ANKARA YILDIRIM BEYAZIT UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES



RANDOMIZED LOW-RANK NONLINEAR RX DETECTOR

M.Sc. Thesis by

Selçuk YAPICI

Department of Computer Engineering

February, 2022

ANKARA

RANDOMIZED LOW-RANK NONLINEAR RX DETECTOR

A Thesis Submitted to the

Graduate School of Natural And Applied Sciences of

Ankara Yildirim Beyazit University

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering, Department of Computer Engineering

by

Selçuk YAPICI

February, 2022

ANKARA

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**RANDOMIZED LOW-RANK NONLINEAR RX DETECTOR**" completed by **SELÇUK YAPICI** under supervision of **Assoc. Prof. Dr. FATİH NAR** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Fatih NAR

Supervisor

Assist. Prof. Dr. Fatma KÜÇÜK

Jury Member

Assist. Prof. Dr. Ayşe Nurdan SARAN

Jury Member

Prof. Dr. Sadettin ORHAN

Director

Graduate School of Natural and Applied Sciences

ETHICAL DECLARATION

I hereby declare that, in this thesis which has been prepared in accordance with the Thesis Writing Manual of Graduate School of Natural and Applied Sciences,

- All data, information and documents are obtained in the framework of academic and ethical rules,
- All information, documents and assessments are presented in accordance with scientific ethics and morals,
- All the materials that have been utilized are fully cited and referenced,
- No change has been made on the utilized materials,
- All the works presented are original,

and in any contrary case of above statements, I accept to renounce all my legal rights.

Date: 2022, 15 February **Signature:** _____

Name & Surname: Selçuk YAPICI

ACKNOWLEDGEMENTS

The author would like to express his sincere gratitude to his advisor Dr. Fatih NAR of the Faculty of Engineering and Natural Sciences at Yıldırım Beyazıt University, for his continuous support and motivation during the study of the author. His immense knowledge and valuable recommendations created the milestones of this study. His guidance helped the author all the time of research and writing this thesis. The author would also like to thank Dr. Hilal KAYA who resigned from the university, for her great help and support the author before starting this thesis.

In addition, thanks to the AYBU family, who made me love the department, lessons, research and learning until I came to the this process.

Finally, many thanks to his intentive, affectionate and supportive wife, Şaylan who was a backstop to safeguard and reinforce the author in his rough times. The author would also like to thank his son Kerem Mirza and his daughter Beren Serra who comforted him with their childlike, innocent and full of life.

February, 2022 **Selçuk YAPICI**

RANDOMIZED LOW-RANK NONLINEAR RX DETECTOR

ABSTRACT

Anomaly detection is an important topic in various application areas including image analysis and network intrusion detection. The Reed–Xiaoli detector is an efficient and accurate anomaly detector that can be used if analyzed data is Gaussian distributed. However, in real-world data is rarely Gaussian distributed. For nonlinear data, kernel RX (KRX) is proposed and widely employed. The biggest issue with KRX method is its high computational complexity which prevents using it for big data or in real-time scenarios. As a remedy, in the literature, Random Fourier Features (RFF) approximation of the KRX method is proposed, namely RRX method. Another weakness of KRX is numerical issues, basically kernel matrix being bad-conditioned, which can be solved by regularization. This situation also applies to RRX method. In this study, we extend the RRX method with randomized SVD as an efficient solution owing to best low-rank approximation being Singular Value Decomposition (SVD). Proposed method, randomized low-rank RX (RLR-RX), provides better computational complexity and better detection performance compared to both RRX and KRX. Efficiency and detection performance of the proposed RLR-RX method is demonstrated using a synthetic dataset and real-world dataset.

Keywords: Anomaly Detection, RX, Kernel RX, Random Fourier Features, Low-Rank, Randomized SVD

RASTGELELEŞTİRİLMİŞ DÜŞÜK RÜTBELİ RX DEDEKTÖRÜ

ÖZ

Anormallik tespiti, görüntü analizi ve ağ saldırı tespiti dahil olmak üzere çeşitli uygulama alanlarında önemli bir konudur. Reed–Xiaoli dedektürü, analiz edilen veriler Gauss dağılımına sahipse kullanılabilecek verimli ve doğru bir anomali dedektörüdür. Ancak, gerçek dünyada veriler nadiren Gauss dağılımına sahiptir. Doğrusal olmayan veriler için çekirdek RX (KRX) önerilmiş ve yaygın olarak kullanılmaktadır. KRX yöntemiyle ilgili en büyük sorun, büyük veri veya gerçek zamanlı senaryolarda kullanılmasını engelleyen yüksek hesaplama karmaşıklığıdır. Çözüm olarak literatürde KRX yönteminin Rastgele Fourier Öznitelikleri (RFF) yaklaşımı, yani RRX yöntemi önerilmiştir. KRX'in bir başka zayıflığı da sayısal sorunlardır, temel olarak çekirdek matrisi kötü koşullandırılmıştır ve düzenlileştirme ile çözülebilir. Bu durum RRX yöntemi için de geçerlidir. Bu çalışmada, en iyi düşük dereceli yaklaşımın Tekil Değer Ayrışımı (SVD) olması nedeniyle verimli bir çözüm olarak RRX yöntemini randomize SVD ile genişletiyoruz. Önerilen yöntem, randomize düşük dereceli RX (RLR-RX), hem RRX hem de KRX'e kıyasla daha iyi hesaplama karmaşıklığı ve daha iyi algılama performansı saşlar. Önerilen RLR-RX yönteminin etkinliği ve algılama performansı, sentetik bir veri kümesi ve gerçek dünya veri kümesi kullanılarak gösterilmiştir.

Anahtar kelimeler: Anomali tespiti, RX, çekirdek RX, rastegele Fourier öznitelikleri, düşük-rank, rastgelileştirilmiş SVD.

CONTENTS

M.Sc THESIS EXAMINATION RESULT FORM ii
ETHICAL DECLARATION iii
ACKNOWLEDGEMENTS iv
ABSTRACT v
ÖZ vi
LIST OF TABLES ix
LIST OF FIGURES x

CHAPTER 1 – INTRODUCTION 1

CHAPTER 2 – RX DETECTOR AND VARIANTS 5

2.1 RX Anomaly Detector 6
2.2 Kernel RX (KRX) Anomaly Detector 6
2.3 Randomized RX (RRX) Anomaly Detector 9

CHAPTER 3 – PROPOSED METHOD 10

3.1 Singular Value Decomposition (SVD) 11
3.2 Proposed Anomaly Detection Method 12

CHAPTER 4 – RESULTS 15

4.1 Synthetic Dataset 15
4.2 Real-world Dataset 17
4.2.1 Detection Performance for Airport Scenes 21
4.2.2 Detection Performance for Beach Scenes 24
4.2.3 Detection Performance for Urban Scenes 26
4.2.4 EDetection Performances versus Execution Time in ABU Dataset 29

CHAPTER 5 – CONCLUSIONS 30

REFERENCES 32

CURRICULUM VITAE 38

LIST OF TABLES

Table 2.1 Space and time complexity for RX, RRX, and KRX methods.....9

Table 3.1 Space and time complexity for RX, RRX, RLR-RX, and KRX methods. 14

Table 4.1 Detection accuracy and execution time on synthetic data..... 17

Table 4.2 Detection performance comparison for the ABU dataset 19

LIST OF FIGURES

Figure 2.1 Sliding window approach for Local RX Detector [1] 5
Figure 2.2 Data distributions of RX and KRX for 2 toy dataset. 8
Figure 3.1 SVD and low-rank approximation for rectangular matrix **A** 11
Figure 3.2 SVD for positive definite symmetrix matrix **A** 11
Figure 3.3 SVD and Low Rank approximation for the matrix **G** 12
Figure 4.1 Synthetic dataset with non-Gaussian distribution 15
Figure 4.2 Execution time with respect to data size on synthetic data 16
Figure 4.3 Airport scenes from ABU dataset 18
Figure 4.4 Beach scenes from ABU dataset 18
Figure 4.5 Urban scenes from ABU dataset 18
Figure 4.6 ROC curve of all ABU scenes 19
Figure 4.7 Condition Number versus Detection Accuracy 20
Figure 4.8 ROC curve of airport-1 scene from ABU dataset 21
Figure 4.9 ROC curve of airport-2 scene from ABU dataset 22
Figure 4.10 ROC curve of airport-3 scene from ABU dataset 22
Figure 4.11 ROC curve of airport-4 scene from ABU dataset 23
Figure 4.12 ROC curve of ABU airport scenes 23
Figure 4.13 ROC curve of beach-1 scene from ABU dataset 24
Figure 4.14 ROC curve of beach-2 scene from ABU dataset 24
Figure 4.15 ROC curve of beach-3 scene from ABU dataset 25
Figure 4.16 ROC curve of beach-4 scene from ABU dataset 25
Figure 4.17 ROC curve of ABU beach scenes 26
Figure 4.18 ROC curve of urban-1 scene from ABU dataset 26
Figure 4.19 ROC curve of urban-2 scene from ABU dataset 27
Figure 4.20 ROC curve of urban-3 scene from ABU dataset 27
Figure 4.21 ROC curve of urban-4 scene from ABU dataset 28
Figure 4.22 ROC curve of urban-5 scene from ABU dataset 28
Figure 4.23 ROC curve of ABU urban scenes 29
Figure 4.24 Detection Performances versus Execution Time for ABU Dataset 29

CHAPTER 1

INTRODUCTION

Anomaly detection (AD) is an important topic to detect outliers as desired targets or to clean the data for further machine learning tasks. Thus, AD is used in various application areas, such as remote sensing [2], finance, network intrusion detection [3, 4], and identification of geochemical anomalies [5]. Basically, AD assumes that anomalies/outliers are rare in the data and tries to find out the data that differs from most of the data [2, 6]. A similar concept to AD is target detection, where the fundamental difference is AD does not assume known target spectra, while target detection uses the known spectra of targets in interest [7].

AD methods employ various approaches. In one popular approach, background is modeled using a probability distribution, such as Gaussian distribution. Since occurrence of anomalies is assumed to be rare, all data is assumed to be background and used to find probability distribution parameters. Finally, data points with very low probability or distant from the distribution center are marked as anomalies. Possible outliers can be censored to eliminate corruption of background statistics because of anomalies in the data [7, 8]. As a popular approach, the Reed-Xiaoli (RX) detector models the background as a multivariate Gaussian distribution. RX detector determines the data points as anomalies that are distant from the data center based on the Mahalanobis metric [9]. So, RX method can find target pixels that have different spectral signature from the background. However, Gaussian distribution assumption cannot model tails and fails, especially in data with complex feature relations. Besides, especially heterogeneous backgrounds are usually not Gaussian distributed. To overcome these limitations, several improvements for RX detector are proposed in the literature. For non-Gaussian distributed data, a nonlinear version of RX has been introduced using kernelization approach, namely kernel RX (KRX) [10, 11]. Downside of the KRX is its huge space and time complexity for large data where space complexity is quadratic and time complexity is cubic regarding the amount of data. This situation limits the practical use or KRX methods in real-world scenarios.

Because of limitations of KRX, several improvements for KRX detector are also proposed. In [12], authors assume spherical covariance matrix to model the scene background as a variant of KRX to overcome the problem of badly conditioned covariance matrix. This situation happens when there are a few high-dimensional samples. A novel cluster KRX method is proposed in [13] where background pixels are grouped into clusters followed by a fast Eigen decomposition leading to anomaly detection index. In [8], a novel background purification algorithm is proposed as a matrix-free approach which also overcomes the problems occurs because of elimination of small eigenvalues. Recently, randomized RX (RRX) which uses random Fourier Features [14], is proposed to reduce to space and time complexity of KRX while keeping its detection performance [15, 16].

In addition to improvements proposed for the KRX method, researchers also proposed improvements to standard RX method. Basically, they try to preserve computational efficiency of RX while improving its detection performance. First and second order statistic estimation are used by [17] while kurtosis-based test is used by [18] to refine RX detection mechanism. Acito et al., analyzes the CFAR property of the RX method, considering the signal-dependent noise in hyperspectral images [19]. In [20], authors estimate the background probability density function using kernel density estimator where bandwidth selection process is obtained via extracted locally adaptive information. As an improvement to RX, in [21], authors use median-mean line metric to deal with degrading effects of anomalous outliers.

Researchers also proposed several approaches and methods to improve computational efficiency of the RX and KRX methods to obtain fast versions or implementations. In first approach, researchers propose algorithmic or mathematical ways of improving the computational efficiency such as using rank-1 update using Woodbury matrix identity of covariance matrix as sliding window moves to next pixel [22–28]. Second group of studies are focused on efficient parallel implementations using multicore CPUs, massively parallel GPUs, specific hardwares such as FPGAs, or even heterogeneous computing environments [1, 29–32]. Also, in [15, 16], randomization based approaches are proposed to ease the huge computational load of the KRX method.

In addition to RX and its variants, in the literature, anomaly detection approaches based on different ideas are also proposed. In [33], authors used Gauss–Markov random field to find small regions in the scene which they assumed as anomalies in a homogeneous background. Some researchers also investigate use of clustering for anomaly detection [13, 34]. In [35], authors use maximum entropy and developed a nonparametric estimation for detecting anomalies in hyperspectral images. In [36], a projection pursuit based method is proposed for anomaly detection in hyperspectral images. Later, local-sparsity-divergence is used by [37] for anomaly detection in hyperspectral images. With an idea of a background pixel can be approximately represented by its spatial neighborhoods while anomalies cannot be used as a novel anomaly detection for hyperspectral images [38]. In the literature, several subspace and sparsity based anomaly detection methods are also proposed [11, 39–42]. In some studies, spatial information is also incorporated for anomaly detection since neighboring pixels may share common characteristics [43, 44]. In [45], authors investigates importance of band selection in the performance of anomaly detectors. As Deep Learning becomes popular in many research areas, it is also applied to anomaly detection where Deep Belief Network (DBN) is used [46]. In this method, unsupervised auto-encoder is used to learn the high-level features as a way of modeling the background / clutter.

In the literature, low-rank approximations are also proposed for the anomaly detection. In [47], authors propose to decompose the model matrix into a low-rank matrix, a sparse matrix, and a noise matrix where background is represented by low-rank matrix and anomalies are assumed to be represented by the sparse matrix. Similarly, in [48], authors assume that low-rank matrix represents the background and sparse matrix represents the anomalies where they used low-rank representation and learned dictionary. Finally, in [16], authors propose use of Random Fourier Features (RFF) [14] and Nyström approximations [49] of KRX as low-rank approximations. Such randomization techniques to speed-up kernel methods are also applied for different kernelized approaches [50]. In [16], authors stated that these low-rank approximations also act as regularizer but our experiments show that they cannot provide strong regularization in all cases.

