

## DEVELOPING PREDICTIVE MODELS USING TYPICAL MACHINE LEARNING AND COMPUTATIONAL TECHNIQUES

Patrick Ozoh<sup>1</sup>, Shapiee Abd-Rahman<sup>2</sup>, Moruf Olayiwola<sup>3</sup>

<sup>1</sup> Osun State University - Nigeria, Department of Information and Communication Technology

<sup>2</sup> Universiti Malaysia Sarawak - Malaysia, Department of Computational Sciences

<sup>3</sup> Osun State University - Nigeria, Department of Mathematical Sciences

Corresponding Author: Patrick Ozoh, patrick.ozoh@uniosun.edu.ng

**ABSTRACT:** This study investigates the accuracy of developing predictive models using machine learning techniques. The machine learning techniques considered in this study include artificial neural network (ANN) and Kalman filter adaptation algorithm. Predictive values are computed based on these techniques. These techniques are tested on daily electricity consumption data and are computed using ANN technique and Kalman filter adaptation algorithm. The accuracy of the predicted values of these techniques are investigated using statistical parameters. This research identified Kalman technique as more accurate in making predictions than ANN technique.

**KEY WORDS:** accuracy, modelling, prediction, machine learning, performance measures

### 1. INTRODUCTION

In order to perform reliable decision making, it is expedient to accurately model real life problems in diverse applications. Prediction models can be developed using machine learning techniques ([Tof15]). In World Energy Outlook ([\*\*16]), the paper identified the need for efficient use of electrical appliances. Ozoh et al. ([OAL14]) utilizes artificial neural network (ANN) technique and Kalman filter adaptation algorithm. In investigating the reliability of ANN technique and Kalman filter adaptation algorithm; this study assesses and compares the techniques in model estimation. In choosing the most appropriate technique for prediction, the techniques were tested on historical data, and their performance tested in terms of statistical parameters. The technique with the smallest error values is selected as the best performing technique as discussed by Zhang et al. ([ZPH98]).

The contributions of this paper are summarized as follows:

1. This empirical study investigates and identifies a reliable approach to modelling and predicting data, and proposes an effective technique for prediction called Kalman filter adaptation algorithm that assures the computation of accurate predictive values.
2. This study confirms Kalman filter adaptation algorithm offers competitive modelling performance

compared to artificial neural network, which is one of the best machine learning techniques. This study confirms that Kalman filter adaptation algorithm (i) outperforms artificial neural network, and in general (ii) has more efficient statistical estimates than artificial neural network.

This paper is organized and classified into the following sections. In Section 1, this paper introduces modelling and predictions, and the need for investigating performance of artificial neural network and Kalman filter adaptation algorithm through the use of statistical parameters. Also, the contributions of the study are discussed. In Section 2, the methods and techniques used in this study are discussed. The mathematical models used in this study are thoroughly explained. Section 3 is devoted to empirical data evaluations and discussion of results obtained from study, and the final section concludes the paper.

### 2. METHODOLOGY

#### 2.1 Artificial Neural Network (ANN)

The models governing artificial neural networks are given as follows:

$$\begin{aligned} y_1 &= f_1(x_t, x_{t-1}, \dots, x_{t-n}) \\ y_2 &= f_2(x_t, x_{t-1}, \dots, x_{t-n}) \\ &\cdot \\ &\cdot \\ &\cdot \\ y_t &= f_t(x_t, x_{t-1}, \dots, x_{t-n}) \end{aligned} \quad (1)$$

where  $f_1, f_2, \dots$  are functions. The input  $f_1, f_2, \dots$  has the following mathematical relationship:

$$y_t = \alpha_0 + \sum_{j=1}^q \alpha_j (\beta_{0j} + \sum_{i=1}^p \beta_{ij} y_{t-1}) = \varepsilon_t \quad (2)$$

where  $\alpha_j (j = 0, 1, \dots)$  and  $\beta_{ij} (i = 0, 1, \dots, p; j = 0, 1, \dots)$  are parameters.

