

A Hybrid Model of Autoregressive Integrated Moving Average and Artificial Neural Network for Load Forecasting

Lemuel Clark P. Velasco¹, Daisy Lou L. Polestico², Gary Paolo O. Macasieb³
Michael Bryan V. Reyes⁴, Felicisimo B. Vasquez Jr⁵
Mindanao State University-Iligan Institute of Technology
Premier Research Institute of Science and Mathematics
Iligan City, The Philippines

Abstract—The complementary strengths and weaknesses of both statistical modeling paired with machine learning has been an ongoing technique in the development and implementation of forecasting models that analyze the dataset's linear as well as non-linear components in the generation of accurate prediction results. In this paper, autoregressive integrated moving average (ARIMA) and artificial neural networks (ANN) were implemented as a hybrid forecasting model for a power utility's dataset in order to predict the next day's electric load consumption. ARIMA and ANN models were serially developed resulting to the findings that out of the twelve evaluated ARIMA models, ARIMA (8,1,2) exhibited the best forecasting performance. After identifying the optimal ANN layers and input neurons, this study showed that out of the six evaluated supervised feedforward ANN models, the ANN model which employed Hyperbolic Tangent activation function and Resilient Propagation training algorithm also exhibited the best forecasting performance. With Zhang's ARIMA and ANN hybridization technique, this study showed that the hybrid model delivered Mean Absolute Percentage Error (MAPE) of 4.09% which is within the 5% internationally accepted forecasting error for electric load forecasting. Through the findings of this research, both the ARIMA statistical model and ANN machine learning approaches showed promising results in being implemented as a forecasting model pair to analyze the linear as well as non-linear properties of a power utility's electric load data.

Keywords—Hybrid model; autoregressive integrated moving average; electric load forecasting; Artificial Neural Network (ANN)

I. INTRODUCTION

Use of individual machine learning and statistical modeling has been in the forefront of predictive analytics due to their promising abilities to deliver close to accurate forecasting results. Autoregressive Integrated Moving Average (ARIMA) is a statistical modeling approach which has been widely used in forecasting with promising accuracy brought about by efficient linear representation exhibited by non-linear systems [1-4]. With its strength in modeling time series data such as consumed electric load as well as datasets with validated linearity properties, ARIMA forecasting applications is a growing body of researches with various applications in different fields [3]. Artificial Neural Networks (ANN) is a machine learning tool that finds patterns in the same way that

biological neural networks develop association along with mathematical equivalent elements capable of processing like that of the human brain [2, 4, 5]. Compared to that of ARIMA, ANN has the ability to learn from non-linear datasets due to its strength of being adaptively formed from the implemented features of its own dataset. Despite their differences in the kind of data that they can accommodate, ARIMA and ANN hybrid forecasting methodologies as well as modelling techniques are being widely developed due to the potential of generating better predictive performance than individually utilizing each model [2, 6]. For the purpose of optimal predictive performance, the main challenge in the hybridization of these machine learning and statistical modeling approaches relies on the optimal match between the data they are processing along with the forecasting ability that they enforce in their inherent unique advantages. This gives data modelers the challenge beyond the functions of data preparation and explore on the performance analysis of the ARIMA and ANN hybridization technique that can yield optimal predictive results.

Datasets such as electric load data bearing recorded consumption behavior through time has linear along with non-linear properties [5, 7, 8]. A combinatory modelling technique of these two models with ARIMA to handle the dataset's linearity and ANN to handle the non-linearity can result to more efficient forecasting outputs than just independently using one of them. This combined model is also suitable for both one-step ahead and multistep ahead predictions in generating better hybrid model performance for natural and economic datasets [1, 9, 10]. Power systems from the functions of generation, transmission and distribution can make use of historical load data in the development of load forecasting models that can aid decision makers in operations planning. Forecasting the load consumption in different time frame granularities whether it be week-ahead, day-ahead or hour-ahead predictions can benefit power systems in the provision of demand information used to carry out required actions and planning processes that ensure reliable power systems [5, 11]. With historical electric load consumption data bearing both linear and non-linear properties, a specified hybrid model imploring the unique strengths of both ARIMA and ANN can be a potential implementation in the generation of close to accurate electric load forecast. The presence of historical load datasets as well as the absence of forecasting models that can

process these datasets for tactical and strategic operations planning is a common problem faced by power generation, transmission and distribution entities. This research aims to present a hybrid model of ARIMA and ANN in the processing of electric load data in order to generate optimal day-ahead forecasting results. Despite availability of literatures that explore the hybridization of ARIMA and ANN among non-electricity related datasets, this study aims to present a personalized hybrid model specific in the processing and prediction of electric load [1, 4, 6, 10]. With an exploration on electric load data preparation and hybridization technique in ARIMA and ANN, the results of this study hopes to aid data modelers and decision makers in the development and implementation of a forecasting model that utilizes both machine learning and statistical modeling for better management of power systems.

II. METHODOLOGY

A. ARIMA Modeling

Historical electric load data of a power utility from December 27, 2013 to October 21, 2014 for a total of 28, 704 records were chosen as inputs to create both ARIMA and ANN models. As shown in Table I, the dataset comes in 15-minute records of the date, time, kilowatt delivered (KW_DEL), kilowatt per hour delivered (KWH_DEL) and kilo volt amps reactive hours delivered (KVARH_DEL). The column utilized as input to the models was kilowatt delivered (KW_DEL) since this represents the consumed electric load which can be used to determine the predicted values for the next day. The entire 15-minute data of October 21, 2014 was used as the testing set for the overall testing in ARIMA forecast, ANN forecast, and Hybrid forecast.