All the efforts in the literature show that an anomaly detection method which can provide high detection accuracy for homogeneous and heterogeneous data in a robust and computationally efficient manner is in great demand. However, these requirements are known to be contradicting to each other since accurate anomaly detection requires more computational resources. Also, anomaly detection method should be simple and adaptation to data must be easily done with a few parameters if possible. Instead of developing a completely new anomaly detection with such characteristics, some researchers tried to increase computational efficiency of an existing detector that already provides satisfactory detection accuracy [16]. More detailed comparison and review of anomaly detection for hyperspectral Images are given in [2, 6, 7].

In this thesis, RRX method [16] is extended with randomized Singular Value Decomposition (SVD) [51] as an efficient and accurate approximation of KRX method, which can detect anomalies for Gaussian and non-Gaussian distributed data. Proposed RLR-RX method first uses RFF [14] to map original nonlinear data to high dimensional space where it becomes linear. Then, covariance matrix is constructed using the mapped high-dimensional data. In this thesis, instead of inverting this covariance matrix for detecting anomalies as done in RRX, the covariance matrix is decomposed using a randomized SVD method [51] and eigenvectors with low or zero eigenvalues are eliminated, and this smaller low-rank approximated covariance matrix is efficiently inverted. Therefore, RLR-RX method becomes faster than RRX while an implicit regularization being enforced due to employed low-rank approximation.

CHAPTER 2

RX DETECTOR AND VARIANTS

The RX detector is easy-to-understood and implement hence becomes a popular anomaly detection method in the literature. Therefore, the RX anomaly detector is applied to various application areas such as remote sensing, network intrusion detection, and finance. In remote sensing, RX detector is employed for detecting anomalies in multispectal and hyperspectral images since they contain spectral signature (spectral channels) of the materials to be observed [11]. The RX detector essentially uses Mahalanobis distance to detect anomalies assuming they are distant from the image background. Here, it is assumed that anomalies are rare in the background and whole image is used as background where RX also assumes that background is Gaussian distributed. However, these assumptions over simplifies the real-life scenarios and RX detector usually fails in complex real-world data. Because of such limitation, many variants of the RX method are also proposed [2, 7, 37].

RX anomaly detector can assume whole image as background and then apply detection to each pixel based on this global statistics which is known as Global RX Detector. However, image can be heterogeneous which leads to violation of Gaussian distribution assumption of the RX for the global background. In that case, a sliding window (Figure 2.1) approach can be used where surrounding pixels for each pixel, beyond a safe guard region, are assumed as clutter to model the background statistics [7].

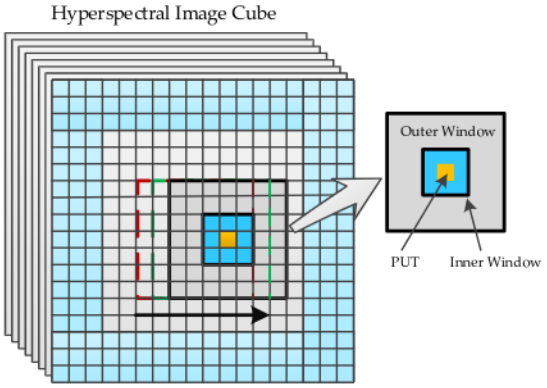


Figure 2.1 Sliding window approach for Local RX Detector [1]

2.1 RX Anomaly Detector

Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be a data in the matrix form where n is the size of the data and d is the dimension of the data. For the sake of simplicity, we assume that $\mathbf{X}$ is centered or mean subtracted (so $\mathbf{X}$ has zero mean vector). In RX, background is characterized by the covariance matrix $\mathbf{\Sigma}$ where RX simply calculates the distance of test data x_* towards the distribution mean using the squared Mahalanobis distance [9]:

$$D_{RX}(x_*) = x_*^\top (\mathbf{\Sigma})^{-1} x_* \tag{2.1}$$

$$\mathbf{\Sigma} = \frac{1}{n} \mathbf{X}^\top \mathbf{X} \tag{2.2}$$

Inverse of the covariance matrix $\mathbf{\Sigma}^{-1}$ can be efficiently computed for moderate data dimensions, i.e., $d < 250$ is typical for multispectral and hyperspectral images [16]. Note that, dimension of the data is significantly lower than the size of the data ($d \ll n$).

2.2 Kernel RX (KRX) Anomaly Detector

Computational efficiency and detection accuracy of the RX method is perfect if anomalies are rare in the scene and $\mathbf{X}$ is Gaussian distributed. However, more flexible models are required since real-life data is rarely Gaussian distributed. KRX is an elegant solution which kernelize the RX method by mapping nonlinear features $\mathbf{X} \in \mathbb{R}^{n \times d}$ into linear features $\mathbf{\Phi} \in \mathbb{R}^{n \times d_\mathcal{H}}$ in a Hilbert space $\mathcal{H}$ via mapping $\phi : x \in \mathbb{R}^d \rightarrow \phi(x) \in \mathbb{R}^{d_\mathcal{H}}$. Note that, now a linear solution can be applied in $\mathbf{\Phi} \in \mathbb{R}^{n \times d_\mathcal{H}}$ [10]. Finally, test data $x_* \in \mathbb{R}^d$ is mapped with $\phi(x_*) \in \mathbb{R}^{d_\mathcal{H}}$ and anomalousness is measured by the nonlinearized RX formula in Equation **2.1**:

$$D_{KRX}(x_*) = \phi(x_*)^\top (\mathbf{\Phi}^\top \mathbf{\Phi})^{-1} \phi(x_*) \tag{2.3}$$

However, $(\mathbf{\Phi}^\top \mathbf{\Phi}) \in \mathbb{R}^{d_\mathcal{H} \times d_\mathcal{H}}$ matrix is huge since $d_\mathcal{H}$ is a large number or infinite for Hilbert space $\mathcal{H}$ which makes it infeasible to execute the defined nonlinear detector in Equation **2.3**. Thanks to the Riesz theorem, the dot product between data samples in $\mathcal{H}$ becomes $K(x, x') = \langle \phi(x), \phi(x') \rangle \in \mathbb{R}$ which can be represented as kernel matrices [52]. Accordingly, in [10], authors proposed derivation of KRX based on kernelization of

Equation **2.3** using eigenvector decomposition where eigenvalues are assumed to be nonzero. Here we present novel derivation of KRX using the linear algebra, where RX formulation in infinite space $\mathcal{H}$ in the Equation **2.3** is kernelized. Let's define $\mathbf{I} \in \mathbb{R}^{d_{\mathcal{H}} \times d_{\mathcal{H}}}$ and then add it into the Equation **2.3** shown as below

$$D_{KRX}(x_*) = \phi(x_*)^\top(\mathbf{I})(\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\phi(x_*) \tag{2.4}$$

then expanding $\mathbf{I}$ as $(\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}(\mathbf{\Phi}^\top\mathbf{\Phi})$ in the Equation **2.4** leads to

$$D_{KRX}(x_*) = \phi(x_*)^\top(\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}(\mathbf{\Phi}^\top\mathbf{\Phi})(\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\phi(x_*) \tag{2.5}$$

Lets arrange Equation **2.5** as below

$$D_{KRX}(x_*) = \phi(x_*)^\top(\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top\mathbf{\Phi}(\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\phi(x_*) \tag{2.6}$$

Using the matrix inverse identity $\mathbf{B}(\mathbf{CB})^{-1} = (\mathbf{BC})^{-1}\mathbf{B}$ leads to

$$D_{KRX}(x_*) = \phi(x_*)^\top\mathbf{\Phi}^\top(\mathbf{\Phi}\mathbf{\Phi}^\top)^{-1}(\mathbf{\Phi}\mathbf{\Phi}^\top)^{-1}\mathbf{\Phi}\phi(x_*) \tag{2.7}$$

which can be rearranged as

$$D_{KRX}(x_*) = \phi(x_*)^\top\mathbf{\Phi}^\top(\mathbf{\Phi}\mathbf{\Phi}^\top\mathbf{\Phi}\mathbf{\Phi}^\top)^{-1}\mathbf{\Phi}\phi(x_*) \tag{2.8}$$

then we define $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^\top$ and $k_* = \mathbf{\Phi}\phi(x_*)$ which leads KRX as shown in Equation **2.9**:

$$D_{KRX}(x_*) = k_*^\top(\mathbf{KK})^{-1}k_* \tag{2.9}$$

where similarities between data point x_* and all the data points in $\mathbf{X}$ represented in $k_* = [K(x_*, x_1), \ldots, K(x_*, x_n)]^\top \in \mathbb{R}^n$ using the kernel function K. Finally, kernel matrix

$\mathbf{K} \in \mathbb{R}^{n \times n}$ can be defined as [15]:

$$\mathbf{K} = \mathbf{X}_{\Phi}^{\top}\mathbf{X}_{\Phi} = \begin{bmatrix} K(x_1, x_1) & \dots & K(x_1, x_n) \\ \vdots & \ddots & \vdots \\ K(x_n, x_1) & \dots & K(x_n, x_n) \end{bmatrix} \tag{2.10}$$

Note that, the KRX method also requires centered data as in the linear RX method. For KRX, this centering must be done in $\mathcal{H}$, which can be done with $\tilde{\mathbf{K}} \leftarrow \mathbf{HKH}$, where $H_{ij} = \delta_{ij} - \frac{1}{n}$, δ is the Kronecker delta ($\delta_{i,j} = 1$ if $i = j$ and zero otherwise).

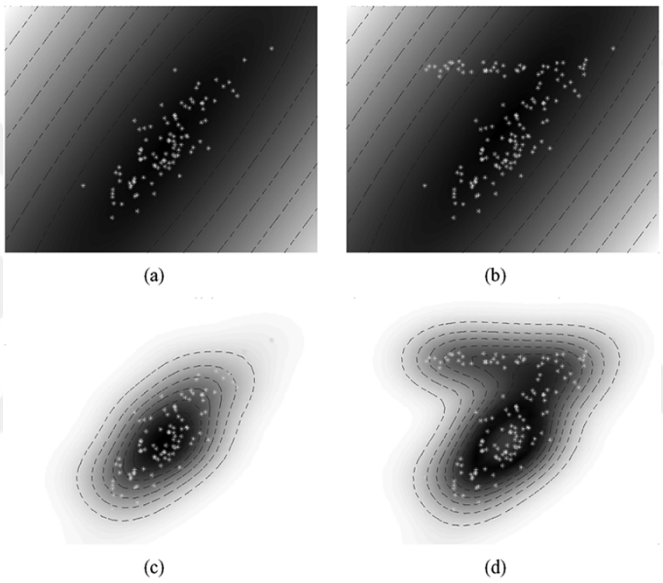


Figure 2.2 Data distributions of RX and KRX for 2 toy dataset.

Note that $(\mathbf{KK})^{-1}$ for large n (i.e., $n > 5000$) is computationally demanding since its space complexity is $O(n^2)$ and time complexity is $O(n^3)$. A simple approach to deal with this problem is to get random r samples ($r \ll n$) and use them as an input to KRX, known as sparse KRX or sub-sampling KRX, namely SRX, which is defined as

$$D_{SRX}(x_*) = k_{*:r}^{\top}(\hat{\mathbf{K}}\hat{\mathbf{K}})^{-1}k_{*:r} \tag{2.11}$$

where $\hat{\mathbf{X}} \in \mathbb{R}^{r \times d}$ is randomly sampled data, $k_{*:r} = [K(x_*, x_1), \dots, K(x_*, x_r)]^{\top} \in \mathbb{R}^r$ is the calculated similarity values between x_* and $\hat{\mathbf{X}}$, and $\hat{\mathbf{K}} \in \mathbb{R}^{r \times r}$ is the constructed kernel matrix. However, since r can not be large ($r < 5000$) most of the data is unused.

2.3 Randomized RX (RRX) Anomaly Detector

KRX can deal with nonlinear data, but it does not scale well for the large data as the inversion of kernel matrix $\mathbf{K}$ has cubic time complexity, where $\mathbf{K}$ is an $n \times n$ matrix as n is the size of data. In [16], RRX method is proposed which uses Random Fourier Features (RFF) to approximate KRX efficiently using the Bochner's theorem. Interested readers can see [14] for the details on Bochner's theorem and its relation to approximation of Radial Basis Function (RBF) kernel. Following the Bochner's theorem and RFF approach, n data points can be mapped as $\mathbf{Z} = [z_1 \dots z_n]^\top \in \mathbb{R}^{n \times 2D}$ to approximate the kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$. Subsequently, approximation of the kernel matrix $\mathbf{K}$ becomes $\hat{\mathbf{K}} \approx \mathbf{Z}\mathbf{Z}^\top$. However, in RRX, the RX (Equation **2.1**) is used with mapped $\mathbf{Z}$ instead of approximated $\hat{\mathbf{K}}$ to obtain computational efficiency. This method, RRX anomaly detector, is defined as:

$$D_{RRX}(x_*) = z_*^\top (\mathbf{G})^{-1} z_* \tag{2.12}$$

where $\mathbf{G} = \frac{1}{n}\mathbf{Z}^\top\mathbf{Z}$ and $z(x) = \frac{1}{\sqrt{2D}}[cos(w_1^\top x), sin(w_1^\top x), \dots, cos(sinw_D^\top x), sin(w_D^\top x)]^\top$ which approximates the RBF kernel. Note that, here, a 2D-dimensional randomized feature map $z : \mathbb{R}^d \to \mathbb{R}^{2D}$ is explicitly constructed and the data x_i is mapped into a higher dimensional space using the mapping $z(x_i)$. Finally, linear RX formula is used on the mapped data matrix $\mathbf{Z}$ where this data matrix has now size of $n \times 2D$. As in RX and KRX, data matrix $\mathbf{Z}$ is also centered as well. Final covariance matrix will be in the size of $2D \times 2D$. Therefore, unlike KRX, space and time complexity of the RRX becomes in the order D of instead of n. Typically, $d < D \ll n$ so RRX method becomes much faster than KRX method while RX method being the fastest. Computational complexity of the RX, RRX, and KRX are given in Table 2.1.

Table 2.1 Space and time complexity for RX, RRX, and KRX methods.

	Space		Time			
Method	T	$\mathbf{C}^{-1}$	T	$\mathbf{C}$	$\mathbf{C}^{-1}$	AD
RX	–	d^2	–	nd^2	d^3	nd^2
RRX	nD	D^2	ndD	nD^2	D^3	nD^2
KRX	n^2	n^2	n^2d	n^3	n^3	n^3

T is mapping operation of the data into another space. $\mathbf{C}$ is covariance/kernel matrix.

