IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429 Volume 6, Issue 1, August 2026 Impact Factor: 8.2

Modified ANFIS (MOD_ANFIS) Approach for Predicting Farmers'Income Levels

Shekh T. N[1], Dr Vishwajeet S. Goswami[2], Dr Vairal K. L[3]

Research Scholar, S.J.J.T University, Jhunjunu, Rajshtan[1]

Research Guide, S.J.J.T University, Jhunjunu, Rajshtan[2]

Research Co-guide, S.J.J.T University, Jhunjunu, Rajshtan[3]

Abstract: *The agricultural sector plays an important role in the national economy. Farmers' income is uncertain and depends on crop yields and agricultural prices. This Final Project develops a Neuro-Fuzzy system using the Modified Adaptive Neuro-Fuzzy Inference System (Mod_ANFIS), a modification of standard ANFIS, to predict farmers' income levels.*

The system uses agricultural data for rice and secondary crops, with three harvest periods per year and calculations expressed per hectare. Training data consist of farmers' income data obtained from the Central Bureau of Statistics (Badan Pusat Statistik/BPS). The Mod_ANFIS learning process consists of a forward stage and a backward stage using a modified error back-propagation correction rule. The forward stage uses fuzzy inference and the Least-Squares Estimator (LSE), while the backward stage updates premise parameters.

The resulting system is used for learning, validation, and testing. The thesis reports prediction errors using RMSE and MAPE and compares the Mod_ANFIS results with forecasts produced by the Department of Agriculture (Deptan).

Keywords: *Modified Adaptive Neuro-Fuzzy Inference System*.

I. INTRODUCTION

The agricultural sector plays an important role in the development of the national economy. Its importance to the Indonesian economy can be seen from food provision, contribution to Gross Domestic Product (GDP), employment, and contribution to foreign-exchange earnings through exports. In addition to providing food for society, agriculture supplies raw materials for industry and employment for tens of millions of people. The agricultural sector, particularly agro-industry or agriculture-based industry, has made a significant contribution to national economic growth. Therefore, placing agriculture as a priority in government development policy is an appropriate strategy for improving public welfare.

Uncertain farmers' income, which depends on harvest results, has an impact on changes in the prices of agricultural products. Farmers face many challenges in farming, including land, agricultural inputs, and the sale of harvested products. If farmers' income levels can be predicted early, farmers can plan various budgets to obtain better harvest results.

Based on the discussion above, a system is required that can predict farmers' income levels. The developed system is based on fuzzy logic theory, in which rules are created from membership functions. The reliability of a fuzzy system is determined by membership functions that represent the real system. Therefore, the membership-function parameters can be selected with the help of an Artificial Neural Network (ANN) using learning rules based on training data. A fuzzy system whose membership-function parameters are determined using an artificial neural-network architecture is known as a Neuro-Fuzzy system.

This Final Project develops a Neuro-Fuzzy system with a Modified Adaptive Neuro-Fuzzy Inference System (Mod_ANFIS) structure, which is a modification of standard ANFIS, for predicting farmers' income levels

DOI: 10.48175/IJARSCT-38077

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

II. FUZZY THEORY

2.1. Fuzzy Set Theory

Fuzzy set theory was introduced by Lotfi A. Zadeh (1965). It is based on fuzzy logic, in which logic values between 0 and 1 express degrees of truth. Zadeh argued that the true/false logic of Boolean logic cannot handle the gradations found in the real world. To address infinitely fine gradations, Zadeh developed fuzzy sets.

Unlike Boolean logic, fuzzy logic has continuous values. Fuzzy values are expressed as degrees of membership and degrees of truth. Consequently, something can be partly true and partly false at the same time.

Let V be a collection of objects, generally represented by {v}, which may have discrete or continuous values. V is called the universe of discourse, and v represents elements of V. A fuzzy set A in the universe V can be represented by a membership function μA that assigns a value in [0,1] to every v in V:

$\mu A : V \rightarrow [0, 1]$ (2.1)

All elements v in V for which $\mu A > 0$ form the support of the corresponding fuzzy set. If $\mu A = 0.5$, v is called a crossover point, and a fuzzy set whose support has value 1.0 is called a fuzzy singleton.

