DFRWS
DIGITAL FORENSIC RESEARCH CONFERENCE

A Comparative Study of Support Vector Machine and Neural Networks for File Type Identification using n-gram analysis

By:

Joachim Sester, Darren Hayes, Mark Scanlon and Nhien-An Le-Khac

From the proceedings of

The Digital Forensic Research Conference

DFRWS EU 2021

March 29 - April 1, 2021

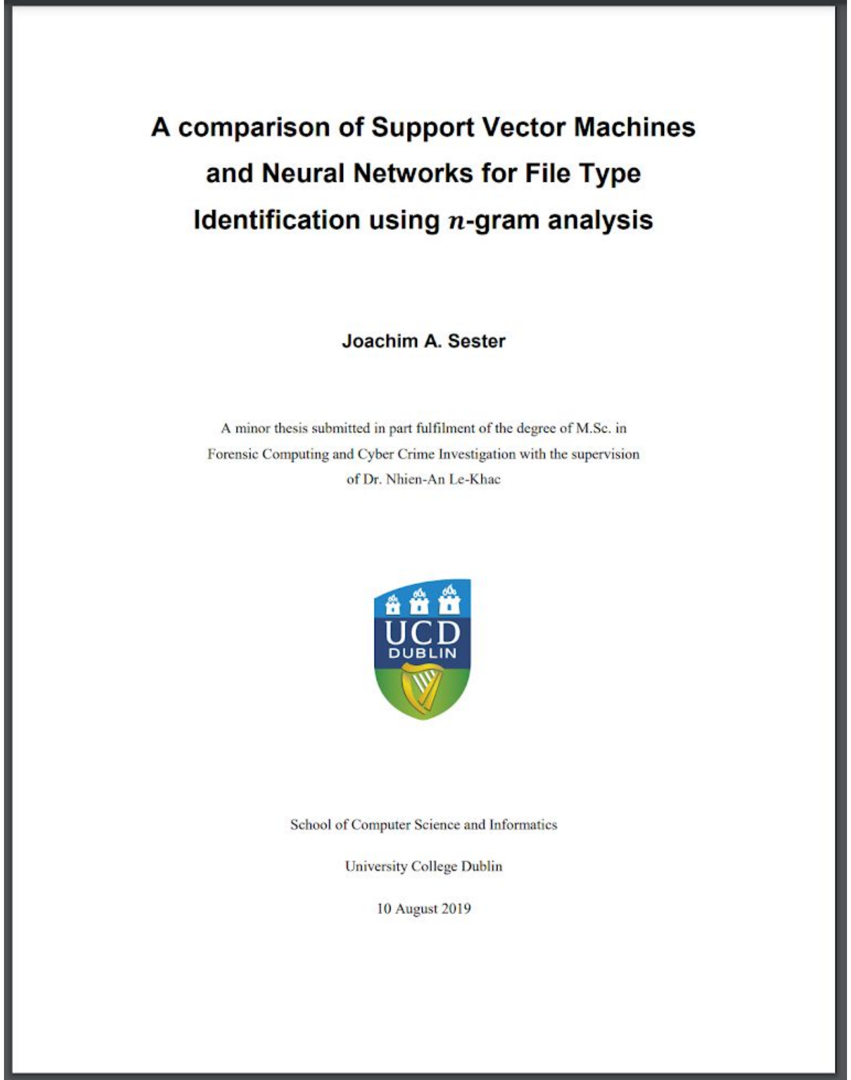
DFRWS is dedicated to the sharing of knowledge and ideas about digital forensics research. Ever since it organized the first open workshop devoted to digital forensics in 2001, DFRWS continues to bring academics and practitioners together in an informal environment.