Since the data contains unscheduled power interruptions, several zero values were found in the historical data causing the dataset to become inefficient and out-of-range. Data correction was then performed in order to replace outlying values that could possibly alter the behavior and final results of the electric load forecasting model and might produce poor results. Outlying values of the electric load consumption data was replaced with the respective preceding day with the same time frame of the outlying value. This process of replacing outlying values by its preceding day's respective data was supported by studies since the missing data per day has a tolerable occurrence [4, 5, 12]. Since the electric load consumption data was recorded in fifteen-minute interval, the maximum consumed electric load among the hour's four fifteen-minute recordings was chosen to represent the hour's consumption as suggested by researchers [5, 13]. By doing this the new number of observation would be 7, 176 with 7, 152 observations to be used for training the model and the last 24-hourly observation for testing the model. The number of observations to be used is just efficient for the hybrid model because a larger amount could lead to overfitting the ARIMA model while a smaller and inappropriately minimal amount could possibly lead to underfitting the ANN model [9, 14].

ARIMA modeling was conducted through model identification, model estimation, diagnostic checking, and forecasting phases. The model identification phase involves determining the order of the ARIMA model p , d , and q , where

p represents the autoregressive terms, d represents the non-seasonal differences needed for stationarity, and q represents the lagged forecast errors in the prediction equation [10, 12, 14]. Modeling an ARMA (p, q) process requires stationarity in order to fit this model easily. A time series data is said to be stationary if its statistical properties do not depend on the time at which the series is observed having both mean and variance that do not change over time with the process not having trends [6, 15]. A method called differencing where differences are computed between consecutive observations must be done to achieve stationarity. Equation 1 shows the process for first order differencing where t is the differenced variable and y_t is the time series variable.

$$\Delta y_t = y_t - y_{t-1} \tag{1}$$

A stationarity test called the Dickey-Fuller test was then used to determine if the variable is stationary or not [14]. The Dickey-Fuller test tests the null hypothesis of whether a unit root or non-stationary is present in an autoregressive model. First regular differencing as seen in Equation (1) was applied to meet the condition for stationarity and if by doing this the electric load data is still not in a stationary condition, the second regular differencing will be applied. Once stationarity have been addressed, the next step is to identify the order p and q of the autoregressive and moving average terms respectively [14, 15]. This involves plotting the data over time and the corresponding Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF). By plotting the ACF and PACF, the researchers can come up to possible ARMA models that will be used later in estimation. After the nomination of one or more appropriate models to describe the view of the time series data, parameters of the model was estimated using an estimation method used in the original ARIMA Box and Jenkins methodology. This estimation method followed a guiding principle of parsimony that is the total number of parameters of the model should be as small as possible which makes a model a good fit [14]. In addition to estimating a model for the electric load consumed data, the model with the smallest parameters is more promising to the efficient forecast due to more stable parameters obtained. Using a high quality statistical software, the coefficients of the parameters was determined to come up with the final ARIMA model that would fit the original data. These ARIMA models were further examined to check if all the parameters are significant using the assumptions of the criteria called the Akaike Information Criterion (AIC) and p-value. The AIC is an index used in a number of areas as an aid to choosing between competing models which defined as where L_m is the maximized log-likelihood and m is the number of parameters in the model. Among the set of suggested models that are being estimated, a model with the smallest AIC and has significant coefficients was chosen as the final model that will be validated in the diagnostic checking phase.

TABLE. I. ELECTRIC LOAD DATA

DATE	TIME	KW_DEL	KWH_DEL	KVARH_DEL
XXX	XXX	XXX	XXX	XXX
XXX	XXX	XXX	XXX	XXX

Once a model has been identified, the result was then considered as the final model and forecasts was obtained accordingly. In a normal diagnostic check in an ARIMA Box and Jenkins methodology, fitted models was checked if it is a satisfactory one to protect against forecasting errors. This was implemented by the use of standard diagnostic checks that examined the correlogram of the residuals from the fitted model to see if the residuals exhibit white noise. A good forecasting method should yield residuals with a white noise in which residuals are uncorrelated and have zero mean. If the fitted model is a good model for the data, the residuals should satisfy these assumptions. If these assumptions are not satisfied, a more appropriate model should be fitted and the whole process will go back to the model identification step and try to develop a better model. In the case of passing the model for these tests, a final model was adopted which is used to estimate linear predictions of the electric load data. Moreover, the residual series was analyzed using the Box-Pierce Q-statistics as recommended by studies to make clear that the values are normally distributed [6, 14, 15]. When the residual series finally meets the condition of normal distribution, then the model could be used in predicting future values. After the model has been estimated and validated, this model was identified as the final model for ARIMA forecasting. Using the provided electric load dataset, the final model will obtain the corresponding linear forecast L_t at time t and the residual of the observed value where y_t is the observation at time t . The residuals dataset was then modeled by ANN.