The fuzzy-logic approach is broadly implemented in three stages:

1. Fuzzification: mapping crisp inputs to fuzzy sets.
2. Inference: generating fuzzy rules.
3. Defuzzification: transforming a fuzzy output into a crisp value.

2.2 Membership Functions

- Triangular membership function - has parameters a, b, and c.
- Trapezoidal membership function - has parameters a, b, c, and d.
- Gaussian membership function - has parameters a and σ.
- Bell membership function - has parameters a, b, and c. With b positive it has the usual bell shape; with b negative it becomes an inverted bell membership function.

The original document gives the mathematical formulations as Equations (2.2)–(2.5). The formulas are retained conceptually here; the source PDF's formula extraction is fragmented in the text layer.

2.3 Rule Base

The rule base is the core of a fuzzy system because it contains a set of rules. Fuzzy If–Then rules or fuzzy conditional statements have the form IF A THEN B, where A and B are labels of fuzzy sets according to L. A. Zadeh (1965), represented through membership functions. In compact form, fuzzy If–Then rules are commonly used to capture imprecise modes of reasoning that play an important role in human decision-making under uncertain and imprecise conditions.

III. NEURO-FUZZY SYSTEM WITH A MODIFIED LEARNING ALGORITHM (MODIFIED ANFIS)

The agricultural sector is a vital component of the national economy; however, farmers' income remains uncertain due to fluctuations in crop productivity and agricultural commodity prices. This study proposes a Modified Adaptive Neuro Fuzzy Inference System (Mod-ANFIS), an enhanced version of the conventional ANFIS model, for predicting farmers' income levels. The proposed system is developed using agricultural data related to rice and secondary crops, considering three harvesting periods annually and expressing the calculations on a per-hectare basis.

An Adaptive Neuro-Fuzzy Inference System (ANFIS) integrates the learning capabilities of artificial neural networks with the reasoning and interpretability of fuzzy logic. In ANFIS, neural network learning algorithms are employed to automatically identify and adjust the parameters of the fuzzy inference system, such as the parameters of membership functions. This reduces the need for manual parameter tuning and can enhance the overall accuracy and efficiency of the system. The combination of neural learning and fuzzy rule-based reasoning enables ANFIS to effectively utilize previous observations and improve decision-making or classification performance. Unlike conventional neural

ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

networks, where the model structure is primarily developed through the training process, a neuro-fuzzy system is initially formulated using fuzzy rules and membership functions and is subsequently optimized through neural network-based learning algorithms.

The training dataset comprises farmers' income information obtained from the Lasalgaon Market yardThe Mod-ANFIS learning procedure involves two primary stages: a forward-learning phase and a backward-learning phase. During the forward phase, fuzzy inference is combined with the Least-Squares Estimator (LSE) to determine the consequent parameters. In the backward phase, the premise parameters are adjusted using a modified error back-propagation algorithm. The developed model is subsequently evaluated through learning, validation, and testing processes.

The performance of the proposed Mod_ANFIS model is assessed using Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) as prediction accuracy measures. Furthermore, the prediction performance of Mod_ANFIS is compared with income forecasts provided by the Lasalgaon Market yard to evaluate its effectiveness and applicability in estimating farmers' income.

The major advantages of ANFIS can be summarized as follows:

Automatic refinement of fuzzy rules: ANFIS can optimize fuzzy if - then rules to effectively represent the behavior of complex and nonlinear systems.

Reduced dependence on expert knowledge**:** It can learn system characteristics directly from available data, thereby minimizing the need for extensive prior human expertise.

Effective use of membership functions**:** ANFIS can employ suitable membership functions to model and approximate relationships within the given dataset.

Flexibility in membership function selection**:** It provides several options for choosing membership functions according to the characteristics of the problem and dataset.

Rapid convergence**:** The learning mechanism of ANFIS generally allows the model parameters to converge quickly during the training process.

3.1 Neuro-Fuzzy System with a Modified Learning Algorithm (Modified ANFIS)

The Neuro-Fuzzy system based on the Modified Adaptive Neuro-Fuzzy Inference System (Mod_ANFIS) offers an advantage over the conventional ANFIS approach by incorporating a network-minimization mechanism that can reduce computational complexity and improve processing efficiency. In the conventional ANFIS framework, the learning procedure is performed through two main stages, namely the forward stage and the backward stage (Jang, 1997).

