Inferences from Some Hybrid Prediction Models with Applications

by

Tanujit Chakraborty

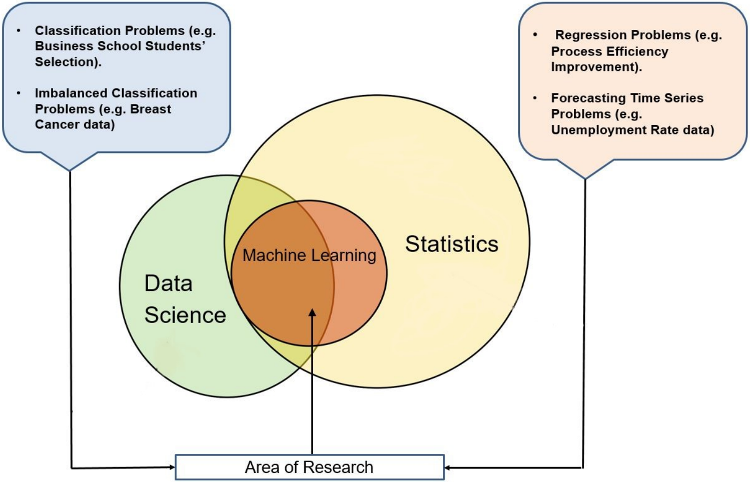
Senior Research Fellow
Statistical Quality Control & Operations Research Unit
Indian Statistical Institute, Kolkata.

June 14, 2019

Outline of the talk

1. Area of Research
2. Motivation of the Thesis
3. Thesis Overview
4. Acknowledgements
5. References

Area of Research



- Classification Problems (e.g. Business School Students' Selection).
- Imbalanced Classification Problems (e.g. Breast Cancer data)
- Regression Problems (e.g. Process Efficiency Improvement).
- Forecasting Time Series Problems (e.g. Unemployment Rate data)
Data Science
Machine Learning
Statistics
Area of Research

"**Statistics** is the universal tool of inductive inference, research in natural and social sciences, and technological applications. Statistics, therefore, must always have purpose, either in the pursuit of knowledge or in the promotion of human welfare" - **P.C. Mahalanobis, 1956**

Motivation of the Thesis

- "Prediction is very difficult, especially if it's about the future"
  - **Niels Bohr**, Father of Quantum Mechanics.

- Predictive modelling approaches are used in the fields of statistics and machine learning, mainly for their accuracy and ability to deal with complex data structures.

- This thesis studies the use of multiple models (hybrid or ensembles) in prediction problems from the area of Business Analytics, Quality Control, and Data Science.

- We developed new hybrid models for finding solutions to these problems:

  1. Feature Selection cum Classification Problem
  2. Imbalanced Classification Problem
  3. Nonparametric Regression Problem
  4. Time Series Forecasting Problem

- Both theoretical (statistical inferences) and practical (computational) aspects of combining models are studied.

Thesis Overview

- Chapter 1: Introduction
- Chapter 2: Dean's Dilemma Problem: A Hybrid Classifier
- Chapter 3: Imbalanced Classification Problem: Hellinger Nets Model
- Chapter 4: Process Efficiency Improvement Problem: A Hybrid Model
- Chapter 5: Forecasting Time Series: A Hybrid Approach
- Chapter 6: Conclusions

Chapter 1: Introduction

Introduction: Developments of Prediction Models

- Linear Regression (Galton, 1875).
- Linear Discriminant Analysis (R.A. Fisher, 1936).
- Logistic Regression (Berkson, JASA, 1944).
- k-Nearest Neighbor (Fix Hodges, 1951).
- Parzens Density Estimation (E Parzen, AMS, 1962)
- Classification and Regression Tree (Breiman et al., 1984).
- Artificial Neural Network (Rumelhart et al., 1985).
- Perceptron Trees (Paul Utgoff, 1989, Connection Science).
- MARS (Friedman, 1991, Annals of Statistics).
- SVM (Cortes Vapnik, Machine learning, 1995)
- Random forest (Breiman, 2001).
- Deep Convolutional Neural Nets (Krizhevsky, Sutskever, Hinton, NIPS 2012).
- Generative Adversarial Nets (Ian Goodfellow et al., NIPS 2014).
- Deep Learning (LeCun, Bengio, Hinton, Nature 2015).
- Bayesian Deep Neural Network (Yarin Gal, Islam, Zoubin Ghahramani, ICML 2017).

Introduction: A Classification Problem

- Statistical learning theory (SLT) studies mathematical foundations for machine learning models, originated in late 1960s.
- Input space (object space): X ; Output space (label space): Y
- The task: to classify objects in X into categories in Y
- Binary classification: to classify objects in X into 2 classes in label space Y $= \{0, 1\}$.
- Given (object, label), The goal: to find a classifier $f : X \to Y$ to predict the label of new object X
- A learning algorithm L: inputs training data, outputs a classifier f
- No assumption is made on the joint probability distribution of data μ.
- The goal is to learn a classifier $f : X \to Y$: "how good" a function f is when used as a classifier?

Introduction: Loss Function and Risk

- Introduce a loss function: Given $(X, Y) \in \mathbb{R}^d \times \{0, 1\}$, an unknown μ and a classifier $f : \mathbb{R}^d \to \{0, 1\}$, the loss function is defined by: $l_\mu(X, Y, f(X)) = \mu\{f(X) \neq Y\}$.

- The risk or misclassification error is the average loss over all $X \in \mathbb{R}^d$ $R(f) := E(l(X, Y, f(X)))$

- The risk counts how many elements of the instance space X are mis-classified by the classifier f. Smaller the risk, better the classifier.

- The Bayes error is the smallest possible risk over all possible classifiers: $R^* = R^*(\mu) = inf_f\{R_\mu(f)\}$.

  Given μ, the optimal classifier - Bayes classifier is defined as:

$$f_{Bayes}(x) = \begin{cases} 0, & \text{if } \psi(x) \geq 1/2. \\ 1, & \text{otherwise.} \end{cases} \tag{0.1}$$

- It is impossible to compute the Bayes classifier: μ is unknown, but we need to evaluate $\psi(x) = \mu(Y = 1|X = x)$.

Introduction: Consistency and Universal Consistency

Consistency: A learning rule, when presented more and more training examples, → the optimal solution.

Definition (Consistent)

Given an infinite sequence of training points $(X_i, Y_i)_{i \in N}$ with μ. For each $n \in N$, let f_n be a classifier for the first n training points. The learning algorithm is called consistent with respect to μ if the risk $l(f_n)$ converges to the risk $l(f_{Bayes})$, that is for all $\epsilon > 0$,

$$\mu(R(f_n) - R(f_{Bayes}) > \epsilon) \rightarrow 0 \textit{ as } n \rightarrow \infty.$$

Definition (Universally Consistent)

The learning algorithm is called universally consistent if it is consistent for all probability distributions μ.

Introduction: Decision Trees

- Decision tree is defined by a hierarchy of rules (in form of a tree).
- Rules from the internal nodes of the tree are called root nodes
- Each rule (internal node) tests the value of some feature.
- Labeled training data is used to construct the Decision tree. The tree need not to be always a binary tree.
- CART (Breiman et al., 1984), RF (Breiman, 2001), BART (Chipman et al., 2010).

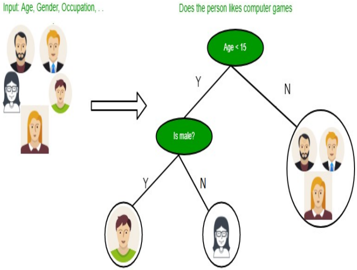

Input: Age, Gender, Occupation, . .
Does the person likes computer games
Age < 15
Y
N
Is male?
Y
N

Fig: An example of a Classification Tree

Introduction: Decision Trees

- CART is a greedy divide-and-conquer algorithm. Trees are constructed in a top-down recursive manner based on selected attributes.
- Attributes are selected on the basis of an impurity function (e.g., IG for Classification MSE for Regression).
- **Pros:** Built-in feature selection mechanism, Comprehensible, easy to design, easy to implement, good for structural learning.
- **Cons:** But may become large for complex problems, too many rules loose interpretability, risk of over-fitting, sticking to local minima.

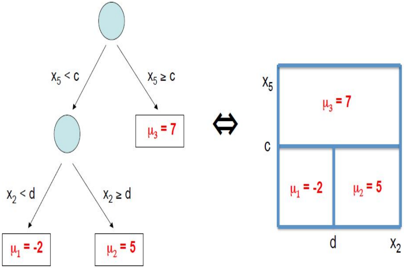


Fig: An example of a Regression Tree

Introduction: Statistical Theory to Decision Trees

- Consistency of data driven histogram methods (Nobel, 1996, Annals of Statistics).
- A Fast, Bottom-Up Decision Tree Pruning Algorithm with Near-Optimal Generalization (Kearns, Mansour, ICML, 1998)
- Generalization Bounds for Decision Trees (Mansour et al., 2000, COLT).
- Analysis of a Complexity-Based Pruning Scheme for CT (Nobel, 2002, IEEE Information Theory).
- Consistency of Online Random Forest (Denil et al., 2013, ICML).
- Consistency of Random Forest (Scornet et al., 2015, Ann. Stat.).

Introduction: Consistency of data driven histogram methods

Theorem (Lugosi, Nobel, 1996, Annals of Statistics)

Let $(\underline{X}, \underline{Y})$ be a random vector taking values in $\mathbb{R}^p \times C$ and L be the set of first n outcomes of $(\underline{X}, \underline{Y})$. Suppose that Φ is a partition and classification scheme such that $\Phi(L) = (\psi_{pl} \circ \phi)(L)$, where ψ_{pl} is the plurality rule and $\phi(L) = (L)_{\tilde{\Omega}_n}$ for some $\tilde{\Omega}_n \in \mathcal{T}_n$, where $\mathcal{T}_n = \{\phi(\ell_n) : P(L = \ell_n) > 0\}$. Also suppose that all the binary split functions in the question set associated with Φ are hyperplane splits. As $n \to \infty$, if the following regularity conditions hold:

$$\frac{\lambda(\mathcal{T}_n)}{n} \to 0 \tag{0.2}$$

$$\frac{log(\triangle_n(\mathcal{T}_n))}{n} \to 0 \tag{0.3}$$

and for every $\gamma > 0$ and $\delta \in (0, 1)$,

$$\inf_{S \subseteq \mathbb{R}^p : \eta_x(S) \geq 1-\delta} \eta_x(x : diam(\tilde{\Omega}_n[x] \cap S) > \gamma) \to 0 \tag{0.4}$$

with probability 1. then Φ is risk consistent.

- Equation (0.2) is the sub-linear growth of the number of cells, Equation (0.3) is the sub-exponential growth of a combinatorial complexity measure, and Equation (0.4) is the shrinking cell condition.
- The process defined above is binary in the sense that each application of the function ϕ splits each node in a partition into two or fewer child nodes.

Introduction: Developments of Neural Nets

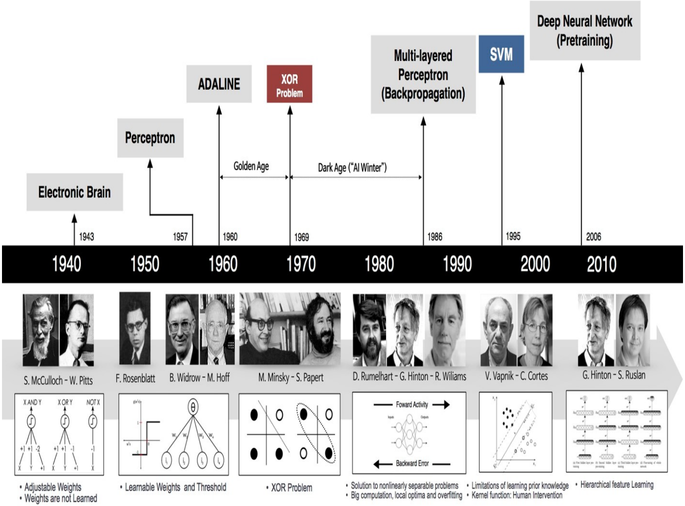


Fig: Developments of Neural Network Models

Introduction: Artificial Neural Networks

- ANN is composed of several perceptron-like units arranged in multiple layers.
- Consists of an input layer, one or more hidden layer, and an output layer.
- Nodes in the hidden layers compute a nonlinear transform of the inputs.
- Also called a Feedforward Neural Network (since there is no backward connections between layers, viz., no loops).
- **Universal Approximation Theorem (Hornik, 1989)**: A one hidden layer FFNN with sufficiently large number of hidden nodes can approximate any function.

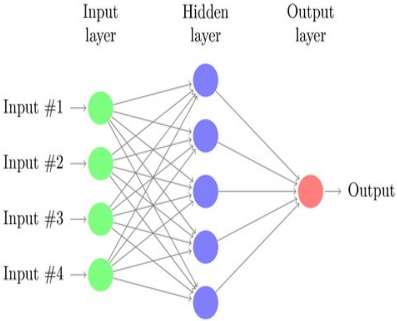


Fig: FFNN Model with one hidden layer with 5 hidden units

Introduction: Pros & Cons of Neural Nets

- Able to learn any complex nonlinear mapping or approximate any continuous function.
- As a nonparametric model, it doesn't make any prior assumption about the data distribution or input-output mapping function.
- ANN are very flexible with respect to incomplete, missing and noisy data. ANN are "fault tolerant".
- Neural network models can be easily updated / are suitable for dynamic environment.

