other characteristics of data have huge impact on the efficacy and effectiveness of the ML models. The quality and quantity aspect of data in ML, in a generalized sense or for a specific domain/application, are well documented in archive literature, such as [1]–[3]. In this paper, we focus on the performance of ML with input data in complex-valued format.

Complex-valued data contains information from both real and imaginary axes. This kind of data is present in many scientific applications and areas, such as signal processing [4] in communication systems, Magnetic Resonance Imaging (MRI) [5] in biomedical imaging, seismic monitoring [6] in geosciences, etc. ML can be a great tool in research and technological development in these areas; however, ML algorithms typically do not handle complex numbers well [7]. Thus, complex data can be pre-processed for ML in these applications by either a) taking only the real component and ignoring the imaginary component or b) combining the real and imaginary components in some way to produce real-valued data or c) separating the real and imaginary components and feeding them simultaneously to the same ML model. Approach a) is generally not desirable because of the loss of information caused by ignoring the imaginary component completely. Interestingly, for approaches b) and c), we are unable to find general guidelines in the literature which describe performance differences or areas of optimality. Hence, the objective of this paper is to make a comparison of these two complex-valued data pre-processing methods for ML with the aim of setting a general guideline when dealing with complex datasets in training ML models.

There is a fourth approach as well, which uses novel Neural Network (NN) models like Complex-Valued Neural Network (CVNN) that can handle the complex dataset. In this approach, the complex-valued data does not need any format change during pre-processing. However, CVNN and the general use of complex numbers in "deep learning" seems to be an active research area, with a lot of different concepts [6]–[12]. Interestingly, the predominant use of real-valued weights in neural networks seems to derive from the focus on real-valued optimization problems. As shown in [8], the use of complex-valued networks (CVNN) on datasets with phase characteristics results in better performance. However, contemporary CVNN are very complicated and sensitive, and the added complexity might not be worth the disruption to the toolchain for most applications. Here, we focus on the use of conventional technology and tools in a way that does not involve a complete re-structuring of the toolset. Hence, our study is limited to the comparison of pre-processing methods of complex-valued dataset for use in real-valued NN.

This paper is organized as follows. Section 2 describes the experimental setup we used to leverage two pre-processing methods for complex-valued data. Section 3 provides the result of the performance comparison between these pre-processing methods when used in a simple NN. Section 4 provides the conclusion of this experimentation.

II. EXPERIMENTAL SETUP

To be able to control, finetune and vary the different data parameters for our comparison, we leverage an artificial dataset. This artificial dataset is based on simple single-bit detection/classification, which is a fundamental component of digital communication. The general findings from the experiments in this dataset should be applicable beyond this domain as well.

Figure 1 shows the flow of our experiment, divided into three main steps. The first step is the creation of artificial time-series data. From a randomly generated binary class digital information {length=10,000 bits}, analog time-series signals were created using Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK), and Phase Shift Keying (PSK) {sampling rate=10,000; number of bits per second=100}.

 ISBN: 978-1-61208-835-8

Then, varying levels of AWGN {from SNR= -21dB to 21dB in increments of 3dB} were added to the clean signals to make our raw time-series data (Test 1). In the second step, this 1D raw time-series data was transformed into a 2D complex-valued spectrogram using Short-Time Fourier Transform (STFT) {frame length=50, frame overlap=50%, window=Hanning}. Then, the complex-valued data was transformed into magnitude-only spectrograms (described as Approach 'b' in Section I: 'Introduction'), as well as real/imaginary spectrograms. The real / imaginary spectrograms were stacked to make a 3D dataset as described in [13] (Approach 'c'). Finally, the 1D time-series dataset, the 2D magnitude spectrogram dataset and the 3D real-imaginary dataset (referred henceforth as rectangular dataset) were flattened and fed into a fully connected NN with one-hidden layer. Although our main objective was to compare the magnitude spectrogram and rectangular spectrogram in a NN, we used time-series dataset as a control input to evaluate and compare the complexity and performance of NN for the other two datasets. In all cases, the hyper-parameters of this NN, listed in Table I, were kept identical. For 'Test 2', an ideal power signal {60Hz, 120V RMS} was added to the noisy time-series signal at the end of 'Step 1', and steps 2 & 3 were repeated. The purpose of 'Test 2' was to simulate the presence of a dominant interfering signal in the raw data. The performance of the NN models from both tests were evaluated primarily using the accuracy metric. This is because accuracy in our experimental context characterizes the Bit Error Ratio (BER), which is an important metric in digital communication. BER is the ratio of wrongly classified bits (or error bits) to the total number of transmitted or evaluated bits. Thus, BER is the "unit complement" of accuracy, i.e., BER + accuracy =100%. Other performance metrics, such as precision and recall, were also measured (see [14] for the data file containing these metrics) but are not evaluated in this paper.