Both standard ANFIS and Mod_ANFIS generally follow a similar learning architecture, consisting of these two learning stages and a network comprising five computational layers. However, the principal distinction between the two approaches lies in the error-correction mechanism employed during the learning process. Mod_ANFIS incorporates a modified Error Back-Propagation (EBP) rule to adjust the network parameters, with the aim of improving learning efficiency and prediction performance.

3.2 Mod-ANFIS Learning Rule

The main modification in Mod-ANFIS is the error-correction rule of EBP. The learning algorithm has two stages: Forward Stage and Backward Stage.

A. Forward Stage

The forward stage uses a fuzzy-inference mechanism with ANFIS structure and the Least-Squares Estimator (LSE).

The Mod-ANFIS architecture is illustrated with two inputs and one output.

DOI: 10.48175/IJARSCT-38077

IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429 Volume 6, Issue 1, August 2026 Impact Factor: 8.2

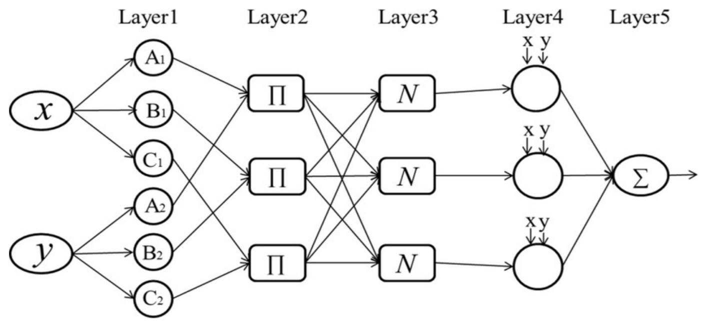

Figure 1 Mod-ANFIS

Layer 1: Defines membership-function parameters (a1…a4, b1…b4, c1…c4) and implements the membership function. In this Final Project the bell function is selected. The output nodes are denoted n1a through n4a. The 'a' distinguishes forward-stage outputs from the corrected 'b' values. This layer is adaptive because its parameters change; these parameters are called premise parameters.

Layer 2: If fuzzy AND logic is applied as the node function, the output nodes are the minimum of the incoming membership values:

n5a = min(n1a, n3a)

n6a = min(n2a,n4a).

Layer 3: Normalizes the incoming signals.

If ntot_a = n5a + n6a,

then n7a = n5a/ntot_a and n8a = n6a/ntot_a.

Layer 4: This layer is adaptive because parameters change.

The incoming signals form matrix A.

A = [(n7a x) (n7a y) n7a (n8a x) (n8a y) n8a]

Using the Least-Squares Estimator, the consequent parameters (p1, q1, r1, p2, q2, r2) are obtained.

$\theta = [\ A^T A\ ]^{-1}\ A^T\ U$

The functions are f1 = p1x + q1y + r1 and f2 = p2x + q2y + r2.

The node outputs are n9a = n7a f1 and n10a = n8a f2.

Layer 5: The final layer is the network output. It sums all incoming signals:

n11a = n9a + n10a.

B. Backward Stage

After the forward stage is completed and the outputs of all signals are obtained, the network output error is propagated backward using the new error-correction rule based on modified EBP. The source derives the squared-error expression and defines the correction values d11, d9, d10, d7, and d8, followed by updating the normalized total and the first-layer membership-function parameters.

The source states that the Bell membership-function derivative is applied to parameters a, b, and c. The input x itself is not used as the update reference because the changes are applied to the premise parameters in Layer 1. The updated parameters are then used in the next Mod_ANFIS process until a minimum error is obtained.

3.3 Artificial Neural Network Architecture

In an Artificial Neural Network (ANN), processing elements known as neurons are organized into distinct layers, with neurons within the same layer typically having similar connection patterns. Based on their architecture, neural networks can generally be categorized as single-layerand multilayer networks (Setiawan &Kuswara, 2003).

In a single-layer neural network, the neurons are primarily divided into input and output units. The input units receive information from the external environment, while the output units generate responses based on the received inputs. In

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, August 2026 Impact Factor: 8.2

contrast, a multilayer neural networkconsists of input and output layers along with one or more intermediate hidden layers. The number of hidden neurons and layers can be selected according to the complexity of the problem being addressed.

