House Price Prediction Using Advanced Regression Techniques

Lokesh Gupta

***Abstract*— With recent growth in amount of data around the world and increase in the computational capabilities, machine learning has flourished itself in tackling and solving many day to day problems in the industry. We take one such problem as part of our project. We take advanced regression techniques to predict sales prices of house based on given set of features. We train our models with best features selected by the feature selection approaches and derive the results and compare the accuracy of various models.**

I. INTRODUCTION

Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But there are many other factors that influence the price negotiations of the house other than the number of bedrooms or a white-picket fence. This project uses advanced regression techniques on Ames housing data set which consists of 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa. Size of the dataset used in this project is $[1460 \times 79]$, where 79 represents the features of 1460 observations.

The report is further explained as follows. In Section II we explain the models that we use and the methodology we adopt to tackle this problem. In Section III we show our results and the detailed analysis w.r.t the data and it's features. Finally, in Section IV we present our concluding remarks.

II. PROJECT METHODOLOGY

To start with the project, at the initial state, we read the data from out Dataset and perform data pre-processing which involved data cleaning and analysis. The analysis part focused on understanding the features and get an intuition of how features can effect the prices of the house. This is followed by Feature Selection to get best features out of pre-processed dataset. Based on the selected features, we perform our regression techniques and cross validate it to get Root Mean Squared Error (RMSE) and R^2 values. This has been graphically shown in Figure 1.

A. Reading the Dataset

Dataset is available in form of *.csv* files and for this project implemetation we use Python as our technology stack. So initially, we import this data and load it in from *.csv* files using Pandas library into dataframe (D) and perform all operations on it.

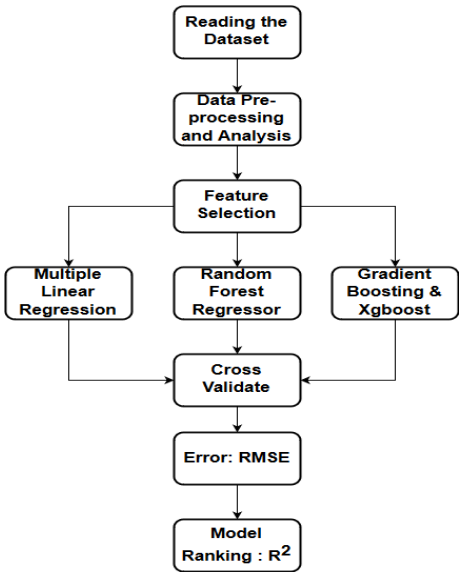

Fig. 1. Workflow Diagram

B. Data Pre-Processing & Analysis

We did a comprehensive, but not exhaustive, analysis of the data. This involves following steps:

1) **Univariable study**. We'll just focus on the dependent variable '*SalePrice*' and try to know a little bit more about it
2) **Multivariate study**. We'll try to understand how the dependent variable and independent variables relate.
3) **Basic cleaning**. We'll clean the dataset and handle the missing data, outliers and categorical variables.
4) **Test assumptions**. We'll check if our data meets the assumptions required by most multivariate techniques.

Based on the Univariate study of *SalePrice* we come to know that:

- *SalesPrice* deviates from the normal distribution.
- *SalesPrice* have appreciable positive skewness.
- Showed peakedness.

For this we calcuate numeric values and plot *SalesPrice* and we get following results which are shown in Figure 2 & 3. We also calculate the *Skewness*: 1.882876 & *Kurtosis*: 6.536282

After analysis of *SalesPrice*, we perform the trend of sales price with some intuitively important variables, i.e *GrLivArea*, *TotalBsmtSF*, *OverallQual* and *YearBuilt* for which we see following results.