As a non-profit, volunteer organization, DFRWS sponsors technical working groups, annual conferences and challenges to help drive the direction of research and development.

https://dfrws.org

Presentation of Thesis on DFRWS EU 2021

Filetype-Identification using SVM vs NN



A comparison of Support Vector Machines and Neural Networks for File Type Identification using n-gram analysis

Joachim A. Sester

A minor thesis submitted in part fulfilment of the degree of M.Sc. in Forensic Computing and Cyber Crime Investigation with the supervision of Dr. Nhien-An Le-Khac

School of Computer Science and Informatics

University College Dublin

10 August 2019

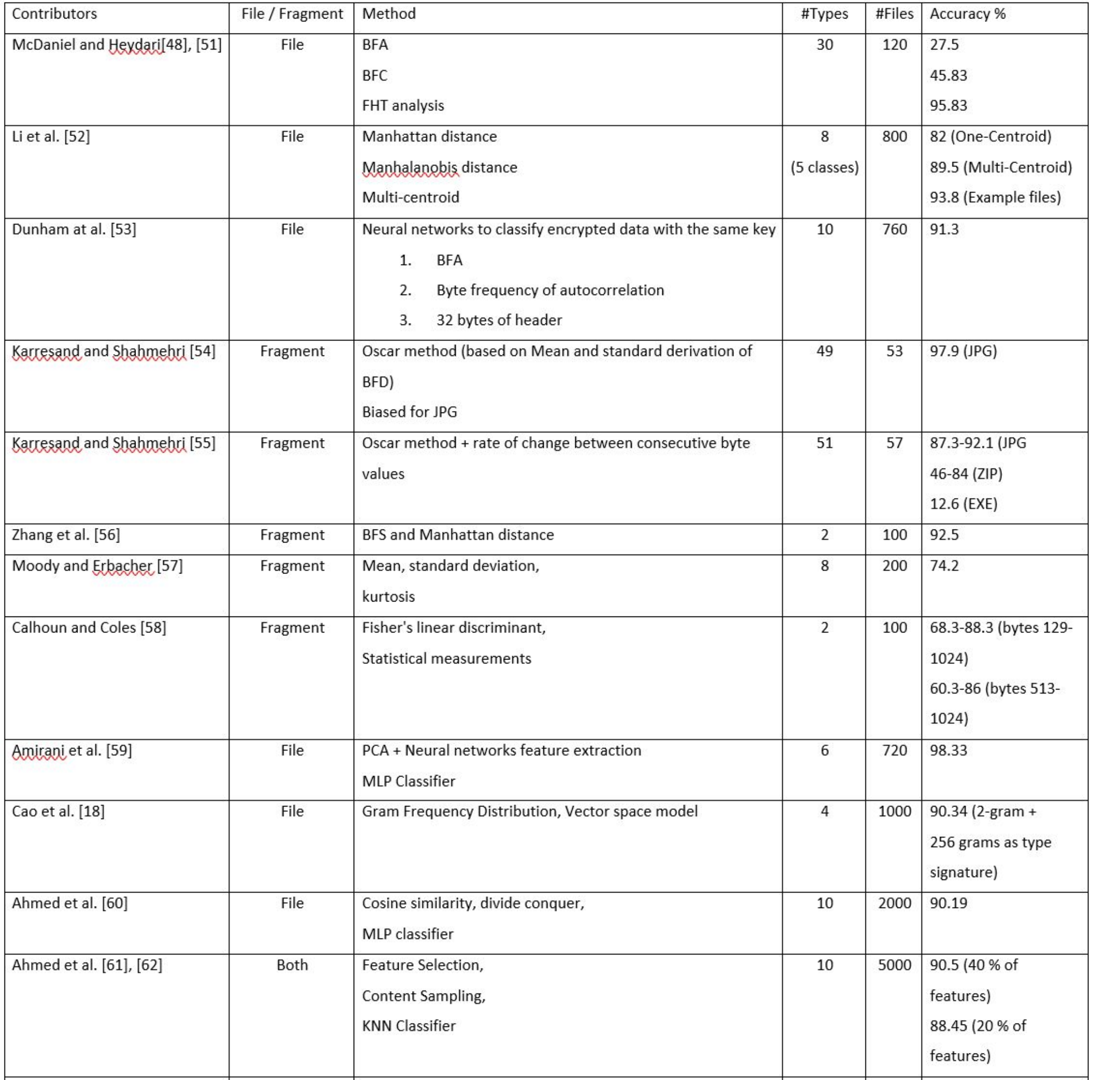
The idea

- Idea of Classical monogram-statistics a.k.a. histograms = 1-byte-statistic
- If we go for 2-bytes, 3-bytes etc., can AI help us classify better then?