## 2.2 Kalman Filter Adaptation Technique

Kalman filter algorithm is represented as ([Hox02]):

$$\begin{aligned} y_{1i} &= y_{10} + y_{10}x_{0i} + \beta_{11}y_{1i} + \varepsilon_{oi} \\ y_{2i} &= y_{20} + y_{21}x_{1i} + y_{22}x_{2i} + \beta_{21}y_{1i} + \varepsilon_{1i} \\ y_{3i} &= y_{30} + y_{32}x_{2i} + \beta_{31}y_{1i} + \beta_{32}y_{2i} + \varepsilon_{2i} \\ &\vdots \\ &\vdots \\ &\vdots \\ y_{ii} &= y_{i0} + y_{(t,t-1)}x_{(t-1,i)} + \beta_{t1}y_{1i} + \beta_{t,2}y_{2i} + \dots + \beta_{(t,t-1)}y_{(t-1,i)} + \varepsilon_{(t-1,i)} \end{aligned} \quad (3)$$

$y_{(t-1,i)}$  and  $\beta_{(t,t-1)} (t = 0, 1, 2, \dots)$  are parameters,  $y_{ii}$  is output, and  $\varepsilon_{(t-1,i)}$  are constants.  $x_t$  is given as price; temperature and humidity are given as  $y_t$ . The dependent variable is given as  $y_{ii}$ .

Equation (3) gives

$$y_{ii} = b_{i0} + b_{(t,t-1)}x_{(t-1,i)} + \alpha_{t1}y_{1i} + \dots + \alpha_{(t,t-1)}y_{t-1} \quad (4)$$

$$\varepsilon_{ii} = y_{ii} - \hat{y}_{ii} = b_{i0} - b_{(t,t-1)}x_{(t-1,i)} - \alpha_{t1}y_{1i} - \alpha_{t2}y_{2i} - \dots - \alpha_{(t,t-1)}y_{(t-1,i)} \quad (5)$$

for  $t = 1, 2, \dots, n$

Equation (5) gives values between observed and the predicted values.

$$SSE = \sum_{i=1}^n (y_{ii} - \hat{y}_{ii})^2 = \sum_{i=1}^n (y_{ii} - b_{i0} - b_{(t,t-1)}x_{(t-1,i)} - \alpha_{t1}y_{1i} - \alpha_{t2}y_{2i} - \dots - \alpha_{(t,t-1)}y_{(t-1,i)})^2 \quad (6)$$

The Standard deviation (SD) for the values is:

$$SD = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n}} \quad (7)$$

$x_i$  are the actual values,  $\mu$  is the mean for the values, and  $n$  is sample size of proceeding values. The computation for standard error for values used in this study is given as:

$$SE = \frac{SD}{\sqrt{n}} \quad (8)$$

The accuracy of the results of this study are tested using the *root mean square error (RMSE)* and the *mean average percentage error (MAPE)*. These statistical parameters are computed using the following relationships:

$$\begin{aligned} RMSE &= \sqrt{\frac{\sum_{t=1}^n (e_t)^2}{N}} \quad (9) \\ MAPE &= \frac{1}{N} \sum_{i=1}^n \left| \frac{e_t}{y_t} \right| * 100 \quad (10) \end{aligned}$$

where,  $e_t$  is the individual estimated error,  $y_t$  is the actual value, and  $N$  is total number of sample size. The results for this study are implanted using MATLAB programming language.

## 3. RESULTS AND DISCUSSION

The results for this study are presented in this section. The electricity prices used for the period understudy are displayed in Figure 1. Figure 1 indicates that electricity consumption demonstrates a regular behaviour when compared with price.

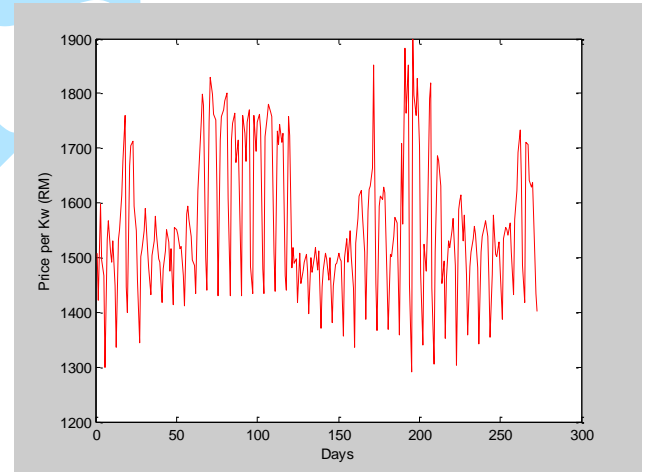


Fig. 1. Consumption relating to price

Table 1 investigates the accuracy of the predicted values for artificial neural network (ANN) and Kalman technique by computing and comparing their respective *root mean square error (RMSE)* and the *mean average percentage error (MAPE)*.