From plot analysis, it seems that *SalePrice* and *GrLivArea* are having a linear relationship with each other. *SalesPrice*

```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```

Fig. 2. Summary of SalesPrice

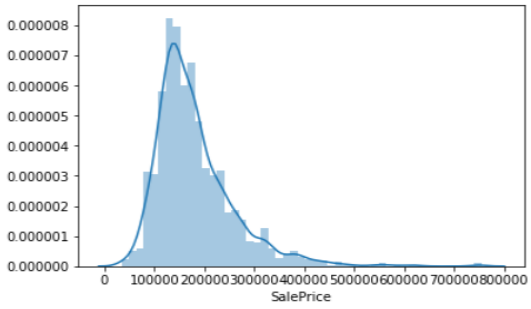

Fig. 3. Histogram of SalesPrice

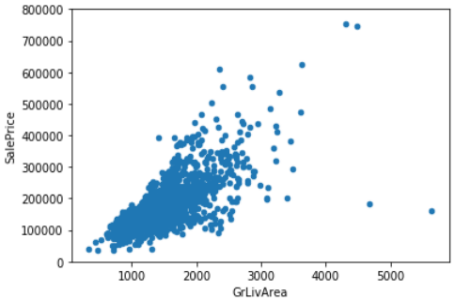

Fig. 4. SalesPrice vs GrLivArea

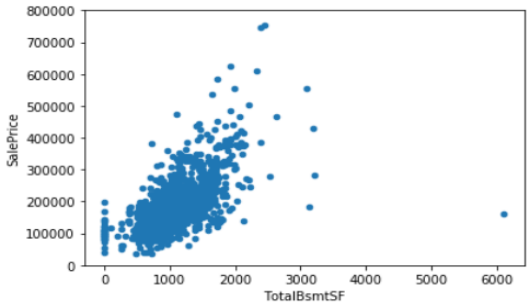

Fig. 5. SalesPrice vs TotalBsmtSF

shows strong exponential trend with *TotalBsmntSF*. We see a steady increase trend of *SalesPrice* with *OverallQual*. However, there is no as such trend observed in *YearBuild* variable. This concludes our multivariate analysis.

Now we head towards cleaning of missing data. For this we consider two major aspects:

- How prevalent is the missing data?

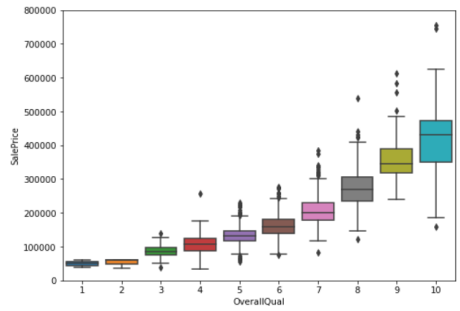

Fig. 6. SalesPrice vs OverallQual

- Is missing data random or does it have a pattern?

The answer to these questions is important for practical reasons because missing data can imply a reduction of the sample size. This can prevent us from proceeding with the analysis. Moreover, from a substantive perspective, we need to ensure that the missing data process is not biased and hiding an inconvenient truth. We'll consider that when more than 15% of the data is missing, we should delete the corresponding variable and pretend it never existed. This means that we will not try any trick to fill the missing data in these cases. According to this, there is a set of variables (e.g. 'PoolQC', 'MiscFeature', 'Alley', etc.) that we should delete. Based on the analysis we get following features to have major amount of data to be missing. And we continue to delete such data from the data-frame and refine our data. Some numerical values of missing data are presented in Table I.

Feature Name	Total	Percent ($\times 100$)
PoolQC	1453	0.995205
MiscFeature	1406	0.963014
Alley	1369	0.937671
Fence	1179	0.807534
FireplaceQu	690	0.472603
LotFrontage	259	0.177397
GarageCond	81	0.055479
GarageType	81	0.055479
GarageYrBlt	81	0.055479
GarageFinish	81	0.055479
GarageQual	81	0.055479

TABLE I
MISSING DATA STATISTICS

C. Feature Selection

Feature Selection is a prime paradigm which is used by a majority of statisticians and computer scientists to select a subset of relevant features (variables, predictors) which can be further used for model construction. This approach is mainly used because of the following reasons:

- There is a need to simplify a model for easy interpretation by researchers
- To reduce computational complexity while training the model

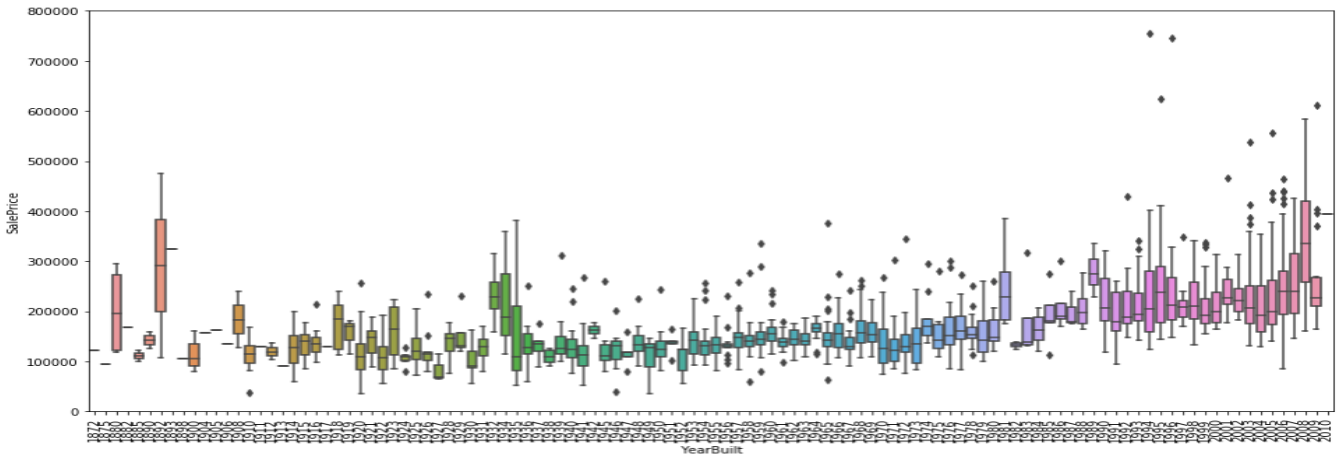

Fig. 7. SalesPrice vs YearBuilt

- To avoid the curse of dimensionality: A phenomena that arise when analyzing and organizing data in high-dimensional spaces (often with hundreds or thousands of dimensions) that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience
- To enhance generalization of features to reduce over-fitting.

For this project we consider **Random forest feature importance** as part of our feature selection process.

1) Random forest feature importance: Random forests are among the most popular machine learning methods thanks to their relatively good accuracy, robustness and ease of use. Random forest consists of a number of decision trees. Every node in the decision trees is a condition on a single feature, designed to split the dataset into two so that similar response values end up in the same set. The measure based on which the (locally) optimal condition is chosen is called impurity. For classification, it is typically either Gini impurity or information gain/entropy and for regression trees it is variance. Thus when training a tree, it can be computed how much each feature decreases the weighted impurity in a tree. For a forest, the impurity decrease from each feature can be averaged and the features are ranked according to this measure.