- SVMs are good for multi-dimensional classification
- NN allow “deep learning” classification

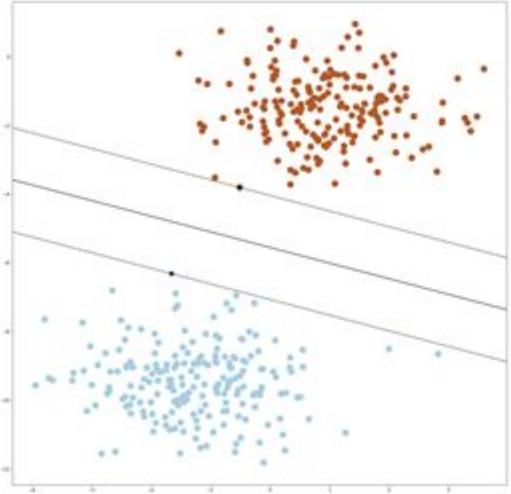
Which is better?

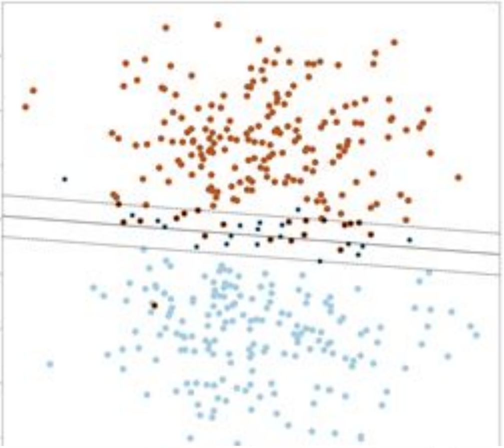
Can they be improved using n-grams ?

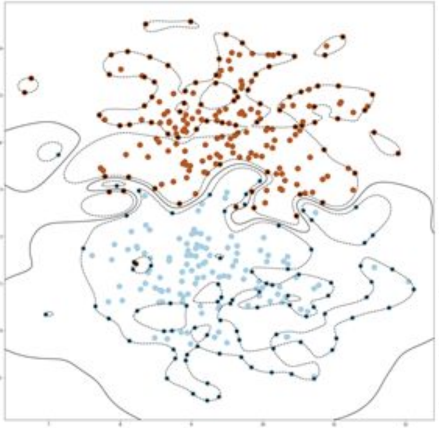


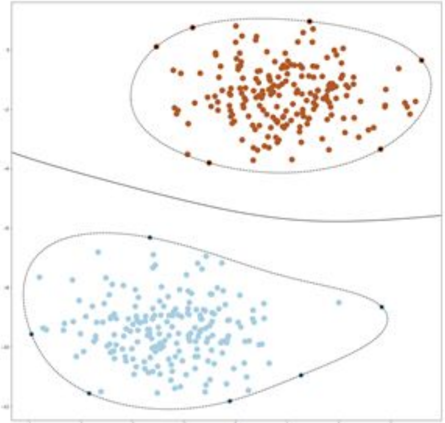
Contributors	File / Fragment	Method	#Types	#Files	Accuracy %
McDaniel and Heydari[48], [51]	File	BFA BFC FHT analysis	30	120	27.5 45.83 95.83
Li et al. [52]	File	Manhattan distance Manhalanobis distance Multi-centroid	8 (5 classes)	800	82 (One-Centroid) 89.5 (Multi-Centroid) 93.8 (Example files)
Dunham at al. [53]	File	Neural networks to classify encrypted data with the same key 1. BFA 2. Byte frequency of autocorrelation 3. 32 bytes of header	10	760	91.3
Karresand and Shahmehri [54]	Fragment	Oscar method (based on Mean and standard derivation of BFD) Biased for JPG	49	53	97.9 (JPG)
Karresand and Shahmehri [55]	Fragment	Oscar method + rate of change between consecutive byte values	51	57	87.3-92.1 (JPG 46-84 (ZIP) 12.6 (EXE)
Zhang et al. [56]	Fragment	BFS and Manhattan distance	2	100	92.5
Moody and Erbacher [57]	Fragment	Mean, standard deviation, kurtosis	8	200	74.2
Calhoun and Coles [58]	Fragment	Fisher's linear discriminant, Statistical measurements	2	100	68.3-88.3 (bytes 129-1024) 60.3-86 (bytes 513-1024)
Amirani et al. [59]	File	PCA + Neural networks feature extraction MLP Classifier	6	720	98.33
Cao et al. [18]	File	Gram Frequency Distribution, Vector space model	4	1000	90.34 (2-gram + 256 grams as type signature)
Ahmed et al. [60]	File	Cosine similarity, divide conquer, MLP classifier	10	2000	90.19
Ahmed et al. [61], [62]	Both	Feature Selection, Content Sampling, KNN Classifier	10	5000	90.5 (40 % of features) 88.45 (20 % of features)