CHAPTER 3

PROPOSED METHOD

The RRX method provides significant execution efficiency compared to the KRX method. However, both RRX method and KRX method may fail at providing good detection accuracy if the covariance/kernel matrix becomes ill-conditioned, i.e., having high condition number. This happens if the largest eigenvalue is much larger than the smallest eigenvalue (ill-conditioned) or if there is a zero eigenvalue (not invertible) [53]. In such cases, inversion of these matrices becomes not possible or any operation with this matrix becomes numerically unstable. One solution is to regularize the RRX and the KRX methods as shown below [16, 47]:

$$D_{RRX}(x_*) = z_*^\top (\mathbf{G} + \lambda \mathbf{I})^{-1} z_* \tag{3.1}$$

$$D_{KRX}(x_*) = k_*^\top (\mathbf{KK} + \lambda \mathbf{I})^{-1} k_* \tag{3.2}$$

where $\mathbf{I}$ is the identity matrix and λ is a small positive scalar which practically determines the amount of applied regularization (larger λ means more regularization). Note that, $\mathbf{I}$ is a square identity matrix having the size of $\mathbf{G}$ for the RRX detector ($2D \times 2D$) and having the size of $\mathbf{K}$ for the KRX detector ($n \times n$).

This regularization approach is simple, computationally efficient, and improves the condition number of the matrix. Even non-invertible matrices become invertible once this regularization is applied. However, this regularization approach also causes change on the matrix leading to corruption on modelled background statistics. One way to deal with this problem is using low-rank approximation of the covariance / kernel matrix to be inverted. As stated in the literature, best low-rank approximation is obtained by SVD [51]. In low-rank approximation, low or zero eigenvalues are removed and r (as rank) eigenvectors and eigenvalues are preserved which improves the matrix condition number and also acts as a regularizer [47, 51]. Complexity of the matrix inversion is significantly reduced after matrix is decomposed by SVD. Yet, complexity of applying SVD is still cubic so no computational gain is obtained.

3.1 Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) factorizes the matrix **A** into **S**, **U**, and **V** where **S** is a diagonal matrix and **U** and **V** are orthonormal matrices as shown below:

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^{\top} \tag{3.3}$$

where the matrix **A** not need to be a square matrix as seen in Figure 3.1. Here, low rank matrix $\hat{\mathbf{A}}$ (of rank r) gets first r eigenvectors (from **U** and **V**) and eigenvalues (from **S**) as a good approximation to the data matrix **A**.

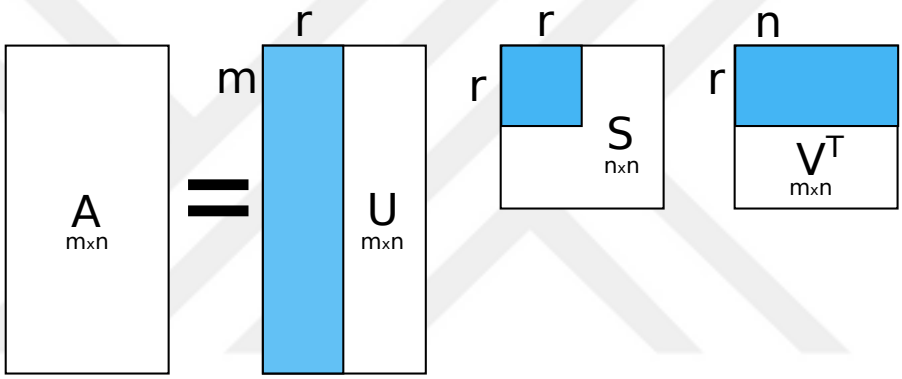


Figure 3.1 SVD and low-rank approximation for rectangular matrix **A**

Due to orthonormality of the **U** and **V**, inverse of the **A** is defined as $\mathbf{A}^{-1} = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^{\top}$ which can be executed efficiently since **S** is a diagonal matrix and its inverse can be calculated by the taking inverse of the each scalar numbers in the diagonal.

If **A** is positive definite and symmetrix matrix then SVD can be defined as $\mathbf{A} = \mathbf{V}\mathbf{S}\mathbf{V}^{\top}$ (Figure 3.2) where inverse of **A** is now defined as $\mathbf{A}^{-1} = \mathbf{V}\mathbf{S}^{-1}\mathbf{V}^{\top}$.

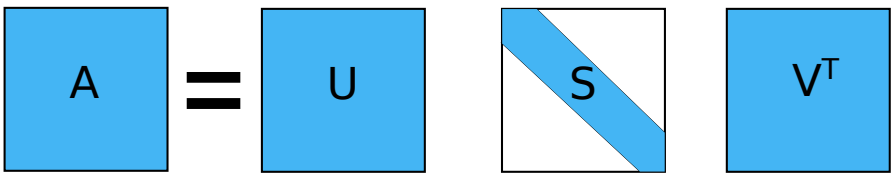


Figure 3.2 SVD for positive definite symmetrix matrix **A**

3.2 Proposed Anomaly Detection Method

Let decompose the symmetric and positive definite covariance matrix in the RRX as $\mathbf{G} = \mathbf{V}\mathbf{S}\mathbf{V}^\top$ using SVD. Here, $\mathbf{V}$ is a square matrix containing eigenvectors and $\mathbf{S}$ is a diagonal matrix with eigenvalues on the diagonal (largest to smallest, largest being at top left cell). All the matrices ($\mathbf{G}$, $\mathbf{V}$, and $\mathbf{S}$) are size of $2D \times 2D$ (Figure 3.2).

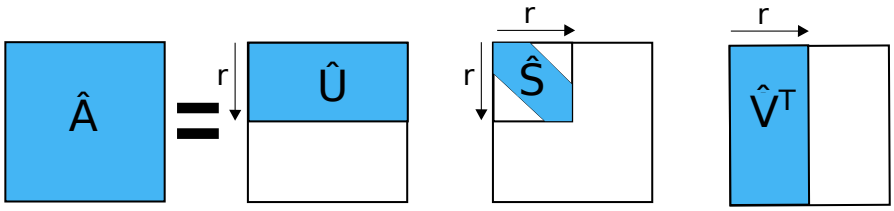


Figure 3.3 SVD and Low Rank approximation for the matrix $\mathbf{G}$

In the low-rank approximation of the matrix $\mathbf{G}$ (Figure 3.3), $\hat{\mathbf{V}}$ is $2D \times r$ and $\hat{\mathbf{S}}$ is $r \times r$. Note that, randomized SVD is used [51] for obtaining computational efficiency. Here $\hat{\mathbf{S}}$ is the top-left $r \times r$ part of the $\mathbf{S}$ corresponding to the largest r eigenvalues and $\hat{\mathbf{V}}$ is the leftmost r columns of the $\mathbf{V}$ which corresponds to these eigenvalues (Figure 3.3). Therefore, most of the information in $\mathbf{G}$ is preserved in $\hat{\mathbf{G}}$ while low and zero eigenvalues are eliminated leading to better conditioned $\hat{\mathbf{G}}$ matrix. Using the low-rank definition $\mathbf{G} \approx \hat{\mathbf{G}} = \hat{\mathbf{V}}\hat{\mathbf{S}}\hat{\mathbf{V}}^\top$, detector $D_{RRX}(x_*)$ becomes $D_{RLR-RX}(x_*)$ as below:

$$D_{RLR-RX}(x_*) = z_*^\top (\hat{\mathbf{G}})^{-1} z_* \tag{3.4}$$

Then using the low-rank definition, Equation **3.4** becomes as below:

$$D_{RLR-RX}(x_*) = z_*^\top (\hat{\mathbf{V}}\hat{\mathbf{S}}\hat{\mathbf{V}}^\top)^{-1} z_* \tag{3.5}$$

Using the inverse of the low-rank definition $\hat{\mathbf{G}}^{-1} = \hat{\mathbf{V}}\hat{\mathbf{S}}^{-1}\hat{\mathbf{V}}^\top$ we obtain:

$$D_{RLR-RX}(x_*) = z_*^\top (\hat{\mathbf{V}}\hat{\mathbf{S}}^{-1}\hat{\mathbf{V}}^\top) z_* = z_*^\top \hat{\mathbf{V}} (\hat{\mathbf{S}})^{-1} \hat{\mathbf{V}}^\top z_* \tag{3.6}$$

Using the mapping $v = \hat{\mathbf{V}}^\top z_*$, Equation **3.6** finally becomes

$$D_{RLR-RX}(x_*) = v_*^\top (\hat{\mathbf{S}})^{-1} v_* \tag{3.7}$$

Here v is $r \times 1$ vector which is mapped from d dimensional space (x_*) to $2D$ dimensional space (z_*) and finally mapped to r dimensional space (v_*). Since $\hat{\mathbf{S}}$ is a diagonal matrix, its inverse is just the inverse of the eigenvalues in the diagonal which is done practically with just r scalar division operation. Most computational part in RLR-RX is decomposition of $\hat{\mathbf{G}}$ using SVD which is done in a much faster way using randomized SVD algorithm [51]. Computation of $v_* = \hat{\mathbf{V}}^\top z_*$ is also fast so proposed RLR-RX detector becomes faster than RRX detector and much faster than KRX detector.

In the literature, in order to deal with large data or large matrix computations, several randomization approaches are proposed [51]. Such methods provides significants computational advantages while loosing accuracy with a low probability. Although result is not deterministic and there may be small deviations from expected result, in practice these differences can be ignored for some applications. In [51], a randomized approach for SVD is proposed to reduce the cubic computational complexity of SVD. In the below algorithm, pseudocode of the randomized SVD [51] method is given.

Algorithm 1 Randomized Singular Value Decomposition (Randomized SVD)

Inputs: $\mathbf{A}$, r ▷ $\mathbf{A}$: input matrix, r: desired rank
$[h,w] \leftarrow$ size($\mathbf{A}$) ▷ h: rowCount($\mathbf{A}$), w: columnCount($\mathbf{A}$)
$p \leftarrow min(2r, w)$ ▷ p is at most twice of rank r
$\mathbf{X} \leftarrow randn(w, p)$ ▷ $\mathbf{X}$ matrix is $w \times p$ with random values
$\mathbf{Y} \leftarrow \mathbf{AX}$ ▷ $\mathbf{Y}$ marix is $w \times p$ with random values
$\mathbf{W}_1 \leftarrow orth(\mathbf{Y})$ ▷ Orthogonalized $\mathbf{Y}$ is assigned to W_1
$B \leftarrow \mathbf{W}_1^\top \mathbf{A}$ ▷ $\mathbf{B}$ is formed by the inverse of W_1
$[\mathbf{W}_2, \mathbf{S}, \mathbf{V}] \leftarrow svd(\mathbf{B},$ 'econ') ▷ economy-size decomposition of $\mathbf{B}^{-1}$
$\mathbf{U} \leftarrow \mathbf{W}_1\mathbf{W}_2$ ▷ $\mathbf{U}$ is matrix with size $w \times p$
$r \leftarrow min(r,$ column count of $\mathbf{U})$
$\mathbf{U} \leftarrow \mathbf{U}_{*,1:r}$ ▷ Left r-columns of $\mathbf{U}$
$\mathbf{S} \leftarrow \mathbf{S}_{1:r,1:r}$ ▷ Top-left $r \times r$ part of $\mathbf{S}$
$\mathbf{V} \leftarrow \mathbf{V}_{*,1:r}$ ▷ Left r-columns of $\mathbf{V}$
Return $\mathbf{U}$, $\mathbf{S}$, $\mathbf{V}$

Note that, once randomized SVD is used to decompose matrix $\mathbf{S}$, inverse of $\mathbf{S}$ become computationally efficient since randomized SVD has computational complexity lower than $O(n^3)$ and computational complexity of $\mathbf{S}^{-1}$ is just $O(r)$.

Resulting algorithm of the proposed RLR-RX detector is given in the Algorithm 2 where $\mathbf{X}$ is input data, D is number of bases in random Fourier Features, r is the rank for low-approximation using randomized SVD, and τ is distance threshold such that pixels with distances larger than the τ are marked as anomalies. In RLR-RX algorithm, $\mathbf{X}$ is mapped to higher dimension as $\mathbf{Z}$ using random Fourier Features which approximates the KRX. Then, linear RX is used with the covariance matrix $\mathbf{G}$ that is obtained using the mapped data $\mathbf{Z}$. Finally, low-rank approximation of $\mathbf{G}$ is obtained using randomized SVD then Equation **3.7** is used for applying RLR-RX anomaly detection.

Algorithm 2 Randomized Low-Rank RX (RLR-RX) Anomaly Detector

Inputs: $\mathbf{X}$, D, r, τ
$d \leftarrow$ columnSize($\mathbf{X}$)
$n \leftarrow$ rowSize($\mathbf{X}$)
$\mathbf{W} \in \mathbb{R}^{2D \times d} \leftarrow$ create Random Fourier Features
$\mathbf{Z} \leftarrow \mathbf{X}\mathbf{W}^\top$
$\mathbf{Z} \leftarrow \mathbf{Z} - \mathbf{Z}_\mu$ ▷ Center the mapped data
$\mathbf{G} \leftarrow \frac{1}{n}\mathbf{Z}^\top\mathbf{Z}$ ▷ Calculate the covariance matrix
$\hat{\mathbf{U}}, \hat{\mathbf{S}}, \hat{\mathbf{V}} \leftarrow$ RandomizedSVD($\mathbf{G}$,r)
$v_* \leftarrow \hat{\mathbf{V}}^\top z_*$ ▷ Map D-dimensional space to r-dimensional space
$\delta_{RLR-RX} \leftarrow v_*^\top(\hat{\mathbf{S}})^{-1}v_*$ ▷ Find distances using Equation **3.7**
Return $\delta_{RLR-RX} > \tau$ ▷ Apply thresholding on distances

Computational complexity of the proposed RLR-RX method compared to RX, KRX, and RRX methods are given in Table 3.1. As seen in Table 3.1, RLR-RX provides same space complexity with RRX method while has better time complexity compared to RRX method. Note that, actual execution time of the RLR-RX can be further reduced using efficient code optimization techniques, especially in memory allocations.

Table 3.1 Space and time complexity for RX, RRX, RLR-RX, and KRX methods.

	Space		Time			
Method	T	$\mathbf{C}^{-1}$	T	$\mathbf{C}$	$\mathbf{C}^{-1}$	AD
RX	–	d^2	–	nd^2	d^3	nd^2
RLR-RX	nD	D^2	ndD	nr^2	r^3	nr^2
RRX	nD	D^2	ndD	nD^2	D^3	nD^2
KRX	n^2	n^2	n^2d	n^3	n^3	n^3

T is mapping operation of the data into another space. $\mathbf{C}$ is covariance/kernel matrix.