By using this Feature Selection approach we have got 8 important features out of 79 features which were originally present in the dataset. The list of features are given in Table II.

Moreover, we plotted an Heat Map between the important features and SalePrice which is shown in figure 8.

Feature Name	Feature Importance Score
OverallQual	0.5842086374734382
BsmtFinSF1	0.10878896114806227
TotalBsmtSF	0.042326149277154376
1stFlrSF	0.03811364385443915
2ndFlrSF	0.029863051439533567
GrLivArea	0.0231618310331448
GarageCars	0.021524472926555238
GarageArea	0.017626279963344014

TABLE II
FEATURE IMPORTANCE SCORES

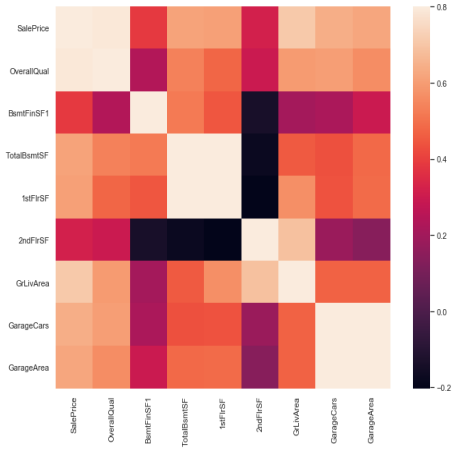

Fig. 8. Correlation Heat Map

D. Regression Techniques

In this project we have used the following regression techniques for *SalesPrice* prediction:

1) Multiple Linear Regression (MLR): Multiple linear regression is the most common form of linear regression analysis. As a predictive analysis, the multiple linear regression is used to explain the relationship between one continuous dependent variable and two or more independent variables. The independent variables can be continuous or categorical. Regression residuals must be normally distributed. A linear relationship is assumed between the dependent variable and the independent variables. At the center of the multiple linear regression analysis is the task of fitting a single line through a scatter plot. More specifically the multiple linear regression fits a line through a multi-dimensional space of data points. The simplest form has one dependent and two independent variables. The dependent variable may also be referred to

as the outcome variable or regressand. The independent variables may also be referred to as the predictor variables or regressors.

There are 3 major uses for multiple linear regression analysis. First, it might be used to identify the strength of the effect that the independent variables have on a dependent variable. Second, it can be used to forecast effects or impacts of changes. That is, multiple linear regression analysis helps us to understand how much will the dependent variable change when we change the independent variables. Third, multiple linear regression analysis predicts trends and future values. The multiple linear regression analysis can be used to get point estimates. When selecting the model for the multiple linear regression analysis, another important consideration is the model fit. Adding independent variables to a multiple linear regression model will always increase the amount of explained variance in the dependent variable (typically expressed as RÂš). Therefore, adding too many independent variables without any theoretical justification may result in an over-fit model.

2) Regularized Regression for MLR: Ridge and Lasso regression are powerful techniques generally used for creating parsimonious models in presence of a âĂŸlargeâĂŹ number of features. Here âĂŸlargeâĂŹ can typically mean either of two things:

- Large enough to enhance the tendency of a model to overfit (as low as 10 variables might cause overfitting).
- Large enough to cause computational challenges. With modern systems, this situation might arise in case of millions or billions of features.

Though Ridge and Lasso might appear to work towards a common goal, the inherent properties and practical use cases differ substantially. The key difference is in how they assign penalty to the coefficients.

a) Ridge Regression: Ridge regression is an extension for linear regression. ItâĂŹs basically a regularized linear regression model. The λ parameter is a scalar that should be learned as well, using a method called cross validation. A super important fact we need to notice about ridge regression is that it enforces the β coefficients to be lower, but it does not enforce them to be zero. That is, it will not get rid of irrelevant features but rather minimize their impact on the trained model.

- Performs L2 regularization, i.e. adds penalty equivalent to square of the magnitude of coefficients.