- Neural network when applied to limited data can overfit the training data and lose generalization capability.
- Training process of ANN is very time-consuming due to having huge number of weights and hyperparameters.
- The selection of the network topology and its parameter lack theoretical background, it is often a "trial and error" matter.
- Advanced ANNs lack theoretical background concerning explanatory capabilities and results in "black-box" model.

Introduction: Statistical Theory to Neural Networks

- Strong Universal Consistency of ANN Classifier (Farago, Lugosi, IEEE IT 1993).
- Approximation properties of ANN (Mhaskar, Advances in Computational Mathematics, 1993).
- Prediction Intervals for Artificial Neural Networks (Hwang, Ding, 1997, JASA)
- Nonasymptotic bounds on the L_2 error of ANN regression estimates (Hamers & Kohler, AISM, 2006).
- Provable approximation properties for DNN (Shaham et al., Applied & Computational Harmonic Analysis, 2018).
- On Deep Learning as a remedy for the curse of dimensionality (Bauer, Kohler, Annals of Statistics, 2019).

Introduction: Consistency Results for Neural Network Classifier

Definition ($L_1 error$)

Define the L_1 error of a function $f : \mathbb{R}^d \to \mathbb{R}$ by

$$J(f) = \mathbb{E}\{|f(X) - Y|\big|Data\}$$

Theorem (Lugosi & Zeger, 1995, IEEE Information Theory)

Consider a neural network with one hidden layer with bounded output weight having k hidden neurons and let σ be a logistic squasher. Let $F_{n,k}$ be the class of neural networks defined as

$$F_{n,k} = \left\{ \sum_{i=1}^{k} c_i \sigma(a_i^T z + b_i) + c_0 : k \in \mathbb{N}, a_i \in \mathbb{R}^{dm}, b_i, c_i \in \mathbb{R}, \sum_{i=0}^{k} |c_i| \leq \beta_n \right\}$$

and let ψ_n be the function that minimizes the empirical L_1 error over $\psi_n \in F_{n,k}$. It can be shown that if k and β_n satisfy

$$k \to \infty, \quad \beta_n \to \infty, \quad \frac{k\beta_n^2 log(k\beta_n)}{n} \to 0$$

then the classification rule

$$g_n(z) = \begin{cases} 0, & \text{if } \psi_n(z) \leq 1/2. \\ 1, & \text{otherwise.} \end{cases} \tag{0.5}$$

is universally consistent.

Introduction: Ensemble Models [Murphy Book, 2012]

- Problem: Single classifiers have the drawbacks of sticking to local minimum or over-fitting the data set, etc.
- Ensemble models are such where predictions of multiple models are combined together to build the final model.
- Examples: Bagging, Boosting, Stacking and Voting Method
- Caution: But ensembles dont always improve accuracy of the model but tends to increase the error of each individual base classifier.

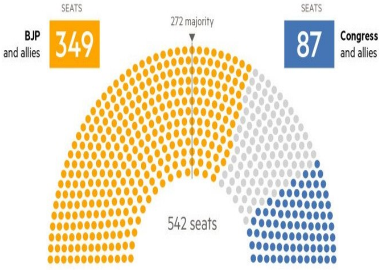


Fig: Election Result of 2019 in India in which alliance failed

Introduction: Hybrid Models [Kuncheva Book, 2013]

- Hybrid models are such where more than one models are combined together.
- It overcomes the limitations of single models and reduce individual variance bias, thus improve the performance of the model.
- Caution: To build a good ensemble classifier the base classifier needs to be simple, as accurate as possible, and distinct from the other classifier used.
- Desired: Interpretability, Less Complexity, Less Tuning Parameters, **high accuracy**.

Fig: "Alone we can do so little; together we can do so much". - Helen Keller

Introduction: Popular Hybrid Prediction Model

- Perceptron Trees (Utgoff, AAAI, 1988).
- Entropy Nets (Sethi, Proceeding of IEEE,1990).
- Neural trees (Sirat & Nadal, Network, 1990).
- Sparse Perceptron Trees (Jackson, Craven, NIPS, 1996).
- SVM Tree Model (Bennett et al., NIPS, 1998)
- Hybrid DT-ANN Model (Jerez-Aragones et al., 2003, AI in Medicine)
- Flexible Neural Tree (Chen et al., Neurocomputing, 2006)
- Hybrid DT-SVM Model (Sugumaran et al,, Mechanical Systems and Signal Processing, 2007).
- Hybrid CNNSVM Classifier (Niu et al., PR, 2012).
- Convolutional Neural Support Vector Machines (Nagi et al., IEEE ICMLA, 2012).
- Hybrid DT model utilizing local SVM (Dejan et al., IJPR, 2013).
- Neural Decision Forests (Bulo, Kontschieder, CVPR, 2014).
- Deep Neural Decision Forests (Kontschieder, ICCV, 2015).
- Soft Decision Tree (Frosst, Hinton, Google AI, 2017).
- Deep Neural Decision Trees (Yang et al., ICML, 2018).

Chapter 2: Dean's Dilemma Problem: A Hybrid Classifier

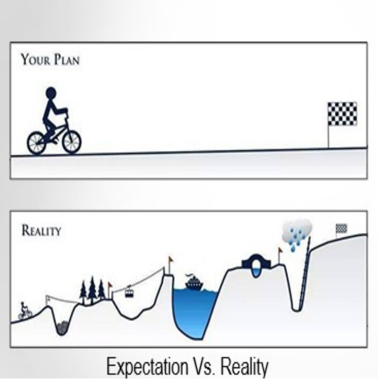
Publications:

1. Tanujit Chakraborty, Swarup Chattopadhyay, and Ashis Kumar Chakraborty. "A novel hybridization of classification trees and artificial neural networks for selection of students in a business school", **Opsearch**, Springer. 55 (2018): 434-446.

2. Tanujit Chakraborty, Ashis Kumar Chakraborty, and C. A. Murthy. "A nonparametric ensemble binary classifier and its statistical properties", **Statistics & Probability Letters**, Elsevier. 149 (2019): 16-23.

Problem Statement

- Placement of MBA student is a serious concern for Private B-Schools.
- The data is collected from a private business school which receives applications from across the country for the MBA program and admits a pre-specified number of students every year.
- Authorities want us to come up with a model that can help them to predict whether a student will be placed or not on certain characteristics of that students provided at the time of admission.
- Selecting a wrong student may increase the number of unplaced students. Also, more the number of unplaced students more is the negative impact on the institutes reputation.



Business School Data

- The data set comprises of several parameters of passed out students profile (collected at the time of admission) along with their placement information (collected at the end of the MBA program).
- The data set comprise of several parameters of passed out students' profile along with their placement information (on average 60% students got placed in last 5 years).
- It is desired to build a classifier which can also find out a set of important student characteristics from the data set.
- The data contains 24 explanatory variables out of which 7 are categorical variables. The response variable (Placement) indicate whether the student got placed or not.

Table: Sample business school data set.

ID	Gender	SSC Percentage	HSC Percentage	DEGREE Percentage	E.Test Percentile	SSC Board	HSC Board	HSC Stream	Placement
1	M	68.4	85.6	72	70	ICSE	ISC	Commerce	Y
2	M	59	62	50	79	CBSE	CBSE	Commerce	Y
3	M	65.9	86	72	66	Others	Others	Commerce	Y
4	F	56	78	62.4	50.8	ICSE	ISC	Commerce	Y
5	F	64	68	61	24.3	Others	Others	Commerce	N
6	F	70	55	62	89	Others	Others	Science	Y
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.

Scope of the Problem

- **Goal**: We would like to come up with a model that can help the authorities of a business school to predict whether a student will be placed or not based on certain characteristics of that student at the time of admission to the professional course.
- **Scope**: Feature Selection (selection of important students' characteristics) cum data classification (a system that will give judgements based on the characteristics of new applicants to their MBA program).
- **Previous works**: Dean's dilemma problem is very popular in Educational data mining. There are various literature available in the field where data mining techniques like logistic regression, LDA, DT, ANN, kNN, SVM, RF, etc have been employed to model students' admission, students' placements.
- Pena-Ayala A (2014) **Educational data mining: A survey and a data mining-based analysis of recent works**. Expert systems with applications, Elsevier, 41(4):14321462 provides a survey of all the techniques used in similar problems.

Proposed Hybrid Model

- First, apply classification tree algorithm to train and build a decision tree model that extracts important features.
- Feature selection model is generated by decision tree and it also shortlists the important features and filters out the rest.
- The prediction result of CT algorithm is used as an additional feature in the input layer of ANN model.
- Export important input variables along with additional input variable to the appropriate ANN model and network is generated.
- Run ANN algorithm till satisfactory accuracy is reached by optimizing weights and number of hidden layer neurons. Then the classifier will be ready to use.

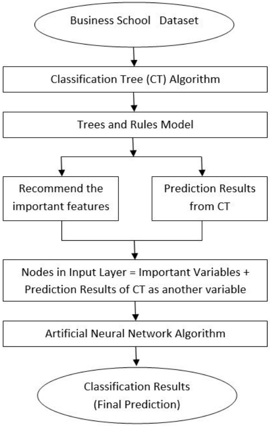

Business School Dataset
Classification Tree (CT) Algorithm
Trees and Rules Model
Recommend the important features
Prediction Results from CT
Nodes in Input Layer = Important Variables + Prediction Results of CT as another variable
Artificial Neural Network Algorithm
Classification Results (Final Prediction)

Fig: Flowchart of the Proposed Hybrid Model

Experimental Evaluation on Business School Data

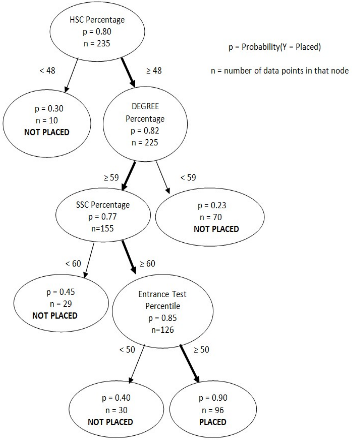


Fig: Decision Tree Diagram

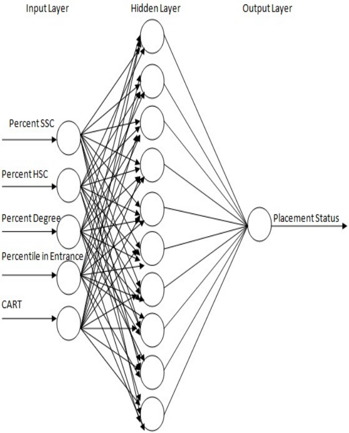


Fig: Hybrid CT-ANN Model Diagram

Performance Evaluation

Popularly used performance metric are:

Precision=$\frac{TP}{TP+FP}$; Recall=$\frac{TP}{TP+FN}$;

F-measure $=2\frac{(Precision.Recall)}{(Precision+Recall)}$; Accuracy $=\frac{(TP+TN)}{(TP+TN+FP+FN)}$;

TP (True Positive): correct positive prediction; FP (False Positive): incorrect positive prediction; TN (True Negative): correct negative prediction; FN (False Negative): incorrect negative prediction.

Table: Quantitative measure of performance for different classifiers.

Classifier	Precision	Recall	F-measure	Accuracy (%)
LR	0.964	0.794	0.871	77.143
LDA	0.964	0.794	0.871	77.143
kNN	0.800	1.000	0.889	80.000
SVM	0.964	0.771	0.857	75.000
RF	0.823	1.000	0.903	82.857
CART	0.823	1.000	0.903	83.333
ANN	0.928	0.812	0.867	77.142
Neural Trees	0.918	0.894	0.906	85.169
Entropy Nets	0.839	0.928	0.881	80.555
Proposed Hybrid CT-ANN	**0.942**	**0.970**	**0.956**	**91.667**

Merits & Possible Extensions

- **Merits**:

  1. Can select important features from the data set;
  2. Performs better than CART & ANN and easy interpretability;
  3. Suitable for Feature Selection cum Classification Problems with limited data sets;
  4. Useful for high dimensional feature spaces in the data sets;
  5. Easy interpretability, “white-box” model, fast in implementing.

- **Possible Extensions**:

  1. Theoretical Consistency of the Model?
  2. Optimal Choice of the number of hidden nodes for the model?
  3. Can this model be useful for practitioner working in other disciplines but on similar types of problems?

Improved Version of the Proposed Model

- First, apply the CT algorithm to train and build a decision tree and record important features.
- Using important input variables obtained from CT along with an additional input variable (CT output), a neural network is generated.
- The optimum number of neurons in the hidden layer of the model to be chosen as $O\left(\sqrt{n/d_m log(n)}\right)$ [to be discussed], where n, d_m are number of training samples and number of input features in ANN model, respectively.