For relatively simple problems, a single-layer architecture may be sufficient. However, problems involving complex relationships and nonlinear patterns generally require a multilayer neural network, where additional hidden neurons and layers enable the network to learn more sophisticated input-output relationships.

3.4 Forecasting Techniques

Forecasting methods are widely used in future-oriented business and economic activities, including finance, human resource management, production, and marketing. Since future conditions cannot be determined with complete certainty, forecasting techniques are employed to reduce uncertainty and support more informed decision-making (Aritonang R. Lerbin R., 2002).

Data constitute a fundamental component of the forecasting process, and the underlying data pattern can significantly influence the accuracy of the resulting predictions. According to the time dimension, data can generally be classified into time-series data and cross-sectional data. Time-series data are observations recorded sequentially over a specific period, such as monthly agricultural or farmer-price data collected over one year. In contrast, cross-sectional data are observations collected at a particular point in time across different individuals, locations, or entities and do not primarily involve a temporal sequence.

Forecasting approaches are commonly categorized into qualitative and quantitative methods. Qualitative forecasting depends largely on the knowledge, experience, intuition, and judgment of experts or forecasters rather than on extensive historical numerical data. Quantitative forecasting, on the other hand, uses historical data and mathematical or statistical techniques to generate predictions, thereby minimizing the influence of subjective judgment.

Quantitative forecasting is appropriate when three basic conditions are satisfied:

(1) relevant information about past events or observations is available;

(2) the available information can be expressed numerically and analyzed quantitatively; and

(3) it is reasonable to assume that certain patterns or relationships observed in the past will continue to some extent in the future. These conditions make quantitative forecasting particularly useful for analyzing historical agricultural data and estimating future trends.

3.5 Data Patterns

Horizontal pattern: Occurs when data values fluctuate around a constant mean; the series is stationary with respect to its mean.

Seasonal pattern: Occurs when a series is influenced by seasonal factors, such as particular quarters, months, or days of the week.

Cyclical pattern: Occurs when data are influenced by long-term economic fluctuations associated with business cycles.

Trend pattern: Occurs when there are long-term secular increases or decreases in data.

3.6 Measures of Forecasting Accuracy

In time-series modeling, part of the known data can be used to forecast subsequent data. Forecasting error is the difference between forecast data and actual data. Forecast error is important for evaluating the accuracy of a forecasting method (Makridakis, 1993).

1. Root Mean Squared Error (RMSE): the square root of the mean of squared differences between actual and forecast values.

2. Mean Absolute Percentage Error (MAPE): the average absolute error in each period divided by the actual value for that period, expressed as a percentage.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

Notation: Xt = actual value at period t; Ft = forecast value at period t; Xt − Ft = forecasting error; n = number of observations.

4 Prediction of Farmers' Income Level

According to agricultural economic theory, several factors determine farmers' income. First, farmers' welfare can be improved through higher prices of agricultural products. Higher agricultural-product prices directly affect and increase farmers' income.

Second, an increase in the supply of agricultural products will lower prices and total farmer revenue. Third, from a macroeconomic perspective, higher agricultural prices can reduce consumers' purchasing power and trigger inflation, which can eventually reduce farmers' real income.

Farmers' income depends on the productivity of the agricultural products they produce. The prices received by farmers also change over time. Expenditures must be carefully considered so that optimal net profit can be obtained. Even when the production value of a harvest is high, very large expenditures can prevent farmers from achieving maximum net profit.

IV. SYSTEM DESIGN

4.1 Problem Analysis

Before the Green Revolution, farmers practiced an agricultural system close to natural systems, including recycling crop residues without inorganic inputs. At that time there were no chemical efforts to control pests, diseases, or weeds, and capital was limited, so productivity was relatively low. This affected farmers' income, which can be described as low.

Because farmers' income is important, a system is required that can predict farmers' income levels and crop productivity using data from previous years. The network output and its error are compared with forecasts produced by the Department of Agriculture (Deptan). Using a forecasting-accuracy measure, the network output and forecast results can be compared. Input and output data are connected by a function generated by the Neuro-Fuzzy system. Matching inputs to outputs produces a function that maps inputs to outputs. If the function's error is small, the obtained function is considered optimal.