III. RESULTS AND DISCUSSIONS

A. Modulation Intensity

While converting the binary information to an analog signal using either of the three modulation schemes (ASK, FSK or PSK), the differentiating parameter between bit values or states can impact detection. Here, we define the modulation intensity as the difference between these parameter values. For example, if the amplitude of the 'high' is set to '1' and 'low' to '0.4' in ASK, the modulation intensity is 60%. In practice, this modulation intensity is dependent on various external factors, which are not the focus of this study. We chose the intensity parameter based on a subjective 'inflection point' in a NN model-accuracy versus 'low values' graph as shown in Figure 2. Based on these plots (see [14] for similar FSK and PSK figures), we set the intensity values as shown in Table II. Clearly, as intensity decreases, differentiation between bit values or states becomes more difficult, and thus, the NN model classification accuracy drops.

B. Training Time Comparison

The architecture of the NN for all signal types and datasets was the same. However, the size of the datasets was different as shown in Table III. Since the size of each sample was different, the training time was also bound to be different. Figure 3 shows the distribution of the total training time for each type of dataset (ASK, FSK and PSK). The time-series dataset had the quickest training time due to small size and single dimensionality. The rectangular spectrogram dataset was the slowest, with the training time almost twice as much as magnitude spectrogram dataset. Depending on the application, the training time of a NN or ML model can have vital consideration.