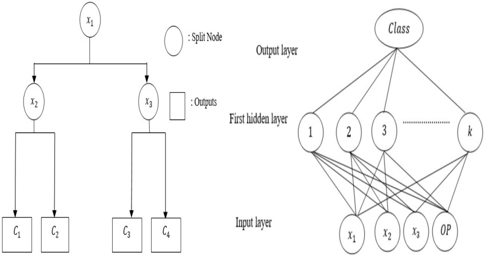


Figure: Graphical Presentation of the proposed Hybrid Model

On Theoretical Consistency

- A consistent rule guarantees us that taking more samples essentially suffices to roughly reconstruct the unknown distribution of (X, Y).
- In other words, infinite amounts of information can be gleaned from finite samples. Without this guarantee, we would not be motivated to take more samples.
- We should be careful and not impose conditions on (X, Y) for the consistency of a rule, because such conditions may not be verifiable. If a rule is consistent for all distributions of (X, Y), it is said to be universally consistent.
- A binary tree-based classification and partitioning scheme Φ is defined as an assignment rule applied to the limit of a sequence of induced partitions $\phi^{(i)}(L)$, where $\phi^{(i)}(L)$ is the partition of the training sample L induced by the partition $(\phi_i \circ \phi_{i-1} \circ .... \circ \phi_1)(\underline{X})$.
- We need to show that CT scheme are well defined, which will be possible only if there exists some induced partition $L^{'}$ such that $\lim_{i \to \infty} \phi^{(i)}(L) = L^{'}$. In fact we need to show that the following lemma holds:

On Theoretical Consistency

Lemma (Chakraborty et al., 2019, Statistics & Probability Letters)

If L is a training sample and $\phi^{(i)}$ is defined as above, then there exists $N \in \mathbb{N}$ such that for $n \geq N$

$$\phi^{(n)}(L) = \lim_{i \to \infty} \phi^{(i)}(L)$$

- For a wide range of partitioning schemes, the consistency of histogram classification schemes based on data-dependent partitions was shown in the literature (Nobel, 1996, Annals of Statistics).

- Now instead of considering histogram-based partitioning and classification schemes, we are going to show the risk consistency of CT as defined above. We can produce a simultaneous result with (Nobel, 1996, Annals of Statistics) replaced by more simple condition. Though the shrinking cell condition is still assumed.

On Theoretical Consistency

Theorem (Chakraborty et al., 2019, Statistics & Probability Letters)

Suppose $(\underline{X}, \underline{Y})$ *be a random vector in* $\mathbb{R}^p \times\ C$ *and* L *be the training set consisting of* n *outcomes of* $(\underline{X}, \underline{Y})$. *Let* Φ *be a classification tree scheme such that* $\Phi(L) = (\psi_{pl} \circ \lim_{i \to \infty} \phi^{(i)})(L)$ *where,* ψ_{pl} *is the plurality rule and* $\phi(L) = (L)_{\tilde{\Omega}_n}$ *for some* $\tilde{\Omega}_n \in \mathcal{T}_n$, *where*

$$\mathcal{T}_n = \{\lim_{i \to \infty} \phi^{(i)}(\ell_n) : P(L = \ell_n) > 0\}.$$

Suppose that all the split function in CT *in the question set associated with* Φ *are axis-parallel splits. Finally if for every* n *and* $w_i \in \tilde{\Omega}_n$, *the induced subset* L_{w_i} *has cardinality* $\geq k_n$, *where* $\frac{k_n}{log(n))} \to \infty$ *and shrinking cell condition holds true, then* Φ *is risk consistent.*

- Note that no assumptions are made on the distribution of the pair $(\underline{X}, \underline{Y}) \in \mathbb{R}^p \times C$. Also sub-linear growth of the number of cells and sub-exponential growth of a combinatorial complexity measure are not required.
- Instead a more flexible restriction such as if each cell of L_{ω_i} has cardinality $\geq k_n$ and $\frac{k_n}{log(n))} \to \infty$, then CT is said to be risk consistent.
- Above Theorem along with the consistency results of FFNN model ensures the universal consistency of the proposed hybrid model.

On the choice of Number of Hidden Neurons

Lemma (Chakraborty et al., 2019, Statistics & Probability Letters)

Assume that there is a compact set $E \subset \mathbb{R}^{d_m}$ such that $Pr\{Z \in E\} = 1$ and the Fourier transform $\widetilde{P_0}(w)$ of $P_0(z)$ satisfies $\int_{\mathbb{R}^{d_m}} |\omega||\widetilde{P_0}(\omega)|d\omega < \infty$ then $\inf_{\psi \in F_{n,k}} E\Big(f(Z,\psi) - P_0(Z)\Big)^2 \leq \frac{c}{k}$, where c is a constant depending on the distribution.

Proposition (Chakraborty et al., 2019, Statistics & Probability Letters)

For a fixed d_m, let $\psi_n \in F_c$. The neural network satisfying regularity conditions of strong universal consistency and if the conditions of the above lemma holds, then the optimal choice of k is $O\left(\sqrt{\frac{n}{d_m log(n)}}\right)$.

- For practical use, if the data set is limited, the recommendation is to use $k = \left(\sqrt{\frac{n}{d_m log(n)}}\right)$ for achieving utmost accuracy of the propose model.

Applications: Data Sets

Data Sets: The proposed model is evaluated using six publicly available medical data sets from Kaggle (`https://www.kaggle.com/datasets`) and UCI Machine Learning repository (`https://archive.ics.uci.edu/ml/datasets.html`) dealing with various diseases. These binary classification data sets have limited number of observations and high-dimensional feature spaces.

Table: Characteristics of the data sets used in experimental evaluation

Data set	Classes	Objects (n)	Number of feature (p)	Number of $(+)$ve instances	Number of $(-)$ve instances
breast cancer	2	286	9	85	201
heart disease	2	270	13	120	150
pima diabetes	2	768	8	500	268
promoter gene sequences	2	106	57	53	53
SPECT heart images	2	267	22	55	212
wisconsin breast cancer	2	699	9	458	241

Applications: Experimental Results

Table: Results (and their standard deviation) of classification algorithms over 6 medical data sets

Classifiers	Data set	The number of (reduced) features after feature selection	Classification accuracy (%)	F-measure
CT	breast cancer	7	68.26 (6.40)	0.70 (0.07)
	heart disease	7	76.50 (4.50)	0.81 (0.03)
	pima diabetes	6	71.85 (4.94)	0.74 (0.03)
	promoter gene sequences	17	69.43 (2.78)	0.73 (0.01)
	SPECT heart images	9	75.70 (1.56)	0.78 (0.00)
	wisconsin breast cancer	8	94.20 (2.98)	0.89 (0.01)
ANN (with 1HL)	breast cancer	9	61.58 (5.89)	0.64 (0.04)
	heart disease	13	73.56 (5.44)	0.79 (0.02)
	pima diabetes	8	66.78 (4.58)	0.69 (0.04)
	promoter gene sequences	57	61.77 (3.46)	0.65 (0.02)
	SPECT heart images	22	79.69 (0.23)	0.81 (0.01)
	wisconsin breast cancer	9	94.80 (2.01)	0.96 (0.01)
Entropy Nets	breast cancer	7	69.00 (6.25)	0.72 (0.05)
	heart disease	7	79.59 (4.78)	0.83 (0.01)
	pima diabetes	6	69.50 (4.05)	0.72 (0.02)
	promoter gene sequences	17	66.23 (1.98)	0.70 (0.01)
	SPECT heart images	9	76.64 (1.70)	0.78 (0.01)
	wisconsin breast cancer	8	95.96 (2.18)	0.96 (0.00)
DNDT	breast cancer	8	66.12 (7.81)	0.68 (0.08)
	heart disease	7	81.05 (3.89)	0.86 (0.02)
	pima diabetes	6	69.21 (5.08)	0.72 (0.05)
	promoter gene sequences	17	69.06 (1.75)	0.71 (0.01)
	SPECT heart images	10	75.50 (0.89)	0.77 (0.00)
	wisconsin breast cancer	7	94.25 (2.14)	0.95 (0.00)
Proposed Model	breast cancer	7	**72.80** (6.54)	**0.77** (0.06)
	heart disease	7	**82.78** (4.78)	**0.89** (0.02)
	pima diabetes	6	**76.10** (4.45)	**0.79** (0.04)
	promoter gene sequences	17	**75.40** (1.50)	**0.79** (0.01)
	SPECT heart images	9	**81.03** (0.56)	**0.82** (0.00)
	wisconsin breast cancer	8	**97.30** (1.05)	**0.98** (0.00)

Chapter 3: Imbalanced Classification Problem: Hellinger Nets Model

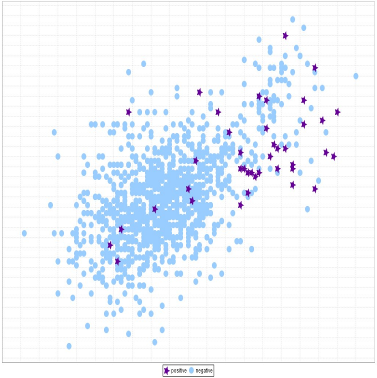
Publications:

1. Tanujit Chakraborty and Ashis Kumar Chakraborty. "Superensemble Classifier for Improving Predictions in Imbalanced Datasets". arXiv preprint arXiv:1810.11317. **(Under Revision)**

2. Tanujit Chakraborty, Ashis Kumar Chakraborty, and C. A. Murthy. "Consistency of Perceptron Trees". **(Under Review)**

Imbalanced Classification Problem

- Real-world data sets are usually skewed, in that many cases belong a larger class and fewer cases belong to a smaller yet usually more exciting class
- For example, consider a binary classification problem with the class distribution of 90 : 10. In this case, a straightforward method of guessing all instances to be positive class would achieve an accuracy of 90%.
- Learning from an imbalanced data set presents a tricky problem in which traditional learning algorithms perform poorly.
- Traditional classifiers usually aim to optimize the overall accuracy without considering the relative distribution of each class.



Sampling Techniques

- One way to deal with the imbalanced data problems is to modify the class distributions in the training data by applying sampling techniques to the data set
- Sampling technique either oversamples the minority class to match the size of the majority class or undersamples the majority class to match the size of the minority class.
- Synthetic minority oversampling technique (SMOTE) is among the most popular methods that oversamples the minority class by generating artificially interpolated data (Chawla et al., 2002, JAIR).
- TL (Tomek links) and ENN (edited nearest neighbor) are popular undersampling approaches (Batista et al., 2004, ACM SIGKDD).
- But these approaches have apparent deficiencies, such as undersampling majority instances may lose potentially useful information of the data set and oversampling increases the size of the training data set which may increase computational cost.
- To overcome these problems, "imbalanced data-oriented" algorithms are designed which can handle class imbalance without any modification to class distribution.

Effect of Class Imbalance on Decision Tree

Let X be attribute and Y be the response class. Here Y^+ denotes majority class, Y^- denotes minority class and n is the total number of instances. Also, let $X^{\geq} \longrightarrow Y^+$ and $X^< \longrightarrow Y^-$ be two rules generated by CT. Table below shows the number of instances based on the rules created using CT.

Table: An example of notions of classification rules

class and attribute	$X^{\geq}$	$X^<$	sum of instances
Y^+	a	b	$a+b$
Y^-	c	d	$c+d$
sum of attributes	$a+c$	$b+d$	n

In the case of imbalanced data set the majority class is always much larger than the size of the minority class and thus we will always have $a+b >> c+d$. It is clear that the generation of rules based on confidence in CT is biased towards majority class.

Various measures, like information gain (IG), gini index (GI) and misclassification impurity (MI) expressed as a function of confidence, are used to decide which variable to split in the important feature selection stage, get affected by class imbalance.

Effect of Class Imbalance on Distance Measures

Table: An example of notions of classification rules

class and attribute	$X^{\geq}$	$X^{<}$	sum of instances
Y^{+}	a	b	$a+b$
Y^{-}	c	d	$c+d$
sum of attributes	$a+c$	$b+d$	n

Using Table 1, we compute the following:

$$P(Y^{+}/X^{\geq}) = \frac{a}{a+c} = \text{Confidence}(X^{\geq} \longrightarrow Y^{+})$$

For an imbalanced data set, Y^{+} will occur more frequently with $X^{\geq}$ & $X^{<}$ than to Y^{-}. So the concept of confidence is a fatal error in an imbalanced classification problem.

Entropy at node t is defined as:

$$\text{Entropy}(t) = -\sum_{j=1,2} P(j/t) log\big(P(j/t)\big)$$

Effect of Class Imbalance on Distance Measures

In binary classification, information gain for splitting a node t is defined as:

$$\text{IG} = \text{Entropy}(t) - \sum_{i=1,2} \frac{n_i}{n}\text{Entropy}(i) \tag{0.6}$$

where i represents one of the sub-nodes after splitting (assuming we have two sub nodes only), n_i is the number of instances in sub-node i and n is the total number of instances. The objective of classification using CT is to maximize IG which reduces to:

$$\text{Maximize}\left\{ - \sum_{i=1,2} \frac{n_i}{n}\text{Entropy}(i) \right\} \tag{0.7}$$

The maximization problem in eqn. (1.7) reduces to:

$$\begin{aligned} \text{Maximize}\Bigg\{ &\frac{n_1}{n}\Big[P(Y^+/X^{\geq})log\big(P(Y^+/X^{\geq})\big) + P(Y^-/X^{\geq})log\big(P(Y^-/X^{\geq})\big)] \\ &+ \frac{n_2}{n}[P(Y^+/X^{<})log\big(P(Y^+/X^{<})\big) + P(Y^-/X^{<})log\big(P(Y^-/X^{<})\big)\Big]\Bigg\} \end{aligned} \tag{0.8}$$

The task of selecting the "best" set of features for node i are carried out by picking up the feature with maximum IG. As $P(Y^+/X^{\geq}) >> P(Y^-/X^{\geq})$, we face a problem while maximizing eqn. (0.8).