4.2 Development Model

This Final Project develops a system that can predict farmers' income levels using a Neuro-Fuzzy system with the Mod_ANFIS structure.

The system-development model is divided into two flows:

1. Learning Process Flow - used to obtain new, updated premise parameters. It contains the Forward Process sub-flow consisting of Layers 1 through 5.
2. Testing Process Flow - contains only the forward process (Layers 1 through 5), using the premise parameters obtained from learning.

Figure 3.2 describes the forward-stage sub-process: Start → Layer 1 (Fuzzification) → Layer 2 (AND logic) → Layer 3 (Normalization) → Layer 4 (Defuzzification using LSE) → Layer 5 (Network output) → Finish.

Layer 1 converts crisp/actual input values into fuzzy values using the bell membership function. Layer 2 applies fuzzy AND logic by selecting the smallest incoming node. Layer 3 performs normalization. Layer 4 performs defuzzification using LSE. Layer 5 produces the network output.

4.3 Architecture Design

The designed Mod_ANFIS Neuro-Fuzzy system is adjusted to the training-data format: a network with two inputs and one output.

Layer descriptions: 1) Fuzzification; 2) AND logic; 3) Normalization; 4) Defuzzification using LSE; 5) Network output. The backward architecture applies the modified error-correction process.

4.4 System Design

Before creating the application program, system design is performed so that the application functions as expected and can produce accurate predictions of farmers' income. The design stages include System Flow, Data Flow Diagram (DFD), Entity Relationship Diagram (ERD), and Database Structure.

4.5 System Flow

System Flow shows the overall flow of the application program. It helps in application development because it shows the program and user flow and also identifies the database required by the application. Figure 3.6 presents the system flow for predicting farmers' income levels.

4.6 Data Flow Diagram

The DFD for the application of Neuro-Fuzzy Mod_ANFIS for predicting farmers' income consists of a Context Diagram, DFD Level 1, and DFD Level 2 for data maintenance.

Context Diagram entities/processes include User and BPS. The system receives training data, learning data, validation data, testing data, productivity data, production data, farmer-price data, and maintenance updates. The system returns learning results, prediction results, updated price and income data, and validation results.

In DFD Level 1 there are four major processes: Data Maintenance, Neuro-Fuzzy Mod_ANFIS, Validation Process, and Testing Process. The user enters maintenance data, selects training data for Neuro-Fuzzy learning, performs validation using training data and learned results, and performs testing using selected test data.

DFD Level 2 for Data Maintenance contains processes for adding data and updating data.

4.7 Entity Relationship Diagram

An Entity Relationship Diagram (ERD) is a system design used to represent, determine, and document system requirements for database processing. The ERD contains a Conceptual Data Model (CDM) and Physical Data Model (PDM).

The main entities shown in the CDM/PDM are Farmer Price Data, Agricultural Data, Learning Details, and Testing Data. Fields include crop code, crop name, month, year, price, productivity, production value, expenses, seed, pesticide, fertilizer, equipment rental, irrigation costs, labor wages, other expenses, profit, premise parameters, consequent parameters, system output, data output, and difference/error.

4.9 Input/Output Design

Several forms are required to run the application.

4.10 Data Maintenance Form

The Data Form is used to input and maintain production data. Users select a category such as Farmer Price Data or Agricultural Production Data, select a crop type such as rice or secondary crops, display the data, add or save records, edit records, and cancel operations.

Figure 3.12 contains controls such as Add, Edit, Cancel, Exit, Category, Type, Date, Grid, Code, Name, Value, Select, and Display.

4.11 Learning Process Form

The Learning Process Form performs learning and produces new premise parameters through Neuro-Fuzzy Mod_ANFIS, selecting the minimum error from a number of iterations. The user selects a crop type, enters the desired number of iterations, and can enter premise parameters manually or obtain system defaults using the Default Premise button. The Process button starts learning, and the Testing button proceeds to testing.

4.12 Testing and Validation Form

This form performs testing or validation using learned premise parameters. The user selects Validation or Testing, selects the type of test, presses Process, and can generate a report.

4.13 Detail Form

The Detail Form calculates production value, expenses, and farmer profit. The user selects the crop type for each harvest season, can reset calculations using Delete, and presses Process to calculate.