B. ANN Modeling

Simple perceptron can only identify sets of data that are linearly separable, and when the input data to be classified are not linearly separable, learning and classification will never reach an optimum point of distinct separation [16]. In situations like this, Multilayered Perceptron (MLP) model is used in learning and classification. Thus, a MLP having input, hidden and outputs units as a type of ANN was used in formulating an ANN Model to resolve the problem of electric load forecasting [5, 7]. As a feed-forward ANN having input neurons, hidden neurons and outputs neurons, it allows signals to travel one way only, from input to output [7, 16-17]. In this study, the number of inputs of the neurons depends on the residuals of the ARIMA model, which has non-linear relationship. The hidden layer, on the other hand, serves to encode the input and map it to the output. Identifying the ANN's hidden neurons in the hidden layer does not have any standardized or theoretical approach but there are some empirically-derived rule-of-thumb approaches [5, 18]. Trial and error was then used to determine the optimum neurons in the hidden layer of the network using formulas from different researchers on how to crosscheck the efficient number of hidden neurons [4, 17-19]. Since the output layer is where the outcome of the network can be seen, the number of output neurons solely depends on the problem that the neural network wants to learn.

Backpropagation is considered as one of the original training algorithms for feedforward neural networks that uses both learning rate and momentum with learning rate as the variable of learning agility and the momentum as the variable that helps the network get out of the established local minima [5, 12]. Manhattan Propagation on the other hand uses a delta

value in updating its weight values. Resilient propagation training algorithm is unique compared to the two earlier mentioned ANN training algorithms since it does not require training parameters making it much easier to model and utilize, but has shown better performance efficiency than that of Manhattan Propagation and Backpropagation [6, 17]. As shown in Table II, this study used different pairs of training algorithms with Hyperbolic Tangent and Logarithmic activation functions since these activation functions has the ability to produce results between -1 and 1 fit to the datasets that were transformed in a scale of -1 and 1. This was done conservatively for the purpose of avoiding ambiguous values.

After the architecture of the ANN model, the next phase involves training the forecasting model. During the training process, test inputs were implemented to the electric load consumed training dataset using the training algorithms. These training algorithms were used to update the network weights and adjust biases of the network until the error is less than the desired limit [9, 17, 19]. The learning parameters used in this study were the desired error, and number of iterations. The desired error was set to 0.0001 to help the network reach an optimal solution with the smallest amount of error. The standard range for the desired error should be between 0.0005 and 0.0001. The lower the desired error implies optimal the result. Since the goal of this study is to get an error rate below the international error rate of 0.0005 and to have efficient and optimal results, the desired value was set to 0.0001. In choosing the number of iteration, the number of records to be used is accounted for resulting to 10, 000 as the identified number of iterations.. If the number of records is 500 then the iteration is between the range of 500+250 and 500-250 or 750 to 250. In this study, since the amount of records to be used is 7,152 for training, then the iteration to be used is within the range 10, 728 and 3576. The researchers then decided to use the value of 10000. These parameters were used in order to have an equal credibility in comparison with the forecasting models. To measure accuracy, error measure in terms of Mean Square Error (MSE) was calculated to determine the predictive capability of the models. For this error measure, resulting values will always be non-negative and values closer to zero are better [7]. Thus, smaller value from the resulting calculation would indicate a consistency of performance in the neural network.

TABLE. II. FORMULATED MLP MODELS

Model	Training Algorithm	Activation Function	Model	Training Algorithm	Activation Function
Model 1	Backpropagation	Hyperbolic Tangent	Model 4	Backpropagation	Logarithmic
Model 2	Manhattan Propagation	Hyperbolic Tangent	Model 5	Manhattan Propagation	Logarithmic
Model 3	Resilient Propagation	Hyperbolic Tangent	Model 6	Resilient Propagation	Logarithmic

C. Hybrid Model Implementation

In implementing the ARIMA model, Dickey-Fuller test was said to test stationarity followed by the identification of a candidate model to be used. The chosen model was estimated along with variations of the model for comparison. The residuals of the chosen model underwent a diagnostic check by doing a portmanteau test. The result of the final model was a linear forecast of the hybrid model, along with the residuals calculated by subtracting the actual dataset with resulting dataset which was then fed to the ANN model. The resulting linear forecasted data from the ARIMA model would be stored in a database. The ANN model was implemented in Encog, a java-based system for ease of simulation and calculation of the training and testing. The residuals from the ARIMA model would be read in Encog which training set and testing set being partitioned from the residuals. After partitioning the datasets, the next step is configuring the neural network and adding a hidden layer. After the hidden layer is added, the network would be structured and undergo a reset. After the network undergo a reset, it would then be trained along with the training set and then iterated. Once the iteration is finish, a finalized neural network is created. The last step is to calculate the error of the finalized neural network on the testing set. After the calculation, Encog would store the data in a database and exit. The result from neural network would be the nonlinear forecast of the hybrid model.

In the Hybrid Implementation phase, the linear and nonlinear forecasted dataset obtained from the previous phases stored in the database would be added manually. As shown in Equation (2), the process of adding the linear and nonlinear forecasted dataset was based on Zhang's ARIMA-ANN hybridization in which it is assumed that the given time series data is a sum of two components: linear and non-linear [4].

$$y_t = L_t + N_t \quad (2)$$

On the given time series (y_t), ARIMA is fit and the linear predictions are obtained, (\hat{L}_t) by Equation (3).

$$\hat{L}_t = a_1 y_{t-1} + \dots + a_p y_{t-p} + b_1 e_{t-1} + \dots + b_q e_{t-q} + e_t \quad (3)$$

The difference series is obtained by Equation (4) on which ANN is fit and the predictions (\hat{N}_t) are obtained using Equation (5).

$$n_t = y_t - \hat{L}_t \quad (4)$$

$$\hat{N}_t = f(n_t, n_{t-2}, \dots, n_{t-A}) + v_t \quad (5)$$

The hybrid model predictions are now obtained by summing the ARIMA and ANN predictions as shown in Equation (6).

$$\hat{y}_t = \hat{L}_t + \hat{N}_t \quad (6)$$

To summarize the hybridization process, a block diagram is shown in Fig. 1. The sums were stored in the same database and was read in the Java-based system. In the Java Interface, it would show the hybrid forecasted dataset along with the specific time and date for each data.