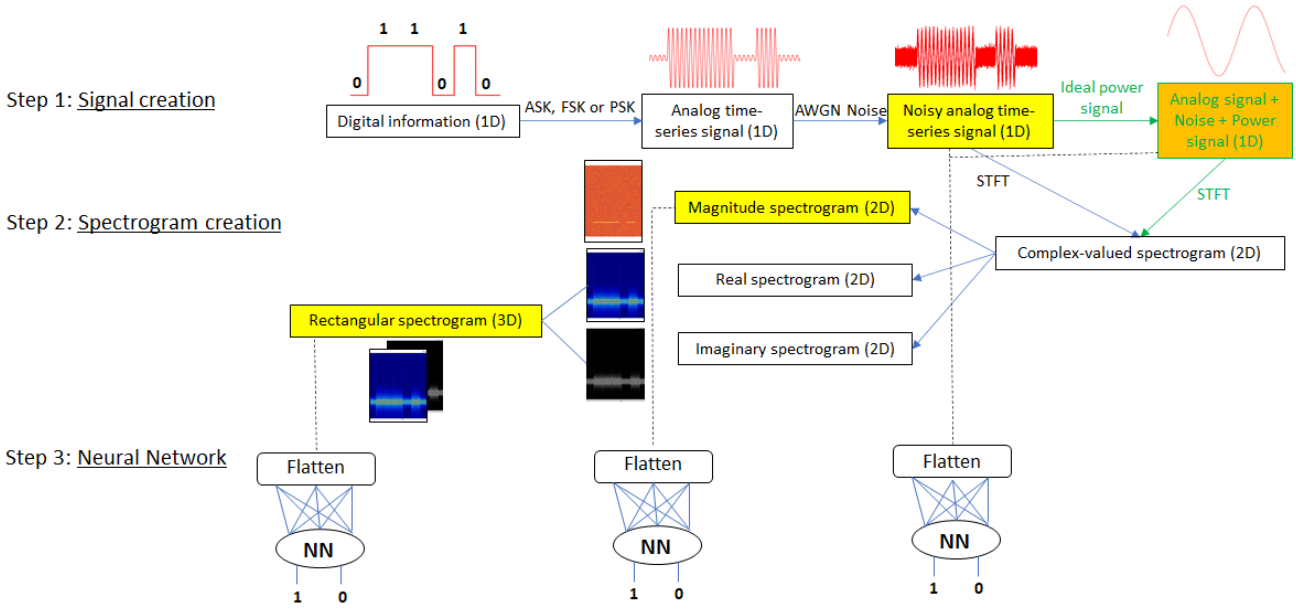

Figure 1. Flow of experiment showing creation of raw time-series modulated signal, transformation to various spectrograms and the use of the three datasets (highlighted) in NN. The extra sub-step of addition of power signal for 'Test 2' is shown in green.

TABLE I. HYPER-PARAMETERS OF THE NEURAL NETWORK

Total number of samples	10,000
Training to Test ratio	70:30
No. of hidden layers	1
No. of nodes in the hidden layer	64
No. of nodes in the output layer	2
Activation function for the hidden layer	Relu
Activation function for the output layer	Softmax
Optimizer	RMSProp
Loss function	Categorical Entropy
No. of training epochs	10
Batch size for training	16

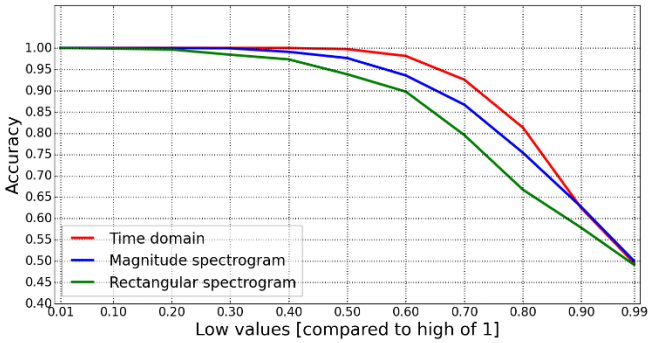

Figure 2. The NN model's test accuracy for a range of 'low values' (compared to a high of '1') for ASK signal with SNR=0dB. The plot shows general decrease in accuracy as 'low-values' get closer to the high-value, i.e., as modulation intensity decreases.

TABLE II. MODULATION INTENSITY VALUES

	High	**Low**
ASK	1 V	0.7 V
FSK	1000 Hz	950 Hz
PSK	0°	25°

TABLE III. DATASET SIZE

Dataset	**Size (each sample)**
Time-series (1D)	100
Magnitude spectrogram (2D)	3 x 1024
Rectangular spectrogram (3D)	3 x 1024 x 2

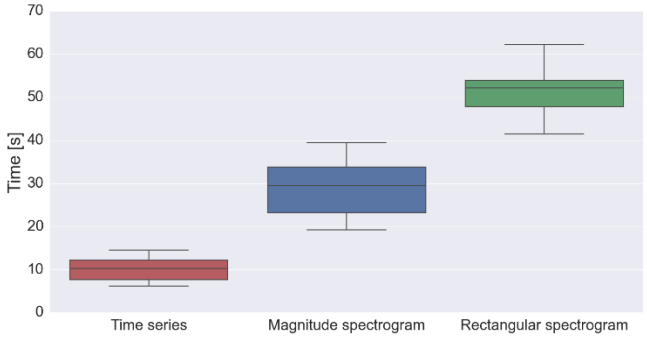

Figure 3. Boxplot showing the total training time distribution for the time-series (red), magnitude spectrogram (blue) and rectangular spectrogram (green) NN models.

C. Test accuracy

1) Test 1

a) ASK signal

Figure 4 shows the accuracy of the NN models for the time-series (1D), magnitude spectrogram (2D) and rectangular spectrogram (3D) datasets containing ASK signals with SNR ranging from -21dB to 21dB. For all models, there is a general trend of increase in testing accuracy when the SNR increases. This is, again, a fairly intuitive behavior since higher SNR means the dataset is 'cleaner' and the NN models can better differentiate between the 'highs' and 'lows' of the core signal.