4.14 Report Form

The Report Form previews prediction-result data and prints it. The user selects a category such as Agricultural Production Data or Farmer Price Data, uses Preview to view the report, and Print to print it.

4.15 Data Design

Training-data pairs are constructed from secondary data obtained from BPS using the format [$x(t-24)$; …; $x(t-1)$; $x(t)$; $x(t+1)$; …; $x(t+12)$]. Here $x(t-24)$ is Input 1, $x(t)$ is Input 2, and $x(t+12)$ is the output or actual data.
Production data used for learning are four-month harvest-period data. Data for 1994–1997 are used as Input 1, data for 1998–2001 as Input 2, and subsequent data through 2005 as Output/Input 3. The updated premise parameters produced by learning are used for validation and testing.

V. IMPLEMENTATION AND EVALUATION

5.1 Program Installation

Before implementing and running the application, the main computer components supporting each process must be installed.

5.2 Hardware Requirements

- Pentium II 333 MHz CPU or higher
- Minimum 32 MB memory
- Minimum 4.3 GB hard disk
- 4 MB VGA card
- SVGA monitor with 800 × 600 resolution
- Keyboard, mouse, and printer

5.3 Software Requirements

- Microsoft Windows XP
- Power Designer
- Microsoft SQL Server 7
- Borland Delphi 6.0

5.4 System Implementation

After all components supporting the application are installed, the system design is implemented as the Farmers' Income Level Prediction program. The important modules are described below.

5.5 Learning Process

A. Input Training Data
Training data are placed in array variables to simplify data access.

```
type
  in1, in2, out1 : Array[1..12] of real;
  Input1, Input2, Output : real;
maks_iterasi : integer;
Tgl : TDateTime;
end;
var
  DL : array of DataLatih;

while not adoquery3.Eof do
begin
  in1[i] := adoquery1.FieldByName('harga').AsFloat;
  in2[i] := adoquery2.FieldByName('harga').AsFloat;
  out1[i] := adoquery3.FieldByName('harga').AsFloat;
inc(i);
  Next;
end;
```

The system takes learning data consisting of input1, input2, and output and stores them in the array variables in1, in2, and out1. The number of iterations is entered by the user.

B. Forward Stage

1. Layer 1 — Fuzzification. Crisp/actual input values are converted into membership degrees. Bell membership parameters a1…a4, b1…b4, and c1…c4 are used. The source implementation includes:

```
Bantu := abs((in1[i]-(c1))/(a1));
Bantu := exp(ln(Bantu)*2*(b1))+1;
```

The membership parameters can be entered manually or obtained from the system using Mathlab (as written in the original thesis). Forward-stage node outputs are denoted with 'a'.

2. Layer 2 — AND Logic. The fuzzy AND operation selects the smallest incoming value:

```
n5a := Min(n1a,n3a);
n6a := Min(n2a,n4a);
```

3. Layer 3 — Normalization:

```
ntot_a := n5a+n6a;
if ntot_a=0 then
begin
  n7a := 0;
  n8a := 0;
end
else
begin
  n7a := n5a/ntot_a;
  n8a := n6a/ntot_a;
end;
```

4. Layer 4 — Defuzzification. Using LSE, consequent parameters p1, q1, r1, p2, q2, and r2 are obtained. The matrix calculations and inverse operation in the source implement $\theta = (A^TA)^{-1}A^TU$.

```
A := [(n7a*in1) (n7a*in2) n7a
      (n8a*in1) (n8a*in2) n8a];
```


DOI: 10.48175/IJARSCT-38077

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, August 2026 Impact Factor: 8.2

```
A[i,1] := n7a*in1; A[i,2] := n7a*in2; A[i,3] := n7a;
A[i,4] := n8a*in1; A[i,5] := n8a*in2; A[i,6] := n8a;
f1[i] := ((p1*in1[i])+(q1*in2[i])+r1);
f2[i] := ((p2*in1[i])+(q2*in2[i])+r2);
n9a := n7a*f1;
n10a := n8a*f2;
```

5. Layer 5 — Network Output:

```
N11a = n9a+n10a;
eror := output-n11a;
```

The result n11a is the network output and the error is calculated at this stage.

