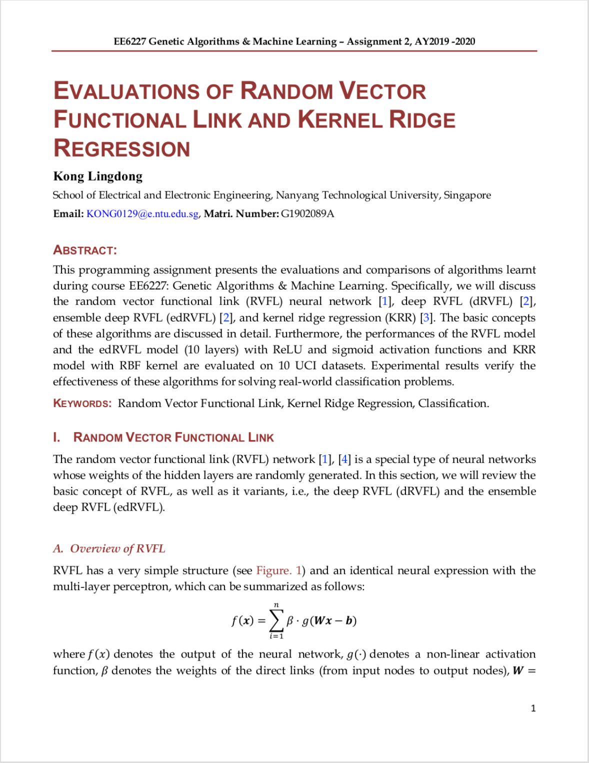
Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	EEE KONG LINGDONG
Assignment title:	Assignment 2 (Deadline 4th May)
Submission title:	EE6227 Programming Assignment 2
File name:	EE6227_Assignment2_Kong_Lingd…
File size:	632.4K
Page count:	11
Word count:	2,859
Character count:	13,720
Submission date:	04-May-2020 09:47AM (UTC+0800)
Submission ID:	1315079940



EVALUATIONS OF RANDOM VECTOR FUNCTIONAL LINK AND KERNEL RIDGE REGRESSION

Kong Lingdong
School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
Email: KONG0129@e.ntu.edu.sg, **Matri. Number:** G1902089A

ABSTRACT:

This programming assignment presents the evaluations and comparisons of algorithms learnt during course EE6227: Genetic Algorithms & Machine Learning. Specifically, we will discuss the random vector functional link (RVFL) neural network [1], deep RVFL (dRVFL) [2], ensemble deep RVFL (edRVFL) [2], and kernel ridge regression (KRR) [3]. The basic concepts of these algorithms are discussed in detail. Furthermore, the performances of the RVFL model and the edRVFL model (10 layers) with ReLU and sigmoid activation functions and KRR model with RBF kernel are evaluated on 10 UCI datasets. Experimental results verify the effectiveness of these algorithms for solving real-world classification problems.

KEYWORDS: Random Vector Functional Link, Kernel Ridge Regression, Classification.

I. RANDOM VECTOR FUNCTIONAL LINK

The random vector functional link (RVFL) network [1], [4] is a special type of neural networks whose weights of the hidden layers are randomly generated. In this section, we will review the basic concept of RVFL, as well as it variants, i.e., the deep RVFL (dRVFL) and the ensemble deep RVFL (edRVFL).

A. Overview of RVFL

RVFL has a very simple structure (see Figure. 1) and an identical neural expression with the multi-layer perceptron, which can be summarized as follows:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \beta \cdot g(\boldsymbol{W}\boldsymbol{x} - \boldsymbol{b})$$

where $f(x)$ denotes the output of the neural network, $g(\cdot)$ denotes a non-linear activation function, β denotes the weights of the direct links (from input nodes to output nodes), $\boldsymbol{W} =$