RESEARCH ARTICLE

International Journal of Agriculture, Environment and Food Sciences 2024; 8(2): 327-346
DOI: https://doi.org/10.31015/jaefs.2024.2.9

Innovation in the dairy industry: forecasting cow cheese production with machine learning and deep learning models

Yunus Emre GÜR[1]

[1] Management Information Systems, Faculty of Economics and Administrative Sciences, Fırat University, Elazığ, Türkiye

Abstract

This study focuses on the use of deep learning and machine learning models to forecast cow cheese production in Turkey. In particular, our research utilizes the LSTM (long short-term memory) model to forecast cow cheese production for the next 12 months by extensively utilizing deep learning and machine learning techniques that have not been applied in this field before. In addition to LSTM, models such as GRU (Gated Recurrent Unit), MLP (Multi-Layer Perceptron), SVR (Support Vector Regression), and KNN (K-Nearest Neighbors) were also tested, and their performances were compared using RMSE (Root Mean Square Error), MSE (Mean Squared Error), MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error), and (Coefficient of Determination) metrics. The findings revealed that the LSTM model performed significantly better than the other models in terms of RMSE, MSE, MAE, and MAPE values. This result indicates that the LSTM model provides high accuracy and reliability in forecasting cow cheese production. This achievement of the model offers important applications in areas such as supply chain management, inventory optimization, and demand forecasting in the dairy industry.

Keywords: Deep Learning, Machine Learning, LSTM, Cow Cheese Production, Futurue Prediction

Type: Research Article
Subject: Zootechny

Citation: Gur, Y.E. (2024). Innovation in the dairy industry: forecasting cow cheese production with machine learning and deep learning models. International Journal of Agriculture, Environment and Food Sciences, 8(2), 327-346.
https://doi.org/10.31015/jaefs.2024.2.9

Submission Date: *November 16, 2023*
Acceptance Date: *May 25, 2024*
Early Pub Date: *June 11, 2024*
Publication Date: *June 29, 2024*

Corresponding Author: Yunus Emre GÜR
E-mail: yegur@firat.edu.tr

Available at:
https://dergipark.org.tr/jaefs/issue/84099/1392122

INTRODUCTION

With the evolution of world conditions, traditional or random eating habits have been replaced by a regular and balanced diet, especially in many developed societies. This means that the focus is on getting the required amounts of energy and nutrients needed by the human body every day. Milk and dairy products, which are vital for the body, are one of the foods that should be consumed at every stage of life, from infancy to old age. Milk and dairy products can be classified into various types. However, when milk and dairy products are mentioned, products such as cheese, yogurt, kefir, buttermilk, butter, milk powder, and cream come to mind (Kahraman, 2012: 49).

The long-term preservation of milk by turning it into cheese through fermentation is a method dating back thousands of years. Today, no matter how industrially advanced cheese production has become, the basic stages of cheesemaking are almost unchanged. However, the variety of cheeses is due to a combination of factors. For example, the type of milk used, the coagulation method, whether the milk is pasteurized or not, fat content, texture, salt content, additives, and ripening time are all important in creating this diversity. Depending on each country's culture and level of development, it offers a wide variety of cheeses. In the case of Turkey, feta cheese, cheddar cheese, and tulum cheese are widely consumed and are accompanied by local products such as curd, cottage cheese,

tongue cheese, Circassian cheese, and herbed cheese (Durlu-Özkaya and Gün, 2007: 488).

Livestock and animal products have become an economic industry and an integral part of economic life. This development emphasizes the importance of recognizing animal husbandry as a strategic sector at the national level. Products of animal origin play an important role in the human diet. They are unique sources of animal protein, containing eight essential amino acids that support the body's health, bone growth, and mental development. It is recommended for humans to have a daily intake of 1 gram of protein per kilogram of body weight, of which approximately 42% should be of animal origin. The protein ratio in animal products is approximately 3-4% in products such as milk (Yıldırım and Altunç, 2020: 138).

Turkey is a country with seas around it such as the Mediterranean, the Black Sea, and the Aegean Sea, where every season is distinctly experienced and where different cultures have met throughout history, with rich vegetation and a strategic location at the intersection of the continents of Asia, Europe, and Africa. These characteristics have undoubtedly provided Turkish cuisine with a wide range of products and influenced cheese diversity. Each region of Turkey is home to its own unique classical and traditional cheese production. This diversity has created a rich variety of cheeses in Turkish cuisine. From the easternmost part of Turkey to the westernmost part, from the north to the south, and even in settlements close to each other, there are similar but different types of cheese (Güllü, 2022: 48). Therefore, cheese is a common foodstuff in Turkey, and estimating production quantities plays a critical role in guiding consumption patterns and food security.