- Minimization objective $= LSObj + \alpha \times$ (sum of square of coefficients)

b) Lasso Regression: The only difference from Ridge regression is that the regularization term is in absolute value. But this difference has a huge impact on the trade-off. Lasso method overcomes the disadvantage of Ridge regression by not only punishing high values of the coefficients β but actually setting them to zero if they are not relevant. Therefore, you might end up with fewer features included in the model than you started with, which is a huge advantage.

- Performs L1 regularization, i.e. adds penalty equivalent to absolute value of the magnitude of coefficients
- Minimization objective $= LSObj + \alpha \times$ (sum of absolute value of coefficients)

Note that here *LSObj* refers to âĂŸleast squares objectiveâĂŹ, i.e. the linear regression objective without regularization.

3) Random Forest Regressor: The Random Forest is one of the most effective machine learning models for predictive analytics, making it an industrial workhorse for machine learning. Random forest acts as an additive model that can make predictions by combining decisions from a sequence of base models. We can describe this as class of models as we can see from equation 1.

$$g(x) = f_0(x) + f_1(x) + f_2(x) + ... \quad (1)$$

where the final model **g** is the sum of simple base models. Here each base classifier is a simple decision tree. This broad technique of using multiple models to obtain better predictive performance is called model ensembling. In random forests, all the base models are constructed independently using a different subsample of the data. Thus, in ensemble terms, the trees are weak learners and the random forest is a strong learner. Here is how such a system is trained, for some number of trees **T**.

1) Sample *N* cases at random to create a subset of the data. The subset should be approximately more than half of the total set.
2) At each node:
   a) For some number m, *m* predictor variables are selected at random from all the predictor variables.
   b) The predictor variable that provides the best split, according to the residual sum of squares shown in equation 2, is used to do a binary split on that node.

$$RSS = \Sigma_{left}(y_i - y_L^*)^2 + \Sigma_{right}(y_i - y_R^*)^2 \quad (2)$$

   c) At the next node, choose another *m* variable at random from all predictor variables and do the same.

Depending upon the value of *m*, there are three slightly different systems:

- Random splitter selection: m = 1
- Breiman's bagger: m = total number of predictor variables.
- Random Forest: m << number of predictor variables.Briemann suggests three possible values for m: $\frac{1}{2}\ \sqrt{m}, \sqrt{m}, and 2\sqrt{m}$

When a new input is entered into the system, it is run down all of the trees. The result may either be an average of weighted average of all of the terminal nodes that are reached in the case of regression.However we should note that:

- With a large number of predictors, the eligible predictor set will be quite different from node to node.

- The greater the inter-tree correlation, the greater the random forest error rate, so one pressure on the model is to have the tree as uncorrelated as possible.
- As *m* goes down, both inter-tree correlation and the strength of individual trees go down. So some optimal value of *m* must be discovered.

4) Gradient Boosting: Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function. The objective of any supervised learning algorithm is to define a loss function and minimize it. So in the case of gradient boosting we have a mean squared error (MSE) as loss defined as:

$$Loss = MSE = \Sigma(y_i - y_i^p)^2 \tag{3}$$

where, y_i is the ith target value, y_i^p is the ith prediction, and $L(y_i, y_i^p)$ is the loss function. The predictions produced should be such that the loss function (MSE) is minimum. By using gradient descent and updating our predictions based on a learning rate, we can find the values where the MSE is minimum.