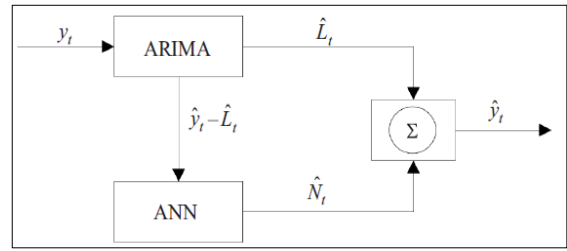


Fig. 1. A Block Diagram of the ARIMA and ANN Hybridization.

After the implementation of the ARIMA and ANN hybrid model, the results from the hybrid model were assessed using MSE. Electric load consumption data for October 21, 2014 was used in testing the hybrid model.

III. RESULTS AND DISCUSSION

A. ARIMA Modeling Results

The results of the ARIMA modeling were divided into four stages from model identification, model estimation, diagnostic checking, and forecasting phases. The model identification phase involves determining the order of the ARIMA model p , d , and q . Before modeling the data, the entire 7,152 hourly observations from the raw electric load data from December 2013 to October 2014 was first plotted in a basic plot to visualize the behavior of the data. The plot as shown in Fig. 2 revealed compactness of the series with huge randomness which signifies the nonlinearity of the data. This constant trend shows that the series is in nonstationary condition and there's a need to apply differencing technique to make the series stationary [15].

Moreover, to support this claim, the test for stationarity called the Dickey-Fuller test which tested the null hypothesis of whether a unit root is present was applied which obtained a p-value of 0.81 from a lag of 24 which is equivalent to the 24 hourly values per day data. A p-value more than 0.01 signifies a non-stationary data [14]. Thus, the test signifies a nonstationary condition and the null hypothesis was not rejected. The dataset, in order to achieve stationarity of the series, was applied with the differencing technique of $y_d = y_t - y_{t-1}$ where y_t is the load at time t and y_d is the differenced load. After the technique was applied, the Dickey-Fuller test showed a result of 0.01 p-value with a lag value of 24 which signifies a stationary condition. The data was plotted as shown in Fig. 3.

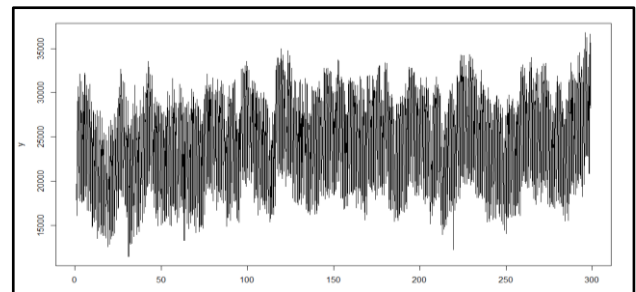


Fig. 2. Electric Load Data Plot.

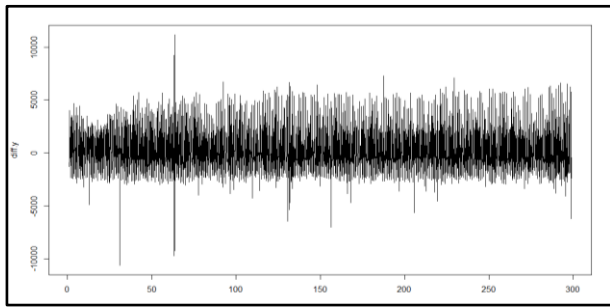


Fig. 3. Differenced Series.

The differenced data as clearly seen in the plot is already in a stationary condition with a constant mean and variance and so the series was ready for model identification. The ACF and PACF of the differenced series was being plotted as shown in Fig. 4 and Fig. 5, respectively which shows a tail-off behavior in the ACF and the PACF cuts off after the 9th lag. This behavior shows a stationary data and a tentative autoregressive-moving average model [15]. Thus, this behavior signifies stationarity of the series and an autoregressive (AR) and moving average (MA) model. The feasible number of AR and MA term is the lag that is zero or approximately equivalent to zero [12]. Based on the ACF, the AR term can be lags 5, 6, 7, or 8 which can be determined by looking at which lag is zero or close to zero. While based on the PACF plot, MA term can be identified the same as the AR do, so the MA term could be 2, 4, or 5. Based on the ACF and PACF results, the model that will be estimated will be ARIMA(5,1,2), ARIMA(5,1,4), ARIMA(5,1,5), ARIMA(6,1,2), ARIMA(6,1,4), ARIMA(6,1,5), ARIMA(7,1,2), ARIMA(7,1,4), ARIMA(7,1,5), ARIMA(8,1,2), ARIMA(8,1,4), and ARIMA(8,1,5) [14-15].