While cow's milk is generally used in cheese and yogurt production in Turkey, sheep, goat, and buffalo milk and their powders are also used in cheese and yogurt production as different milk types. In addition, products containing live microorganisms other than cheese and yogurt, especially fermented milk products such as kefir, are widely produced (Güllü, 2022: 48). Therefore, cheese production in Turkey is considered an important part of the livestock sector and contributes to the country's rural economy. At the same time, Turkey is an important exporter of cheese and dairy products. Therefore, accurately estimating production quantities can influence the foreign trade balance and export strategies. To summarize, cheese production forecasts are of great importance both economically and nutritionally. Especially in countries like Turkey, where agriculture and animal husbandry are the mainstays, these forecasts are seen as an important tool for strategic planning, economic performance, and food security.

Cow cheese production forecasting is a complex forecasting problem with multiple independent variables and a large number of data points. Therefore, deep learning and machine learning methods can be ideal tools to manage this complexity (Bulut, 2024). On the other hand, these predictions require accurate and precise results. Deep learning and machine learning models often have the potential to provide high precision and accuracy, which can improve the reliability of forecasts. Deep learning models are capable of learning complex relationships and can recognize unique patterns using large amounts of data (Şimşek, 2024). This can be advantageous for improving cheese production forecasts. In summary, benchmarking deep learning and machine learning methods is becoming a necessity to obtain more accurate, reliable, and precise predictions. Determining under which conditions these methods perform best can improve the quality of predictions and provide valuable insights. Comparing the performance of multiple models provides the opportunity to make the best prediction.

This study aims to forecast monthly cow cheese production from October 2023 to September 2024. For this purpose, deep learning and machine learning models such as GRU, LSTM, kNN, SVR, and MLP are trained using cow cheese production data from past periods and other independent variables. Then, the model with the best forecasting performance was selected, and future forecasts for cow cheese production were made. Such studies are expected to be of great importance and provide valuable results for dairy producers and other relevant stakeholders. In the rest of the paper, the literature focusing on the forecasting of agricultural and livestock products is reviewed. Then, the methodology of the study is explained, and a conclusion section is presented based on the findings.

Although there are many studies in the Turkish literature on the forecasting of milk, dairy products, livestock, and other agricultural products, there are no studies focusing on the forecasting of cheese production. This is the most important factor that distinguishes this study from other studies. Accordingly, what makes this study important in the literature is that deep learning and machine learning prediction models have not been applied in this sector before.

One of the studies conducted in Turkey is Yıldırım and Altunç (2020). In the study, the ARIMA (Box-Jenkins) model was used to forecast the future milk production of Muş province. The results of the study showed that milk production in Muş province will be approximately 336 thousand tons in 2020 and approximately 368 thousand tons in 2023.

In the study of Goyal and Goyal (2013), feed-forward multilayer artificial neural network (ANN) models were developed to predict the shelf life of processed cheese. These ANN models were used to predict how long processed cheese can maintain its freshness at a given temperature at which it is stored. The input variables used in the study represented the chemical and microbiological characteristics of processed cheese samples. In the study, different ANN model

combinations were used to predict the shelf life of processed cheese. Metrics such as mean square error (MSE), root mean square error (RMSE), coefficient of determination (), and Nash-Sutcliffe coefficient were used to compare the predictive capabilities of these models. The findings of the study showed that the feed-forward ANN model in the combination of 5-16-16-1 gave the best result in predicting the shelf life of processed cheese with a high value. This emphasizes that multilayer machine learning models can successfully predict the shelf life of processed cheese.

Liseune et al. (2021) proposed a model to estimate the time of birth of dairy cows. In this model, they measured the movement behaviors of the cows, such as lying down, standing up, walking, ruminating, and walking, by sensors attached to their necks and feet and recorded them in minutes. To predict the last 24, 12, 6, 3, and 1 hour of calving, they used machine learning methods such as logistic regression (LR) and random forest (RF) and deep learning methods combining long-short-term memory (LSTM), convolutional neural networks (CNN), and a specially developed CNN and LSTM model. As a result, calving was predicted to occur within 24 hours, and the CNN algorithm gave the best result.