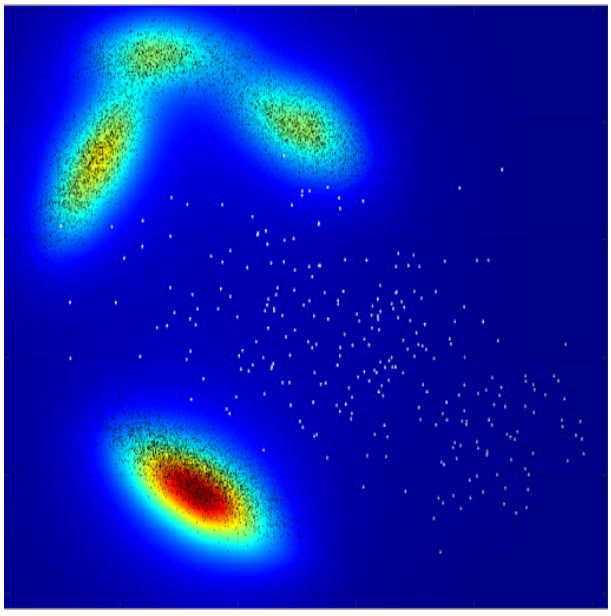
CHAPTER 4

RESULTS

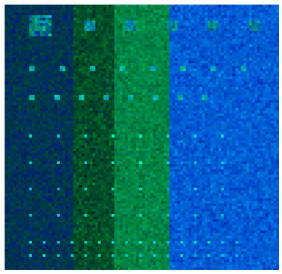
This section compares the detection performance and the execution time efficiency of the RX, KRX, RRX, and proposed RLR-RX methods, in both synthetically generated data and real-world Airport-Beach-Urban (ABU) dataset. All the experiments are executed on Intel i7 4700HQ 2.4 GHz PC with 16 GB memory.

4.1 Synthetic Dataset

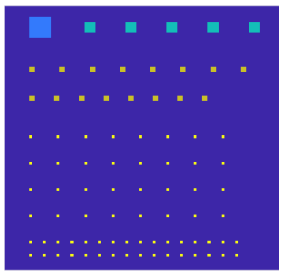
Synthetic image with size of 100×100 and 2 spectral channels (Figure 4.1b) with 2.72% of fused anomalies (Figure 4.1c) is crated to analyze detection performance and execution time efficiency of the RX, KRX, RRX, and RLR-RX detectors. In Figure 4.1b, first and the second channels of the data is assigned to the green and blue channels of the generated image. Synthetic image with only 2 channel is created for the sake of simplicity and to be able to visualize the generated data. Created data is non-Gaussian as the distribution of data and the data points are shown in Figure 4.1a.



(a) Data distribution



(b) Synthetic data



(c) Fused anomalies

Figure 4.1 Synthetic dataset with non-Gaussian distribution

In Figure 4.2, execution time of the RX, KRX, RRX, and proposed RLR-RX detectors are shown with respect to data size. In Figure 4.2a, execution time of the KRX detector is measured up to 2500 data points and shown in log-scale since computational complexity of the KRX detector is cubic. RX, RRX, and RLR-RX are fast detectors since RX is linear and RRX and RLR-RX methods are efficient approximations of the KRX method since they use RX on mapped space. In Figure 4.2b, KRX is excluded, and execution times are shown in linear scale to compare the execution time of the fast detectors. Although RX is extremely fast it can only deal with Gaussian distributed data, which is rare in complex real world scenarios. However, use of KRX becomes impractical as data size gets larger, i.e., for $n > 5000$. As seen in Figure 4.2, RRX and RLR-RX detectors are fast while they can deal with non-Gaussian data as KRX does.

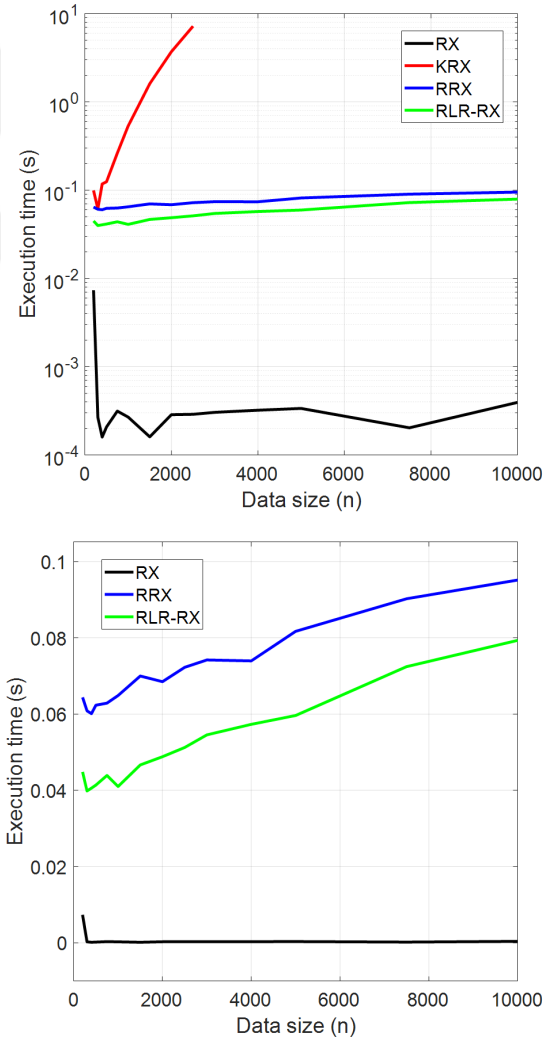


Figure 4.2 Execution time with respect to data size on synthetic data

In Table 4.1, detection accuracy and execution time of the RX, KRX, RRX, and RLR-RX detectors are shown with respect to data size on synthetically generated data. Detection accuracies are given as Area Under the Curve (AUC) values. As seen in Table 4.1, execution time of proposed RLR-RX method is always faster than RRX method while detection performance is being better or similar. The KRX method is the slowest one due to its cubic complexity on data size where RX method is the fastest one as it has cubic complexity on number of data dimensions, i.e., $d < 250$. Note that, since the dimension/channels of synthetically generated data is only 2, it becomes much faster compared to RRX and RLR-RX whereas for hyperspectral data with many channels (generally around hundreds) execution time difference becomes lower.

Table 4.1 Detection accuracy and execution time on synthetic data

n (Data Size)		Detection Performance (AUC)				Execution Time (Seconds)				RLR-RX is x times faster than KRX	RLR-RX is % times faster than RRX
		RX	KRX	RRX	RLR-RX	RX	KRX	RRX	RLR-RX		
Data Size	200	0.6963	**0.9921**	0.9907	0.9864	0.0007	0.0500	0.0867	**0.0480**	1.04	80.64%
	300	0.7246	**0.9859**	0.9856	0.9838	0.0004	0.0824	0.0827	**0.0495**	1.66	67.05%
	400	0.7052	**0.9797**	0.9782	0.9775	0.0004	0.1267	0.0861	**0.0510**	2.48	68.71%
	500	0.7265	0.9895	**0.9898**	0.9893	0.0004	0.1804	0.0857	**0.0508**	3.55	68.73%
	750	0.7145	**0.9884**	0.9850	0.9875	0.0004	0.3744	0.0847	**0.0525**	7.14	61.41%
	1000	0.7410	**0.9907**	0.9902	0.9896	0.0004	0.6449	0.0877	**0.0534**	12.09	64.38%
	1500	0.7194	**0.9879**	0.9857	0.9857	0.0004	1.5920	0.0901	**0.0569**	27.98	58.41%
	2000	0.7273	**0.9877**	0.9859	0.9861	0.0004	3.8498	0.0939	**0.0589**	65.39	59.54%
	2500	0.7148	**0.9898**	0.9874	0.9880	0.0004	7.6288	0.0984	**0.0619**	123.27	59.00%
	3000	0.7200		0.9881	**0.9881**	0.0004		0.0984	**0.0659**		49.27%
	4000	0.7207		0.9878	**0.9885**	0.0004		0.1032	**0.0702**		46.94%
	5000	0.7183		0.9884	**0.9893**	0.0004		0.1107	**0.0747**		48.09%
	7500	0.7267		**0.9889**	0.9882	0.0004		0.1224	**0.0865**		41.60%
	10000	0.7254		**0.9895**	0.9889	0.0005		0.1317	**0.1019**		29.25%

4.2 Real-world Dataset

In their anomaly detection study, Kang et al. created Airport-Beach-Urban (ABU) test dataset which they also made publicly available for other researchers. This dataset has airport images (size of 100×100, Figure 4.3), beach images (size of 100×100 and 150×150, Figure 4.4), and urban images (size of 100×100, Figure 4.5) which are manually extracted from large images that are acquired from the Airborne Visible / Infrared Imaging Spectrometer (AVIRIS). Numbers of hyperspectral channels are between 102 and 205. ABU dataset contains 5 airport scenes (from Los Angeles and Gulfport), 4 beach scenes (from Cat Island, San Diego, Bay Champagne, and Pavia), and 6 urban scenes (from Texas Coast, Gainesville, and Los Angeles).

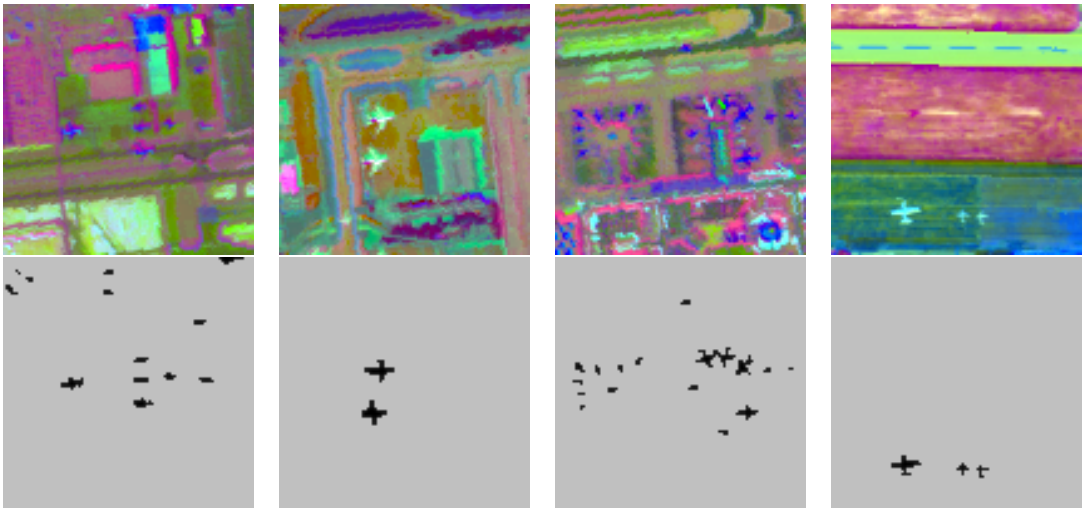


Figure 4.3 Airport scenes from ABU dataset

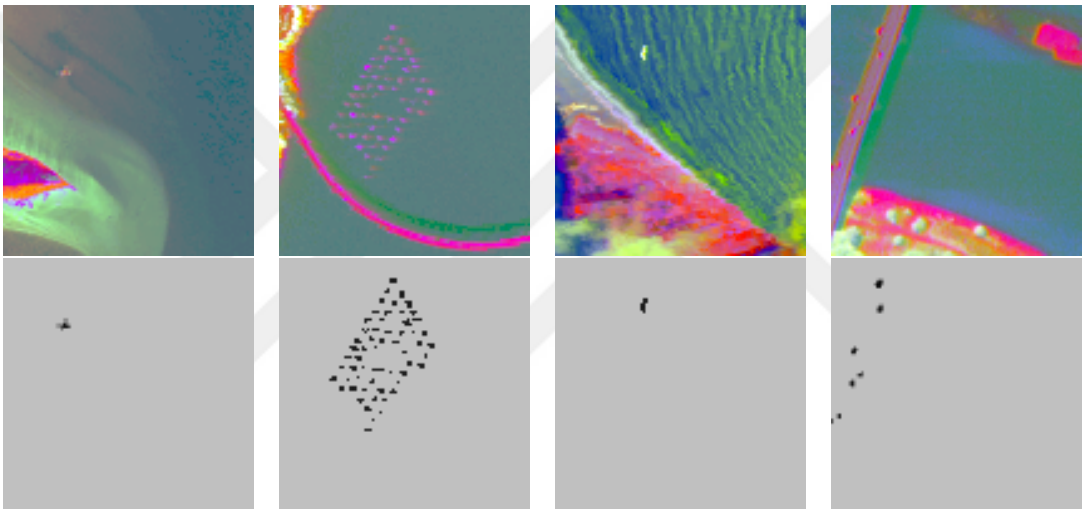


Figure 4.4 Beach scenes from ABU dataset

Compared to beach scences (Figure 4.4), airport (Figure 4.3) and urban (Figure 4.5) scence have more complex backgrounds.

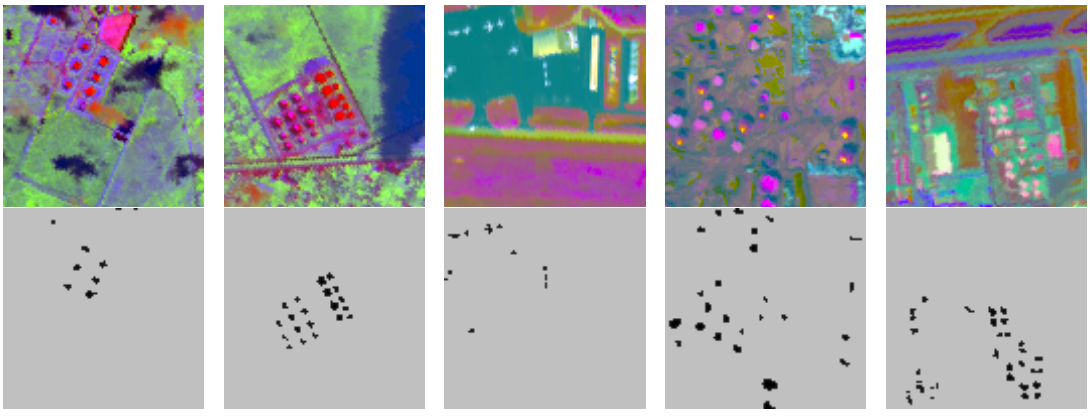


Figure 4.5 Urban scenes from ABU dataset

In Table 4.2, detection performance (as AUC) and execution time (in seconds) are given for all the images in the ABU dataset for the RX, KRX, RRX, and proposed RLR-RX methods. As a nonlinear detector, RLR-RX is the fastest one among the nonlinear detectors and only 4 times slower compared to linear RX (see Table 4.2). Also, for the most of the images, and for each background class (airport, beach, urban), and in general RLR-RX gives the best detection accuracy (highest AUC values).