Hellinger Distance

Let (Θ, λ) denote a measurable space. Let us suppose that P and Q be two continuous distributions with respect to the parameter λ having the densities p and q in a continuous space Ω, respectively. Define HD as follows:

$$d_H(P,Q) = \sqrt{\int_\Omega (\sqrt{p} - \sqrt{q})^2 d\lambda} = \sqrt{2\left(1 - \int_\Omega \sqrt{pq} d\lambda\right)}$$

where $\int_\Omega \sqrt{pq} d\lambda$ is the Hellinger integral. It is noted that HD doesn't depend on the choice of the parameter λ.

For the application of HD as a decision tree criterion, the final formulation can be written as follows:

$$HD = d_H(X_+, X_-) = \sqrt{\sum_{j=1}^{k} \left(\frac{|X_{+j}|}{|X_+|} - \frac{|X_{-j}|}{|X_-|} \right)^2}, \qquad (0.9)$$

where $|X_+|$ indicates the number of examples that belong to the majority class in training set and $|X_{+j}|$ is the subset of training set with the majority class and the value j for the feature X. The bigger the value of HD, the better is the discrimination between the features (Hellinger Distance Decision Tree, Chawla et al. 2008, ECML).

Proposed Model: Hellinger Nets

- Hellinger Nets are composed of three basic steps:

  (a) Converting a DT into rules (HD is used as criterion);
  (b) Constructing a two hidden layered NN from the rules;
  (c) Training the MLP using gradient descent backpropagation (Rumelhart, Hinton (1988).

- In decision trees, the overfitting occurs when the size of the tree is too large compared to the number of training data.

- Instead of using pruning methods (removing child nodes), HN employs a backpropagation NN to give weights to nodes according to their significance.

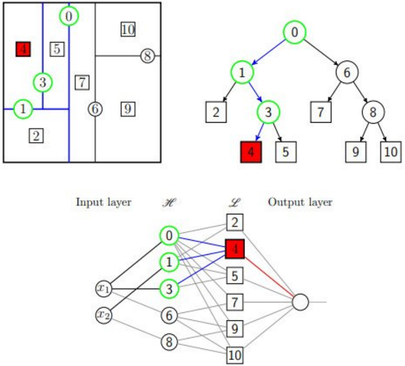


Fig: Graphical Representation of Hellinger Nets

The idea of the this approach is inspired from the idea of Perceptron Trees [Paul E Utgoff, 1988, AAAI]

Hellinger Nets Algorithm

- Build a HDDT with $(k_n - 1)$ split nodes and k_n leaf nodes. HDDT is mapped into a two hidden layered MLP model having $(k_n - 1)$ and k_n hidden neurons in first hidden layer $(HL1)$ and second hidden layer $(HL2)$, respectively.

- The first hidden layer is called the partitioning layer which partitions the input feature spaces into different regions. It corresponds to the internal nodes of the DT. In $HL1$, the neurons compute all the tree split decisions and indicate the split directions for the inputs.

- Further, $HL1$ passes the information to $HL2$. The neurons in the second hidden layer represent the terminal nodes of the DT.

- The final layer is the output class label of the tree. Train the tree structured neural network using gradient descent backpropagation algorithm.

Merits & Possible Extensions

- **Merits**:

  1. The additional training using backpropagation potentially improves the predictions of the HDDT and can deny tree pruning steps vis-a-vis the risk of overfitting.;

  2. Hellinger Nets give weight to nodes according to their significance as determined by the gradient backpropagation algorithm.;

  3. In Hellinger Nets, the neural network follows the built-in hierarchy of the originating tree since connections do not exist between all pairs of neurons in any two adjacent layers.;

  4. Since the number of neurons in the hidden layers are fixed, thus the training time is less.

- **Possible Extensions**:

  1. Theoretical Consistency?

  2. Rate of Convergence?

Theoretical Analysis of Hellinger Nets

- Hellinger Net uses sigmoidal activation function instead of the relay-type activation function $\tau(u)$ with a hyperbolic tangent activation function $\sigma(u) = \tanh(u)$ which has a chosen range from -1 to 1.

- More precisely, the model uses $\sigma_1(u) = \sigma(\beta_1 u)$ at every neuron of the first hidden layer for better generalization, where β_1 is a positive hyper-parameter that determines the contrast of the hyperbolic tangent activation function.

- The larger the value of the parameters β_1 and β_2, the sharper is the transition from -1 to 1. As β_1 and β_2 approach to infinity, the continuous functions σ_1 and σ_2 converge to the threshold activation function.

- The use of hyperbolic tangent activation functions instead of threshold activation function provide better generalization, smooth decision boundaries, and fast implementation. They also support the differentiability of the empirical loss function with respect to its parameters due to continuous property of the tangent activation function.

On Universal Consistency

Theorem (Chakraborty et al., 2019, Submitted Manuscript)

Assume X is uniformly distributed in $[0,1]^p$, $Y = \{0,1\}$, and $\psi \in F_{n,k_n}$. As $n \to \infty$ and for any $k_n, \beta_1, \beta_2 \to \infty$ if the following conditions are satisfied:

$$
\begin{aligned}
&(A1) \quad \frac{k_n^4 log(\beta_2 k_n^4)}{n} \to 0,\\
&(A2) \quad \textit{there exists} \quad \delta > 0 \quad \textit{such that} \quad \frac{k_n^2}{n^{1-\delta}} \to 0,\\
&(A3) \quad \frac{k_n^2}{e^{2\beta_2}} \to 0, \quad \textit{and}\\
&(A4) \quad \frac{k_n^3 \beta_2}{\beta_1} \to 0,
\end{aligned}
$$

then Hellinger Nets classifier is universally consistent.

The above Theorem states that with certain restrictions imposed on the number k_n of terminal nodes and the parameters β_1, β_2 being properly regulated as functions of n, the empirical L_1 risk-minimization provides strong universal consistency of the Hellinger Nets classifier.

Rate of Convergence

Theorem (Chakraborty et al., 2019, Submitted Manuscript)

Assume that X is uniformly distributed in $[0,1]^p$ and $Y = \{0,1\}$ and a function $m : C^p \to \{0,1\}$ satisfies $|m(x) - m(z)| \leq c\|x - z\|^{\delta}$ for any $\delta \in [0,1]$ and $z \in [0,1]^p$. Let m_n be the estimate that minimizes empirical L_1-risk and the network activation function σ_i satisfies Lipschitz property. Then for any $n \geq max\{\beta_2, 2^{p+1}L\}$, we have

$$E \int_{[0,1]^p} \big|m_n(X) - m(X)\big| \mu(dx) = O\left(\frac{log(n)^6}{n}\right)$$

- The proof of the Theorem is using Complexity Regularization Principles.
- The rate of convergence doesn't depend on the data dimension and hence the model will be able to circumvent the so-called problem of "curse of dimensionality".
- In practice, the larger the value of k_n, β_1, and β_2, the better the model performance is.

Applications: Data Sets

Data Sets: The proposed model is evaluated using five publicly available data sets from a wide variety of application areas such as management, business, and medicine, available at UCI Machine Learning repository. To measure the level of imbalance of these data sets, we compute the coefficient of variation (CV) which is the proportion of the deviation in the observed number of samples for each class versus the expected number of examples in each class. We have chosen thee data sets with a CV more than equal to 0.30− a class ratio of 2 : 1 on a binary data set as imbalanced data. Table 3 gives an overview of these data sets.

Table: Characteristics of the data sets used in experimental evaluation

Data set	Classes	Objects	Number of	Number of	Number of	CV
		(n)	feature (p)	(+)ve instances	(−)ve instances	
breast cancer	2	286	9	201	85	0.41
german credit card	2	1000	20	700	300	0.40
Indian business school	2	480	17	400	80	0.56
page blocks	2	5473	10	4913	560	0.80
pima diabetes	2	768	8	500	268	0.30

Applications: Performance Evaluation

The performance evaluation measure used in our experimental analysis is based on the confusion matrix in Table 2. Area under the receiver operating characteristic curve (AUC) is a popular metric for evaluating performances of imbalanced data sets and higher the value of AUC, the better the classifier is. AUC $= \frac{\text{Sensitivity}+\text{Specificity}}{2}$; where, Sensitivity $= \frac{TP}{TP+FN}$; Specificity $= \frac{TN}{FP+TN}$.

Table: Average AUC value for balanced data sets (using SMOTE and SMOTE+ENN) on different classifiers

Data	Sampling Techniques	kNN	CT	RF	ANN (with 1HL)	ANN (with 2HL)	RBFN
breast cancer	SMOTE	0.700	0.665	**0.722**	0.605	0.680	0.704
	SMOTE+ENN	0.685	0.650	0.708	0.600	0.652	0.700
german credit card	SMOTE	0.758	0.745	0.762	0.740	0.735	0.764
	SMOTE+ENN	0.760	**0.778**	0.770	0.750	0.720	0.765
indian business school	SMOTE	0.783	0.845	0.859	0.765	0.798	0.905
	SMOTE+ENN	0.801	0.850	0.875	0.798	0.807	**0.914**
page blocks	SMOTE	0.927	0.965	**0.967**	0.933	0.942	0.954
	SMOTE+ENN	0.935	0.952	0.966	0.925	0.937	0.949
pima diabetes	SMOTE	0.770	0.758	0.753	0.698	0.719	0.745
	SMOTE+ENN	**0.788**	0.760	0.761	0.712	0.725	0.748

Applications: Performance Evaluation

Highest AUC value in both the tables are highlighted with dark black for all the data sets. It is clear from computational experiments that our model stands as very much competitive with the current state-of-the-art models.

Table: AUC results (and their standard deviation) of classification algorithms over original imbalanced test data sets

Classifiers	breast	German credit	Indian business	page	pima
	cancer	card	school	blocks	diabetes
CT	0.603 (0.04)	0.665 (0.03)	0.810 (0.04)	0.950 (0.00)	0.724 (0.02)
RF	0.690 (0.06)	0.725 (0.03)	0.850 (0.04)	0.964 (0.00)	0.747 (0.04)
k-NN	0.651 (0.03)	0.727 (0.01)	0.750 (0.03)	0.902 (0.02)	0.730 (0.05)
RBFN	0.652 (0.06)	0.723 (0.04)	0.884 (0.05)	0.935 (0.01)	0.725 (0.04)
HDDT	0.625 (0.04)	0.738 (0.04)	0.933 (0.02)	0.974 (0.00)	0.760 (0.02)
HDRF	0.636 (0.04)	0.742 (0.03)	0.939 (0.02)	**0.988** (0.00)	0.760 (0.03)
CCPDT	0.618 (0.05)	0.712 (0.05)	0.912 (0.03)	0.971 (0.00)	0.753 (0.01)
ANN (with 1HL)	0.585 (0.03)	0.700 (0.03)	0.768 (0.05)	0.918 (0.02)	0.649 (0.03)
ANN (with 2HL)	0.621 (0.02)	0.715 (0.02)	0.820 (0.04)	0.925 (0.01)	0.710 (0.03)
Hellinger Nets	**0.720** (0.06)	**0.798** (0.04)	**0.964** (0.01)	0.985 (0.00)	**0.789** (0.05)

Chapter 4: Process Efficiency Improvement Problem: A Hybrid Regression Model

Publications:

1. Tanujit Chakraborty, Ashis Kumar Chakraborty, and Swarup Chattopadhyay. "A novel distribution-free hybrid regression model for manufacturing process efficiency improvement", **Journal of Computational and Applied Mathematics**, Elsevier. 362 (2019): 130-142.

2. Tanujit Chakraborty, Swarup Chattopadhyay, and Ashis Kumar Chakraborty. "Radial basis neural tree model for improving waste recovery process in a paper industry", **Applied Stochastic Models in Business and Industry**, Wiley. (2019).

Motivation of the Problem

- This work is motivated by a particular problem in a modern paper manufacturing industry, in which maximum efficiency of the process fiber-filler recovery equipment, also known as Krofta supracell, is desired.
- As a by-product of the paper manufacturing process, a lot of unwanted materials along with valuable fibers and fillers come out as waste materials.

Fig: Krofta supracell

Problem Statement

- The job of an efficient Krofta supracell is to separate the unwanted materials from the valuable ones so that fibers and fillers can be reused in the manufacturing process.

- The Krofta recovery percentage was around 75%. The paper manufacturing company wants to improve the recovery percentage to 90%.

- To identify the important parameters affecting the Krofta efficiency, a failure mode and effect analysis (FMEA) was performed with the help of process experts.