The purpose of ARIMA model estimation is to select a parsimonious model from the generated models which will base on the lowest total number of parameters and AIC. When comparing models fitted by maximum likelihood to the same data, the smaller the AIC implies better fit [1, 3]. After the nomination of the appropriate models based on the ACF and PACF, the AIC test was done on the generated models. The one with the smallest AIC had been chosen as the final model for forecasting. This estimation method follows a guiding principle of parsimony that is the total number of parameters of the model should be as small as possible which makes a model a good fit [14]. Table IV presents the possible ARIMA models with their corresponding criteria in terms of Root Mean Squared Error (RMSE) and the AIC. During the run, some models produced an error in fitting which outputs no AIC and RMSE which indicates that the model is not good for the feature of the data. These models are ARIMA(6,1,4), ARIMA(6,1,5), ARIMA(7,1,5). In this case, the model with the least criterion as a whole is ARIMA(8,1,2) with an RMSE of 1247.582 and AIC of 122283 as indicated in Table III. Based on the results, the effective order of the AR terms is found to be $p = 8$, the MA terms is equal to $q = 2$, and the differentiation parameter is $i = 1$ since the raw series is differenced in the first order. The final model of order ARIMA(8,1,2) was chosen as the model for validation in the next phase.

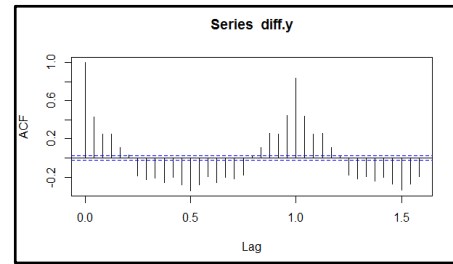


Fig. 4. Autocorrelation Function Plot.

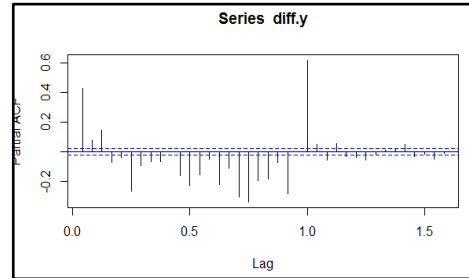


Fig. 5. Partial Autocorrelation Function Plot.

TABLE III. POSSIBLE ARIMA MODELS

Model	RMSE	AIC	Model	RMSE	AIC
ARIMA(5,1,2)	1396.16	123881.9	ARIMA(7,1,2)	1357.84	123487.5
ARIMA(5,1,4)	1438.60	124315.1	ARIMA(7,1,4)	1259.14	122420.7
ARIMA(5,1,5)	1449.25	124419.4	ARIMA(7,1,5)	N/A	N/A
ARIMA(6,1,2)	1357.94	123486.5	ARIMA(8,1,2)	1247.58	122283
ARIMA(6,1,4)	N/A	N/A	ARIMA(8,1,4)	1259.69	122424.4
ARIMA(6,1,5)	N/A	N/A	ARIMA(8,1,5)	1249.38	122312.8

Before a forecast was generated using the ARIMA model, the model was first checked to test the adequacy of the overall model to prevent forecast errors. Residuals of the model was generated and checked by examining the correlogram to see if the residuals process a white noise. A residual in forecasting is the difference between an observed value and its forecast based on other observations [4, 15]. The residuals were plotted in a standard plot and autocorrelation plot as shown in Fig. 6 and Fig. 7, respectively. The residuals standard and ACF plot of the fitted model did not satisfy the diagnostic checking phase. The standard plot shown in Fig. 6 shows a slight pattern of data which rejects the white noise assumption. While the ACF plot in Fig. 7 shows significant spikes especially in lag 24 which correlates to the 24 hourly data and not all lags fall outside the confidence interval.

The Box-Ljung test was also applied to the residuals and the test showed a p-value of 2.2e-1. Since this study is a hybrid process of ARIMA and ANN for linear and nonlinear components respectively, the results of the diagnostic checking phase shows that it does not satisfy the claim to be a random

distribution, thus was fixed by feeding the residuals to ANN for it to model and forecast a residual data for hybridization. Based on the results, the researchers found out that there was no evidence to claim that the residuals are random. Thus, the assumption that the electric load data comprised of nonlinear component was proved and the residuals data needs to utilize ANN modeling for residuals forecasting to later be joined in ARIMA forecast. The final chosen model as a whole in ARIMA modeling phase was ARIMA(8,1,2) which has a coefficients per term shown in Table IV.

The model equation is where B is the backshift operator and a_t is the white noise. A 24-hour forecast was generated using this model which is shown in Fig. 8. The blue line is the point forecast from ARIMA(8,1,2) with its high and low boundaries. While the red line represents the actual 24-hour value of October 21, 2014. As shown in Fig. 5, the 24-hour forecast values of this model were imported in a database for later use in comparison.

While residuals of the fitted model for a total of 7,152 data points were passed on to the next process for ANN modelling and then will forecast residuals to be merged in the hybrid process. Fig. 9 shows that the raw data of 7,152 values being fed to the ARIMA generated the same number of residuals to be used then in ANN modeling and a 24-ahead forecast to be test in the hybridization process.

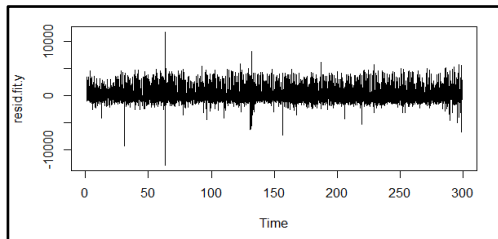


Fig. 6. Residuals Standard Plot.