Ma et al. (2021) reported in a study that deep learning models have been used for maize yield forecasting with successful results, but existing models do not quantify the uncertainty associated with the forecasts and often require a large training dataset. To address these limitations, this study develops a district-level maize yield forecasting model that incorporates a large number of publicly available data sources. By training the model for forecasts since 2001, the study showed that the developed Bayesian Neural Network (BNN) model achieved an average coefficient of determination () value of 0.77 for late season forecasts in the US Corn Belt during the test years from 2010 to 2019, outperforming five other leading machine learning models.

Li et al. (2021a) conducted research addressing the complexity of industrial cream cheese production. Cream cheese production involves a process that requires the complex scheduling of multiple batches of fermenters and various fluidic units. This study used an artificial neural network (Long-Short Term Memory Network, LSTM) to address this challenge. It is also combined with a mechanistic model that describes changes in biomass, lactose, and lactic acid concentrations. The LSTM network/mechanistic modeling approach showed a difference of 3 minutes for batch durations of 6 to 7 hours compared to the laboratory experiment, and the overall accuracy () was above 0.99.

Keskinbıçak (2023) conducted a study focusing on species classification and yield estimation of chickpea plants in order to increase productivity in the agricultural sector. This study was handled in two stages. In the first stage, the classification of chickpea species was carried out with machine learning methods using the characteristics of chickpea plants. In the second stage, yield predictions for these classified species were made by the regression method. In the classification process, machine learning methods such as decision trees (DT), support vector machines (SVM), and k nearest neighbors (kNN) were used. The accuracy rate was used as a measure of success. The results showed that the highest accuracy rate of 90.6% was achieved by SVM in the classification with raw data. Similarly, the classification success of the dataset with a combination of raw data and synthetic data was recorded as the highest by SVM with 100% accuracy. When only synthetic data was used, the highest success rate was achieved by kNN, with an accuracy rate of 95.4%.

Li and Liu (2023), in their study, aim to detect food fraud to ensure the quality and safety of milk. For this purpose, hyperspectral images of pure and adulterated milk samples were collected using a hyperspectral imaging system (400–1000 nm). Then, the best preprocessing and characteristic wavelength selection methods were selected using the calibration model SVR, and the best combination of data processing was used to process the spectral data. Finally, the LSTM model optimized by the whale optimization algorithm was used to predict the content of additives in milk. Experimental results show that the WOA-LSTM model can accurately predict the content of additives. This study has the potential to provide an effective solution against food fraud and represents an important research area to improve the safety of dairy products.

Gandotra et al. (2023), in a study, examined the performance of different machine learning models for wheat, rice, and maize yield forecasting in the Jammu region. These models are: Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Bi-directional Long Short-Term Memory (Bi-LSTM), classical Deep Neural Network (DNN), and the basic models: Support Vector Regression (SVR), Random Forest Regression, and Ensemble Method AdaBoost. The data used for the study included environmental data such as temperature, precipitation, humidity, solar radiation, and sunshine hours from 2009 to 2019. The results showed that the Long Short-Term Memory (LSTM) model outperformed the other models in this study. This means that it has low values for the error measures RMSE, MAE, and MAPE (0.30, 0.20, and 0.23, respectively).

MATERIALS AND METHODS

This study aims to forecast monthly cow cheese production from November 2023 to September 2024. For this purpose, deep learning and machine learning models such as GRU, LSTM, kNN, SVR, and MLP are trained using cow

cheese production data from previous periods and other independent variables. The dataset used in the study covers a total period of 165 months, from January 2010 to September 2023. In the dataset, monthly cow cheese production data is used as the dependent variable. The independent variables are monthly cheese exports, monthly cattle feed production, monthly average dairy feed price, monthly number of cultured dairy cows, and monthly cow milk production. A total of 966 data entries were provided. Data were obtained through the Central Bank's EVDS system. Table 1 displays a subset of the dataset that was used in this study.