- **Goal**: We would like to come up with a model that can help the manufacturing process industry to achieve an efficiency level of about 90% from the existing level of about 75% to improve the Krofta supracell recovery percentage.

Understanding the Process: Process Flow & Ishikawa Diagram

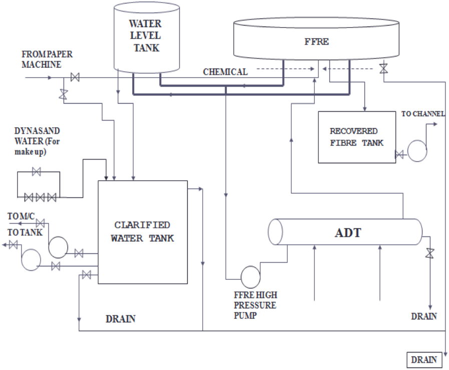


Fig: Process Flow Diagram of Krofta supracell

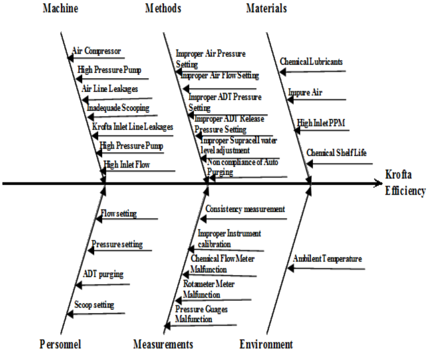
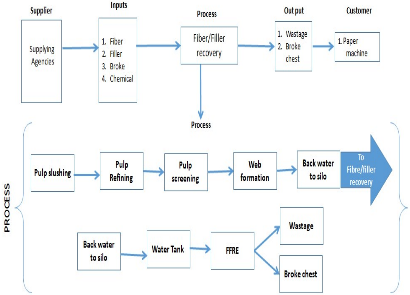

Machine
Methods
Materials
Air Compressor
High Pressure Pump
Air Line Leakages
Inadequade Scooping
Krofta Inlet Line Leakages
High Pressure Pump
High Inlet Flow
Improper Air Pressure Setting
Improper Air Flow Setting
Improper ADT Pressure Setting
Improper ADT Release Pressure Setting
Improper Supracell water level adjustment
Non compliance of Auto Purging
Chemical Lubricants
Impure Air
High Inlet PPM
Chemical Shelf Life
Krofta Efficiency
Flow setting
Pressure setting
ADT purging
Scoop setting
Consistency measurement
Improper Instrument calibration
Chemical Flow Meter Malfunction
Rotameter Meter Malfunction
Pressure Guages Malfunction
Ambilent Temperature
Personnel
Measurements
Environment

Fig: Ishikawa Diagram of Fiber and Filler Recovery Process

SIPOC Diagram & FMEA Analysis



Supplier

Supplying Agencies

Inputs

1. Fiber
2. Filler
3. Broke
4. Chemical

Process

Fiber/Filler recovery

Out put

1. Wastage
2. Broke chest

Customer

1. Paper machine

Process

PROCESS

Pulp slushing

Pulp Refining

Pulp screening

Web formation

Back water to silo

To Fibre/filler recovery

Back water to silo

Water Tank

FFRE

Wastage

Broke chest

Fig: SIPOC Diagram of Fiber and Filler Recovery Process

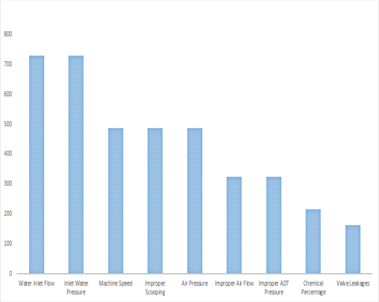


Fig: Summary of the FMEA.

Process Data Set

- The data set collected for a year from the process on the following causal variables: Inlet Flow, Water Pressure (water inlet pressure to ADT), Air Pressure, Pressure of Air-Left, Pressure of Air-Right, Pressure of ADT-D Left, Pressure of ADT-D Right and Amount of chemical lubricants.
- The response variable (FFRE recovery percentage) lies between 20 to 100.
- Sample data set for the paper tissue is presented in Table below.
- This data set will be used for finding crucial process parameters and also finding a prediction model that can help the company for forecasting future recovery percentage of FFRE.

Table: Sample data set

Inlet Flow	Water Pressure	Air Pressure	Air-Left	Air-Right	ADT-D	ADT-D	Amount of	Recovery
						Left	Right	chemical
Percentage								
1448	6.4	5.8	1.0	2.1	3.2	4.0	2.0	96.80
1794	5.2	5.6	2.4	1.6	3.6	4.0	3.0	97.47
2995	6.0	6.0	1.5	4.5	4.0	4.8	4.0	28.87
1139	6.5	6.0	1.2	1.7	3.0	4.6	2.0	33.05
2899	6.2	5.7	2.0	1.2	3.1	4.0	2.0	97.91
1472	6.6	6.8	3.7	3.1	5.2	4.8	4.0	57.77
1703	6.2	6.0	2.9	1.0	3.0	4.2	2.0	26.94
1514	5.5	5.0	2.0	2.1	3.8	4.7	2.0	67.01
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

Proposed Hybrid Model

- Apply RT algorithm to train and build a decision tree. Use the tree to extract the important features and find the splits between different adjacent values of the features.
- Choose the features that have minimum mean squared error as important input variables and record RT predicted outputs.
- Export important input variables along with an additional feature (prediction values of RT algorithm) to the RBFN model and a neural network is generated.
- RBFN model uses Gaussian kernel as an activation function, and parameter optimization is done using gradient descent algorithm. Finally, we obtain the final outputs.

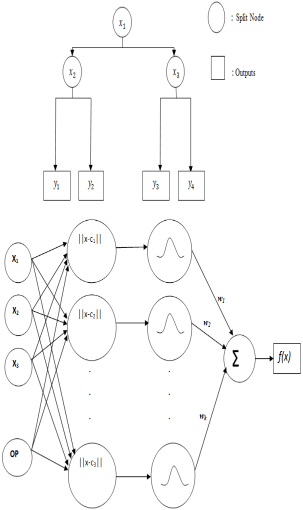


Fig: Flowchart of the Proposed Hybrid RT-RBFN Model

Experimental Evaluation

Popularly used performance metric are:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y}_i\right|;\ RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2};\ MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \widehat{y}_i}{y_i}\right|;$$

$$R^2 = 1 - \left[\frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}\right];\ AdjR^2 = 1 - \left[\frac{(1-R^2)(n-1)}{n-d_m-1}\right];$$

where, $y_i, \overline{y}, \widehat{y}_i$ denote the actual value, average value and predicted value of the dependent variable, respectively for the i^{th} instant. Here n and d_m denote the number of data points and independent variables used for performance evaluation, respectively.

Table: Quantitative measure of performance for different regression models. Results are based on 10 fold cross validations. Mean values of the respective measures are reported with standard deviation within the bracket.

Models	**MAE**	**RMSE**	**MAPE**	R^2	**Adj(R^2)**
RT	11.691 (0.45)	16.927 (0.89)	29.010 (1.02)	59.028 (3.25)	55.304 (1.95)
ANN	12.334 (0.25)	17.073 (0.56)	27.564 (1.85)	58.310 (2.98)	54.529 (2.08)
SVR	12.460 (0.28)	20.362 (1.23)	40.010 (1.81)	40.174 (2.05)	35.325 (2.64)
BART	12.892 (0.59)	16.010 (1.25)	30.038 (1.95)	59.380 (2.50)	56.458 (1.75)
RBFN	13.926 (2.50)	18.757 (3.25)	32.48 (3.45)	49.689 (5.45)	46.335 (3.95)
Tsai Neural tree	10.895 (0.78)	16.012 (0.50)	24.021 (1.85)	65.120 (2.89)	62.946 (1.78)
Proposed Model	**9.226** (0.35)	**14.331** (0.82)	**20.187** (1.45)	**70.632** (2.00)	**68.675** (2.13)

Theoretical Consistency

Theorem (Chakraborty et al., 2019, Applied Stochastic Models)

Suppose $(\underline{X}, \underline{Y})$ *be a random vector in* $\mathbb{R}^p \times [-K, K]$ *and* L_n *be the training set of* n *outcomes of* $(\underline{X}, \underline{Y})$*. Finally if for every* n *and* $w_i \in \tilde{\Omega}_n$*, the induced subset* $(L_n)_{w_i}$ *contains at least* k_n *of the vectors of* $X_1, X_2, ..., X_n$*, then empirically optimal regression trees strategy employing axis parallel splits are consistent when the size* k_n *of the tree grows as* $o(\frac{n}{log(n)})$*.*

Theorem (Chakraborty et al., 2019, Applied Stochastic Models)

Consider a RBF network with Gaussian radial basis kernel having one hidden layer with k (> 1) *nodes. If* $k \to \infty$, $b \to \infty$ *and* $\frac{kb^4 log(kb^2)}{n} \to 0$ *as* $n \to \infty$*, then RBFN model is said to be universally consistent for all distribution of* $(\underline{Z}, \underline{Y})$*.*

On the choice of Number of Hidden Neurons

- RBFN is a family of ANNs, consists of only a single hidden layer and uses radial basis function as an activation function, unlike feed forward neural network. RBF network with one hidden layer having k nodes for a fixed Gaussian function is given by the equation:

$$f(z_i) = \sum_{j=1}^{k} w_j \; exp\left(- \frac{\parallel z_i - c_i \parallel^2}{2\sigma_i^2} \right) + w_0,$$

where $\sum_{j=0}^{k} |w_j| \leq b\ (> 0)$ and $c_1, c_2, ..., c_k \in \mathbb{R}^{d_m}$.

- For practical use, if the data set is limited, the recommendation is to use $k = \left(\sqrt{n/d_m log(n)}\right)$ for achieving utmost accuracy of the propose model.

Proposition (Chakraborty et al., 2019, Journal of Comp. & Appl. Mathematics)

For any fixed d_m and training sequence ξ_n, let $Y \in [-K, K]$, and $m, f \in F_{n,k}$, if the neural network estimate m_n satisfies the above-mentioned regularity conditions of strong universal consistency and f satisfying $\int_{S_r} f^2(z)\mu(dz) < \infty$ where, S_r is a ball with radius r centered at 0, then the optimal choice of k is $O\left(\sqrt{\frac{n}{d_m log(n)}}\right)$.

Other Experiments

Data Sets: The proposed model is evaluated using six publicly available from UCI Machine Learning repository (`https://archive.ics.uci.edu/ml/datasets.html`). These regression data sets have limited number of observations.

Table: Data set characteristics: number of samples and number of features, after removing observations with missing information or nonnumerical input features.

Sl. No.	Data	Number of samples	Number of features
1	Auto MPG	398	7
2	Concrete	1030	8
3	Forest Fires	517	10
4	Housing	506	13
5	Wisconsin	194	32

Table: Average RMSE results for each of the models across the different data sets

Data	RT	ANN	SVR	BART	RBFN	Neural Tree	Our Model
Auto MPG	3.950	4.260	5.720	3.220	4.595	3.300	**3.215**
Concrete	8.700	10.180	11.588	**5.540**	10.210	7.420	7.063
Forest Fires	75.138	90.702	91.985	65.890	82.804	**62.478**	64.411
Housing	4.980	9.054	12.520	3.978	7.871	4.590	**3.077**
Wisconsin	41.059	34.710	41.220	32.054	38.495	40.700	**23.659**

Chapter 5: Forecasting Time Series: A Hybrid Approach

Publications:

1. Tanujit Chakraborty, Swarup Chattopadhyay, and Indrajit Ghosh. "Forecasting dengue epidemics using a hybrid methodology." **Physica A: Statistical Mechanics & its Applications**, Elsevier. (2019).

2. Tanujit Chakraborty, Shramana Bhattacharya, Sayak Banerjee, Munmun Biswas, and Ashis Kumar Chakraborty. "Forecasting the unemployment rates of European Countries" **(To be Submitted)**.

Linear Vs. Nonlinear Models in Time Series Forecasting

- Conventional statistical methods, the autoregressive integrated moving average (ARIMA) (Box and Jenkins, 1976) is extensively utilized in constructing a forecasting model.
- ARIMA cannot be utilized to produce an accurate model for forecasting nonlinear time series.
- Machine Learning algorithms have been successfully utilized to develop a nonlinear model for forecasting time series.
- Determining whether a linear or nonlinear model should be fitted to a real-world data set is difficult.
- The ARIMA model is used for prediction non-stationary time series when linearity between variables is supposed.
- However, in many practical situations supposing linearity is not valid.

Background: ARIMA Model

- The ARIMA model, introduced by Box and Jenkin, is a linear regression model indulged in tracking linear tendencies in stationary time series data.
- The model is expressed as ARIMA(p,d, q) where p, d, and q are integer parameter values that decide the structure of the model.
- More precisely, p and q are the order of the AR model and the MA model respectively, and parameter d is the level of differencing applied to the data.
- The mathematical expression of the ARIMA model is as follows:

$$y_t = \theta_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q},$$

- where y_t is the actual value, ε_t is the random error at time t, ϕ_i and θ_j are the coefficients of the model.
- It is assumed that ε_{t-l} $(\varepsilon_{t-l} = y_{t-l} - \hat{y}_{t-l})$ has zero mean with constant variance, and satisfies the i.i.d condition.
- Three Steps: Model identification, Parameter Estimation, and Diagnostic Checking.