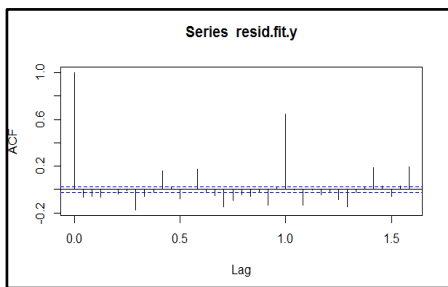


Fig. 7. Residuals Autocorrelation Plot.

TABLE IV. ARIMA COEFFICIENTS

AR1	AR2	AR3	AR4	AR5	AR6	AR7	AR8	MA 1	MA 2
2.0165	-1.3577	0.4737	-0.4029	0.3233	-0.4254	0.5849	-0.3244	-1.9283	0.9956

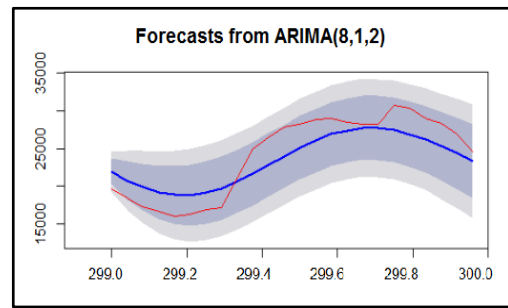


Fig. 8. Actual vs. Predicted Plot.

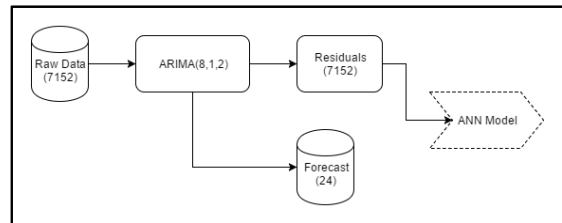


Fig. 9. ARIMA Modeling Data Flow.

B. ANN Modeling Results

The results of the ANN model were divided into four phases namely ANN data preparation, ANN model formulation, ANN model training, and residuals forecasting. The model formulated in the ARIMA phases generated residual data for the ANN to model. After the residuals were generated in the ARIMA modeling, the data were plotted in the graph as shown in Fig. 10.

The plotted residual values shows a random distribution of values with minimum value of -2680.893454503 and maximum value of 3652.1309335015 as these values are within the range of -2680.893454503 and 3652.1309335015, the boundaries for the residual transformation process. It is common in ANN modeling to undergo data transformation between a specified range e.g. -1 to 1 or 0 to 1 since this makes the training of the network efficient to yield accurate predictive results [6-8]. It was found out that the residuals generated from the ARIMA model did not satisfy this requirement resulting to a need for the data to be transformed. Shown in Table V are sample of the residuals dataset that underwent a transformation technique using Min-Max normalization which scaled down the dataset to a range of -1 to 1.

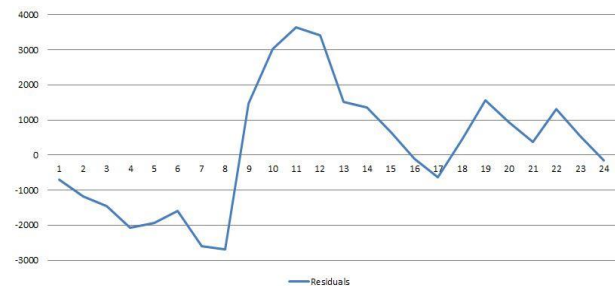


Fig. 10. Residuals Dataset.

TABLE. V. SAMPLE ACTUAL RESIDUALS AND NORMALIZED RESIDUALS

Actual	Normalized	Actual	Normalized	Actual	Normalized
-1459.63138	-0.524629072	3652.130934	1	1564.143068	0.340603245
-2596.55235	-0.973364668	648.3317892	0.051385575	522.33606	0.011595509
-2680.89345	-1	110.760525	-0.09525267	-154.224759	-0.80737623

Designing the architecture of an ANN model includes the identification of the number and size of the input, hidden, and output layers. Although neural networks are used for the purpose of unsupervised learning, classification, or regression, in this study, regression was the appropriate application for the problem since residuals have time series structure [1, 7, 10]. A regressor neural network used two window sizes for the model training which are the input window size and the output window size [9, 12]. The input window used the number of input nodes while the output window size is equivalent to the number of the forecasting results being 24 hourly values in a day. The block diagram shown in Fig. 11 implies that all the models that were created were of a feedforward type of ANN where there is 1 layer containing 24 input neurons for the input layer to represent the 24 data points of the 24 hourly residuals.

If the neural network is used for the purpose of regression, then the output layer has a single node [1]. This was also supported in a study which also used 1 output neuron for the output layer [19]. Since the data used in this study is a time series data, the output layer used contains 1 neuron. A neural network with one hidden layer has the tendency to perform very well depending on the problem [5, 19]. The researchers used 1 layer for the hidden layer. In order to select the appropriate number of hidden neurons, the researchers conducted a test with the result shown in Table VI. Each identified number of hidden neurons was tested using the final ANN model and yielded error measures in terms of MSE. From the 4 hidden neurons results, 17 hidden neurons had the lowest MSE while 4 hidden neurons had the highest MSE. Thus, this study used 1 hidden layer containing 17 of hidden neurons.