The comparison between the three datasets is more interesting. In low SNR conditions (less than -15dB), the rectangular spectrogram model seems to perform slightly better than the magnitude spectrogram model. However, as the SNR increases, the performance of the magnitude spectrogram model improves rapidly and overtakes the rectangular spectrogram model after about -6dB. This can be explained by the type and quantity of information each dataset contains. Magnitude spectrogram, by definition, contains the magnitude or energy information of the signal, which is directly proportional to the signal amplitude. So, for ASK signals, the magnitude spectrogram more clearly represents modulation transitions in higher SNR conditions, thus simplifying the task of the NN as compared to the rectangular spectrogram. In contrast, the rectangular spectrogram holds more information about the signal, which is a boon in low SNR conditions but also the cause of data dilution resulting in lower performance compared to magnitude spectrogram in high SNR condition.

b) FSK signal

Figure 5 shows a similar NN model accuracy versus SNR plot as Figure 4, but for FSK signals. In contrast with the ASK signals of Figure 4, the comparison between the time-series, rectangular spectrogram and magnitude spectrogram models is more consistent across all SNR levels. The time-series and rectangular spectrogram models are evenly matched, and better than magnitude spectrogram models (until convergence occurs) in terms of accuracy.

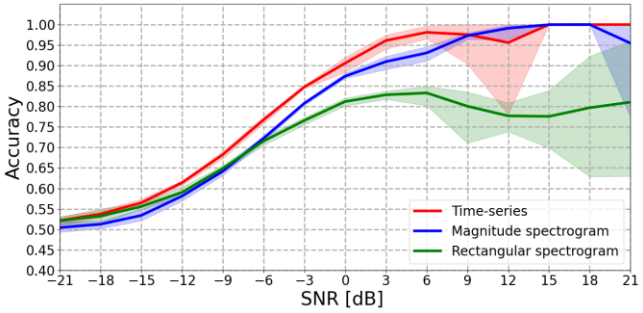

Figure 4. Test accuracy of NN models trained with time domain, magnitude spectrogram and rectangular spectrogram datasets containing ASK signals (high=1, low=0.7) with SNR ranging from -21dB to 21dB. Time-series model had generally highest accuracy while the rectangular spectrogram model shows better performance than magnitude spectrogram model only in low SNR conditions.

 ISBN: 978-1-61208-835-8

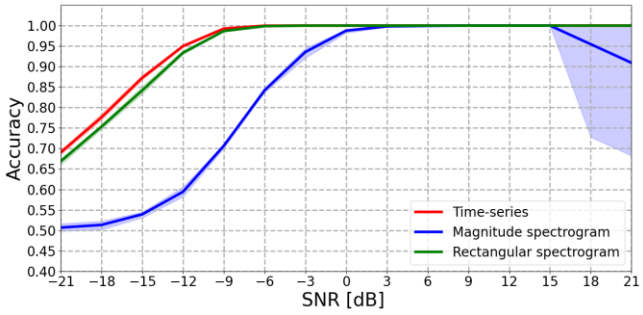

Figure 5. Accuracy versus SNR plot for NN models of FSK signals (high=1000 Hz, low=950 Hz) showing similar performance of time-series and rectangular spectrogram models while the magnitude spectrogram model performed worst across all SNR levels.

The higher accuracy of rectangular spectrogram compared to magnitude spectrogram can again be explained by the quality and quantity of information represented by the magnitude and rectangular spectrograms. The real, imaginary and magnitude spectrograms all contain the frequency shift information. By stacking the real and imaginary parts together, the quantity of information is doubled in rectangular spectrogram. However, unlike ASK, this does not dilute the dataset because the quality of information in relation to frequency is the same in all three sets. Therefore, the 3D rectangular spectrogram performs better than the 2D magnitude spectrogram over all SNR values.

c) PSK signal

Figure 6 shows a similar NN model accuracy versus SNR plot as Figures 4 and 5 but for PSK signals. As with the FSK models, the time-series and rectangular spectrogram model accuracies are evenly matched across all SNR levels. However, the magnitude spectrogram models were stuck at around 50% accuracy regardless of the SNR level. By definition, magnitude spectrogram completely ignores the phase information of the complex-valued spectrogram, and thus, the corresponding model can't distinguish between the different phases of the PSK signals. In contrast, the rectangular data retains this phase information indirectly, as indicated by the improved rectangular spectrogram accuracy curve in Figure 6.