Problems - Overfitting

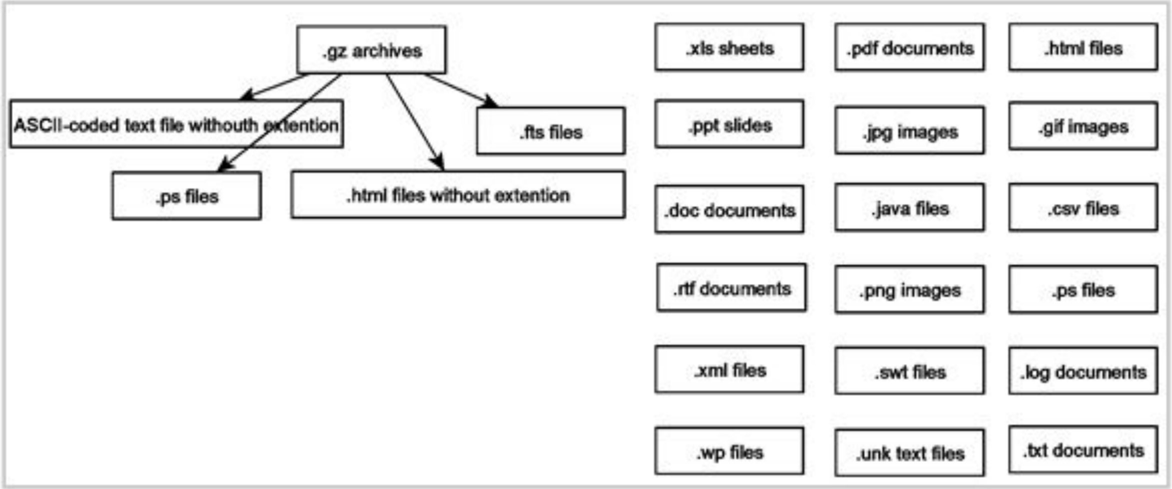








Problems - Dataset



.gz archives

ASCII-coded text file withouth extention

.fts files

.ps files

.html files without extention

.xls sheets	.pdf documents	.html files
.ppt slides	.jpg images	.gif images
.doc documents	.java files	.csv files
.rtf documents	.png images	.ps files
.xml files	.swt files	.log documents
.wp files	.unk text files	.txt documents

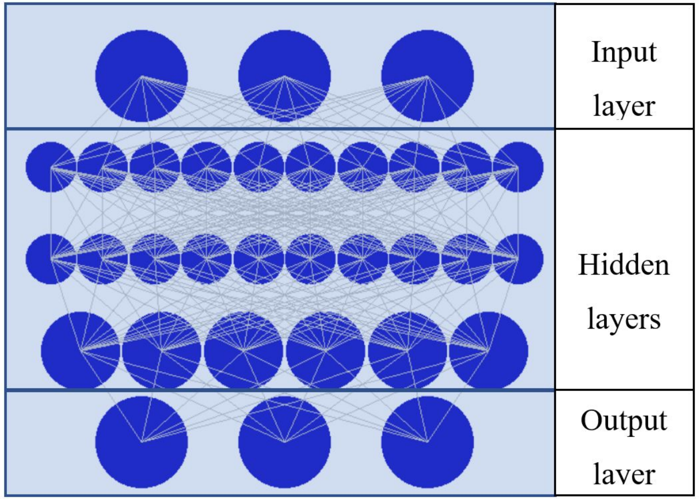
Problems NeuralNetwork

Formula for DeltaNN-Backpropagation:

