

1. Record Nr.	UNINA9911009387303321
Autore	Ansari Irshad Ahmad
Titolo	Signal Processing with Python : A Practical Approach
Pubbl/distr/stampa	Bristol : , : Institute of Physics Publishing, , 2024 ©2024
ISBN	9780750359313 0750359315
Edizione	[1st ed.]
Descrizione fisica	1 online resource (297 pages)
Collana	IOP Ebooks Series
Altri autori (Persone)	BajajVarun
Soggetti	Signal processing Python (Computer program language)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Intro -- Acknowledgements -- Editor biographies -- Irshad Ahmad Ansari -- Varun Bajaj -- List of contributors -- Contributor biographies -- Outline placeholder -- Mosabber Uddin Ahmed -- Meena Anandan -- Zahra Ghanbari -- Aakash Kumar Jain -- K P Madhavan -- Manas Kumar Mishra -- Ishrat Jahan Mohima -- Mohammad Hassan Moradi -- Poorya Moradi -- Rohini Palanisamy -- Nakul Kishor Pathak -- K K Mujeeb Rahman -- Mozhdeh Saghalaini -- Deepika Sainani -- Urvashi Prakash Shukla -- Abhishek Kumar Singh -- Mahdi Taghaddossi -- Pandiyarasan Veluswamy -- Chapter Automatic feature extraction using deep learning for automatic modulation classification implemented with Python -- 1.1 Introduction -- 1.2 Proposed AMC based on DL framework -- 1.3 System model for dataset generation -- 1.4 Generalized feature extraction system using DL -- 1.4.1 Spatial feature extraction using CNNs -- 1.4.2 CLDNN-based approach -- 1.4.3 MCLDNN based approach -- 1.4.4 Complement of multichannel structure from bimodal information using BMCCLDNN models -- 1.5 Data preparation -- 1.5.1 Python code (libraries, file paths) -- 1.5.2 Python code (random shuffling of data) -- 1.5.3 Python code (data preparation for training) -- 1.5.4 Python code (data reshaping) -- 1.5.5 Python code (model training) -- 1.5.6 Python code (model evaluation) -- 1.5.7 Python code (model prediction) -- 1.6 Conclusion --

Acknowledgments -- Bibliography -- Chapter Applying B-value and empirical equivalence hypothesis testing to intellectual and developmental disabilities electroencephalogram data -- 2.1 Introduction -- 2.2 Materials -- 2.2.1 Dataset -- 2.2.2 Pre-processing -- 2.3 Feature extraction -- 2.3.1 DWT -- 2.3.2 PSD -- 2.4 Statistical method -- 2.4.1 The two-stage hypothesis testing based on EEB -- 2.4.2 Implementation (Python code) -- 2.5 Results. 2.5.1 Comparison of two groups of IDD and TDC under the same conditions -- 2.5.2 Comparison of rest-state and music stimuli for each group of IDD and TDC -- 2.6 Conclusion -- Bibliography -- Chapter Filter design and denoising technique for ECG signals -- 3.1 Introduction -- 3.2 Filter types -- 3.2.1 Time domain filters -- 3.2.2 Frequency selective filters -- 3.3 Python libraries for filter design -- 3.4 Filter specifications -- 3.5 Mapping of the digital frequency -- 3.6 Time domain filters -- 3.6.1 Moving average window -- 3.6.2 Derivative filter -- 3.7 Frequency selective filters -- 3.7.1 FIR filter -- 3.7.2 IIR filters -- 3.7.3 Adaptive filter -- 3.8 Conclusion -- Bibliography -- Chapter Electroencephalogram signal processing with Python -- 4.1 Introduction -- 4.2 Principal and primary actions in EEG signal processing -- 4.2.1 Importing EEG signals to Python environment -- 4.2.2 Saving the details of EEG data on other variables -- 4.2.3 Modifying the imported EEG data -- 4.2.4 Extracting data from a raw object -- 4.2.5 Saving objects -- 4.2.6 Working with events -- 4.3 Exposure signals in time and frequency -- 4.3.1 Displaying data information in time and frequency -- 4.3.2 Topomap displaying -- 4.3.3 Saving MNE-produced plots and images -- 4.4 EEG signals preprocessing -- 4.4.1 Setting an EEG reference -- 4.4.2 Removing bad channels and data spans -- 4.4.3 Filtering the data and resampling -- 4.4.4 Artifact removal by ICA -- 4.5 EEG signal processing -- 4.5.1 Evoke responses analysis -- 4.5.2 Time-frequency analysis -- 4.5.3 Source localization -- Bibliography -- Chapter AG-PSO: prediction of heart diseases for an unbalanced dataset using feature extraction -- 5.1 Introduction -- 5.2 Literature survey -- 5.3 Proposed methodology -- 5.3.1 Data augmentation and imbalance (SMOTE) -- 5.3.2 Feature subset selection (BPSO). 5.3.3 Prediction (various ML algorithms) -- 5.4 Parameter settings for the simulation study -- 5.4.1 Experimental dataset -- 5.4.2 Experimentation platform -- 5.4.3 Evaluation metric -- 5.5 Results and analysis -- 5.6 Conclusion -- Bibliography -- Chapter Python based bio-signal processing: mitigation of baseline wandering in pre-recorded electrooculogram -- 6.1 Introduction -- 6.1.1 Baseline wander noise -- 6.1.2 Muscle tremor noise -- 6.1.3 Powerline interference -- 6.1.4 EOGs and their clinical use -- 6.2 Introduction to Google Colab -- 6.2.1 How to use Google Colab -- 6.3 Algorithm for correction of baseline wandering in pre-recorded EOG signals -- 6.3.1 Steps 1-3 -- 6.3.2 Step 4 -- 6.3.3 Steps 5-6 -- 6.3.4 Step 7 -- 6.3.5 Step 8 -- 6.3.6 Step 9 -- 6.3.7 Step 10 -- 6.3.8 Steps 11 and 12 -- 6.4 Conclusion -- Bibliography -- Chapter Efficient nanoscale device modeling using artificial neural networks with TensorFlow and Keras libraries in Python -- 7.1 Introduction -- 7.1.1 Motivation and background -- 7.1.2 Problem statement and objectives -- 7.2 Brief literature survey -- 7.2.1 MuGFETs and their characteristics -- 7.2.2 ANNs and their application in compact modeling -- 7.2.3 Previous work using ANNs for modeling MuGFETs -- 7.3 Methodology -- 7.3.1 Overview of proposed approach -- 7.3.2 Implementation details -- 7.3.3 Hypertuning for optimizing performance -- 7.3.4 DNN modeling and training -- 7.4 Results and analysis -- 7.4.1 Model evaluation and comparison with TCAD simulations -- 7.4.2 Potential and limitations of