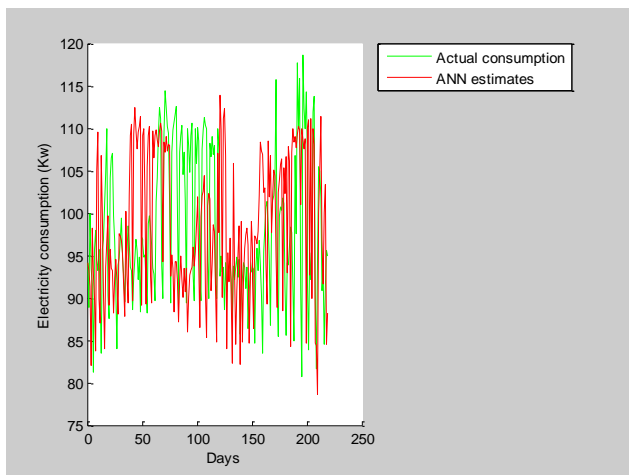
Table 1. Performance of ANN and Kalman algorithm

Parameters	By ANN	By Kalman
RMSE	5.5	7.3E-15
MAPE	4.1	3.9E-15

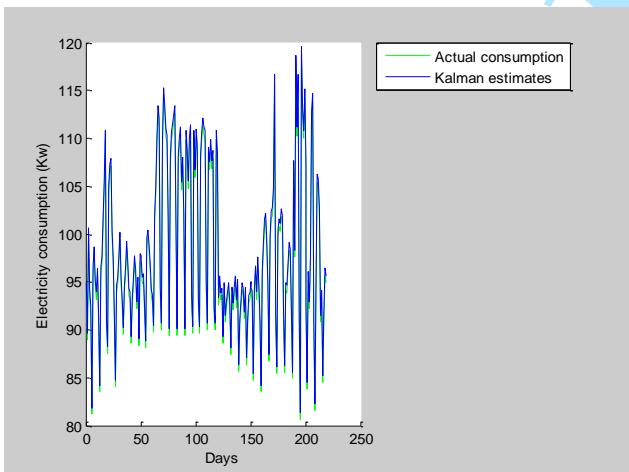
The respective *RMSE* and *MAPE* of Kalman filter adaptation algorithm are lower than that of artificial neural network technique. Table 1 show that the

*RMSE* for Kalman filter adaptation algorithm is  $7.3E-15$ , and  $3.9E-15$  for *MAPE*, which are very much better than artificial neural network, i.e. 5.5 for *RMSE* and 4.1 for *MAPE*. Therefore, Kalman filter adaptation algorithm performs better than artificial neural network.

Figure 2 and Figure 3 predicts daily electricity consumption by artificial neural network (ANN) technique and Kalman filter adaptation algorithm respectively. Figure 3 shows the closeness of predicted values to actual values using Kalman technique compared with ANN. This confirms results obtained in Table 1.



**Fig. 2. Actual consumption compared with predicted values by ANN**



**Fig. 3. Actual consumption compared with predicted values by Kalman method**

The statistical parameters artificial neural network (ANN) and Kalman technique are computed to investigate statistical inference of the results of the study. The standard deviation values of the techniques are compared to show the closeness of the predicted values to actual consumption, and as a result to investigate the accuracy of the results. Table 2 shows standard deviation of Kalman filter adaptation algorithm (0.1%) is lower compared with ANN technique (11.3%). This confirms results

obtained in Table 1, Table 2, Figure 2 and Figure 3 respectively. Thus, Kalman filter adaptation algorithm has a better forecasting accuracy, compared with ANN technique.

**Table 2. Statistical parameters**

Technique	Mean	Std. Dev.	Min.	Max.
Kalman	98.5	0.1%	78.6	113.9
ANN	98.7	11.3%	81.3	119.6

#### 4. CONCLUSIONS

A reliable approach is necessary in modeling and predictions, otherwise inappropriate models and poor estimates may occur. This study investigates the accuracy of developing predictive models using artificial neural network (ANN) and Kalman filter adaptation algorithm. These models are tested on historical data using these techniques. This study indicates that Kalman filter adaptation algorithm is more effective compared to the artificial neural network. This enabled Kalman filter adaptation algorithm to be more effective in predicting data. For future work, a study will be undertaken to develop predictive models of factors affecting household appliance usage. Some of the factors that will be utilized to develop these predictive models include price of electricity usage, atmospheric temperature, and humidity. Models will be developed based on these factors.

#### REFERENCES

- [Hox02] **J. Hox** - *Introduction to structural equation modeling*, Family science review, vol. 11: 354-373, 2002.
- [OAL14] **P. Ozoh, S. Abd-Rahman, J. Labadin** - *A comparative analysis of techniques for forecasting electricity consumption*, International journal of computer applications, vol. 8(15): 8–12, 2014.
- [Tof15] **C. Tofallis** - *A better measure of relative prediction accuracy for model selection and model estimation*, Journal of the operational research society, vol. 66(8): 1352-1362, 2015.
- [ZPH98] **G. Zhang, B. E. Patuwo, M. Y. Hu** - *Forecasting with artificial neural networks: the state of the art*, International journal of forecasting, vol. 14: 35–62, 1998.
- [\*\*\*16] **\*\*\*** - *World Energy Outlook*, Retrieved from doi:10.1787/weo-2013-en/, 2016.