Table 1. A Part of The Dataset Used In The Study

Period	Monthly Cheese Exports (Million Pieces)	Monthly Cattle Feed Production (Thousand Tons)	Monthly Dairy Feed Price (Ton/TL)	Monthly Number of Cultured Dairy Cows (Million)	Monthly Cow Milk Production (Thousand Tons)	Monthly Cow Cheese Production (Thousand Tons)
2010-1	6	294	440	1,9	494	31990
2010-2	8	295	440	1,91	492	32374
2010-3	7	296	460	1,92	594	35296
2010-4	8	297	460	1,93	610	36143
2010-5	7	298	460	1,94	654	38254
…	…	…	…	…	…	…
…	…	…	…	…	…	…
2023-3	18	452	6810	3,48	911	66526
2023-4	16	459	6810	3,49	879	61755
2023-5	17	454	7125	3,5	944	67235
2023-6	15	500	7365	3,6	866	64085
2023-7	15	582	7460	3,5	835	66557
2023-8	16	600	7575	3,6	823	63451
2023-9	16	607	7520	3,6	852	62567

Resource: Turkish Statistical Institute

Furthermore displayed in Figure 1 is the correlation matrix illustrating the relationship between every variable in the study's data set and other factors. Two variables have a positive link when there is a positive correlation between them, and a negative link when there is a negative correlation. There is a complete connection between the variables when the correlation value is 1 or -1.

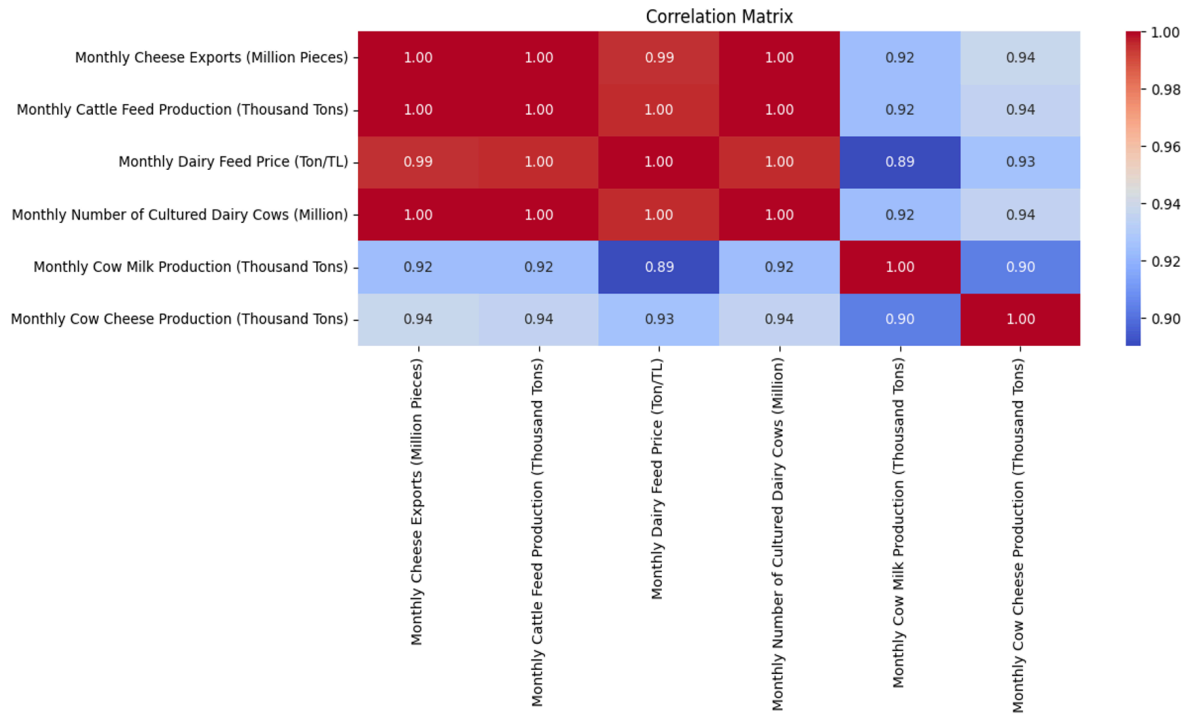

Figure 1. The Correlation Matrix.

As Figure 1 summarizes the results of the correlation matrix, there is a very high positive correlation (0.938506) between "Monthly Cheese Exports (Million Pieces)" and "Monthly Cow Cheese Production (Thousand Tons)". This shows that dairy exports and cow cheese production are highly positively correlated. There are also high positive correlations between "Monthly Cattle Feed Production (Thousand Tons)", "Monthly Dairy Feed Price (Ton/TL)", "Monthly Number of Cultured Dairy Cows (Million)", "Monthly Cow Milk Production (Thousand Tons)", and "Monthly Cow Cheese Production (Thousand Tons)". These results suggest that these variables are strongly correlated with cow cheese production, and these variables can play an important role in your forecasting model. Especially the high positive correlation between "Monthly Cheese Exports (Million Pieces)" and "Monthly Cow Cheese Production (Thousand Tons)" emphasizes the strong relationship between these two variables.