C. Backward Stage

The backward process applies the modified error-correction algorithm from Layer 5 to Layer 1. The source implementation is:

```
e11 := -2*(Output-n11a);
d11 := -e11/2;
ftot := f1+f2;
d9 := d11*f1/ftot;
d10 := d11*f2/ftot;
d7 := d9/f1;
d8 := d10/f2;
ntot_b := ntot_a;
n5b1 := (n7a+d7)*ntot_b;
n6b1 := (n8a+d8)*ntot_b;
n1b := n5b1; n3b := n5b1;
n2b := n6b1; n4b := n6b1;
```

The membership-function parameters in Layer 1 are then updated. The source calculates derivatives da1 through dc4. The forward and backward stages continue until minimum error is obtained or the user-specified iteration count is reached.

5.6 Validation Process

Validation tests the system using parameters updated through learning and training-data pairs. Only the forward stage of the Neuro-Fuzzy algorithm is used; no backward error correction is performed. The source uses productivity as the input/output data and retrieves premise and consequent parameters from the learning-detail table.

```
in1[i] := adoquery1.FieldByName('produktivitas').AsFloat;
in2[i] := adoquery2.FieldByName('produktivitas').AsFloat;
out1[i] := adoquery3.FieldByName('produktivitas').AsFloat;
a1[i] := adoquerydetilbelajar.fieldbyname('premis_a1').AsFloat;
b1[i] := adoquerydetilbelajar.fieldbyname('premis_b1').AsFloat;
c1[i] := adoquerydetilbelajar.fieldbyname('premis_c1').AsFloat;
```

The validation process is the same as learning except that only the forward stage is performed; premise parameters and LSE-related values are taken from the learning results.

5.7 Testing Process

After learning, the system is tested to predict farmers' income. Testing uses input data together with the updated membership-function parameters and only the forward stage. The network output is the prediction for 2005. The

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

predicted result is compared with prepared 2005 data. The testing form follows the validation process but uses different data and performs LSE as in the initial learning stage.

5.8 System Testing

Testing is performed to determine whether the developed system conforms to the planned design and represents the system as a whole.

5.9 Main Menu

The Main Menu Form provides access to the forms in the Farmers' Income Level Prediction application.

5.10 Data Form

The Data Form is used to maintain agricultural production data, including entering production data and updating existing data.

5.11 Learning Process

The Learning Process Form performs Neuro-Fuzzy Mod_ANFIS learning using training-data pairs and produces updated premise parameters with minimum error among the performed iterations.

Crop	Year	RMSE	MAPE (%)
Rice	2018	141.392	0.0237
Corn	2018	17.5053	0.0034
Soybean	2018	96.9795	0.0079
Cassava	2018	5.0111	0.0043
Sweet potato	2018	9.0169	8.0505
Peanut	2018	116.991	0.0045

Table 4.1. Results of the learning process.

5.12 Validation Form

The Validation Form evaluates the results of the learning process. The validation results reported in Table 4.2 are the same as those in Table 4.1, indicating that the process performed by the system is valid.

Crop	Year	RMSE	MAPE (%)
Rice	2018	141.392	0.0237
Corn	2018	17.5053	0.0034
Soybean	2018	96.9795	0.0079
Cassava	2018	5.0111	0.0043
Sweet potato	2018	9.0169	8.0505
Peanut	2018	116.991	0.0045

5.13 Report Form

The user can create a prediction report by pressing the Report button in the Testing and Validation Form. Reports can be selected by a start and end date. Preview displays the report on screen and Print sends it to the printer.

5.14Detail Form

The Detail Form calculates production value, expenditure, and net profit.

DOI: 10.48175/IJARSCT-38077

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, August 2026
Impact Factor: 8.2

5.15 Testing Form

The Testing Form performs testing using the updated premise parameters. It uses variables and inputs produced by the learning process and premise parameters obtained during learning.

Crop	Year	RMSE	MAPE (%)
Rice	2018	180.937	4.1501
Corn	2018	112.356	0.0015
Soybean	2018	237.994	0.0049
Cassava	2018	11.3308	0.0012
Sweet potato	2018	50.246	0.0987
Peanut	2018	30.6249	0.0108

Table 4.3. Prediction errors from system testing.