Table 4.2 Detection performance comparison for the ABU dataset

	RX d=2		KRX r=3000		RRX D=256		RLR-RX D=256, r=128	
	AUC	**Time (secs)**	**AUC**	**Time (secs)**	**AUC**	**Time (secs)**	**AUC**	**Time (secs)**
Airport-1, 100x100x205	0.8221	0.063150	0.9311	12.045569	0.9224	0.358812	0.9265	0.277196
Airport-2, 100x100x205	0.8404	0.064029	0.9732	12.109578	0.9628	0.358880	0.9716	0.278251
Airport-3, 100x100x205	0.9288	0.062950	0.9456	12.622220	0.9537	0.350079	0.9529	0.285013
Airport-4, 100x100x191	0.9526	0.059603	0.9857	12.240329	0.9743	0.347297	0.9850	0.284930
All Airport Images	**0.8860**	**0.062433**	**0.9589**	**12.254424**	**0.9533**	**0.353767**	**0.9590**	**0.281348**
Beach-1, 150x150x188	0.9807	0.102275	0.5074	15.237153	0.9843	0.574739	0.9755	0.416774
Beach-2, 100x100x193	0.9106	0.055894	0.5532	16.299159	0.9229	0.363845	0.9296	0.289537
Beach-3, 100x100x188	0.9998	0.055936	0.9985	11.864610	0.9882	0.347647	0.9986	0.267836
Beach-4, 150x150x102	0.9538	0.040958	0.7847	16.463607	0.9623	0.517094	0.9664	0.378158
All Beach Images	**0.9612**	**0.063766**	**0.7110**	**14.966132**	**0.9644**	**0.450831**	**0.9675**	**0.338076**
Urban-1, 100x100x204	0.9907	0.063172	0.9889	12.374027	0.9846	0.352361	0.9945	0.267262
Urban-2, 100x100x207	0.9946	0.066138	0.9491	11.881633	0.9921	0.341110	0.9968	0.270310
Urban-3, 100x100x191	0.9513	0.056752	0.9634	11.863191	0.9560	0.338222	0.9634	0.266350
Urban-4, 100x100x205	0.9887	0.061322	0.9376	12.299675	0.9810	0.345810	0.9853	0.267964
Urban-5, 100x100x205	0.9692	0.061324	0.9761	12.131038	0.9731	0.346614	0.9734	0.270846
All Urban Images	**0.9789**	**0.061742**	**0.9630**	**12.109913**	**0.9774**	**0.344823**	**0.9827**	**0.268546**
All ABU Dataset	***0.9449***	***0.062577***	***0.8842***	***13.033215***	***0.9660***	***0.380193***	***0.9707***	***0.293879***

Overall performances of the RX, KRX, RRX, and RLR-RX detectors for all the scenes are given in the Figure 4.6 where all nonlinear detectors have similar detection performances (see Table 4.2) where RLR-RX provides the highest performance.

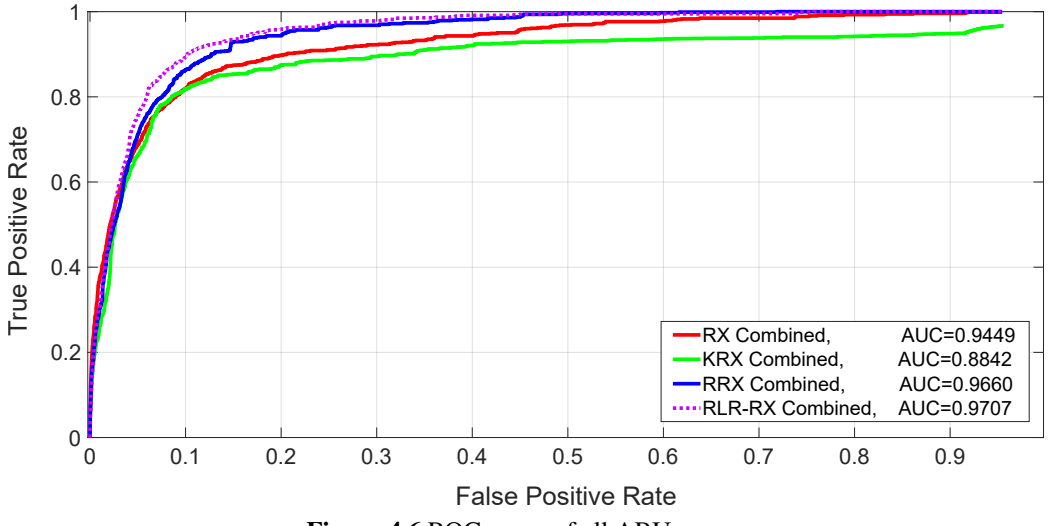


Figure 4.6 ROC curve of all ABU scenes

Condition Numbers of each matrix to be inverted in RX, KRX, RRX, and LRL-RX methods versus the obtained detection accuracy is shown in Figure 4.7 on ABU dataset. In this graph, detection accuracy is shown using AUC value and numerical stability of the detector is shown using condition number. Here condition number shows how stable the numerical system in the detector where lower value means higher numerical stability. Note that, best condition number is 1 since eigenvalues are same and data is distributed equally in all principal axes, i.e. data has circular distribution. If condition number is infinity that means there is at least one zero eigenvalue and such matrix is indeed not invertible. As seen in Figure 4.7, KRX has unexpectedly low detection performances for 2 cases. One can also see that condition number of KRX is worse compared to the other detectors. Proposed RLR-RX detector has best condition number versus detection accuracy characteristics which is followed by the RRX method. This is expected since RLR-RX is kind of extended version of RRX as both methods uses random Fourier features to approximate KRX detector where RLR-RX has stronger regularization due to employed low-rank approximation using SVD. RRX is followed by RX if condition number is considered but detection accuracy of RX is lower for the scenes violating Gaussian background assumption, i.e. for nonlinear data. Therefore, Figure 4.7 shows superiority of RLR-RX compared to other methods in terms of numerical stability and detection accuracy.

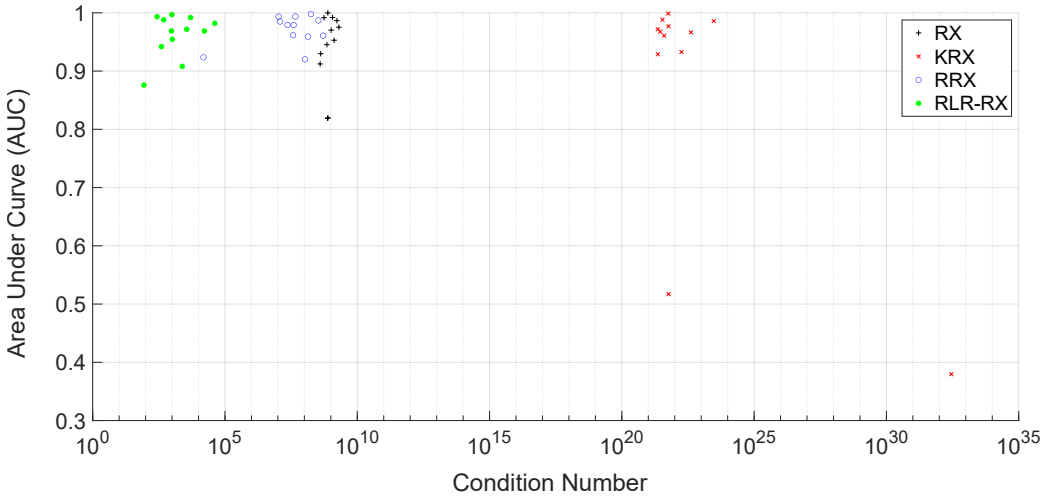


Figure 4.7 Condition Number versus Detection Accuracy

4.2.1 Detection Performance for Airport Scenes

In Figure 4.8, Figure 4.9, Figure 4.10, and Figure 4.11 detection performances of the RX, KRX, RRX, and RLR-RX anomaly detection methods are shown for the airport scenes in the ABU dataset.

As seen in Figure 4.8, KRX method provides the best performance for airport-1 scene where RX method performs worse. Here, detection performance of the methods are given as AUC values of the obtained ROC curves. Nonlinear detectors provide 90% detection rate with 20% false alarm rate and 80% detection rate with 10% false alarm rate where linear RX detector provides much lower detection rate for the same false alarm rates. Since background of the airport-1 scene is not perfectly Gaussian distributed, RX method gives relatively lower performance compared to nonlinear methods, namely KRX and its approximations RRX and RLR-RX. Both RRX and RLR-RX use Random Fourier Features, but RLR-RX performs better since it gets rid of eigenvectors corresponding to low or zero eigenvalues. By doing this, condition matrix of covariance matrix become better and numerical stability is increased. This acts as a regularization thus RLR-RX improves the performance of the RRX. Note that, RLR-RX provides the fastest execution time among the nonliner detectors.

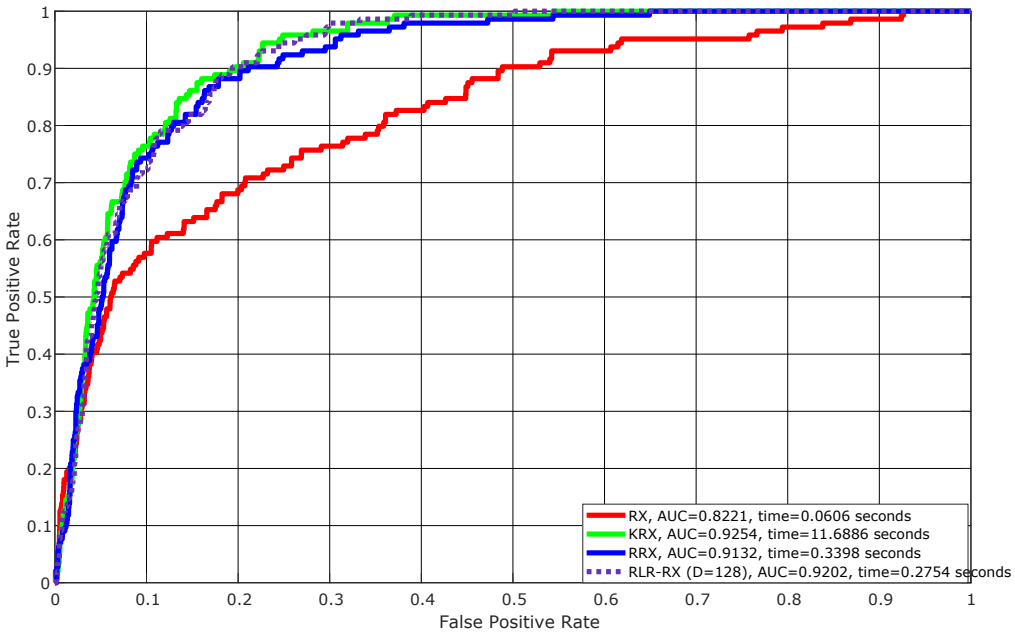


Figure 4.8 ROC curve of airport-1 scene from ABU dataset

As seen in Figure 4.9, for airport-2 scene, RLR-RX provides the best performance where KRX performs similarly, RRX being slightly lower, and RX being worse. Although RLR-RX aims to approximate KRX, in that scene it also provides a minimal performance improvement while being significantly faster.

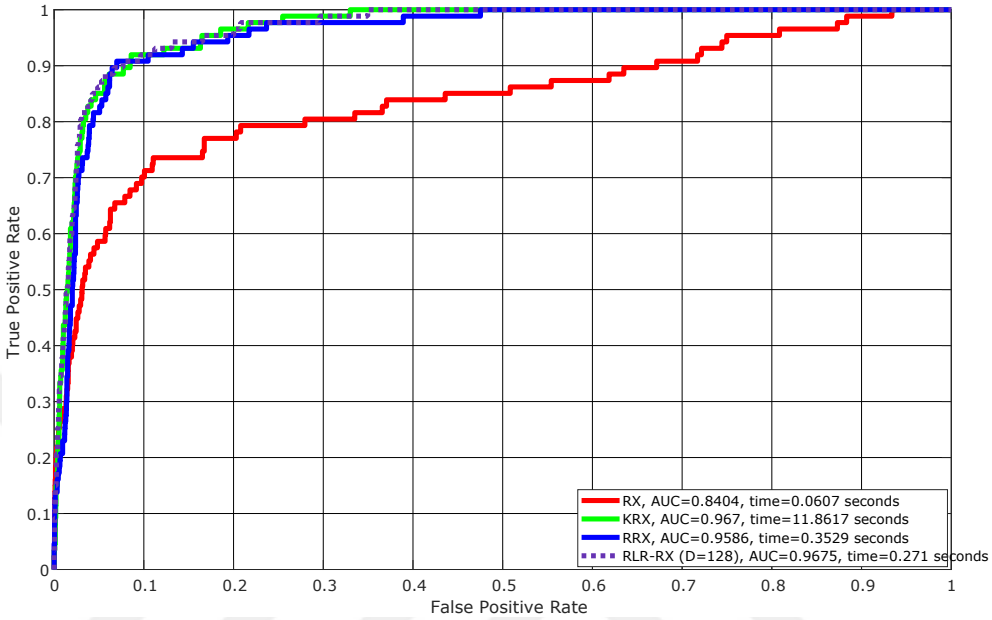


Figure 4.9 ROC curve of airport-2 scene from ABU dataset

As seen in Figure 4.10, for airport-3 scene, RLR-RX provides the fastest execution time among the nonliner detectors while providing slightly better detection accuracy.

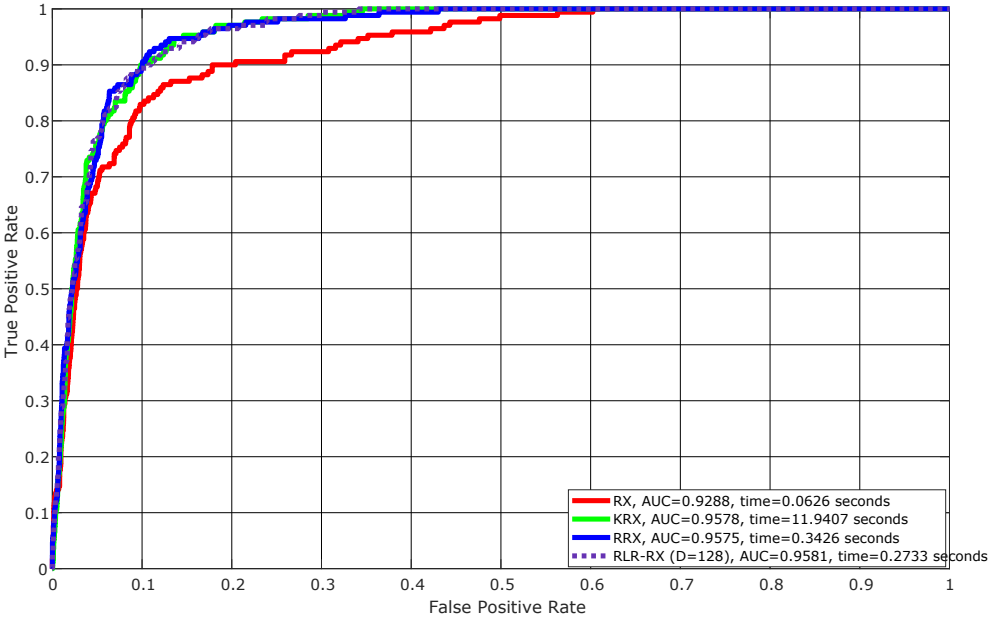


Figure 4.10 ROC curve of airport-3 scene from ABU dataset

As seen in Figure 4.11, RLR-RX and KRX methods provides the best performance for airport-4 scene where RRX is slightly worse and RX method performs worse. Note that, RLR-RX is still the fastest one among the nonlinear methods.

