Machine Learning - Software Project

Component 4

Authors: Liviu-Ștefan Neacșu-Miclea, Răzvan-Gabriel Petec
Specialization: Applied Computational Intelligence
Group: 246/2

1 Support Vector Regression

Support Vector Regression Machines were first introduced by Vapnik et al. as a SVM variant for regression tasks [DBK+96]. The model was tested against three artificial Friedman functions and the Boston Housing dataset, and compared with another regression algorithm, namely a bagging of regression trees [DBK+96]. Each Friedman functions generated 100 different experiments with 200 training samples, 40 validation and 1000 test samples. The used metrics were prediction error ($PE = MSE(y_{pred} - y_{obs}(x)) = MSE(y_{pred} - (ground\_truth(x) + noise)))$ and modelling error ($ME = MSE(y_{pred} - ground\_truth(x)))$, where ground truths where available (the Boston Housing dataset featured a median prediction on house pricing, therefore not the real ground truth value) [DBK+96]. Table 1 summarizes the results of SVR vs bagging on Friedman functions, pointing that SVR has an overall better performance out of the underwent 100 trials [DBK+96]. The same work suggests that better values are obtained for the second degree of feature space polynomial encoder. Vapnik et al. also perform an experiment that highlights the importance of their ϵ-sensitive model and the choice for the problem-dependent regularization parameter. This setup features a contrastive display of the optimized SVR and the "suboptimal" version with $\epsilon = 0$ and very large regularization constant (meaning that little to no attention is accorded to the weights magnitude). As expected, the suboptimal version performed 5%-86% worse than the optimal one on Friedman functions [DBK+96]. Additionally, SVR performed better despite the inability to create a feature representation because of the high dimensionality required for the feature space relative to the number of trained samples [DBK+96]. Moreover, it was noted that the test cases for evaluating the SVR were "too simple" and it was hypothesized that the model works better when the dimensionality of feature space representation is considerably greater than the number of training samples [DBK+96].

Friedman fun. no.	SVR feature space	ME		PE		No. times SVR better than bagging
		bagging	SVR	bagging	SVR	
1	66	2.2600	0.6700	3.3600	1.7500	100/100
2	15	10.1850	4.9440	66.0770	60.4240	92/100
3	15	0.0302	0.0261	0.0677	0.0692	46/100
Boston H.	-	-	-	12.4000	7.2000	71/100

Table 1: SVR vs bagging experimental results on Friedman functions [DBK+96]

Aside from the seminal paper, SVR also caught interest in other domains featuring regression problems. A review on COVID-19 forecasting and diagnosing showcases an in-depth methodology analysis, revealing that SVR is the second most used method in the field [CP22]. SVRs are concluded to outperform other models (e.g. Ribeiro et al., SVR has the lowest average absolute error in predicting COVID-19 daily cases in Brazil), or rival with other models that prove to be slightly better (such as Hazarika and Gupta's MLP based model WCRVFL (e.g. 0.997%/0.999% accuracy, 0.0858/0.0069 RMSE SVR/WCRVFL+sigmoid on Brasil infection cases prediction [HG20]), or Shahie et al.'s outperforming BiLSTM) [CP22].

Another interesting application is the combined Regressive CNN (RCNN) and SVR for forecasting electricity consumption [ZL20]. Table 2 shows the comparison between RCNN and SVR performances, as well as the beneficial effect of combining these two architecture on improving the results.

Method	RCNN-SVR	RCNN	SVR
MSE	0.8564	10.690	11.639
MAPE	0.0197	0.0212	0.0226
CV-RMSE	0.0068	0.0075	0.0080

Table 2: SVR compared with RCNN and combined method for electricity consumption [ZL20]

Moreover, SVR inspired other similar models, such as the extended SVR (X-SVR), to improve on regression capability in reliability analysis (Figure 1 shows boxplots of experimental results on a benchmark Borehole function) [FLW^{+}19], with applications in fracture mechanics [FWSG23].

