

Editor's Comment:

1. The authors need to explain more how the proposed hybrid will work.

Step1: Insert Web Pages.

Step2: Apply Naïve Bayesian for feature selection.

Step3: Apply support vector machine to classify these Web Pages and get the results.

Step4: Determine the Web pages that classified incorrectly.

Step5: Apply K-Nearest Neighbor on the Web pages obtained step 4.

Figure 1, presents a flowchart of the proposed system in the improving phase.

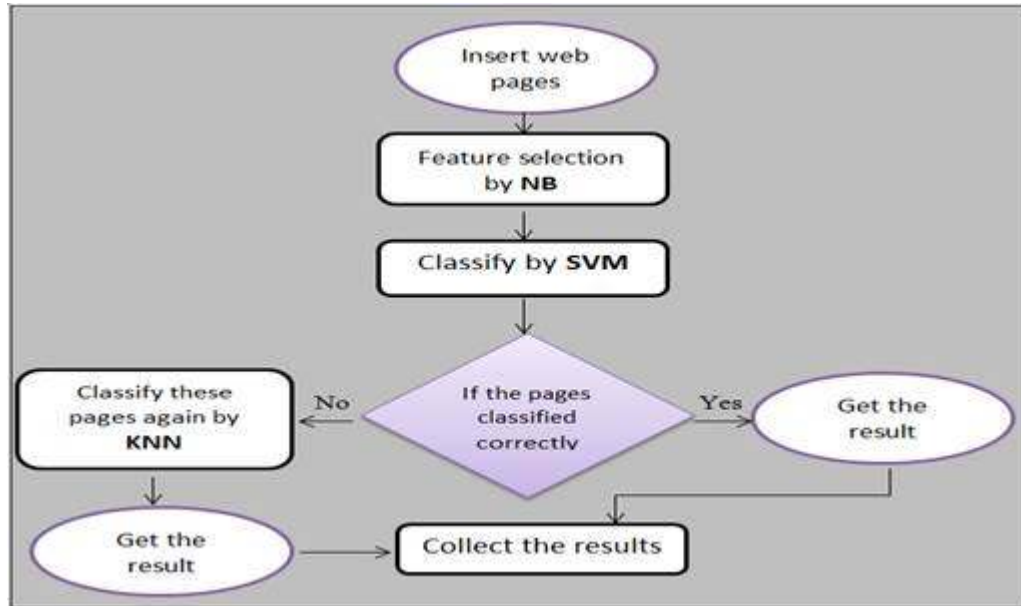


Figure 1: Improving Phase Steps

In the above flowchart, it stated : if pages are classified incorrectly, then use KNN to do the classification again.

This raises some questions:

1. How do you do this in real life classification where the system does not know about the actual correct classes. So how will you decide whether SVM classification is correct or not? Author need to provide more clarifications in this case. I mean how will the hybrid scheme work in the testing phase. Give a flowchart too. Since in testing phase the target classes are not known, so how do you decide whether to use SVM alone or still pass it to KNN to try?
2. What if KNN also classified it as incorrect? The flowchart does not take care of this kind of possibilities.
3. Why not first use KNN before SVM? Does that make any difference?
The authors should address the above issues .

Authors' Feedback:

We modify the flowchart to make the proposed hybrid algorithm clearer.