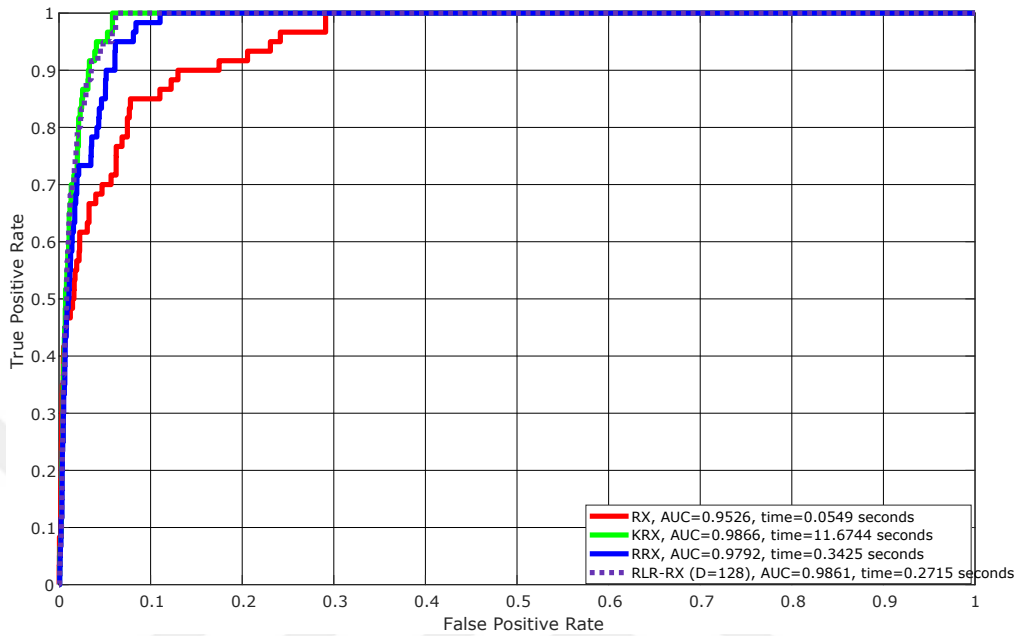


Figure 4.11 ROC curve of airport-4 scene from ABU dataset

Overall performances of the RX, KRX, RRX, and RLR-RX detectors for airport scenes are given in the Figure 4.12 where all nonlinear detectors have similar detection performances (see Table 4.2).

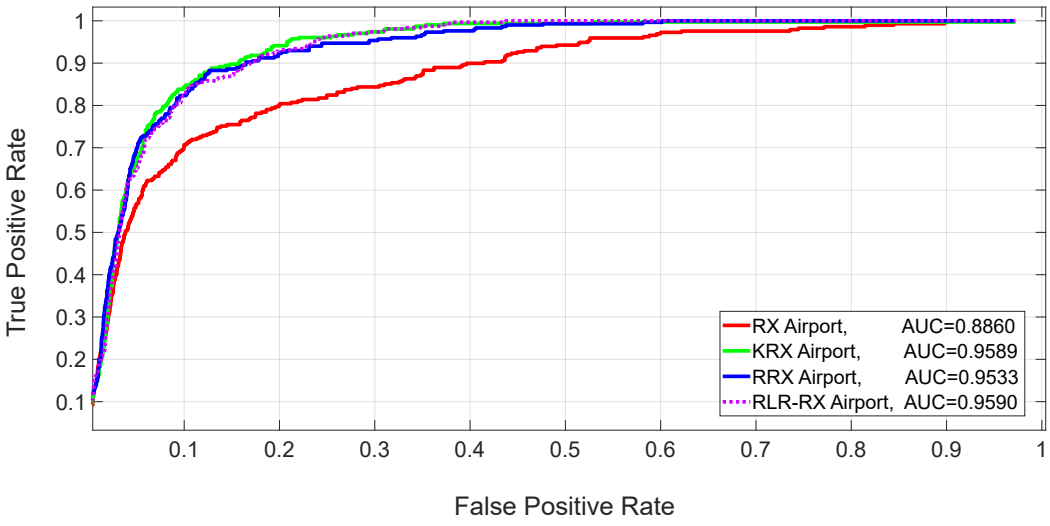


Figure 4.12 ROC curve of ABU airport scenes

4.2.2 Detection Performance for Beach Scenes

In Figure 4.13, Figure 4.14, Figure 4.15, and Figure 4.16 detection performances of RX, KRX, RRX, and RLR-RX methods are shown for the beach scenes. For beach-1 scene (Figure 4.13), RX method provides the best performance while RRX and RLR-RX provide similar performance with RX. In beach-1 (Figure 4.13) and beach-2 scene (Figure 4.14), KRX performs poorly because kernel matrix becomes ill-conditioned in mapped high dimensional space which shows importance of regularization.

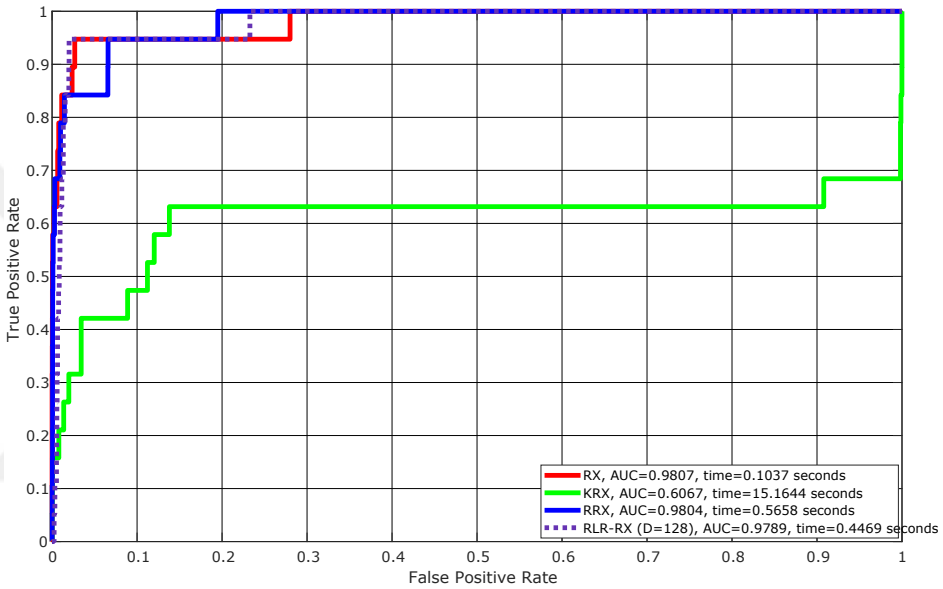


Figure 4.13 ROC curve of beach-1 scene from ABU dataset

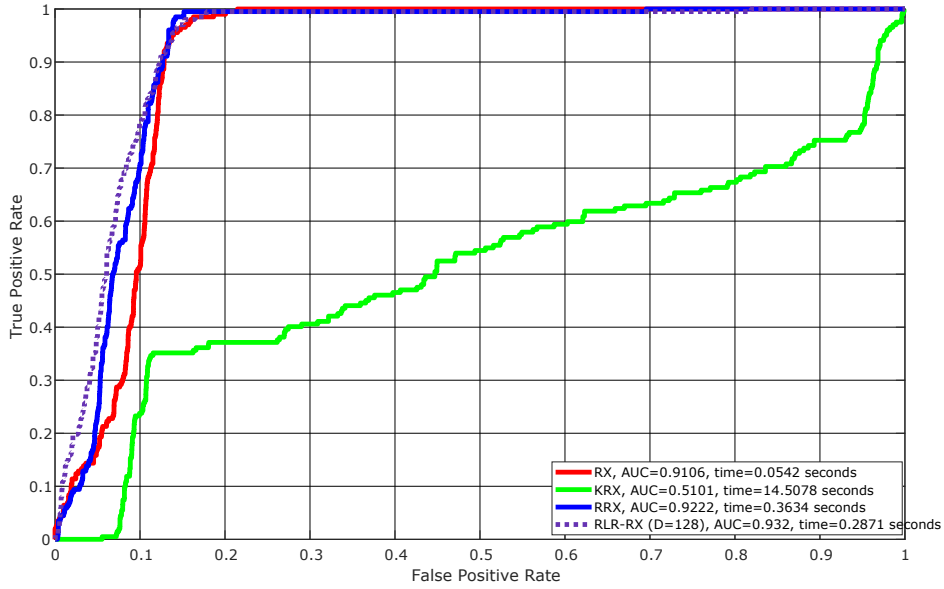


Figure 4.14 ROC curve of beach-2 scene from ABU dataset

Beach-3 scene (Figure 4.15) is very simple so all method provides almost perfect detection performance. So, for this image RX method shines.

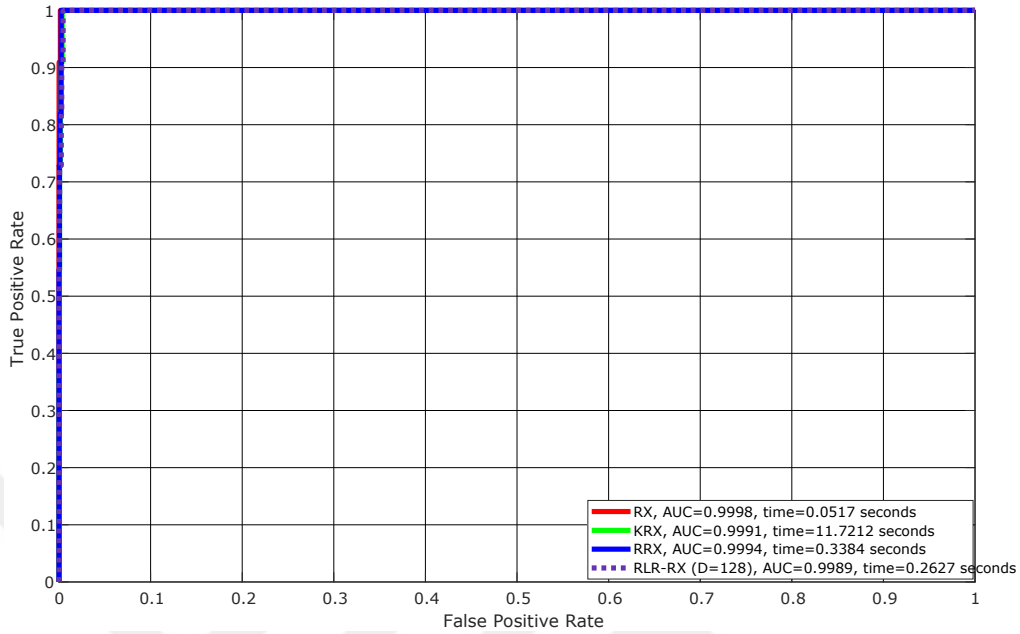


Figure 4.15 ROC curve of beach-3 scene from ABU dataset

For beach-4 scene (Figure 4.16), RLR-RX and RRX methods provide best performance. Here, KRX method performs worse although RLR-RX and RRX approximate the KRX. This is due to kernel matrix being ill-conditioned in KRX.

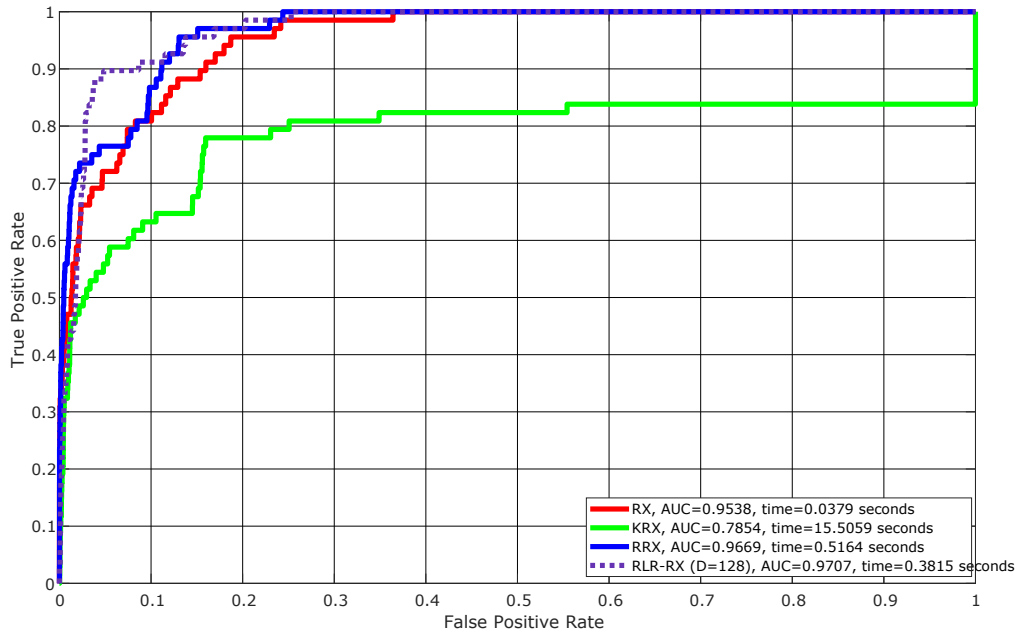


Figure 4.16 ROC curve of beach-4 scene from ABU dataset

Overall detector performances for airport scenes are given in the Figure 4.17 where RRX and RLR-RX detectors have similar detection performances while KRX detector has lowest performance (see Table 4.2). In beach scenes, KRX detector performs even worse then RX detector. The RX detector provide good detection performance since Gaussian background assumption is sufficiently valid. The KRX detector has worse detection performance in some beach scenes since condition number of kernel matrix in KRX is high which causes numerical problems (see Figure 4.13, 4.14, and 4.16).