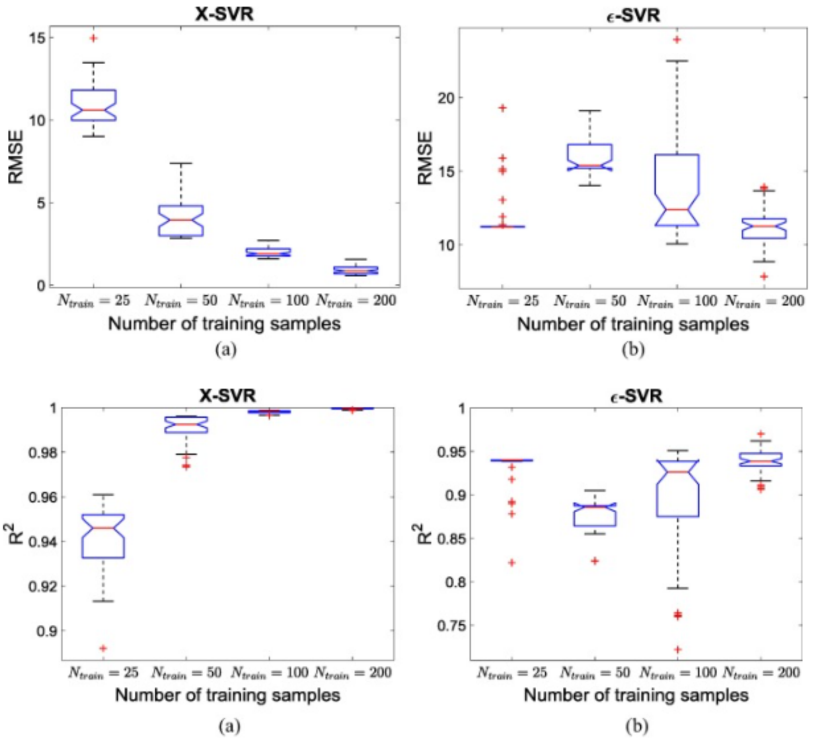

Figure 1: Comparison between SVR and X-SVR on Borehole function

2 Random Forest Classification

Random Forest Trees were first introduced by Breiman as an ensemble learning method for classification and regression tasks [Bre01]. In classification tasks, Random Forests operate by constructing multiple decision trees and aggregating predictions through majority voting, making them robust to overfitting and capable of handling high-dimensional datasets effectively.

Random Forests (RFs) have been extensively evaluated for classification tasks and demonstrated strong performance. RFs proves robust generalization by building ensembles of decision trees using random feature selection and bootstrapping. Comparative experiments show that RFs can achieve greater results than other models in certain scenarios. The metrics that are highlighted in the following experiments are presented in Table 3.

Metric	Equation	Explanation
Test Error (%)	$\text{Error} = \frac{\text{Misclassified Samples}}{\text{Total Samples}}$	Fraction of misclassified samples.
Accuracy (%)	$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$	Fraction of correctly predicted samples.

Table 3: Equations for RFs metrics: Test Error and Accuracy.

Breiman's work [Bre01] evaluated RF on datasets including *Glass*, *Breast Cancer*, *Diabetes*, *Sonar*, and synthetic datasets (e.g., *Twonorm* and *Threenorm*). Table 4 presents the test set error comparisons between RF and Adaboost:

Dataset	Random Forest	Adaboost	Single Decision Tree
Glass	21.2%	22.0%	36.9%
Breast Cancer	2.7%	3.2%	6.3%
Diabetes	24.3%	26.6%	33.1%
Sonar	18.0%	15.6%	31.7%
Twonorm	3.9%	4.9%	24.7%
Threenorm	17.5%	18.8%	38.4%

Table 4: Random Forest vs. Adaboost and Single Decision Tree test errors on classification tasks [Bre01]

The results show that Random Forest consistently outperforms single decision trees by a wide margin on all datasets. For example, the Glass dataset error rate for a single tree is 36.9%, while RF reduces it to 21.2%. Similarly, for the Breast Cancer dataset, RF achieves an error of only 2.7%, compared to 6.3% for a single tree. Compared to Adaboost, RF remains competitive, often outperforming it slightly, as seen on datasets like *Diabetes* and *Twonorm*. This highlights RF's ability to balance bias and variance effectively.

Breiman also analyzed in his paper [Bre01] the robustness of RF to *output noise* by randomly flipping 5% of the class labels in the training set. Table 5 shows the error rate increases compared to Adaboost.

Dataset	RF (5% Noise)	Adaboost (5% Noise)
Glass	+0.4%	+6.2%
Breast Cancer	+0.3%	+43.2%
Diabetes	+1.0%	+10.2%
Sonar	+1.2%	+7.5%

Table 5: Robustness to 5% label noise: Random Forest (RF) vs Adaboost [Bre01].