### Appendix A MATLAB Code for ANN Technique

```
% Data_Inputs=xlread('1.xls'); % Import file
%The training data sample are randomized by using the
function'randperm'
Training_Set=Data_Inputs(1:end,1);%specific training set
Target_Set=Data_Inputs(1:end,2); %specific target set
Input=Training_Set'; %Convert to row
Target=Target_Set'; %Convert to row
X = con2seq(Input); %Convert to cell
T = con2seq(Target); %Convert to cell
%% 2. Data preparation
N = 365; % Multi-step ahead prediction
% Input and target series are divided in two groups of data:
% 1st group: used to train the network
inputSeries = X(1:end-N);
targetSeries = T(1:end-N);
inputSeriesVal = X(end-N+1:end);
targetSeriesVal = T(end-N+1:end);
% Create a Nonlinear Autoregressive Network with External
Input
delay = 2;
inputDelays = 1:2;
feedbackDelays = 1:2;
hiddenLayerSize = 100;
% Prepare the Data for Training and Simulation
% The function PREPARETS prepares timeseries data for a
particular network,
% Using PREPARETS allows you to keep your original time
series data unchanged, while
% easily customizing it for networks with differing numbers of
delays, with
% open loop or closed loop feedback modes.
[inputs,inputStates,layerStates,targets] =
preparets(net,inputSeries,{},targetSeries);
% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
% Train the Network
[net,tr] = train(net,inputs,targets,inputStates,layerStates);
% Test the Network
outputs = net(inputs,inputStates,layerStates);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs)
view(net)
netc = closeloop(net);
netc.name = [net.name ' - Closed Loop'];
view(netc)
[xc,xic,aic,tc] = preparets(netc,inputSeries,{},targetSeries);
yc = netc(xc,xic,aic);
% Early Prediction Network
% For some applications it helps to get the prediction a timestep
early.
% The original network returns predicted y(t+1) at the same time
it is given y(t+1).
% For some applications such as decision making, it would help
to have predicted
% y(t+1) once y(t) is available, but before the actual y(t+1)
occurs.
nets.name = [net.name ' - Predict One Step Ahead'];
view(nets)
[xs,xis,ais,ts] = preparets(nets,inputSeries,{},targetSeries);
ys = nets(xs,xis,ais);
earlyPredictPerformance = perform(nets,ts,ys)
%% 5. Multi-step ahead prediction
inputSeriesPred = [inputSeries(end-
delay+1:end),inputSeriesVal];
targetSeriesPred = [targetSeries(end-delay+1:end),
con2seq(nan(1,N))];
```

```
[Xs,Xi,Ai,Ts]
preparets(netc,inputSeriesPred,{},targetSeriesPred);
yPred = netc(Xs,Xi,Ai);
perf = perform(net,yPred,targetSeriesVal);
ANN = yPred;
```

### Appendix B MATLAB Code for Kalman Filter Adaptation Algorithm

```
% Copyright 2010 The MathWorks, Inc.
function y = kalmanfilter(z)
%#codegen
dt=1;
% Initialize state transition matrix
A=[ 1 0 dt 0 0 0;... % [x ]
    0 1 0 dt 0 0;... % [y ]
    0 0 1 0 dt 0;... % [Vx]
    0 0 0 1 0 dt;... % [Vy]
    0 0 0 0 1 0 ;... % [Ax]
    0 0 0 0 0 1 ]; % [Ay]
H = [ 1 0 0 0 0 0; 0 1 0 0 0 0 ]; % Initialize measurement matrix
Q = eye(6);
R = 1000 * eye(2);
persistent x_est p_est % Initial state conditions
if isempty(x_est)
    x_est = zeros(6, 1); % x_est=[x,y,Vx,Vy,Ax,Ay]'
    p_est = zeros(6, 6);
end
% Predicted state and covariance
x_prd = A * x_est;
p_prd = A * p_est * A' + Q;
% Estimation
S = H * p_prd' * H' + R;
B = H * p_prd';
klm_gain = (S \ B);
% Estimated state and covariance
x_est = x_prd + klm_gain * (z - H * x_prd);
p_est = p_prd - klm_gain * H * p_prd;
% Compute the estimated measurements
y = H * x_est;
Kalman = p_est;
end % of the function
```