The ANN model architecture composed 24 input neurons, 17 hidden neurons and 1 output neuron. This research produced 6 models and used different kinds of training algorithms, namely, Backpropagation, Manhattan Propagation, and Resilient Propagation which were Hyperbolic Tangent and Logarithmic functions. In comparing the different ANN models, the model which had the lowest error is the Model 3 which has the combination of Resilient Propagation and Hyperbolic Tangent and generated a network error of 0.003668849. As shown in Table VII, better precision and accuracy of prediction were seen in Model 3 and Model 6 which uses Resilient Propagation as training algorithm, however Model 3 outperformed Model 6 in terms of lesser

MSE or prediction accuracy. On the other hand, Model 2 and Model 5 which uses Manhattan Propagation as training algorithm produced the neutral values of MSE which means it has the capability of performing predictions, however, these models were not enough to have a better prediction.

Model 1 and Model 4 which used Backpropagation as Training Algorithm produced bigger value of MSE which only means that it has less capability of achieving good prediction or low quality of performance. Moreover, results showed that models which had Resilient Propagation as the training algorithm produced the smallest amount of error which only means that better predictions can be seen in these models. Resilient Propagation exhibited more efficient performance than Manhattan Propagation or Backpropagation for supervised feedforward neural networks. This made an advantage for the Resilient Propagation training algorithm since there were no learning rates, momentum values or update constants that need to be determined. On the other hand, the Backpropagation training algorithm used two parameters in conjunction with the gradient descent which may result to a problem in the algorithm because the gradient descent algorithm should be able to seek out local minima. These local minima are points of low error but may not be a global minimum. With Manhattan Propagation training algorithm, the sign of the gradient and the magnitude is discarded. This means that it is only important if the gradient is positive, negative, or near zero. When all the propagation training algorithms were paired with the Hyperbolic Tangent activation function, it stood out and resulted lesser error than the other activation function. This is because Hyperbolic Tangent activation function works with both negative and positive numbers. It has a derivative function which can be used with propagation training making it a common choice for feedforward and simple recurrent neural networks. Using Model 3, the graph shown in Fig. 12 shows the forecasted values of the residuals for the next 24 hours.

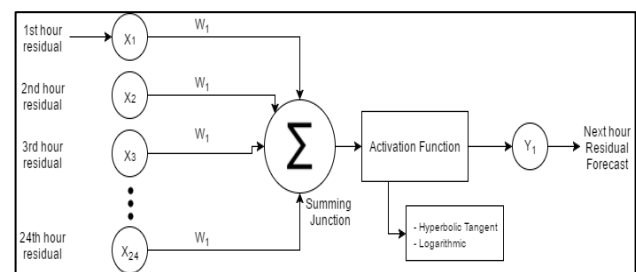


Fig. 11. Block Diagram for ANN Architecture.

TABLE. VI. HIDDEN NEURON RESULTS

Hidden Neurons Technique	Hidden Neuron	MSE
The square root of input neuron plus the output neuron	5	0.000149
The average of input and output neurons	13	0.000119
The square root of input neuron multiplied with the output neuron	4	0.000153
2/3 of the input neuron plus the output neuron	17	0.000110

TABLE. VII. ANN MODEL RESULTS

ANN Models	MSE	ANN Models	MSE	ANN Models	MSE
Model 1	0.00490	Model 3	0.00111	Model 5	0.00426
Model 2	0.00406	Model 4	0.00505	Model 6	0.001273

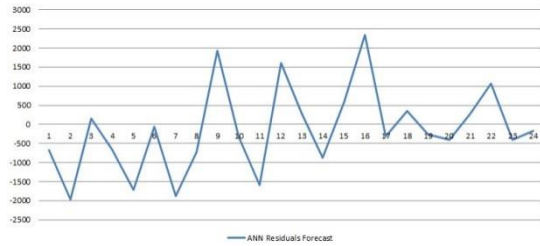


Fig. 12. ANN Residuals Forecast Plot.

TABLE. IX. MAPE OF THE MODELS

Model	Mean Absolute Percentage Error
ARIMA Model	5.11%
ANN Model	16.90%
ARIMA+ANN Hybrid Model	4.09%

IV. CONCLUSION AND RECOMMENDATIONS

This study attempted to present a hybrid model of ARIMA and ANN in load forecasting. The methodology focused in the formulation and performance evaluation of twelve ARIMA models and six ANN models with different combination of training algorithm and activation functions. Since ARIMA (8,1,2) exhibited the best forecasting performance, its residuals were then processed by the best performing ANN model employing Hyperbolic Tangent activation function and Resilient Propagation training algorithm which generated the ANN forecast. The independent results of both ARIMA and ANN models were then processed following Zhang’s hybridization technique that generated a MAPE of 4.09% which is generally lower than the internationally accepted 5% MAPE for electric load forecasting.

This study only focuses on a single hybridization technique of ARIMA and ANN models. Despite the very limited literature that attempted to develop and implement ARIMA and ANN models in forecasting, the researchers still recommend that other hybridization techniques should be explored along with the performance evaluation of these hybridization techniques. The possibility of implementing other hybrid implementation frameworks other than that of Zhang’s ARIMA and ANN hybridization process can yield to a fundamental rethinking of how statistical and machine learning models can process linear and non-linear datasets for forecasting functions. The results of this study clearly suggest that a forecasting model that utilizes both statistical modeling and machine learning can perform promising results that can be used for better management of power systems.