$$\Delta w_{ij_x} = - \varepsilon \frac{\delta E}{\delta w_{ij}} = \varepsilon \delta a_{i_x}$$

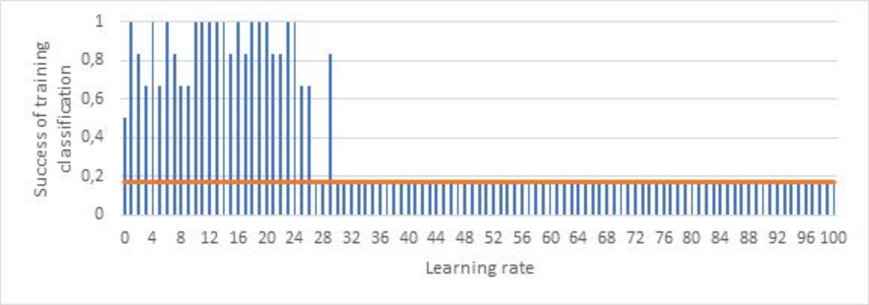
General Explaination of NeuralNetworks’ Setup

Activation funtion: sigmoid



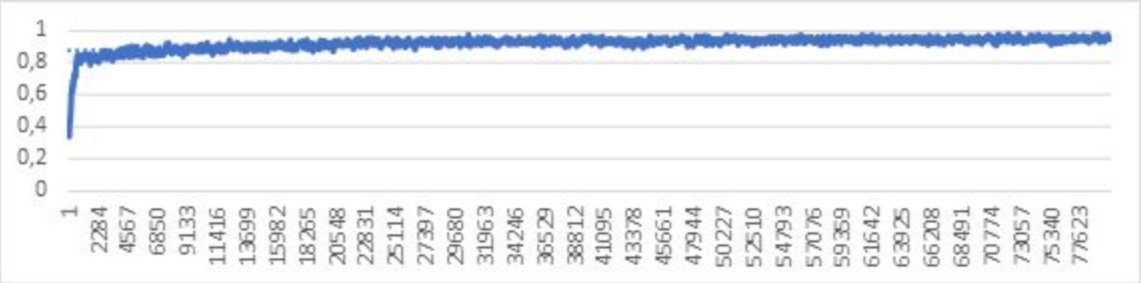
n = 1

10000x Training. Red = estimated random prediction (1/6th because of six filetypes)