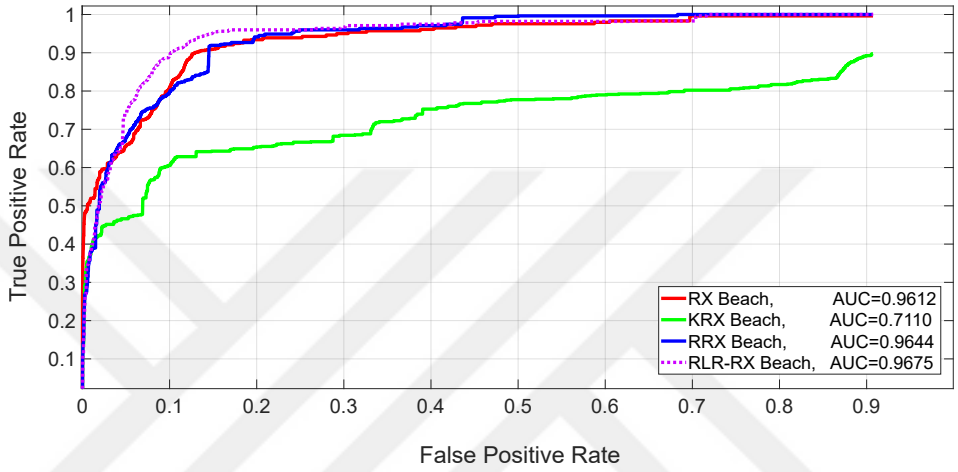


Figure 4.17 ROC curve of ABU beach scenes

4.2.3 Detection Performance for Urban Scenes

In Figure 4.18-4.22 detector performances are shown for the urban scenes. RLR-RX method provides the best detection performance for urban-1 scene (Figure 4.18).

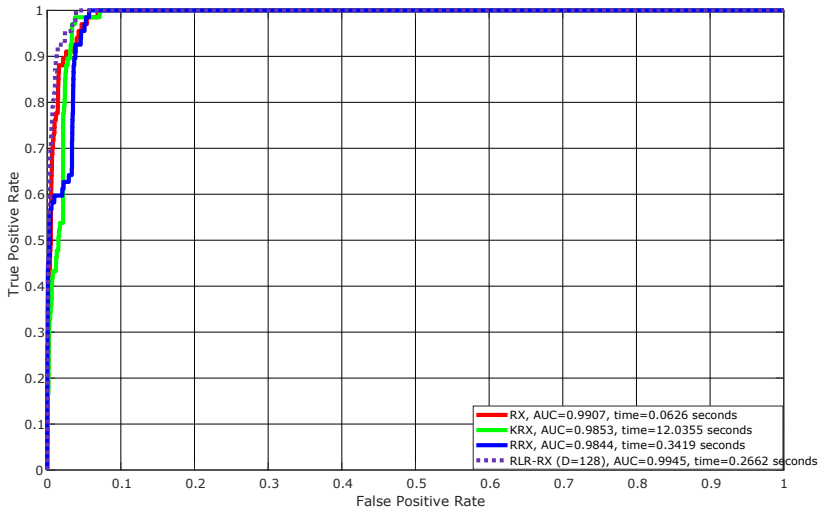


Figure 4.18 ROC curve of urban-1 scene from ABU dataset

As seen in Figure 4.19, RLR-RX and RX methods provides the best performance for urban-2 scene where KRX method does not perform good enough. This is also due to condition number of the kernel matrix in the KRX detector.

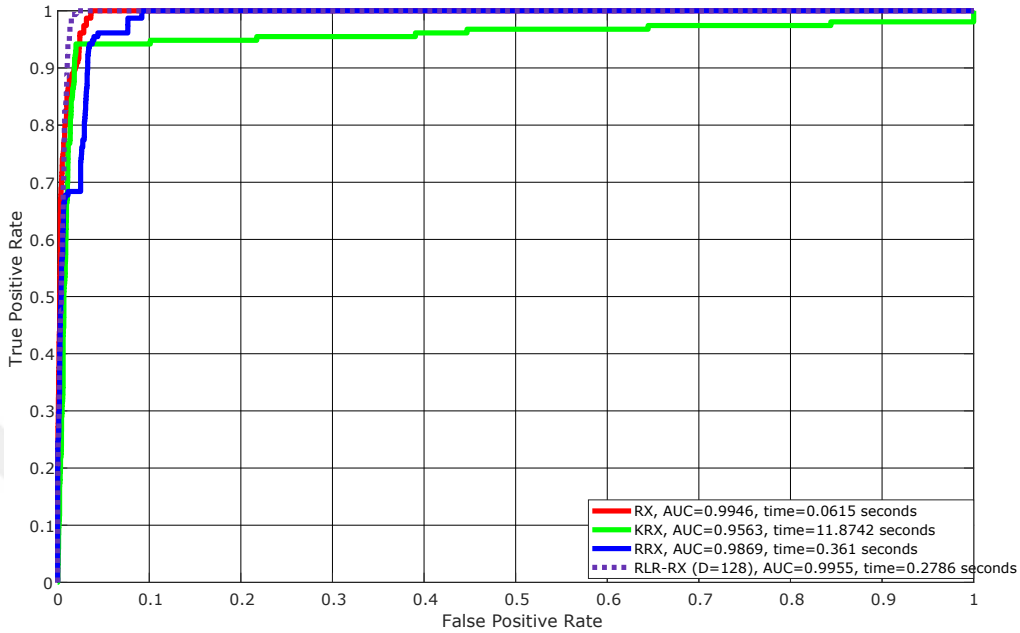


Figure 4.19 ROC curve of urban-2 scene from ABU dataset

As seen in Figure 4.20, KRX and RLR-RX methods provides the best performance for urban-3 scene where RX method performs worse.

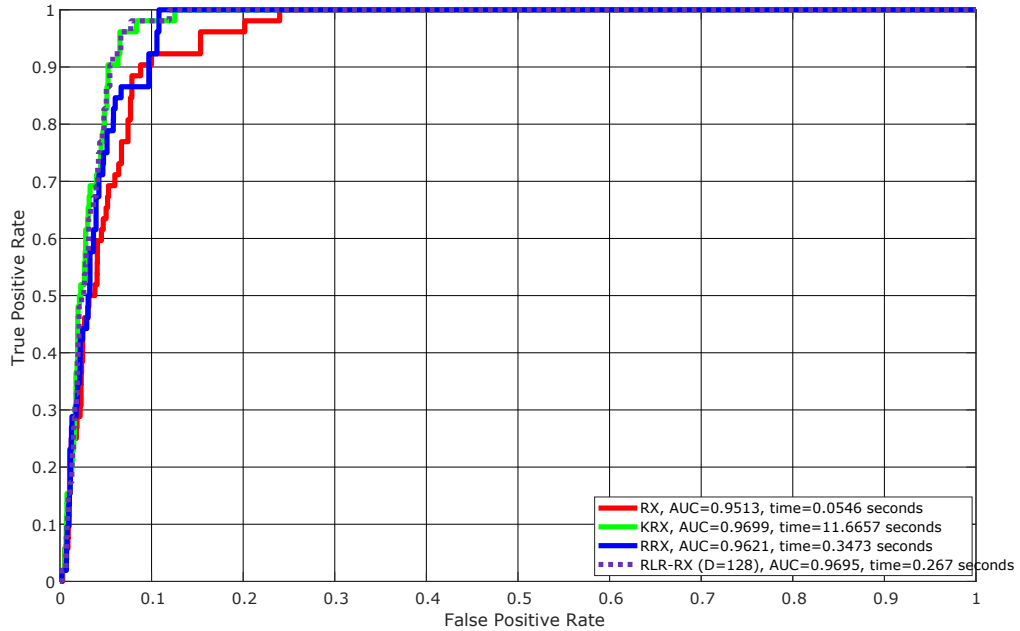


Figure 4.20 ROC curve of urban-3 scene from ABU dataset

RX method provides the best performance for urban-4 scene followed RLR-RX method with similar performance while KRX having the worse performance (see Figure 4.21).

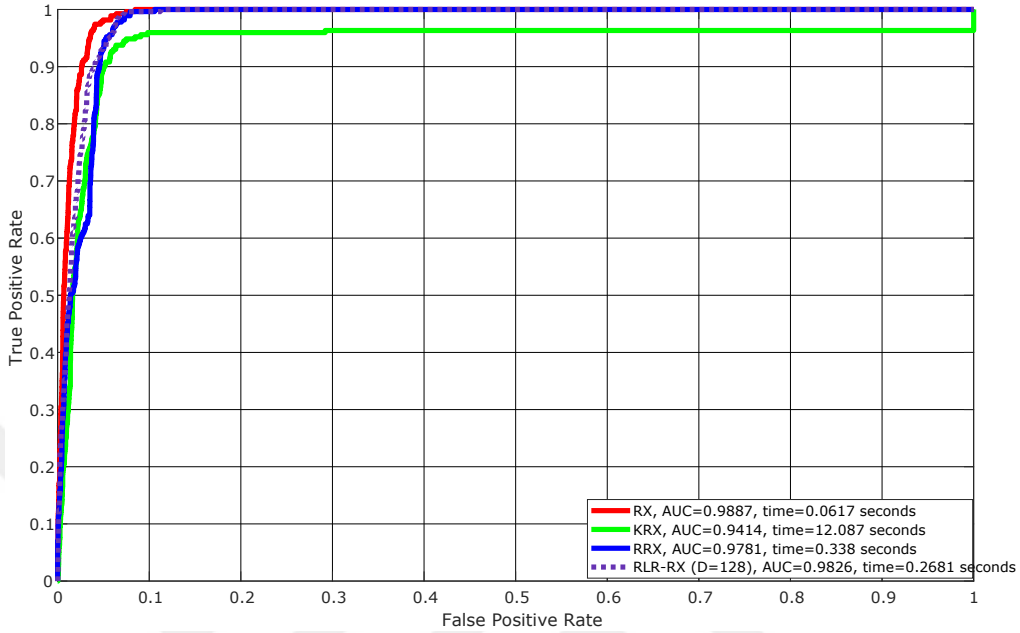


Figure 4.21 ROC curve of urban-4 scene from ABU dataset

As seen in Figure 4.22, for urban-5 scene, KRX performs best but RRX and RLR-RX provides almost the same performance where RX still having high detection accuracy.

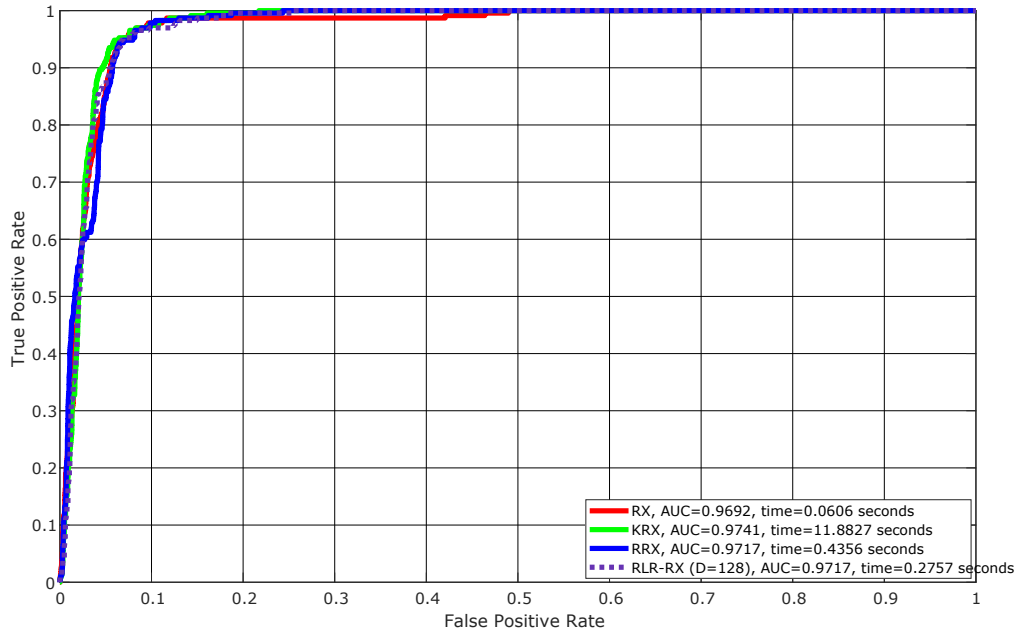


Figure 4.22 ROC curve of urban-5 scene from ABU dataset

Overall performances of the RX, KRX, RRX, and RLR-RX detectors for urban scenes are given in the Figure 4.23 where all nonlinear detectors have similar detection performances (see Table 4.2) while RLR-RX detector provides the highest performance.

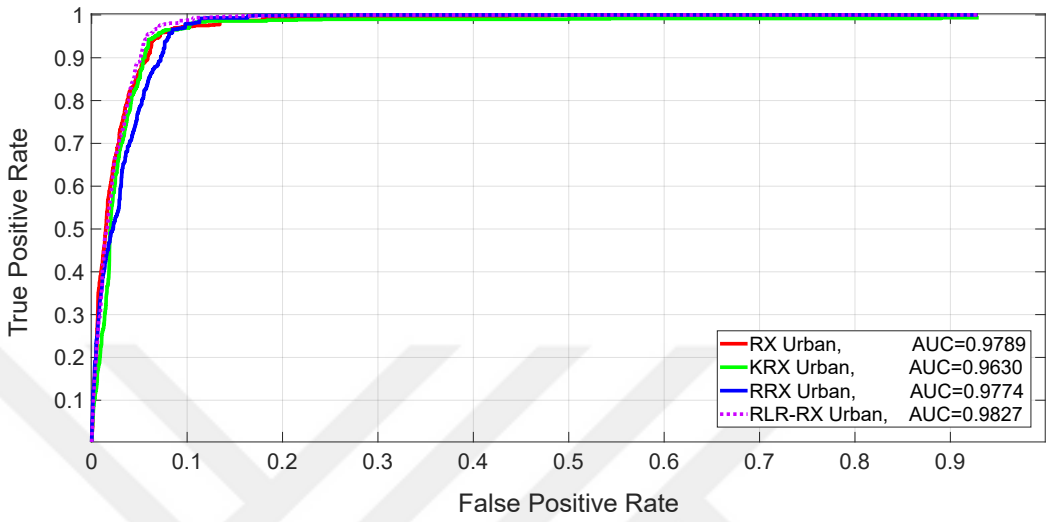


Figure 4.23 ROC curve of ABU urban scenes

4.2.4 EDetection Performances versus Execution Time in ABU Dataset

In Figure 4.24, detection performance of the detectors with respect to execution time is shown. As seen in the Figure 4.24, the fastest detector is RX, slowest detector is KRX, and fastest nonlinear detector is proposed RLR-RX method.