The study forecasts the monthly production of cow cheese using Python software. Importing the required Python libraries is therefore the first step. These libraries enable the creation of models, data processing, and result visualization. NumPy, Pandas, Matplotlib, Scikit-learn, and TensorFlow are the libraries that are used. Pandas was used to load the dataset from an Excel file. Additionally, NaN values were eliminated from the data set using "data.dropna()". Next, the data frame was divided into the independent variables (X) and dependent variables (Y). The min-max scaling method was used to normalize the data. This improves the performance of the model by converting each feature to a value between 0 and 1.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Training and test sets made up 80% and 20% of the overall data set, respectively, after data standardization. This allowed evaluation of the model's training efficacy with an alternative data set. A specific random seed (random_state) was utilized to partition the data set. "random_state" is used to make sure that the data set is randomly split in an identical manner each time. This guarantees reproducibility. In other words, the identical data split is produced each time by utilizing the same "random_state" value. When testing hyperparameter settings or assessing the model's performance, this preserves the comparability of the findings. Given that the goal of this research is to compare the performance of different models, it is imperative to ensure that the results are comparable by using the same data split. This will make determining which model performs better more equitable. As a result, 42 was chosen as the "random state" value and applied to every model.

All of the models that were employed in the study had their training loss tracked during the training phase, and the results were displayed in graph form. This made it feasible to see how the loss reduced and how each model learnt.

Next, predictions were generated for each trained model using the test data. Prior to Min-Max scaling, the forecasts were put back on their original scales. The study included five distinct techniques, including LSTM, GRU, MLP, SVM, and kNN models. Using estimated statistical measures such mean absolute percentage error (MAPE), mean squared error (MSE), coefficient of determination (R^2), and mean absolute error (MAE), the models' performance was assessed. These statistical data may be computed using equations 2, 3, 4, 5, and 6.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)}{N}} \quad (2)$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (3)$$

$$MAE = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} \quad (4)$$

$$MAPE = \frac{\sum_{t=1}^{n}\frac{u_t}{Y_t}}{n} * 100 \quad (5)$$

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \mu)^2} \quad (6)$$

The results were released following the calculation of the several error metrics previously created to measure the difference between the actual and projected numbers. A graph was then created using the projected and actual figures. This allowed for a clearer understanding of the models' functionality. This process includes training, predicting, assessing outcomes, preparing data, and building models. A complete list of the hyperparameters utilized in each of these phases may be seen in the Results section. The models' performance is impacted by these hyperparameters, which comprise configurations and optimization techniques. Furthermore, fundamental details on the models employed in the study are provided under the following subheadings.

However, in the next stage of the study, the model with the best training and testing forecast performance was selected to forecast monthly cow cheese production from October 2023 to September 2024.

Long Short Term Memory (LSTM)

Long short-term memory (LSTM) models are a subset of cyclic neural networks (RNNs) that are especially useful for learning chronic addictions. The cell state (C_t), one of the fundamental elements of the LSTM design, is regarded as the "memory" of the LSTM and retains information over extended periods of time (Li et al., 2021b). Each time step's output state is represented by the stored state (h_t), which is then passed on to the following step. What data should be added to the cell state is determined by the input gate (i_t). Typically, it is computed by multiplying a sigmoid (σ) by the tanh function. Which data from the cell state should be removed is decided by the forget gate (f_t). once more computed with the sigmoid function. Which data is moved to the stored state is decided by the output gate (o_t). The sigmoid function often governs this gate as well. The information chosen by the input gate and the data that the forget gate did not delete are contained in the intermediate state ($\hat{C}_t$), which is utilized to update the cell state (Smagulova and James, 2019).

$$f_t = \sigma(W_f \cdot [h_{(t-1)}, x_t] + b_f) \quad (7)$$

The same inputs are used to calculate both the intermediate state ($\tilde{C}_t$) and the input gate (i_t):

$$i_t = \sigma(W_i \cdot [h_{(t-1)}, x_t] + b_i) \quad (8)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{(t-1)}, x_t] + b_C \quad (9)$$

After that, the cell status is updated:

$$C_t = f_t * C_{(t-1)} + i_t * \tilde{C}_t \quad (10)$$

Ultimately, the new hidden state (h_t) is formed and the output gate (o_t) is calculated:

$$o_t = \sigma(W_o \cdot [h_{(t-1)}, x_t] + b_o) \quad (11)$$

$$h_t = o_t * \tanh(C_t) \quad (12)$$

In order to provide a mathematical explanation of how an LSTM cell works, let's start by calculating the forget gate (f_t) using the input that is being used (x_t) and the stored state that was previously used ($h_{(t-1)}$).