$$y_i^p = y_i^p + \alpha * \delta\Sigma(y_i - y_i^p)^2 l\delta y_i^p \tag{4}$$

$$y_i^p = y_i^p - \alpha * 2 * \Sigma(y_i - y_i^p) \tag{5}$$

We can simplify the gradient descent algorithm shown in equation 4 to a simpler form formally defined in equation 5, where α is the learning rate and $\Sigma(y_i - y_i^p)$ is sum of residuals. So these equations will help update the predictions such that the sum of our residuals is close to 0 (minimum) and predicted values are sufficiently close to actual values. So, the intuition behind gradient boosting algorithm is to repetitively leverage the patterns in residuals and strengthen a model with weak predictions and make it better. Once we reach a stage that residuals do not have any pattern that could be modeled, we can stop modeling residuals otherwise it may lead to overfitting. Algorithmically, we are minimizing our loss function, such that the test loss reaches its minima.

III. RESULTS & ANALYSIS

After performing all the Regression Techniques which are explained in the above section, we have come up with results. Each of Regression technique was evaluated by using **10 - Fold Cross Validation** and the error is reported in terms of **Root Mean Squared Error (RMSE)**. Moreover, we also computed the R^2, coefficient of determination, for each of the regression model. All these results are provided in the tables given below.

R^2 is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

The definition of R^2 is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model. Or:

R^2 = Explained variation / Total variation

R-squared is always between 0 and 100%:

- 0% indicates that the model explains none of the variability of the response data around its mean.
- 100% indicates that the model explains all the variability of the response data around its mean.

	Training Set	Testing Set
RMSE	31863.650	29941.922
R^2	0.816	0.869

TABLE III
RANDOM FOREST REGRESSION RESULTS

	Training Set	Testing Set
RMSE	39714.368	34142.047
R^2	0.722	0.774

TABLE IV
MULTIPLE LINEAR REGRESSION RESULTS

	Training Set	Testing Set
RMSE	39318.730	27542.379
R^2	0.706	0.856

TABLE V
RIDGE REGULARIZED RESULTS

	Training Set	Testing Set
RMSE	39968.510	28355.278
R^2	0.698	0.846

TABLE VI
LASSO REGULARIZED RESULTS

	Training Set	Testing Set
RMSE	35361.419	31063.649
R^2	0.771	0.852

TABLE VII
GRADIENT BOOSTING RESULTS

The **RMSE** is the square root of the variance of the residuals. It indicates the absolute fit of the model to the data how close the observed data points are to the model's predicted values. Whereas R^2 is a relative measure of fit, RMSE is an absolute measure of fit. As the square root of a variance, RMSE can be interpreted as the standard deviation of the unexplained variance, and has the useful property of being in the same units as the response variable. Lower values of RMSE indicate better fit. RMSE is a good measure of how accurately the model predicts the response, and it is the most important criterion for fit if the main purpose of the model is prediction.

Firstly, considering the **RMSE** values which are obtained from various regression models, we can observe that the

magnitude is high. This is because of the magnitude of the Predictor i.e. *SalesPrice*. Since the SalesPrice lies between 34908 to 755000.

Secondly, considering the R^2 values from the above tables we can see that, for all the regression models the R^2 value for testing set is > 75% which signifies that all the models are able to capture the variance in the dataset properly and will be able to predict the future values with reasonable error.

The parameters used for building the above models are given below:

- Random Forest Regressor : n_estimaters = 1000
- Multiple Linear Regressor : degree = 1
- Ridge Regression : $\alpha = 84$
- Lasso Regression : $\alpha = 1.4$
- Gradient Boosting : n_estimators = 1000, $\lambda = 0.1$