the proposed approach -- 7.5 Conclusion -- Bibliography -- Chapter
 A Python-based comparative study of convolutional neural network-
 based approaches for the early detection of breast cancer -- 8.1
 Introduction -- 8.2 Related works -- 8.3 Methodology -- 8.3.1 Dataset
 -- 8.3.2 Image preprocessing -- 8.3.3 Convolutional neural network.
 8.3.4 Model overview -- 8.3.5 Evaluation metrics -- 8.4 Result analysis
 -- 8.5 Conclusion -- Acknowledgments -- Bibliography -- Chapter
 Maximum power point tracking for partially shaded photovoltaic
 system using advanced signal processing -- 9.1 Introduction -- 9.2
 Modeling and characteristics of PV systems -- 9.2.1 Modeling of PV
 cells -- 9.2.2 Modeling of PV modules -- 9.2.3 Characteristics of PV
 modules -- 9.2.4 PV system schematics -- 9.3 MPPT: concept and
 traditional techniques -- 9.3.1 Need and concept of MPPT -- 9.3.2
 Traditional MPPT concepts -- 9.4 Challenges of MPPT -- 9.4.1 Dynamic
 changes in temperature and irradiance -- 9.4.2 PSC and its challenges
 -- 9.4.3 MPPT requirements -- 9.5 Soft computing methods -- 9.5.1
 Fuzzy logic -- 9.5.2 Artificial neural network -- 9.5.3 Adaptive neural
 fuzzy inference system -- 9.5.4 Extreme learning mechanism -- 9.5.5
 Extremum seeking control -- 9.5.6 Reinforcement learning -- 9.5.7
 Bayesian network -- 9.6 Meta-heuristic techniques -- 9.6.1 Swarm
 intelligence -- 9.6.2 Evolutionary algorithms -- 9.6.3 Mathematics-
 and physics-based methods -- 9.6.4 Sociology-based methods -- 9.7
 Exact algorithms -- 9.7.1 Mathematics-based algorithms -- 9.7.2
 Techniques based on P V array characteristics -- 9.8 Hardware
 configuration-based MPP methods -- 9.8.1 Array reconfiguration-
 based methods -- 9.8.2 Power electronics-based methods -- 9.9
 Analysis and comparison of various techniques -- 9.10 Challenges and
 future scope -- 9.11 Conclusion -- Bibliography -- Chapter
 Automating Monte Carlo simulation data analysis using Python in
 Anaconda environment -- 10.1 Advanced nodes design -- 10.1.1 A
 simple memory design -- 10.1.2 Monte Carlo simulations -- 10.1.3
 Challenges in Monte Carlo simulation -- 10.1.4 Fast Monte Carlo
 simulations -- 10.1.5 Validation of design for tool evaluation.
 10.2 Simulation setup and tools overview -- 10.3 Analysis of
 simulation results with Python -- 10.4 Conclusion -- 10.5 Future scope
 -- Bibliography.

Sommario/riassunto

This book explores the domain of signal processing using Python, with the help of working examples and accompanying code and introduces the concepts of Python programming via signal processing with numerous hands-on examples and code snippets.