This table demonstrates Random Forest's robustness to noise compared to Adaboost. For instance, on the Breast Cancer dataset, Adaboost's error increases drastically by 43.2% under 5% label noise, while RF's error increases by only 0.3%. Similarly, for the Glass dataset, RF's error rises marginally by 0.4%, whereas Adaboost's error increases by 6.2%. This highlights RF's resilience to noise, making it a more stable choice for real-world datasets where label inaccuracies are common.

Caruana et al. [CNM06] compared RF to other classifiers, including bagged trees, SVMs, and neural networks. Table 6 summarizes accuracy scores across multiple benchmarks.

Dataset	**Random Forest**	**Bagged Trees**	**SVM**	**Neural Net**
Adult	85.2%	84.1%	83.3%	85.0%
Covtype	94.3%	92.7%	93.2%	93.1%
Letter (balanced)	96.8%	95.5%	94.7%	96.0%
Medis	87.3%	84.5%	82.0%	86.2%

Table 6: Accuracy comparison: Random Forest vs Bagged Trees, SVM, and Neural Networks [CNM06].

Random Forest achieves the highest or near-highest accuracy across multiple datasets. For example, on the *Covtype dataset*, RF achieves 94.3% accuracy, outperforming bagged trees (92.7%) and SVMs (93.2%). Similarly, on the *Medis dataset*, RF achieves 87.3% accuracy, significantly higher than SVM (82.0%). Although neural networks perform comparably on some datasets (e.g., *Letter*), RF remains a strong contender, often requiring less tuning and achieving better generalization.

The results consistently show that Random Forests achieve high accuracy, are robust to noise, and outperform or match alternative methods like Adaboost, SVMs, and neural networks on benchmark datasets. These characteristics make Random Forests a great method for classification tasks.

3 Room Occupancy Estimation dataset

The creators of the Room Occupancy estimation dataset used a handful of supervised (Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Support Vector Machine (SVM), Random Forest (RF)) and unsupervised (Principal Component Analysis (PCA)) methods to train on the dataset [SJC+18]. A 10-fold cross-validation was used, and accuracy and F1 score were measured. The SVM features usage of two kernels (linear and RBF) and hyperparameter tuning [SJC+18]. In the case of random forest, the tuned hyperparameters were number of trees (chosen to be 30) and the minimum number of samples required to split an internal node [SJC+18].

Feature	Metric	LDA	QDA	SVM (Linear)	SVM (RBF)	RF
Temp{1,2,3,4}	A	0.840	0.862	0.866	0.895	0.869
	F1	0.479	0.590	0.554	0.730	0.657
Light{1,2,3,4}	A	0.973	0.919	0.973	0.973	0.972
	F1	0.928	0.854	0.929	0.927	0.925
Sound{1,2,3,4}	A	0.851	0.879	0.875	0.885	0.887
	F1	0.449	0.544	0.542	0.591	0.601
PIR {6,7}	A	0.869	0.869	0.870	0.870	0.870
	F1	0.474	0.474	0.466	0.460	0.460
CO2	A	0.809	0.808	0.812	0.812	0.763
	F1	0.383	0.409	0.286	0.314	0.329
Slope	A	0.852	0.831	0.870	0.870	0.876
	F1	0.387	0.394	0.462	0.510	0.564
CO2, Slope	A	0.891	0.867	0.890	0.888	0.873
	F1	0.556	0.590	0.592	0.635	0.559

Table 7: Experimental results on homogeneous fusion [SJC+18]

Table 7 reports the accuracy and F1 score of employed methods trained on homogeneous subsets of the feature space (e.g. only temperature, only sound, movement etc), regarded

Feature	Metric	LDA	QDA	SVM (Linear)	SVM (RBF)	RF
Temp{1,2,3,4}, CO2, Slope	A	0.903	0.881	0.904	0.912	0.894
	F1	0.653	0.680	0.667	0.750	0.684
Temp{1,2,3,4}, CO2,Slope, Sound{1,2,3,4}	A	0.920	0.908	0.933	0.924	0.918
	F1	0.735	0.749	0.793	0.782	0.731
Temp{1,2,3,4}, CO2, Slope,Sound{1,2,3,4}, PIR{6,7}	A	0.922	0.910	0.934	0.924	0.919
	F1	0.737	0.748	0.793	0.780	0.734
Complete dataset	A	0.980	0.957	0.982	0.984	0.978
	F1	0.946	0.911	0.948	0.953	0.933

Table 8: Experimental results on heterogeneous fusion [SJC+18]