Background: NNAR Model

- Neural nets are based on simple mathematical models of the brain, used for sophisticated nonlinear forecasting.
- NNAR (Faraway and Chatfield, JRSS C, 1998) overcomes the problems of fitting ANN for time series data sets like the choice on the number of hidden neurons, and its black box nature.
- NNAR model is a nonlinear time series model which uses lagged values of the time series as inputs to the neural network.
- NNAR(p,k) is a feed-forward neural network having one hidden layer with p lagged inputs and k nodes in the hidden layer.
- Thus, NNAR model with one hidden layer with the following mathematical form:

  $$\hat{x}_t = \phi_0 \left\{ w_{c_0} + \sum_h w_{h_0} \phi_h \left(w_{c_h} + \sum_i w_{i_h} x_{t-j_i} \right) \right\}$$

  where $\{w_{c_h}\}$ denotes the the connecting weights and ϕ_i is the activation function.
- An NNAR(p,k) model uses p as the optimal number of lags (calculated based on the AIC value) for an AR(p) model and k is set to $k = [\frac{(p+1)}{2}]$ for non-seasonal data sets.

Motivation for Hybrid Techniques

- A Forecaster wants the ARIMA model error series to be composed by i.i.d. random chocks or unpredictable or unsystematic terms with zero mean and constant variance, reflecting the piece of variability for which no reduction is possible.
- However, due to model mis-specification or to disturbances introduced in the stochastic process after forecasters elaboration, this (white noise) assumption may be violated during application phase.
- If the information underlying the error series is modeled, the performance of the original forecaster can be improved.

Table: Popular Hybrid Models in Time Series Forecasting Literature

Hybrid Model	**Author**	**Year**	**Journal**
SARIMA + BPNN	Tseng	2002	TFSC
ARIMA + ANN	Zhang	2003	Neurocomputing
ARIMA + SVM	Pai	2005	Omega
ARIMA + RNN	Aladag	2009	AML
ARIMA + PNN	Khashei	2012	C&IE
VARMA + BNN	Guo	2016	JAS
ARIMA + DNN	Qin	2017	KBS
Hybrid Survey	Khashei	2018	CinS

Error Calculation

There are popularly two types of error structures available in the literature (Mosleh et al., 1986, Risk Analysis).

Definition (Additive error model)

In the additive error model, the analyst treats the expert's estimate as a variable, $\hat{Y}_t$, and thinks of it as the sum of two terms, viz

$$\hat{Y}_t = Y_t + E_t$$

where Y_t is the true value and E_t the additive error term (capital letters indicate random variables).

Definition (Multiplicative error model)

In the multiplicative error model, the analyst treats the expert's estimate $\hat{Y}_t$ as the product of two terms, viz

$$\hat{Y}_t = Y_t \times E_t$$

where Y_t is the true value and E_t the multiplicative error term (capital letters indicate random variables).

Research Objective

Under some assumptions we can consider $\epsilon_t \overset{iid}{\sim} N(0, \sigma^2)$. Then the likelihood function will be

$$L(y_t|\hat{y}_t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left\{\frac{y_t - (\hat{y}_t + \mu)}{\sigma}\right\}^2 \right]$$

for additive error modelling approach.

The additive error model is replaced by $Y_t = \hat{Y}_t \times \epsilon_t$ in multiplicative model. Taking logarithms in both sides, we get $lnY_t = ln\hat{Y}_t + ln\epsilon_t$ which is of the same form of additive error model.

In the multivariate error model, if the assumption of normally distributed $ln\epsilon_t$ is justified, then the likelihood function is now the lognormal distribution, viz.,

$$L(y_t|\hat{y}_t) = \frac{1}{\sqrt{2\pi}\sigma y_t} \exp\left[-\frac{1}{2}\left\{\frac{lny_t - (ln\hat{y}_t + ln\hat{\mu})}{\sigma}\right\}^2 \right]$$

where $\hat{\mu}$ is the median of ϵ_t.

Thus, the forecaster expects additive errors calculated from ARIMA to follow $N(0, \sigma^2)$ and multiplicative errors to follow lognormal distribution but this is violated during practical applications.

Proposed Additive Hybrid Model

- $Z_t = Y_t + N_t$, where Y_t is the linear part and N_t is the nonlinear part of the hybrid model.
- Both Y_t and N_t are estimated from the data set.
- Let, $\hat{Y}_t$ be the forecast value of the ARIMA model at time t and ε_t represent the residual at time t as obtained from the ARIMA model.
- Then $\varepsilon_t = Z_t - \hat{Y}_t$.
- The residuals are modeled by the NNAR model and can be represented as follows $\varepsilon_t = f(\varepsilon_{t-1}, \varepsilon_{t-2}, ..., \varepsilon_{t-n}) + \varsigma_t$, where f is a nonlinear function modeled by the NNAR approach and ς_t is the random error.
- Therefore, the combined forecast is $\hat{Z}_t = \hat{Y}_t + \hat{N}_t$, where, $\hat{N}_t$ is the forecast value of the NNAR model.

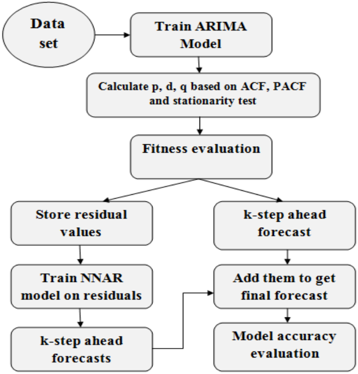

Data set
Train ARIMA Model
Calculate p, d, q based on ACF, PACF and stationarity test
Fitness evaluation
Store residual values
k-step ahead forecast
Train NNAR model on residuals
Add them to get final forecast
k-step ahead forecasts
Model accuracy evaluation

Fig: Graphical Representation of Hybrid ARIMA + NNAR Model

Applications: Dengue Data Sets

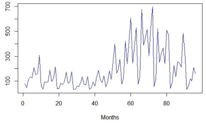
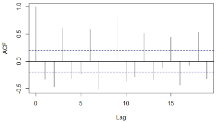
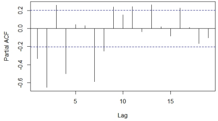
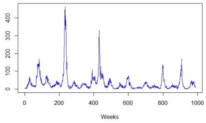
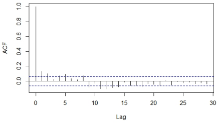
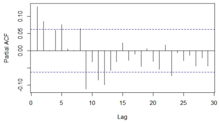
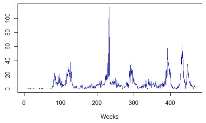
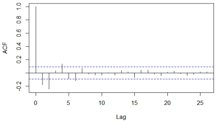
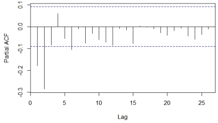
Region	Training data	ACF plot	PACF plot
Philippines			
San Juan			
Iquitos			

Table: Training data sets and corresponding ACF,PACF plots.

Applications: Performance Evaluation

Model	3-Months ahead forecast			6-Months ahead forecast		
	RMSE	MAE	SMAPE	RMSE	MAE	SMAPE
ARIMA	7.801	7.230	0.636	21.68	17.59	0.668
SVM	9.988	7.120	0.612	28.90	22.27	0.798
ANN	9.511	6.991	0.577	26.33	20.79	0.765
LSTM	10.500	7.095	0.630	28.50	23.05	0.800
NNAR	7.635	6.708	0.581	24.49	19.25	0.696
Hybrid ARIMA+SVM	8.150	7.695	0.640	23.01	18.95	0.703
Hybrid ARIMA+ANN	7.781	7.238	0.635	21.48	17.45	0.663
Hybrid ARIMA+LSTM	7.981	7.592	0.643	22.92	19.03	0.690
Hybrid ARIMA+NNAR	**7.438**	**6.569**	**0.570**	**20.73**	**16.56**	**0.612**

Table: Quantitative measures of performance for different forecasting models on San Juan data set

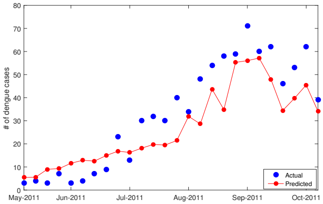
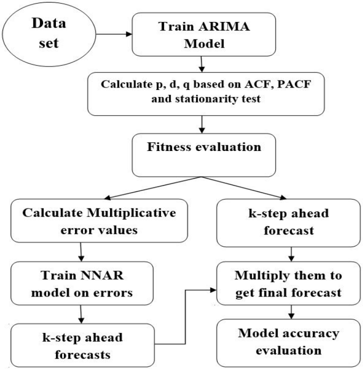


Fig: Actual vs predicted forecasts (using ARIMA+NNAR model) of San Jaun Data set

Proposed Multiplicative Hybrid Model

- $Z_t = Y_t \times N_t$, where Y_t is the linear part and N_t is the nonlinear part of the hybrid model.
- Both Y_t and N_t are estimated from the data set.
- Let, $\hat{Y}_t$ be the forecast value of the ARIMA model at time t and ε_t represent the residual at time t as obtained from the ARIMA model.
- Then $\varepsilon_t = Z_t / \hat{Y}_t$.
- The residuals are modeled by the NNAR model and can be represented as follows $\varepsilon_t = f(\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-n}) + \varsigma_t$, where f is a nonlinear function modeled by the NNAR approach and ς_t is the random error.
- Therefore, the combined forecast is $\hat{Z}_t = \hat{Y}_t \times \hat{N}_t$, where, $\hat{N}_t$ is the forecast value of the NNAR model.



Data set

Train ARIMA Model

Calculate p, d, q based on ACF, PACF and stationarity test

Fitness evaluation

Calculate Multiplicative error values

k-step ahead forecast

Train NNAR model on errors

Multiply them to get final forecast

k-step ahead forecasts

Model accuracy evaluation

Fig: Graphical Representation of Hybrid ARIMA × NNAR Model

Applications: Unemployment Rate Data Sets

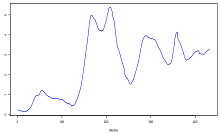
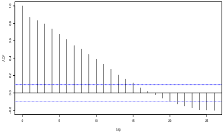
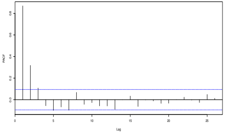
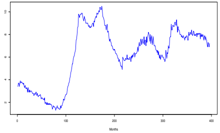
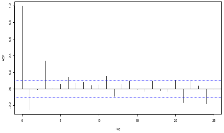
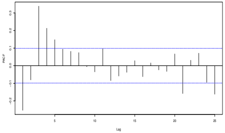
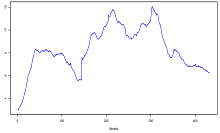
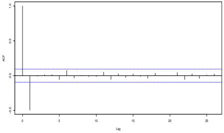
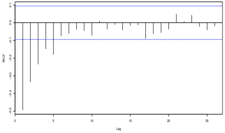
Region	Training data	ACF plot	PACF plot
Switzerland			
Sweden			
Germany			

Table: Training data sets and corresponding ACF,PACF plots.

Applications: Performance Evaluation

Table: Quantitative measures of performance for different forecasting models on the Switzerland data

Model	1-Year ahead forecast			2-Year ahead forecast			3-Year ahead forecast		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
ARIMA	0.047	0.037	1.095	0.153	0.116	3.436	0.437	0.314	9.365
ANN	0.226	0.133	5.394	0.949	0.607	40.363	1.326	0.933	115.176
NNAR	0.073	0.048	1.715	0.209	0.140	4.775	0.498	0.340	10.924
Hybrid ARIMA+ANN	0.045	0.035	1.035	0.151	0.114	3.366	0.435	0.311	9.295
Hybrid ARIMA × ANN	0.048	0.038	1.117	0.154	0.117	3.459	0.438	0.315	9.387
Hybrid ARIMA+NNAR	0.044	0.034	1.010	0.151	0.113	3.342	0.435	0.310	9.273
Hybrid ARIMA × NNAR	**0.036**	**0.028**	**0.838**	**0.142**	**0.104**	**3.093**	**0.427**	**0.301**	**9.017**

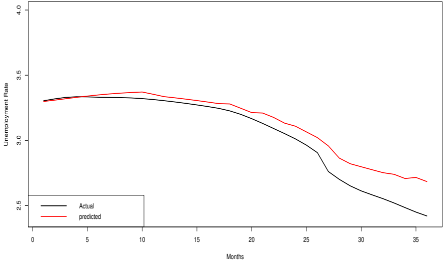


Fig: Actual vs predicted forecasts (using ARIMA × NNAR model) of Switzerland Data set

Merits & Discussion

- The multiplicative hybridization approach studies the relationship between linear and nonlinear components of the econometric time series.

- The multiplicative method is appropriate for explaining variations of economic and business data where there are interactions between linear and nonlinear time series.

- The concept of the multiplicative model is more useful than additive approach because it tends to remain more nearly constant in magnitude relative to the linear component model rather than in absolute terms.