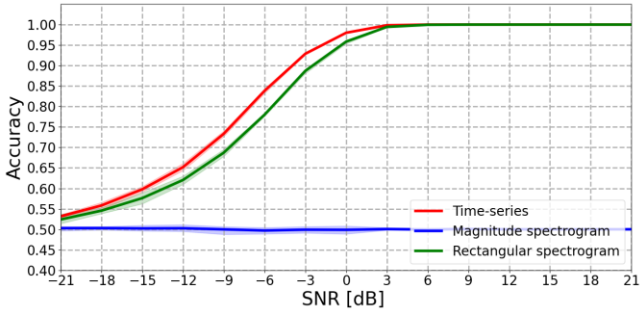

Figure 6. Accuracy versus SNR plot for NN models of PSK signals (high=0°, low=25°) showing the similar performace of time-series and rectangular spectrogram models. The magnitude spectrogram models' accuracy was approximately 50% for all SNR levels because of its inherent inability to retain phase information.

2) Test 2

From Figure 4-6, we observe that time-series NN model performed better than the spectrogram models in all three cases. This can be attributed to the unprocessed information that this raw time-series signal contains. Spectrograms need pre-processing, and each pre-processing step results in some information loss. Hence, the pre-processed spectrograms contained less information than the unprocessed time-series signal. However, the information contained in the time-series signal can be confused in the presence of fake or interfering signals. In such cases, we expect the classification performance of the time-series NN model to suffer. To test this hypothesis, we conducted 'Test 2'.

As explained in Section II, Test 2 is similar to Test 1 except that an ideal power signal is included to the AWGN added modulated signal as a strong out-of-band interferer. In practical terms, this power signal simulates a dominant interfering signal that makes the identification and classification of the desired signal more difficult and can confound the ML training process. This test case directly corresponds to an application scenario of a power line communication medium where the power signal is much stronger than the communication signal and affects the reactive channel in various other ways.

Figure 7 shows the accuracy of the NN models for time-series, magnitude spectrogram and rectangular spectrogram datasets each with ASK, FSK or PSK signals. This figure shows that the time-series and magnitude spectrogram NN models fail to produce any notable result regardless of modulation type. On the other hand, the rectangular spectrogram models are able to compensate for the dominant, out-of-band interfering signal and produce a good classification result. This figure also shows that this plot is not merely a far-left extension (very low SNR) of the plots in Figures 4, 5 and 6, at least not for rectangular spectrogram models, as the accuracies approach 100%. The sharp increase in accuracies with increasing SNR indicates that the rectangular spectrogram model is affected more by the less energetic AWGN than the highly energetic out-of-band interferer.

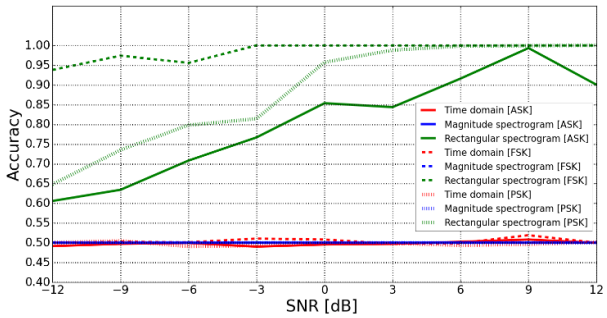

Figure 7. Accuracies of the time-series, magnitude spectrogram and rectangular spectrogram NN models for ASK, FSK and PSK signals with added ideal power signal. The SNR levels in the X-axis of the plot is discounting the power signal (i.e., this SNR=Energy of the core modulated signal/Energy of the AWGN).

