Lab assignment 1

Artificial neural networks using back propagation

**Examiner Email**

Ning Xiong [**ning.xiong@mdh.se**](mailto:ning.xiong@mdh.se)

**Students name Email**

***Robin Calmegård*** [***rcd10002@student.mdh.se***](mailto:rcd10002@student.mdh.se)

***Dennis Stockhaus*** [***dss10002@student.mdh.se***](mailto:dss10002@student.mdh.se)

# **Abstract**

This report is written in relation to Assignment 1 in the course ‘DVA427 – Learning Systems’ at Mälardalens University. The assignment is to develop a competent neural network to predict ‘etch-depth’ for a Reactive-Ion-Etching (RIE) machine based on predefined selected sensor signals gathered in .txt files. The Neural-Network may be implemented with any programming language as well as framework.   
In continuation this report will describe and summarize the following information: the structure of neural-network selected, the learning algorithm used, the evolving of performance (errors with iterations), performance on training data and performance on test data

# Background

Reactive-Ion-Etching  
Reactive-Ion-Etching (RIE) machines are used in some companies for removal of thin layers in the production of magnetic heads. The task of a RIE machine is to etch a specified surface of some Xm depth into sliders. The etch depths of sliders are regarded as indicators for process stability. It would be easy to control the process if the etch depth could be monitored online. Unfortunately such a depth can only be measured at the end of the etching process.

In order to enhance process stability and reliability, attempts have been made to analyze data from process observations and to detect correlation between sensor data and process result, i.e. etch depth. At present, however, there is no explicit mathematical model to calculate etch depth from process parameters. As alternative artificial neural network may be constructed to predict etch depth based on sensor information.

Neural Networks

The software for the assignment is built with C# programming language and uses predefined graphical libraries offered in Microsoft’s Integrated Development Enviroment (IDE) Visual Studio 2015.

This solution is built by C# (C-sharp)[[1]](#footnote-1) programming language and for the moment it’s limited and is only available for Windows[[2]](#footnote-2) systems. The solution uses the predefined graphical libraries that Microsoft [[3]](#footnote-3)offers in their Integrated Development environment (IDE) *Visual Studio*[[4]](#footnote-4) 2015.

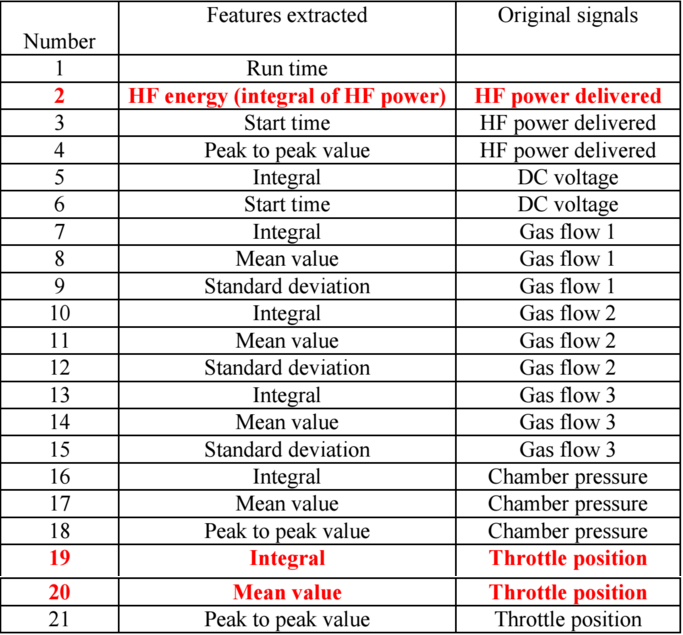
# Implementation

The current implementation of the lab assignment is divided into separate components / model parts and each model have their specific functionality. Also this implementation is early stages of going to be redefined and built after the software architectural pattern *Model-View-View Model* (MVVM). Why we choose this pattern is because to separate *data* (model) from the *view* and make it less clustered and easier for reusability of the code.

## Artificial Neural Network

the learning algorithm used,

### The Reactive-Ion-Etching machines data



The image shows the different steps and there signals in the Reactive-Ion-Etching machines. Notice that there are three of these rows that are marked in red and bold text (rows are 2-19-21). These are the inputs for the ANN and the last row is the answer for the etching machine.

### Structure of neural network

the structure of neural network selected,

### Performance

the evolving of performance (errors with iterations),

#### Traning Data

performance on traning data

#### Test Data

performance on test data

# Result

Display graphs

1. <https://en.wikipedia.org/wiki/C_Sharp_(programming_language)> (Last accessed 2016-09-08) [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)
3. [↑](#footnote-ref-3)
4. [↑](#footnote-ref-4)