IV. CONCLUSION

In the end we would like to conclude that we have found 8 prominent features from the Ames Dataset which are listed in table II. The results obtained by using different regression techniques are given in Section III.

We can also conclude that all the regression models performed very well which is evident from their R^2 score and RMSE values. According to our results, Random Forest Regressor performed the best among the other techniques with $R^2 = 86.9\%$ which is very closely followed by the Gradient Boosting technique with $R^2 = 85.2\%$. All these scores are computed on the Testing dataset.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to our course instructor **Dr. Alioune Ngom** for his continuous support in this course. His guidance helped us throughout our time of research and implementation of this project. We could not have imagined having better mentor for our study.

Secondly, we would like to thank **Dean De Cock** for compiling The Ames Housing dataset. It's an incredible alternative for data scientists looking for a modernized and expanded version of the often cited Boston Housing dataset.

REFERENCES

[1] G. O. Young, ÃŠSynthetic structure of industrial plastics (Book style with paper title and editor),ÃŞ in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15ÃŘ64.

[2] W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123ÃŘ135.

[3] H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.

[4] B. Smith, ÃŠAn approach to graphs of linear forms (Unpublished work style),ÃŞ unpublished.

[5] E. H. Miller, ÃŠA note on reflector arrays (Periodical styleÃŚAccepted for publication),ÃŞ IEEE Trans. Antennas Propagat., to be publised.

[6] J. Wang, ÃŠFundamentals of erbium-doped fiber amplifiers arrays (Periodical styleÃŚSubmitted for publication),ÃŞ IEEE J. Quantum Electron., submitted for publication.

[7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, ÃŠElectron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style),ÃŞ IEEE Transl. J. Magn.Jpn., vol. 2, Aug. 1987, pp. 740ÃŘ741 [Dig. 9th Annu. Conf. Magnetics Japan, 1982, p. 301].

[9] M. Young, The Techincal Writers Handbook. Mill Valley, CA: University Science, 1989.

[10] J. U. Duncombe, ÃŠInfrared navigationÃŚPart I: An assessment of feasibility (Periodical style),ÃŞ IEEE Trans. Electron Devices, vol. ED-11, pp. 34ÃŘ39, Jan. 1959.

[11] S. Chen, B. Mulgrew, and P. M. Grant, ÃŠA clustering technique for digital communications channel equalization using radial basis function networks,ÃŞ IEEE Trans. Neural Networks, vol. 4, pp. 570ÃŘ578, July 1993.

[12] R. W. Lucky, ÃŠAutomatic equalization for digital communication,ÃŞ Bell Syst. Tech. J., vol. 44, no. 4, pp. 547ÃŘ588, Apr. 1965.

[13] S. P. Bingulac, ÃŠOn the compatibility of adaptive controllers (Published Conference Proceedings style),ÃŞ in Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory, New York, 1994, pp. 8ÃŘ16.

[14] G. R. Faulhaber, ÃŠDesign of service systems with priority reservation,ÃŞ in Conf. Rec. 1995 IEEE Int. Conf. Communications, pp. 3ÃŘ8.

[15] W. D. Doyle, ÃŠMagnetization reversal in films with biaxial anisotropy,ÃŞ in 1987 Proc. INTERMAG Conf., pp. 2.2-1ÃŘ2.2-6.

[16] G. W. Juette and L. E. Zeffanella, ÃŠRadio noise currents n short sections on bundle conductors (Presented Conference Paper style),ÃŞ presented at the IEEE Summer power Meeting, Dallas, TX, June 22ÃŘ27, 1990, Paper 90 SM 690-0 PWRS.

[17] J. G. Kreifeldt, ÃŠAn analysis of surface-detected EMG as an amplitude-modulated noise,ÃŞ presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.

[18] J. Williams, ÃŠNarrow-band analyzer (Thesis or Dissertation style),ÃŞ Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

[19] N. Kawasaki, ÃŠParametric study of thermal and chemical nonequilibrium nozzle flow,ÃŞ M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.

[20] J. P. Wilkinson, ÃŠNonlinear resonant circuit devices (Patent style),ÃŞ U.S. Patent 3 624 12, July 16, 1990.