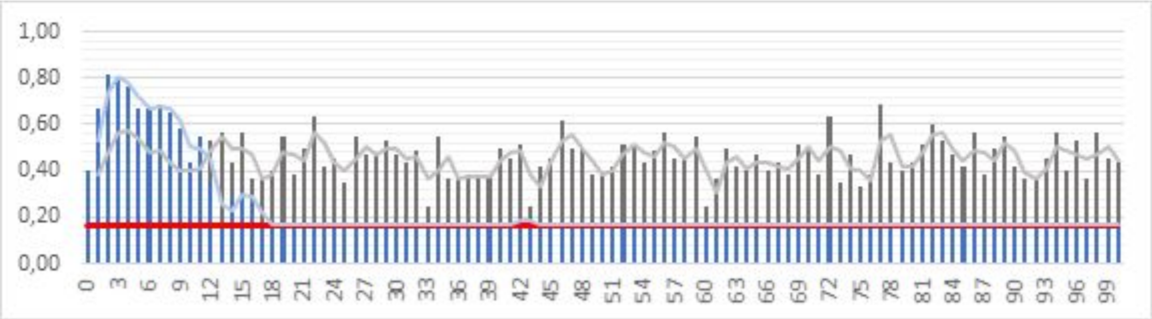
training progress: 12% training rate

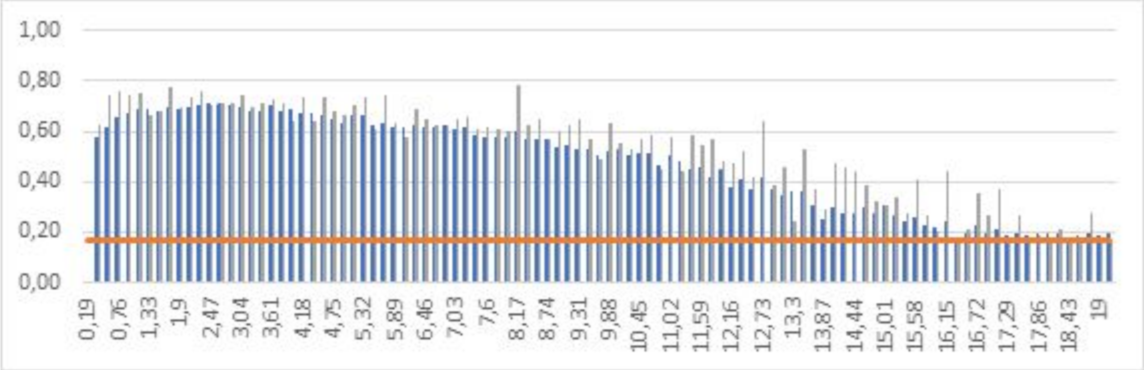
After 80.000 pieces of training, above 96 percent of test data was correctly classified.



n=2

learning rate of was chosen

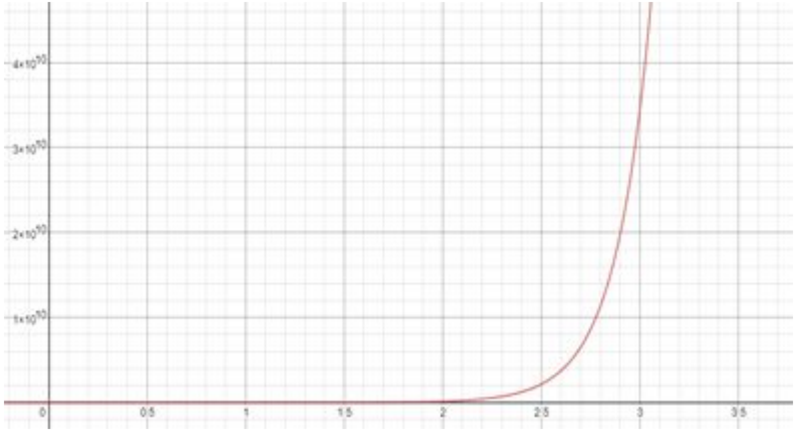




n=3

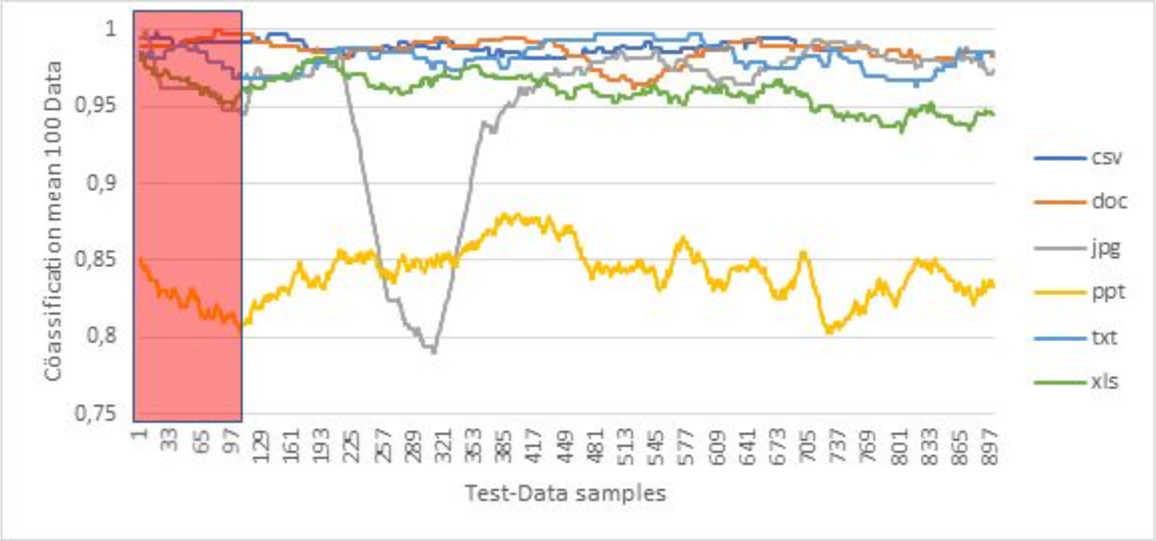
Error… Huh?