5.16 Prediction Results

The reported prediction results for 2005 are: Rice RMSE 141.392 and MAPE 0.0237%; Corn RMSE 17.5053 and MAPE 0.034%; Soybean RMSE 96.9795 and MAPE 0.0079%; Cassava RMSE 5.0111 and MAPE 0.0043%; Sweet potato RMSE 9.0169 and MAPE 8.0505%; and Peanut RMSE 116.030 and MAPE 0.0045%.

Note: the source thesis reports 116.991 for Peanut in Table 4.1/4.2 but 116.030 in the narrative on page 59. Both values are retained here rather than silently correcting the source.

5.17 System Evaluation

The system is evaluated by comparing its test results with another method and measuring prediction errors using forecasting-accuracy measures.

5.18 Comparison of Prediction Results

To demonstrate the prediction performance of Neuro-Fuzzy Mod_ANFIS, its results are compared with forecasts produced by the Department of Agriculture (Deptan).

Crop	Method	MAPE 2005 (%)
Rice	A — Neuro-Fuzzy Mod_ANFIS	0.0237
Rice	B — Deptan forecast	0.0094
Corn	A — Neuro-Fuzzy Mod_ANFIS	0.034
Corn	B — Deptan forecast	8.846
Soybean	A — Neuro-Fuzzy Mod_ANFIS	0.0079
Soybean	B — Deptan forecast	0.916
Cassava	A — Neuro-Fuzzy Mod_ANFIS	0.0043
Cassava	B — Deptan forecast	0.467
Sweet potato	A — Neuro-Fuzzy Mod_ANFIS	8.0505
Sweet potato	B — Deptan forecast	0.925
Peanut	A — Neuro-Fuzzy Mod_ANFIS	0.0045
Peanut	B — Deptan forecast	9.805

Method A = Neuro-Fuzzy with Mod_ANFIS structure. Method B = forecasting by Deptan. MAPE is expressed as a percentage.

According to the thesis, Table 4.4 shows that the system's prediction results have a smaller total error than the predictions by Deptan, and therefore the Neuro-Fuzzy Mod_ANFIS predictions are considered better on the basis of the reported error comparison.

VI. CONCLUSION

• The Neuro-Fuzzy system with the Mod_ANFIS structure can be used to predict farmers' income levels.
• The application can help produce more accurate decisions to support the calculation of farmers' income-level predictions.
• The reported average differences between Deptan forecasting and Neuro-Fuzzy MAPE are: Rice 0.0165%, Corn 4.445%, Soybean 0.461%, Cassava 0.235%, Sweet potato 4.487%, and Peanut 4.904%.

6.1 Suggestions

The system can be developed as an online or mobile application so that it can be accessed by anyone without time restrictions.

REFERENCES

[1] Jang, J.-S. R., Sun, C.-T., and Mizutani, E. (1996). Neuro-Fuzzy and Soft Computing. Prentice Hall.
[2] Kusumadewi, Sri. (2002). Analisis& Desain Sistem Fuzzy Menggunakan Toolbox Matlab. GrahaIlmu, Yogyakarta.
[3] Makridakis, Spyros, Wheelwright, Steven C., and McGee, Victor E. (1993). Metode dan AplikasiPeramalan, second edition, volume one. PT. Binarupa Aksara, Jakarta.
[4] Assauri, Sofjan. (1984). Teknik & Metode Peramalan. Faculty of Economics, Universitas Indonesia, Jakarta.
[5] Ayres, Frank. (1994). Teori dan Soal-soalMatriks. Erlangga, Jakarta.
[6] Jogiyanto. (1999). Analisis&DisainSistemInformasi: PendekatanTerstruktur Teori dan PraktekAplikasiBisnis. Andi, Yogyakarta.
[7] Pranata, Antony. (2000). Pemrograman Borland Delphi. Andi, Yogyakarta.
[8] Setiawan, Kuswara. (2003). ParadigmaSistem Cerdas. Banyumedia Publishing, Malang.
[9] Rahmat, Basuki. (2004). PenerapanSistem Neuro-Fuzzy untukPrediksi Curah Hujan DaerahBanyuwangi. Jurnal Teknik Komputer: GEMATEK, 6(2), 63–72.
[10] Martina, Inge. (2002). Database Client/Server Menggunakan Delphi. PT. Elex Media Komputindo, Jakarta.
[11] Romeo. (2003). Testing dan ImplementasiSistem. STIKOM, Surabaya.