ACKNOWLEDGMENT

The authors would like to thank Mr. Art Louie Bangcale for his technical assistance in the models evaluation. Additionally, the authors would also like to thank the support of the MSU-IIT Office of the Vice Chancellor for Research and Extension and PRISM-Premiere Research Institute in Sciences and Mathematics for their assistance in this study.

REFERENCES

- [1] B. C. Narendra, and R. C. Eswara, “Performance Comparison of Four New ARIMA-ANN Prediction Models on Internet Traffic Data,” Journal of Telecommunications and Information Technology, 2015.
- [2] Y. Yang, J. Wu, Y. Chen, and C. Li, “A New Strategy for Short-Term Load Forecasting,” Abstract and Applied Analysis, 1-9, 2013, doi:10.1155/2013/208964.
- [3] N. Mohamed, M. Ahmad, Z. Ismail, and S. Suhartono, “Short Term Load Forecasting Using Double Seasonal ARIMA Model,” World Applied Sciences Journal 15(2), p 57-73, 2010.

TABLE. VIII. ADDING THE LINEAR AND NONLINEAR FORECASTS

	ARIMA Forecast		ANN Forecast		ARIMA+ANN Forecast
1	26510.853	+	-183.272	=	26327.581
2	25565.625	+	-1245.971	=	24319.654
...	...	+	...	=	...
24	28439.424	+	-463.978	=	27975.445

- [4] P. G. Zhang, "Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model," *Neurocomputing*, 50, p 159-175. 2003, doi:10.1016/s0925-2312(01)00702-0.
- [5] L. C. Velasco, D. L. Polestico, G. P. Macasieb, M. B. Reyes, F. Vasquez, "Load Forecasting using Autoregressive Integrated Moving Average and Artificial Neural Network," *International Journal of Advanced Computer Science and Applications*, 2018.
- [6] M. Kumar and M. Thenmozhi, "A Comparison of Different Hybrid ARIMA-Neural Network Models for Stock Index Return Forecasting and Trading Strategy," *International Journal of Financial Management*, 1(1), 2012.
- [7] H. Chen, C. Canizares, and A. Singh, "ANN-based short-term load forecasting in electricity markets," *Conference Proceedings of the IEEE Power Engineering Society Winter Meeting*, 2001, doi:10.1109/pesw.2001.916876.
- [8] F. Liu, Z. Wang, J. Wu, and J. Wang, "A Hybrid Forecasting Model Based on Bivariate Division and a Backpropagation Artificial Neural Network Optimized Chaos Particle Swarm Optimization for Day-Ahead Electricity," *Abstract and Applied Analysis*, p 1-31, 2014, doi:10.1155/2014/249208.
- [9] I. V. Sousa, F. G. Martins, M. C. M. Alvim-Ferraz, M. C. Pereira, "Multiple linear regression and artificial neural networks based on principle components to predict ozone concentrations," *Environmental Modelling & Software*, Volume 22, Issue 1, p 97-103, 2007.
- [10] L. Barba, N. Rodriguez, and C. Montt, "Smoothing Strategies Combined with ARIMA and Neural Networks to Improve the Forecasting of Traffic Accidents," *The Scientific World Journal*, p 1-12, 2014, doi:10.1155/2014/152375.
- [11] H. Cui, and X. Peng, "Short-Term City Electric Load Forecasting with Considering Temperature Effects: An Improved ARIMAX Model," *Mathematical Problems in Engineering*, p. 1-10, 2015, doi:10.1155/2015/589374.
- [12] W. Wei, J. Jiang, H. Liang, L. Gao, B. Liang, J. Huang, and H. Chen, "Application of a Combined Model with Autoregressive Integrated Moving Average (ARIMA) and Generalized Regression Neural Network (GRNN) in Forecasting Hepatitis Incidence in Heng County, China," *PLOS ONE PLoS ONE*, 11(6), 2016 doi:10.1371/journal.pone.0156768.
- [13] J. E. Gardner and R. L. Lehr, "Enabling the Widespread Adoption of Wind Energy in the Western United States: the Case for Transmission, Operations and Market Reforms," *Journal of Energy & Natural Resources Law*, 31(3), pp. 237-285, 2013, doi:10.1080/02646811.2013.11435333.
- [14] A. Alexandru and I. Dobre, "Modelling Unemployment Rate Using Box-Jenkins Procedure," *Journal of Applied Quantitative Methods*, 2008.
- [15] R. J. Hyndman and G. Athanasopoulos, "Forecasting: Principles and Practice", Monash University, Australia.
- [16] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," *.Informatica* 31, University of Peloponnese, 2007.
- [17] S. Param, M. Chowdhury, D. Lamp, E. Kendall, and E. Nygard, "Energy Demand Prediction Using Neural Networks" *Proceedings of the Conference: 28th International Conference on Computer Applications in Industry and Engineering*, 2016.
- [18] D. Palchak, S. Suryanarayanan, D. Zimmerle, "An Artificial Neural Network in Short-Term Electrical Load Forecasting of a University Campus: A Case Study," *Journal of Energy Resources Technology*, p 135(3), 2013, doi:10.1115/1.4023741.
- [19] J. Lipae, and E. Deligero, "On Forecasting Water Consumption in Davao City Using Autoregressive Integrated Moving Average (ARIMA) Models and the Multilayer Perceptron Neural Network (MLPNN) Processes," *Ateneo de Davao University*, 2012.