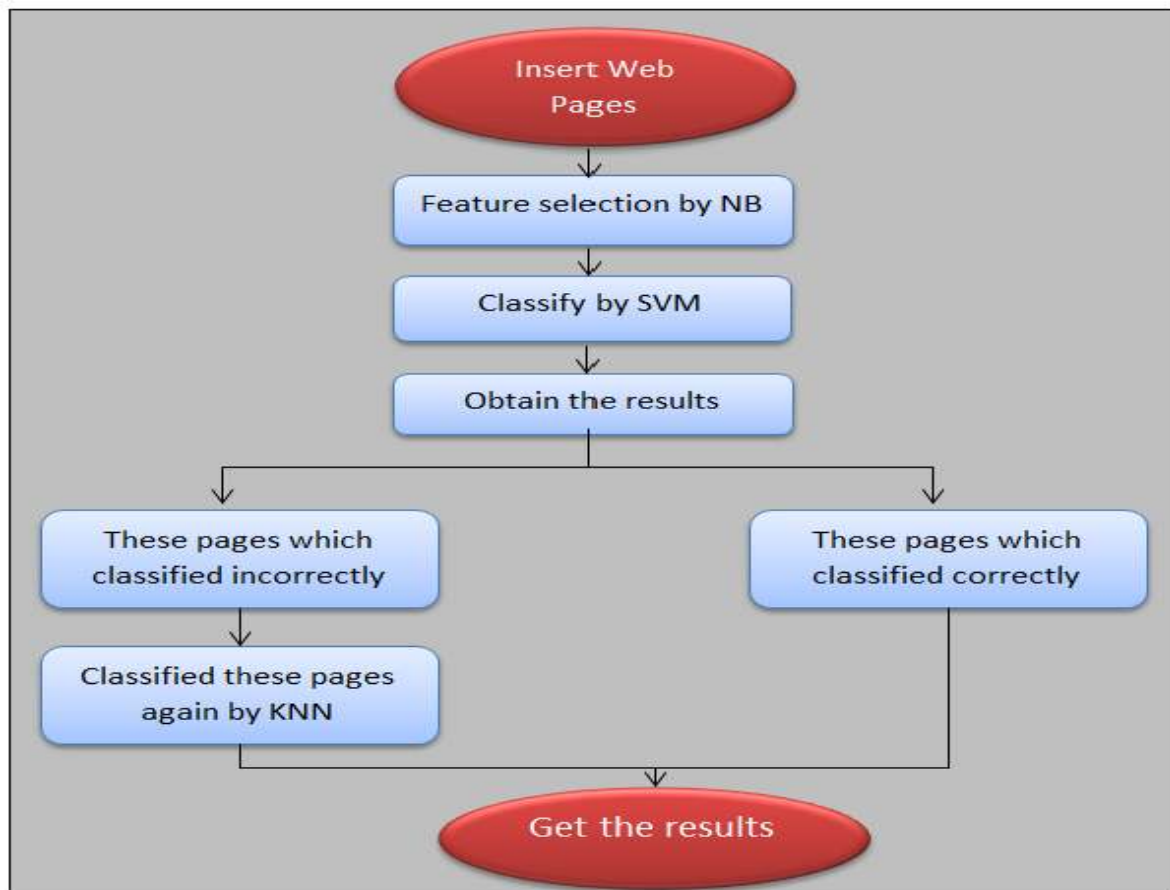


Figure 1: Improving Phase Steps

The Editor asks : how will we decide whether SVM classification is correct or not?

1. We used supervised machine learning.
2. We must pass the training process first.
3. We know the results before going to the improvement phase. So, the web pages obtained from SVM classifier is measured efficiently and no errors.

The Editor asks: Why not first use KNN before SVM? Does that make any difference?

1. The proposed system is for improving the results, efficiency, and run time.
2. We make some experiments; we found that the run time of KNN is less than SVM in the training process but in the classification process we found that the run time of SVM is less than KNN.
3. So, we apply SVM first on all WebPages (whole set) and apply KNN on only incorrect classified pages (subset only) to save time.
4. The run time of the training phase and classification phase is added in the following paragraph and in the paper (Table 1 and Table 9).

Table1. Execution Time of training process measured in seconds

Text Source	BAYES	SVM	KNN
TITLE	0.054	18.649	1.304

BODY	0.163	239.201	35.334
Body and Title (BT)	0.362	466.646	71.246

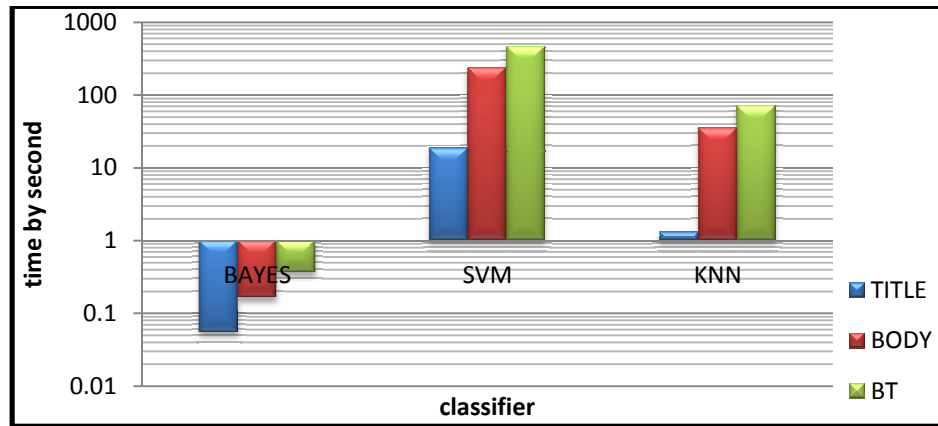


Table 9: A summary for the CPU Time taken by the classification phase (measured by second)

	Execution Time of Title	Execution Time of Body	Execution Time of BT
BAYES	338.25	304.62	318.54
SVM	314.65	351.69	275.28
KNN	300.1700	2184.34	2609.93
NB & SVM	281.83	342	274.32
NB & KNN	297.23	1992.84	2007.80
NB & SVM & KNN	280.31000	576.19	442.51