- But both the models suppose that the residuals from the linear model will contain only the nonlinear relationship. However, one may not always guarantee that the residuals of the linear component may comprise valid nonlinear patterns.

- This model supposes that the linear and nonlinear patterns of a time series can be separately modeled by different models and then the forecasts can be combined together and this may degrade performance, if it is not true.

On Asymptotic Stationarity

ARIMA model has the in-built mechanism to transform a nonstationary time series into a stationary one and then it models the remainder by a stationary process. This is done by simple differencing to transform nonstationary ARIMA into stationary.

Consider the stochastic difference equation:

$$\varepsilon_t \quad = \quad f(\varepsilon_{t-1}, \varepsilon_{t-2}, ..., \varepsilon_{t-p}, \theta) + \varsigma_t, \tag{0.10}$$

where ς_t is an i.i.d. white noise and $f(., \theta)$ is a feedforward neural network with weight parameter θ. This is called an NNAR process of order p and has k hidden nodes in its one hidden layer. Thus, we refer the model as NNAR(p, k) model.

We consider the following architecture:

$$f(\underline{\varepsilon}) \quad = \quad c_0 + \sum_{i=1}^{k} w_i \sigma\left(a_i + b_i' \underline{\varepsilon}\right) \tag{0.11}$$

Let ε_t denote a time series generated by a nonlinear autoregressive process as defined in (0.10). Let $E(\varepsilon_t) = 0$, then f equals to the conditional expectation $E\left(\varepsilon_t | \varepsilon_{t-1}, ..., \varepsilon_{t-p}\right)$ is the best prediction for ε_t in the L_2-minimization sense.

On Geometric Ergodicity

We use the following notation:

$$z_{t-1} = \left(\varepsilon_{t-1}, ..., \varepsilon_{t-p}\right)' ; F(z_{t-1}) = \left(f(z_{t-1}), \varepsilon_{t-1}, ..., \varepsilon_{t-p+1}\right)' ; \hat{\varsigma}_t = \left(\varsigma_t, 0, ..., 0\right)'$$

Then we can write scalar AR(p) model in (0.10) as a first-order vector model,

$$z_t = F(z_{t-1}) + \hat{\varsigma}_t \qquad (0.12)$$

with $z_t, \hat{\varsigma}_t \in \mathbb{R}^p$.

Definition (Geometric ergodicity, Chan & Tong, 1985, AAP)

Let $\{z_t\}$, a markov chain, is said to be geometrically ergodic if there exists a probability measure $\Pi(A) = \lim_{t\to\infty} P(\varepsilon_t \in A)$ on the state space $(\mathbb{R}^p, \mathbb{B}, \mathbb{P})$, where $\mathbb{B}$ are Borel set on $\mathbb{R}^p$ and $\mathbb{P}$ be the Lebesgue measure, and for $\rho > 1$ and for all $z \in \mathbb{R}^p$,

$$\lim_{n\to\infty} \rho^n \| P\{z_{t+n} \in A | z_t = z\} - \Pi(A) \| = 0$$

where $\|.\|$ denotes the total variation and $P\{z_{t+n} \in A | z_t = z\}$ denote the probability of going from point z to set $A \in \mathbb{B}$ in n steps.

On Asymptotic Stationarity

If the markov chain is geometrically ergodic then its distribution will converge to Π and the corresponding time series will be called asymptotically stationary (Chan & Tong, 1985, Advances in Applied Probability).

It is also important to note that all neural network activation functions (like logistics or tan-hyperbolic) are continuous and compact functions and must have a bounded range.

Theorem (Chakraborty et al. (2019) Working Paper)

Let $E|\varsigma_t|^{1+\delta} < \infty$ for all $\delta > 1$ and the probability density function of ς_t is positive everywhere in $\mathbb{R}$ and $\{\varepsilon_t\}$ and $\{z_t\}$ are defined as in (0.10) and (0.12). Then if f is a nonlinear neural network as defined in (0.11), then $\{z_t\}$ is geometrically ergodic and $\{\varepsilon_t\}$ is asymptotically stationary.

Theoretical results on asymptotic stationarity is important for predictions over larger intervals of time, for example, one might train the network on an available sample and then use the trained network to generate new data with similar properties than the training sample.

The asymptotic stationarity guarantees that the the model cannot have growing variance with time.

Chapter 6: Conclusions

Summary of works (Chapter 2)

- A novel nonparametric ensemble classifier is proposed to achieve higher accuracy in classification performance with very little computational cost (by working with a subset of input features).

- Our proposed feature selection cum classification model is robust in nature.

- Ensemble CT-ANN is shown to be universally consistent and less time consuming during the actual implementation.

- We have also found the optimal value of the number of neurons in the hidden layer so that the user will have less tuning parameters to be controlled.

Summary of works (Chapter 3)

- Learning from an imbalanced data set presents a tricky problem in which traditional learning models perform poorly. Simply allocating half of the training examples to the minority class does nt provide the optimal solution in most of the real-life problems.

- If one would like to work with the original data without taking recourse to sampling, our proposed hybrid methodology will be quite handy.

- We proposed ‘Hellinger Nets’, a hybrid learner, that first construct a tree and then simulate it using neural networks.

- The approach depends on the choice of the total number of leaves and certain restrictions imposed on neural network hyper-parameters to ensure the consistency of Hellinger Nets model.

Summary of works (Chapter 4)

- In this chapter, we build a hybrid regression model for improving the process efficiency in a paper manufacturing company.
- Our study presented a hybrid RT-ANN model that integrates RT and ANN algorithm which gives more accuracy than all other competitive models to address the Krofta efficiency improvement problem.
- The proposed model is consistent, and when applied to other complex regression problems, it performed well as compared to other state-of-the-art.
- The usefulness and effectiveness of the model lie in its robustness and easy interpretability as compared to complex "black-box-like" models.

Summary of works (Chapter 5)

- In practice, it is often challenging to determine whether a time series under study is generated from a linear or nonlinear underlying process.
- In this chapter, we have built a novel hybrid model with a multiplicative approach that performs superior for forecasting unemployment rates.
- The proposed hybrid ARIMA$\times$NNAR model filters out linearity using the ARIMA model and predicts nonlinear tendencies with the NNAR approach.
- In this work, we also investigate the asymptotic behavior (stationarity and ergodicity) of the proposed hybrid approach using Markov chains and nonlinear time series analysis techniques.

Publication from the thesis

1. Tanujit Chakraborty, Swarup Chattopadhyay, and Ashis Kumar Chakraborty. "A novel hybridization of classification trees and artificial neural networks for selection of students in a business school", **Opsearch**. Springer. 55 (2018): 434-446.
2. Tanujit Chakraborty, Ashis Kumar Chakraborty, and C. A. Murthy. "A nonparametric ensemble binary classifier and its statistical properties, **Statistics & Probability Letters**. Elsevier. 149 (2019): 16-23.
3. Tanujit Chakraborty, Ashis Kumar Chakraborty, and Swarup Chattopadhyay. "A novel distribution-free hybrid regression model for manufacturing process efficiency improvement, **Journal of Computational & Applied Mathematics**. Elsevier. 362(2019): 130-142.
4. Tanujit Chakraborty, Swarup Chattopadhyay, and Ashis Kumar Chakraborty. "Radial basis neural tree model for improving waste recovery process in a paper industry, **Applied Stochastic Models in Business and Industry**. Wiley. (2019)
5. Tanujit Chakraborty, Swarup Chattopadhyay, and Indrajit Ghosh. "Forecasting dengue epidemics using a hybrid methodology. **Physica A: Statistical Mechanics & its Applications**. Elsevier. (2019).
6. Tanujit Chakraborty and Ashis Kumar Chakraborty. "Superensemble Classifier for Improving Predictions in Imbalanced Datasets. arXiv preprint arXiv:1810.11317. **(Under Revision)**.
7. Tanujit Chakraborty, Ashis Kumar Chakraborty, and C. A. Murthy. "Consistency of Perceptron Trees. **(Under Review)**.
8. Tanujit Chakraborty, Shramana Bhattacharya, Sayak Banerjee, Munmun Biswas,and Ashis Kumar Chakraborty. "Forecasting the unemployment rates of European Countries **(To be Submitted)**.

Acknowledgements

I would like to acknowledge the collaborators of these works:

- Late Prof. C.A. Murthy, Machine Intelligence Unit , ISI Kolkata.
- Dr. Ashis Kr Chakraborty, SQC & OR Unit, ISI Kolkata (Thesis Supervisor).
- Dr. Munmun Biswas, Brahmananda Keshab Chandra College, Kolkata.
- Mr. Swarup Chattopadhyay, MIU, ISI Kolkata.
- Mr. Indrajit Ghosh, AERU, ISI Kolkata.
- Ms. Shramana Bhattacharya, IIPS Mumbai.
- Mr. Sayak Banerjee, IIPS Mumbai.

References I

Galton, Francis. Natural inheritance. Macmillan and Company, 1894.

Fisher, Ronald A. “The precision of discriminant functions.” Annals of Eugenics 10.1 (1940): 422-429.

Berkson, Joseph. “Application of the logistic function to bio-assay.” Journal of the American Statistical Association 39.227 (1944): 357-365.

Fix, Evelyn, and Joseph L. Hodges Jr. Discriminatory analysis-nonparametric discrimination: consistency properties. California Univ Berkeley, 1951.

Parzen, Emanuel. “On estimation of a probability density function and mode.” The annals of mathematical statistics 33.3 (1962): 1065-1076.

Breiman, Leo. Classification and regression trees. Routledge, 2017.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. No. ICS-8506. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

Utgoff, Paul E. “Perceptron trees: A case study in hybrid concept representations.” Connection Science 1.4 (1989): 377-391.

Friedman, Jerome H. “Multivariate adaptive regression splines.” The annals of statistics 19.1 (1991): 1-67.

Cortes, Corinna, and Vladimir Vapnik. “Support-vector networks.” Machine learning 20.3 (1995): 273-297.

Breiman, Leo. ”Random forests.” Machine learning 45.1 (2001): 5-32.

Krizhevsky, A., I. Sutskever., and Hinton. G., “ImageNet Classification with Deep. Convolutional Neural Networks.” NIPS (2012).

Goodfellow, Ian, et al. “Generative adversarial nets.” Advances in neural information processing systems. 2014.

References II

- LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436.
- Gal, Yarin, Riashat Islam, and Zoubin Ghahramani. "Deep bayesian active learning with image data." Proceedings of the 34th International Conference on Machine Learning-Volume 70. 2017.
- Chipman, Hugh A., Edward I. George, and Robert E. McCulloch. "BART: Bayesian additive regression trees." The Annals of Applied Statistics 4.1 (2010): 266-298.
- Lugosi, Gbor, and Andrew Nobel. "Consistency of data-driven histogram methods for density estimation and classification." The Annals of Statistics 24.2 (1996): 687-706.
- Nobel, Andrew. "Histogram regression estimation using data-dependent partitions." The Annals of Statistics 24.3 (1996): 1084-1105.
- Kearns, Michael J., and Yishay Mansour. "A Fast, Bottom-Up Decision Tree Pruning Algorithm with Near-Optimal Generalization." ICML. Vol. 98. 1998.
- Mansour, Yishay, and David A. McAllester. "Generalization Bounds for Decision Trees." COLT. 2000.
- Nobel, Andrew B. "Analysis of a complexity-based pruning scheme for classification trees." IEEE Transactions on Information Theory 48.8 (2002): 2362-2368.
- Denil, Misha, David Matheson, and Nando Freitas. "Consistency of online random forests." International conference on machine learning. 2013.
- Scornet, Erwan, Grard Biau, and Jean-Philippe Vert. "Consistency of random forests." The Annals of Statistics 43.4 (2015): 1716-1741.
- Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators." Neural networks 2.5 (1989): 359-366.

References III

Hinton, E. C., et al. “Neural representations of hunger and satiety in PraderWilli syndrome.” International Journal of Obesity 30.2 (2006): 313.

Farag, Andrs, and Gbor Lugosi. “Strong universal consistency of neural network classifiers.” IEEE Transactions on Information Theory 39.4 (1993): 1146-1151.

Mhaskar, Hrushikesh Narhar. “Approximation properties of a multilayered feedforward artificial neural network.” Advances in Computational Mathematics 1.1 (1993): 61-80.

Hwang, JT Gene, and A. Adam Ding. “Prediction intervals for artificial neural networks.” Journal of the American Statistical Association 92.438 (1997): 748-757.

Hamers, Michael, and Michael Kohler. “Nonasymptotic bounds on the L 2 error of neural network regression estimates.” Annals of the Institute of Statistical Mathematics 58.1 (2006): 131-151.

Shaham, Uri, Alexander Cloninger, and Ronald R. Coifman. “Provable approximation properties for deep neural networks.” Applied and Computational Harmonic Analysis 44.3 (2018): 537-557.

Bauer, Benedikt, and Michael Kohler. “On deep learning as a remedy for the curse of dimensionality in nonparametric regression.” The Annals of Statistics 47.4 (2019): 2261-2285.

Lugosi, Gbor, and Kenneth Zeger. “Nonparametric estimation via empirical risk minimization.” IEEE Transactions on information theory 41.3 (1995): 677-687.