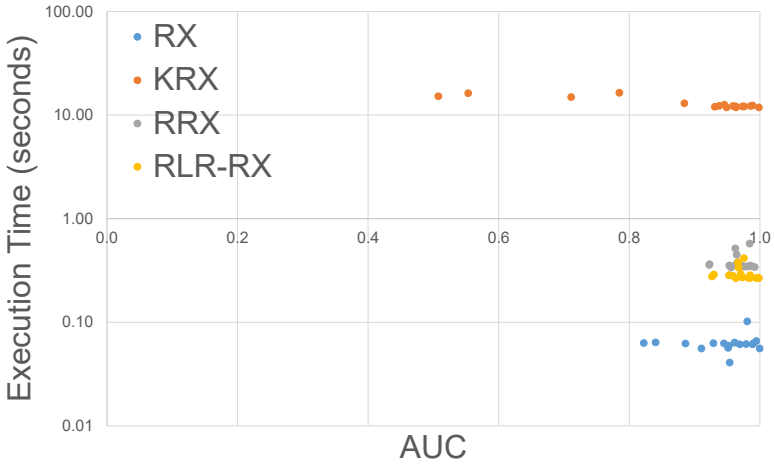


Figure 4.24 Detection Performances versus Execution Time for ABU Dataset

CHAPTER 5

CONCLUSIONS

In this thesis, RRX method is extended with randomized SVD as an efficient and accurate approximation of KRX method which is a well-known nonlinear ananomaly detector. So, proposed RLR-RX is a nonlinear anomaly detection method that can work on non-Gaussian distributed (nonlinear) data in addition to Gaussian distributed (linear) data. In RLR-RX method, first original data is mapped to high dimensional space using the random Fourier Features then covariance matrix is obtained using the mapped data. Instead of inverting this covariance matrix for detecting anomalies as done in RRX, covariance matrix is decomposed using a randomized SVD method and then eigenvectors with low or zero eigenvalues are eliminated as a low-rank approximation of the covariance matrix. By doing this, condition number of the covariance matrix is improved, i.e., its condition number is decreased. Since eigenvalues and eigenvectors are already computed, efficient inverse of low-rank approximated covariance matrix can be computed without no additional cost. Therefore, RLR-RX method becomes faster than RRX while an implicit regularization being enforced leading to stable detection accuracy. Proposed RLR-RX method can be used with tabular data or images. In this thesis, all the experiments are done within a global anomaly detection setting for the sake of simplicity. Accordingly, efficiency of the proposed RLR-RX method is first tested on a synthetic data which confirms that RLR-RX is always faster than RRX and much faster than KRX while detection performance is better or similar to RRX and KRX. Proposed RLR-RX is also tested on real-world dataset, Airport-Beach-Urban (ABU) dataset with 13 hyperspectral images. Experiments show that, RX is always the fastest method such that it works less than 10 milliseconds for 100×100 hyperspectral image while it can give high detection accuracy with only linear data. Note that, proposed nonlinear RLR-RX method is only 4 times slower compared to linear RX method for the hperspectral images that typically have hundreds of channels. Experiments also show that proposed method gives better detection accuracy compared to RX, KRX, and RRX in general while being the fastest nonlinear detector.

One advantage of RLR-RX method is its simplicity. So due to this simplicity several future studies can be easily done such as:

- RLR-RX can be tested on complex scenarios using synthetically generated data,
- RLR-RX can be tested on other real-world data such as multispectral images, other hyperspectral images, infrared images, day-camera images, ultrasound images, and Synthetic Aperture Radar (SAR) images,
- RLR-RX can be tested on tabular data for application domains beyond the image analysis, i.e., for network intrusion detection,
- RLR-RX can be compared with other anomaly detection methods in the literature,
- RLR-RX can be implemented for local anomaly detection on image data,
- RLR-RX can be extended to local-to-global anomaly detector such that best characteristics of local and global anomaly detection can be obtained
- RLR-RX can be extended for the along-track scanners such as pushbroom hyperspectral scanners for real-time anomaly detection,
- RLR-RX can be extended to video images or camera images,
- RLR-RX uses only background data, known spectras can be included to extend RLR-RX as target detection method,
- RLR-RX can be improved with censoring mechanisms (especially for the scenarios where anomalies are not rare or anomalies are big objects in the images),
- RLR-RX can be implemented for embedded platforms,
- RLR-RX can be parallelized using OpenMP for CPU and CUDA for GPU, and
- current RLR-RX code is developed using Matlab, Python and C++ implementations can be done to allow larger community access

REFERENCES

[1] Zhao C., Li J., Meng M., and Yao X., "A weighted spatial-spectral kernel RX algorithm and efficient implementation on GPUs," *Sensors*, vol. 17, no. 3, 2017.

[2] Guo Q., Pu R., and Cheng J., "Anomaly detection from hyperspectral remote sensing imagery," *Geosciences*, vol. 6, no. 4, 2016.

[3] Chandola V., Banerjee A., and Kumar V., "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, 2009.

[4] Nassif A. B., Talib M. A., Nasir Q., and Dakalbab F. M., "Machine learning for anomaly detection: A systematic review," *IEEE Access*, vol. 9, pp. 78 658–78 700, 2021.

[5] Xiong Y., Zuo R., Wang K., and Wang J., "Identification of geochemical anomalies via local RX anomaly detector," *Journal of Geochemical Exploration*, vol. 189, pp. 64–71, 2018.

[6] Borghys D., Achard V., Rotman S. R., Gorelik N., Perneel C., and Schweicher E., "Hyperspectral anomaly detection: A comparative evaluation of methods," in *URSI General Assembly and Scientific Symposium*, Aug 2011, pp. 1–4.

[7] Matteoli S., Diani M., and Corsini G., "A tutorial overview of anomaly detection in hyperspectral images," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 5–28, July 2010.

[8] Zhao C., Yao X.-F., and Yan Y., "Modified kernel RX algorithm based on background purification and inverse-of-matrix-free calculation," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 4, pp. 544–548, April 2017.

[9] Reed I. S. and Yu X., "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution," *IEEE transactions on acoustics, speech, and signal processing*, vol. 38, no. 10, pp. 1760–1770, 1990.

[10] Kwon H. and Nasrabadi N. M., "Kernel RX-algorithm: a nonlinear anomaly detector for hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 2, pp. 388–397, Feb 2005.

[11] Nasrabadi N. M., *Target detection with kernels*. John Wiley & Sons, Ltd Chichester, UK, 2009.

[12] Khazai S. and Mojaradi B., "A modified kernel-RX algorithm for anomaly detection in hyperspectral images," *Arabian Journal of Geosciences*, vol. 8, no. 3, pp. 1487–1495, Mar 2015.

[13] Zhou J., Kwan C., Ayhan B., and Eismann M. T., "A novel cluster kernel RX algorithm for anomaly and change detection using hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 11, pp. 6497–6504, Nov 2016.

[14] Rahimi A. and Recht B., "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 3, no. 4, 2007, p. 5.

[15] Nar F., Pérez-Suay A., Padrón J. A., and Camps-Valls G., "Randomized RX for target detection," in *IEEE International Geoscience and Remote Sensing Symposium*, 2018, pp. 4237–4240.

[16] Hidalgo J. A. P., Pérez-Suay A., Nar F., and Camps-Valls G., "Efficient nonlinear RX anomaly detectors," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 2, pp. 231–235, 2021.

[17] Taitano Y. P., Geier B. A., and Bauer K. W., "A locally adaptable iterative RX detector," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 1, p. 341908, May 2010.

[18] Matteoli S., Diani M., and Corsini G., "A kurtosis-based test to efficiently detect targets placed in close proximity by means of local covariance-based hyperspectral anomaly detectors," pp. 1–4, 2011.

[19] Acito N., Diani M., and Corsini G., "On the CFAR property of the RX algorithm in the presence of signal-dependent noise in hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 6, pp. 3475–3491, June 2013.

[20] Matteoli S., Veracini T., Diani M., and Corsini G., "A locally adaptive background density estimator: An evolution for RX-based anomaly detectors," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 1, pp. 323–327, 2014.

[21] Imani M., "RX anomaly detector with rectified background," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 8, pp. 1313–1317, Aug 2017.

[22] Rossi A., Acito N., Diani M., and Corsini G., "RX architectures for real-time anomaly detection in hyperspectral images," *Journal of real-time image processing*, vol. 9, no. 3, pp. 503–517, 2014.

[23] Chang C.-I., Wang Y., and Chen S.-Y., "Anomaly detection using causal sliding windows," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 7, pp. 3260–3270, July 2015.

[24] Zhao C., Wang Y., Qi B., and Wang J., "Global and local real-time anomaly detectors for hyperspectral remote sensing imagery," *Remote Sensing*, vol. 7, no. 4, pp. 3966–3985, 2015.

[25] Zhao C., Yao X., and Huang B., "Real-time anomaly detection based on a fast recursive kernel RX algorithm," *Remote Sensing*, vol. 8, no. 12, 2016.

[26] Zhao R., Du B., Zhang L., and Zhang L., "A robust background regression based score estimation algorithm for hyperspectral anomaly detection," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 122, pp. 126 – 144, 2016.

[27] Zhao C., Deng W., Yan Y., and Yao X., "Progressive line processing of kernel RX anomaly detection algorithm for hyperspectral imagery," *Sensors*, vol. 17, no. 8, 2017.

[28] Zhao L., Lin W., Wang Y., and Li X., "Recursive local summation of RX detection for hyperspectral image using sliding windows," *Remote Sensing*, vol. 10, no. 1, 2018.

[29] Molero J. M., Paz A., Garzón E. M., Martínez J. A., Plaza A., and García I., "Fast anomaly detection in hyperspectral images with RX method on heterogeneous clusters," *The Journal of Supercomputing*, vol. 58, no. 3, pp. 411–419, 2011.

[30] Molero J. M., Garzón E. M., García I., and Plaza A., "Analysis and optimizations of global and local versions of the RX algorithm for anomaly detection in hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 2, pp. 801–814, April 2013.

[31] Molero J. M., Garzón E. M., García I., Quintana-Ortí E. S., and Plaza A., "Efficient implementation of hyperspectral anomaly detection techniques on GPUs and multicore processors," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2256–2266, June 2014.

[32] Yang B., Yang M., Plaza A., Gao L., and Zhang B., "Dual-mode FPGA implementation of target and anomaly detection algorithms for real-time hyperspectral imaging," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2950–2961, June 2015.

[33] Schweizer S. M. and Moura J. M. F., "Hyperspectral imagery: clutter adaptation in anomaly detection," *IEEE Transactions on Information Theory*, vol. 46, no. 5, pp. 1855–1871, Aug 2000.

[34] Duran O. and Petrou M., "A time-efficient method for anomaly detection in hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 3894–3904, Dec 2007.

[35] He L., Pan Q., Di W., and Li Y., "Anomaly detection in hyperspectral imagery based on maximum entropy and nonparametric estimation," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1392 – 1403, 2008.

[36] Malpica J. A., Rejas J. G., and Alonso M. C., "A projection pursuit algorithm for anomaly detection in hyperspectral imagery," *Pattern Recognition*, vol. 41, no. 11, pp. 3313 – 3327, 2008.

[37] Yuan Z., Sun H., Ji K., Li Z., and Zou H., "Local sparsity divergence for hyperspectral anomaly detection," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 10, pp. 1697–1701, Oct 2014.

[38] Li W. and Du Q., "Collaborative representation for hyperspectral anomaly detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1463–1474, March 2015.

[39] Li J., Zhang H., Zhang L., and Ma L., "Hyperspectral anomaly detection by the use of background joint sparse representation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2523–2533, June 2015.

[40] Sun W., Tian L., Xu Y., Du B., and Du Q., "A randomized subspace learning based anomaly detector for hyperspectral imagery," *Remote Sensing*, vol. 10, no. 3, 2018.

[41] Zhu L. and Wen G., "Hyperspectral anomaly detection via background estimation and adaptive weighted sparse representation," *Remote Sensing*, vol. 10, no. 2, 2018.

[42] Soofbaf S. R., Sahebi M. R., and Mojaradi B., "A sliding window-based joint sparse representation (SWJSR) method for hyperspectral anomaly detection," *Remote sensing*, vol. 10, no. 3, p. 434, 2018.

[43] Du B., Zhao R., Zhang L., and Zhang L., "A spectral-spatial based local summation anomaly detection method for hyperspectral images," *Signal Processing*, vol. 124, pp. 115 – 131, 2016.

[44] Wang L. and Zhao C., *Anomaly Detection Technique of HSI*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 217–256.

[45] Huber-Lerner M., Hadar O., Rotman S. R., and Huber-Shalem R., "Hyperspectral band selection for anomaly detection: The role of data gaussianity," *IEEE Journal*

of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 9, no. 2, pp. 732–743, Feb 2016.

[46] Ma N., Peng Y., Wang S., and Leong P. H. W., "An unsupervised deep hyperspectral anomaly detector," *Sensors*, vol. 18, no. 3, 2018.

[47] Zhang Y., Du B., Zhang L., and Wang S., "A low-rank and sparse matrix decomposition-based mahalanobis distance method for hyperspectral anomaly detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1376–1389, 2015.

[48] Niu Y. and Wang B., "Hyperspectral anomaly detection based on low-rank representation and learned dictionary," *Remote Sensing*, vol. 8, no. 4, 2016.

[49] Drineas P., Mahoney M. W., and Cristianini N., "On the nyström method for approximating a gram matrix for improved kernel-based learning." *journal of machine learning research*, vol. 6, no. 12, 2005.

[50] Achlioptas D., McSherry F., and Schölkopf B., "Sampling techniques for kernel methods," *Advances in neural information processing systems*, vol. 14, p. 335, 2002.

[51] Halko N., Martinsson P.-G., and Tropp J. A., "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[52] Camps-Valls G. and Bruzzone L., *Kernel methods for remote sensing data analysis*. John Wiley & Sons, 2009.

[53] Burden R. L. and Faires J. D., *Numerical Analysis*. PWS-Kent Publishing Company, 1989.

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname : Selçuk YAPICI

EDUCATION

High School : (1997-2001) GPA 4.56/5.00

Giresun Anatolian Vocational High School, Department of Computer

Bachelor : (2002-2006) GPA 2.93/4.00

Hacettepe University, Computer Education and Instructional Technology

Bachelor : (2013-2016) GPA 2.88/4.00

Akhmet Yassawi International Kazakh-Turkish University, Computer Engineering

Master Degree : (2018-2022) GPA 3.38/4.00

Ankara Yildirim Beyazit University, Computer Engineering

WORK EXPERIENCE

Software Developer : April 2009 -

General Directorate of Information Technologies, Ministry of National Education, Republic of Turkey

ICT Teacher : September 2006 - April 2009

Kazan İbrahim Bitik Secondary School, Keçiören Talia Yaşar Bakdur Secondary School

TOPICS OF INTEREST

- Anomaly detection
- Machine learning
- Numerical methods
- Randomization