	Predicted 0	Predicted 1	Predicted 2	Predicted 3
Actual 0	8117	43	41	27
Actual 1	104	336	19	0
Actual 2	65	48	502	133
Actual 3	21	7	154	512

Table 9: Confusion matrix for Linear SVM [SJC+18]

in the original approach as Phase 1 [SJC+18]. It shows that CO_2, light and temperature are key factors in estimating the number of people in the room [SJC+18]. Phase 2 results (heterogeneous fusion, i.e. progressively including more types of physical measurements into the system) are summarized in Table 8. It was noted that the entire dataset (which includes light) provides best results, with an F1 score of 0.95, as opposed to the missing light subsets, whose F1 scores do not get past 0.8 [SJC+18].

Tables 9 and 10 show the confusion matrices of both variants of SVM in Phase 2, complete dataset.

In the last phase of the experiment (Phase 3), a dimensionality reduction using PCA is performed. The study reports that 4 components are enough to achieve over 90% accuracy and 0.72 F1 score with RBF SVM (Figure 2) [SJC+18].

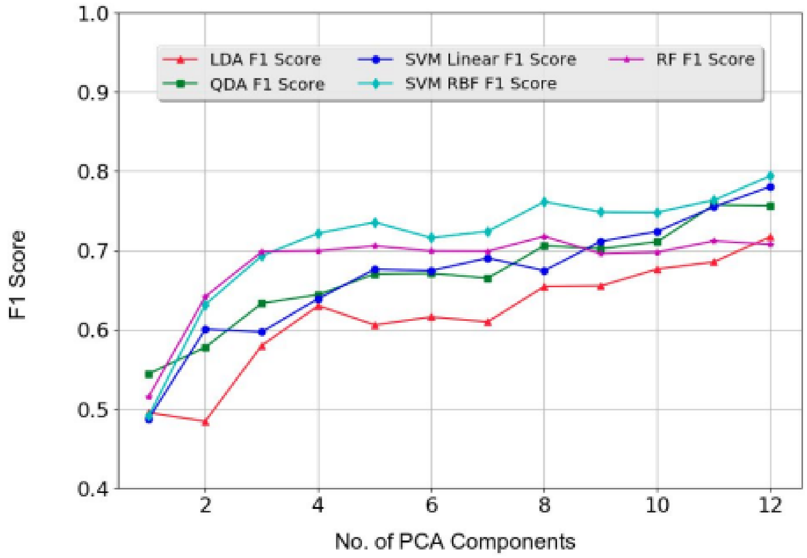

Figure 2: Relation between F1 score and number of PCA components [SJC+18]

	Predicted 0	Predicted 1	Predicted 2	Predicted 3
Actual 0	8196	1	3	28
Actual 1	0	453	6	0
Actual 2	0	0	712	36
Actual 3	10	1	67	616

Table 10: Confusion matrix for RBF SVM [SJC+18]

References

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[CNM06] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168, 2006.

[CP22] Carmela Comito and Clara Pizzuti. Artificial intelligence for forecasting and diagnosing covid-19 pandemic: A focused review. *Artificial Intelligence in Medicine*, 128:102286, 2022.

[DBK+96] Harris Drucker, Christopher J Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. *Advances in neural information processing systems*, 9, 1996.

[FLW+19] Jinwen Feng, Lei Liu, Di Wu, Guoyin Li, Michael Beer, and Wei Gao. Dynamic reliability analysis using the extended support vector regression (x-svr). *Mechanical Systems and Signal Processing*, 126:368–391, 2019.

[FWSG23] Yuan Feng, Di Wu, Mark G. Stewart, and Wei Gao. Past, current and future trends and challenges in non-deterministic fracture mechanics: A review. *Computer Methods in Applied Mechanics and Engineering*, 412:116102, 2023.

[HG20] Barenya Bikash Hazarika and Deepak Gupta. Modelling and forecasting of covid-19 spread using wavelet-coupled random vector functional link networks. *Applied Soft Computing*, 96:106626, 2020.

[SJC+18] Adarsh Pal Singh, Vivek Jain, Sachin Chaudhari, Frank Alexander Kraemer, Stefan Werner, and Vishal Garg. Machine learning-based occupancy estimation using multivariate sensor nodes. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE, 2018.

[ZL20] Youshan Zhang and Qi Li. *A Regressive Convolution Neural Network and Support Vector Regression Model for Electricity Consumption Forecasting*, pages 33–45. 01 2020.