Murphy, Kevin P. Machine learning: a probabilistic perspective. MIT press, 2012.

Kuncheva, Ludmila I. Combining pattern classifiers: methods and algorithms. John Wiley Sons, 2004.

Sethi, Ishwar Krishnan. “Entropy nets: from decision trees to neural networks.” Proceedings of the IEEE 78.10 (1990): 1605-1613.

References IV

- Sirat, J. A., and J. P. Nadal. “Neural trees: a new tool for classification.” Network: computation in neural systems 1.4 (1990): 423-438.
- Jackson, Jeffrey C., and Mark Craven. “Learning sparse perceptrons.” Advances in Neural Information Processing Systems. 1996.
- Bennett, Kristin P., and J. A. Blue. “A support vector machine approach to decision trees.” 1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence. Vol. 3. IEEE, 1998.
- Jerez-Aragons, Jos M., et al. “A combined neural network and decision trees model for prognosis of breast cancer relapse.” Artificial intelligence in medicine 27.1 (2003): 45-63.
- Chen, Yuehui, Ajith Abraham, and Bo Yang. “Feature selection and classification using flexible neural tree.” Neurocomputing 70.1-3 (2006): 305-313.
- Sugumaran, V., V. Muralidharan, and K. I. Ramachandran. “Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing.” Mechanical systems and signal processing 21.2 (2007): 930-942.
- Nagi, Jawad, et al. “Convolutional neural support vector machines: hybrid visual pattern classifiers for multi-robot systems.” 2012 11th International Conference on Machine Learning and Applications. Vol. 1. IEEE, 2012.
- Gjorgjevikj, Dejan, Gjorgji Madjarov, and SAO DEROSKI. “Hybrid decision tree architecture utilizing local svms for efficient multi-label learning.” International Journal of Pattern Recognition and Artificial Intelligence 27.07 (2013): 1351004.
- Rota Bulo, Samuel, and Peter Kontschieder. “Neural decision forests for semantic image labelling.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.
- Kontschieder, Peter, et al. “Deep neural decision forests.” Proceedings of the IEEE international conference on computer vision. 2015.
- Hinton, Geoffrey, and Nicholas Frosst. “Distilling a Neural Network Into a Soft Decision Tree.” (2017).

References V

- Yang, Yongxin, Irene Garcia Morillo, and Timothy M. Hospedales. “Deep neural decision trees.” arXiv preprint arXiv:1806.06988 (2018).
- Pea-Ayala, Alejandro. “Educational data mining: A survey and a data mining-based analysis of recent works.” Expert systems with applications 41.4 (2014): 1432-1462.
- Chawla, Nitesh V., et al. “SMOTE: synthetic minority over-sampling technique.” Journal of artificial intelligence research 16 (2002): 321-357.
- Batista, Gustavo EAPA, Ronaldo C. Prati, and Maria Carolina Monard. “A study of the behavior of several methods for balancing machine learning training data.” ACM SIGKDD explorations newsletter 6.1 (2004): 20-29.
- Cieslak, David A., and Nitesh V. Chawla. “Learning decision trees for unbalanced data.” Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Berlin, Heidelberg, 2008.
- Krzyzak, Adam, Tams Linder, and C. Lugosi. “Nonparametric estimation and classification using radial basis function nets and empirical risk minimization.” IEEE Transactions on Neural Networks 7.2 (1996): 475-487.
- Krzyzak, Adam, and Tams Linder. “Radial basis function networks and complexity regularization in function learning.” Advances in neural information processing systems. 1997.
- Cieslak, David A., et al. “Hellinger distance decision trees are robust and skew-insensitive.” Data Mining and Knowledge Discovery 24.1 (2012): 136-158.
- Liu, Wei, et al. “A robust decision tree algorithm for imbalanced data sets.” Proceedings of the 2010 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2010.
- Su, Chong, et al. “Improving random forest and rotation forest for highly imbalanced datasets.” Intelligent Data Analysis 19.6 (2015): 1409-1432.

References VI

Daniels, Zachary Alan, and Dimitris N. Metaxas. “Addressing imbalance in multi-label classification using structured hellinger forests.” Thirty-First AAAI Conference on Artificial Intelligence. 2017.

Krofta, Milos. “Apparatus for clarification of water.” U.S. Patent No. 4,626,345. 2 Dec. 1986.

Krofta, Milos. “Apparatus and method for clarification of water using combined flotation and filtration processes.” U.S. Patent No. 4,377,485. 22 Mar. 1983.

Krofta, Milos. “Apparatus for clarifying waste water.” U.S. Patent No. 4,184,967. 22 Jan. 1980.

Tsai, Chia-Cheng, Mi-Cheng Lu, and Chih-Chiang Wei. “Decision treebased classifier combined with neural-based predictor for water-stage forecasts in a river basin during typhoons: a case study in Taiwan.” Environmental engineering science 29.2 (2012): 108-116.

Drucker, Harris, et al. “Support vector regression machines.” Advances in neural information processing systems. 1997.

Box, George EP, and Gwilym M. Jenkins. “Time series analysis: Forecasting and control San Francisco.” Calif: Holden-Day (1976).

Faraway, Julian, and Chris Chatfield. “Time series forecasting with neural networks: a comparative study using the air line data.” Journal of the Royal Statistical Society: Series C (Applied Statistics) 47.2 (1998): 231-250.

Hyndman, Rob J., and George Athanasopoulos. Forecasting: principles and practice. OTexts, 2018.

Tseng, Fang-Mei, Hsiao-Cheng Yu, and Gwo-Hsiung Tzeng. “Combining neural network model with seasonal time series ARIMA model.” Technological forecasting and social change 69.1 (2002): 71-87.

Zhang, G. Peter. “Time series forecasting using a hybrid ARIMA and neural network model.” Neurocomputing 50 (2003): 159-175.

Terui, Nobuhiko, and Herman K. Van Dijk. “Combined forecasts from linear and nonlinear time series models.” International Journal of Forecasting 18.3 (2002): 421-438.

References VII

Pai, Ping-Feng, and Chih-Sheng Lin. "A hybrid ARIMA and support vector machines model in stock price forecasting." Omega 33.6 (2005): 497-505.

Yu, Lean, Shouyang Wang, and Kin Keung Lai. "A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates." Computers Operations Research 32.10 (2005): 2523-2541.

Huang, Shian-Chang. "Online option price forecasting by using unscented Kalman filters and support vector machines." Expert Systems with Applications 34.4 (2008): 2819-2825.

Aladag, Cagdas Hakan, Erol Egrioglu, and Cem Kadilar. "Forecasting nonlinear time series with a hybrid methodology." Applied Mathematics Letters 22.9 (2009): 1467-1470.

Khashei, Mehdi, and Mehdi Bijari. "An artificial neural network (p, d, q) model for timeseries forecasting." Expert Systems with applications 37.1 (2010): 479-489.

Faruk, Durdu mer. "A hybrid neural network and ARIMA model for water quality time series prediction." Engineering Applications of Artificial Intelligence 23.4 (2010): 586-594.

Chan, Kung S., and Howell Tong. "On the use of the deterministic Lyapunov function for the ergodicity of stochastic difference equations." Advances in applied probability 17.3 (1985): 666-678.

Khashei, Mehdi, and Mehdi Bijari. "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting." Applied Soft Computing 11.2 (2011): 2664-2675.

Chen, Kuan-Yu. "Combining linear and nonlinear model in forecasting tourism demand." Expert Systems with Applications 38.8 (2011): 10368-10376.

Khashei, Mehdi, and Mehdi Bijari. "A new class of hybrid models for time series forecasting." Expert Systems with Applications 39.4 (2012): 4344-4357.

Wang, Ju-Jie, et al. "Stock index forecasting based on a hybrid model." Omega 40.6 (2012): 758-766.

References VIII

- Khashei, Mehdi, Mehdi Bijari, and Gholam Ali Raissi Ardali. “Hybridization of autoregressive integrated moving average (ARIMA) with probabilistic neural networks (PNNs).” Computers Industrial Engineering 63.1 (2012): 37-45.
- Yolcu, Ufuk, Erol Egrioglu, and Cagdas H. Aladag. “A new linear nonlinear artificial neural network model for time series forecasting.” Decision support systems 54.3 (2013): 1340-1347.
- Firmino, Paulo Renato A., Paulo SG de Mattos Neto, and Tiago AE Ferreira. “Correcting and combining time series forecasters.” Neural Networks 50 (2014): 1-11.
- Mosleh, Ali, and George Apostolakis. “The assessment of probability distributions from expert opinions with an application to seismic fragility curves.” Risk analysis 6.4 (1986): 447-461.
- Trapletti, Adrian, Friedrich Leisch, and Kurt Hornik. “Stationary and integrated autoregressive neural network processes.” Neural Computation 12.10 (2000): 2427-2450.
- -Farias, Mayte Surez, Carlos E. Pedreira, and Marcelo C. Medeiros. “Local global neural networks: A new approach for nonlinear time series modeling.” Journal of the American Statistical Association 99.468 (2004): 1092-1107.
- Tersvirta, Timo, Dick Van Dijk, and Marcelo C. Medeiros. “Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A re-examination.” International Journal of Forecasting 21.4 (2005): 755-774.
- Medeiros, Marcelo C., Timo Tersvirta, and Gianluigi Rech. “Building neural network models for time series: a statistical approach.” Journal of Forecasting 25.1 (2006): 49-75.
- Yajima, Yoshihiro. “Asymptotic properties of the sample autocorrelations and partial autocorrelations of a multiplicative ARIMA process.” Journal of Time Series Analysis 6.3 (1985): 187-201.
- Wong, Wing-keung, and Robert B. Miller. “Repeated time series analysis of ARIMAnoise models.” Journal of Business Economic Statistics 8.2 (1990): 243-250.

References IX

- Yajima, Yoshihiro. “Asymptotic properties of the LSE in a regression model with long-memory stationary errors.” The Annals of Statistics 19.1 (1991): 158-177.
- Guo, Hongyue, Xiaodong Liu, and Zhubin Sun. “Multivariate time series prediction using a hybridization of VARMA models and Bayesian networks.” Journal of Applied Statistics 43.16 (2016): 2897-2909.
- Qin, Mengjiao, Zhihang Li, and Zhenhong Du. “Red tide time series forecasting by combining ARIMA and deep belief network.” Knowledge-Based Systems 125 (2017): 39-52.
- Khashei, Mehdi, and Zahra Hajirahimi. “A comparative study of series arima/mlp hybrid models for stock price forecasting.” Communications in Statistics-Simulation and Computation (2018): 1-16.
- Khashei, Mehdi, and Mehdi Bijari. “Which methodology is better for combining linear and nonlinear models for time series forecasting?.” Journal of Industrial and Systems Engineering 4.4 (2011): 265-285.
- Lematre, Guillaume, Fernando Nogueira, and Christos K. Aridas. “Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning.” The Journal of Machine Learning Research 18.1 (2017): 559-563.
- Loh, WeiYin. “Fifty years of classification and regression trees.” International Statistical Review 82.3 (2014): 329-348.
- Chakraborty, Tanujit, Swarup Chattopadhyay, and Ashis Kumar Chakraborty. “A novel hybridization of classification trees and artificial neural networks for selection of students in a business school.” Opsearch 55.2 (2018): 434-446.
- Chakraborty, Tanujit, Ashis Kumar Chakraborty, and C. A. Murthy. “A nonparametric ensemble binary classifier and its statistical properties.” Statistics Probability Letters 149 (2019): 16-23.
- Chakraborty, Tanujit, Ashis Kumar Chakraborty, and Swarup Chattopadhyay. “A novel distribution-free hybrid regression model for manufacturing process efficiency improvement.” Journal of Computational and Applied Mathematics (2019).

References X

Chakraborty, Tanujit, Swarup Chattopadhyay, and Ashis Kumar Chakraborty. "Radial basis neural tree model for improving waste recovery process in a paper industry." (2018).

Chakraborty, Tanujit, Swarup Chattopadhyay, and Indrajit Ghosh. "Forecasting dengue epidemics using a hybrid methodology." Physica A: Statistical Mechanics and its Applications 527 (2019): 121266.

Chakraborty, Tanujit, and Ashis Kumar Chakraborty. "Superensemble Classifier for Improving Predictions in Imbalanced Datasets." arXiv preprint arXiv:1810.11317 (2018).

Loh, WeiYin. "Classification and regression trees." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1.1 (2011): 14-23.

Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "Statistical and Machine Learning forecasting methods: Concerns and ways forward." PloS one 13.3 (2018): e0194889.

Opitz, David, and Richard Maclin. "Popular ensemble methods: An empirical study." Journal of artificial intelligence research 11 (1999): 169-198.

Sagi, Omer, and Lior Rokach. "Ensemble learning: A survey." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8.4 (2018): e1249.

Woniak, Micha, Manuel Graa, and Emilio Corchado. "A survey of multiple classifier systems as hybrid systems." Information Fusion 16 (2014): 3-17.

Ahmed, Nesreen K., et al. "An empirical comparison of machine learning models for time series forecasting." Econometric Reviews 29.5-6 (2010): 594-621.

Questions?

Thank You