Memory Usage prediction:



NN results

underfitting on jpg



SVM results

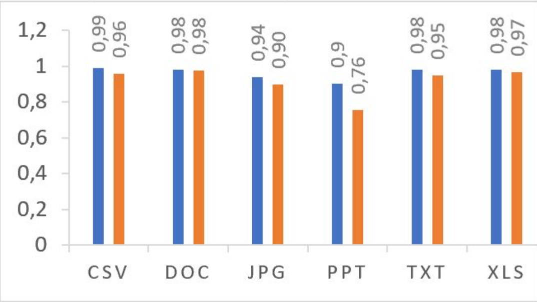


Figure 5-5 SVM n = 1 linear kernel

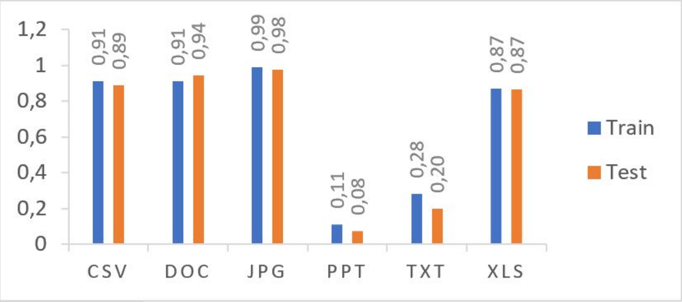


Figure 5-6 SVM n = 1 rbf kernel

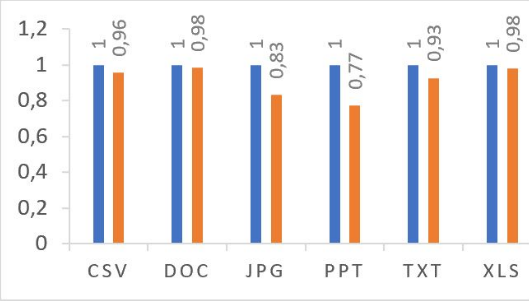


Figure 5-4 SVM n = 2 linear kernel

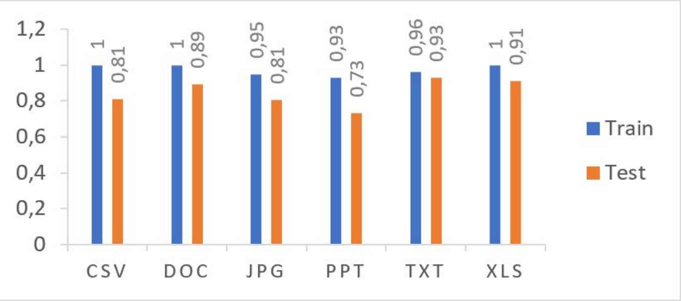
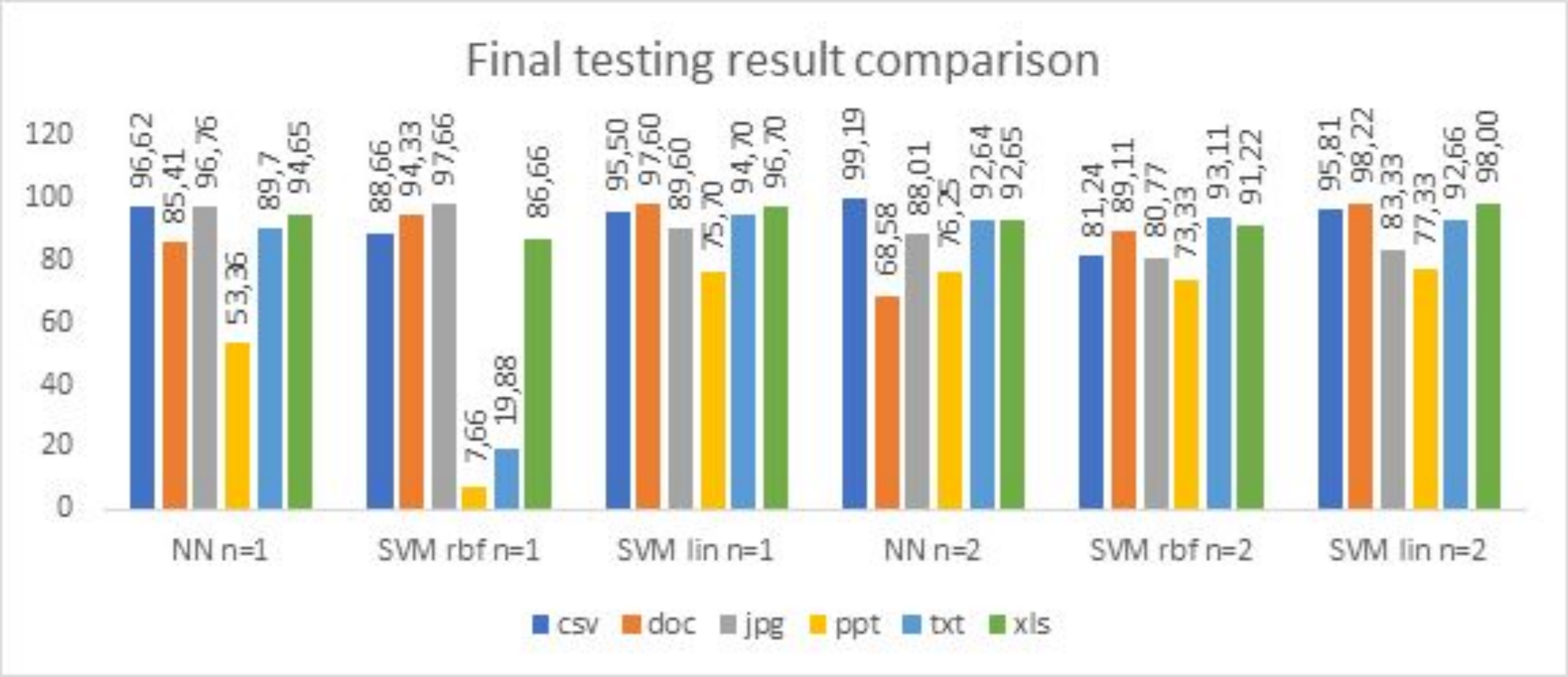


Figure 5-3 SVM n = 2 rbf kernel

Results


Final testing result comparison
120
100
80
60
40
20
0
96,62
85,41
96,76
53,36
89,7
94,65
88,66
94,33
97,66
7,66
19,88
86,66
95,50
97,60
89,60
75,70
94,70
96,70
99,19
68,58
88,01
76,25
92,64
92,65
81,24
89,11
80,77
73,33
93,11
91,22
95,81
98,22
83,33
77,33
92,66
98,00
NN n=1
SVM rbf n=1
SVM lin n=1
NN n=2
SVM rbf n=2
SVM lin n=2
csv
doc
jpg
ppt
txt
xls

Conclusion

- 3-gram is not superior to 2-gram
- n-gram analysis is not useful for further FTI
- SVMs are faster, NNs can provide better results (with deep learning)

In this scenario, both approaches resulted in almost equal ability of both.