 ISBN: 978-1-61208-835-8

IV. CONCLUSION

In this study, we compared the performance of time-series, magnitude spectrogram and rectangular spectrogram datasets in training a simple, fully connected NN. We observed that time-series and rectangular spectrogram training data performed better than magnitude spectrogram data for FSK, PSK and low-SNR ASK signals. We also observed that rectangular spectrogram training data performs significantly better than other formats when there are dominant out-of-band interferers present.

REFERENCES

[1] J. Zhou, R. Cao, J. Kang, K. Guo, and Y. Xu, "An Efficient High-Quality Medical Lesion Image Data Labeling Method Based on Active Learning," IEEE Access, vol. 8, pp. 144331–144342, 2020, doi: 10.1109/ACCESS.2020.3014355.

[2] S. Raschka, "Chapter 4: Building good training dataset," in Python machine learning : machine learning and deep learning with Python, scikit-learn, and TensorFlow., Second edition, Fully revised and Updated., Packt Publishing, 2017, pp. 109–144.

[3] M. C. Sorkun, J. M. V. A. Koelman, and S. Er, "Pushing the limits of solubility prediction via quality-oriented data selection," iScience, vol. 24, no. 1, p. 101961, Jan. 2021, doi: 10.1016/j.isci.2020.101961.

[4] J. Krzyston, R. Bhattacharjea, and A. Stark, "Complex-Valued Convolutions for Modulation Recognition using Deep Learning," in 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Jun. 2020, pp. 1–6, doi: 10.1109/ICCWorkshops49005.2020.9145469.

[5] W. He, Y. Zhang, J. Ding, and L. Zhao, "A Modified Phase Cycling Method for Complex-Valued MRI Reconstruction.," Int. J. Biomed. Imaging, pp. 1–7, Nov. 2020, doi: 10.1155/2020/8846220.

[6] J. S. Dramsch, M. Lüthje, and A. N. Christensen, "Complex-valued neural networks for machine learning on non-stationary physical data," Comput. Geosci., vol. 146, p. 104643, Jan. 2021, doi: 10.1016/j.cageo.2020.104643.

[7] S. Scardapane, S. V. Vaerenbergh, A. Hussain, and A. Uncini, "Complex-Valued Neural Networks With Nonparametric Activation Functions," IEEE Trans. Emerg. Top. Comput. Intell., vol. 4, no. 2, pp. 140–150, Apr. 2020, doi: 10.1109/TETCI.2018.2872600.

[8] H. G. Zimmermann, A. Minin, and V. Kusherbaeva, "Comparison of the complex valued and real valued neural networks trained with gradient descent and random search algorithms," 2010, pp. 213–218.

[9] R. Chakraborty, Y. Xing, and S. X. Yu, "SurReal: Complex-Valued Learning as Principled Transformations on a Scaling and Rotation Manifold," IEEE Trans. Neural Netw. Learn. Syst., pp. 1–12, 2020, doi: 10.1109/TNNLS.2020.3030565.

[10] M. Amin, "Complex-Valued Neural Networks: Learning Algorithms and Applications," Ph.D. Dissertation, University of Fukui, Japan, 2012.

[11] J. A. Barrachina, C. Ren, C. Morisseau, G. Vieillard, and J.-P. Ovarlez, "Complex-Valued vs. Real-Valued Neural Networks for Classification Perspectives: An Example on Non-Circular Data," ArXiv200908340 Cs Stat, Sep. 2020. [Online]. Available from: http://arxiv.org/abs/2009.08340 [retrieved: April, 2021]

[12] C. Trabelsi et al., "Deep Complex Networks," in arXiv:1705.09792 [cs], Feb. 2018, pp. 1–19. [Online]. Available from: http://arxiv.org/abs/1705.09792 [retrieved: April, 2021]

[13] J. Shima, "Weak signal processing systems and methods," United States Patent 10879946, Dec. 29, 2020.

[14] K. Thapa, kushal-thapa/NN_complex-valued-data. 2021. https://github.com/kushal-thapa/NN_complex-valued-data.

 ISBN: 978-1-61208-835-8