Gated Recurrent Unit (GRU)

With a more straightforward architecture, GRU is a kind of artificial neural network that is intended to handle sequential input and shares traits with the LSTM (Long Short-Term Memory) model (Athiwaratkun and Stokes, 2017). GRU gets two input vectors (X_t) and the stored state of the previous time step ($H_{(t-1)}$) at each time step. How much of the stored state keeps past data is decided by the update gate. The update gate is computed using the formula and is represented by the symbol Z_t :

$$Z_t = \sigma(W_z \cdot [H_{(t-1)}, X_t]) \quad (13)$$

The sigmoid activation function is denoted by in this case, while the update gate weights are represented by W_z.

How much of the previous data is moved to the new storage state is decided by the reset gate. It is computed as follows and is represented by the symbol R_t (Nosouhian et al., 2017):

$$R_t = \sigma(W_r \cdot [H_{(t-1)}, X_t]) \quad (14)$$

W_r represents the reset gate's weights in this instance.

On the other hand, the input vector and the previous hidden state are combined to compute the new hidden state, $\hat{H}_t$ (Agarap, 2018). The calculation of an intermediate vector, $\hat{H}_t$, is done first as follows:

$$\tilde{H}_t = \tanh(W \cdot [R_t * H_{(t-1)}, X_t]) \quad (15)$$

Tanh stands for the hyperbolic tangent activation function in this case, while W stands for the hidden state weights. Next, the following formula is used to determine the new concealed state:

$$H_t = Z_t * H_{(t-1)} + (1 - Z_t) * \tilde{H}_t \quad (16)$$

Subsequently, the updated stored state is transmitted to the subsequent layer or output at every time step.

Multilayer Perceptron (MLP)

Machine learning and artificial neural networks are built on a sort of artificial neural network called Multi-Layer Perceptrons (MLP). The following is a summary of the fundamental elements that comprise the MLP architecture: The model first accepts information and begins processing it at the input layer. Every node, also known as a neuron, represents a feature in the dataset. The layers that sit between the input and output layers are the hidden layer(s). Weights and activation functions are used in these levels to process the incoming data. Multiple hidden layers are possible in MLP models (Ramchoun et al., 2017). The output layer represents the final layer at which the network generates output. For a classification job, each neuron in the output layer represents a class, and for a regression task, it represents a value. The weights stand for each connection's strength. These weights are updated to allow the network to learn. Each neuron's activity threshold is determined by biases. Biases enable neurons to become more or less responsive to a particular stimulus by shifting the activation function. Lastly, the output of neurons is computed using activation functions. The network may learn several activation functions, including sigmoid, tanh, and ReLU, to overcome non-linear complexity (Desai and Shah, 2021).

The actions at each layer of the network are covered by the mathematical formulation of the MLP, which typically comprises of the successive application of an activation function and a linear transformation (applying weights and biases). These steps may be described in depth mathematically as follows: In a linear transformation, biases () are applied to each layer and inputs (x) are multiplied by weights (W):

$$z^l = W^l x^{l-1} + b^l \quad (17)$$

where W^l is the weight matrix in the 1th layer, $x^{(l-1)}$ denotes the outputs of the (l-1)th layer (or the original inputs for the input layer), b^l denotes the bias vector in the lth layer, and z^l denotes the linear transformation result of the neurons in the lth layer.

The activation function is then applied to the outcome of the linear transformation:

$$a^l = f(z^l) \quad (18)$$

where f stands for the selected activation function and σ^l is the activation result of the neurons in the 1th layer. An activation function that may be used is a sigmoid (σ), tanh, or ReLU (rectified linear unit). After applying the final linear transformation and activation function, the following is the final output generated in the last layer of the network:

$$z^L = W^L a^{L-1} + b^L \quad (19)$$
$$\tilde{y} = f(z^L) \quad (20)$$

where ỹ is the network's anticipated output and L is the total number of layers.

From the input layer to the final output layer, these mathematical procedures are repeated at every layer of the network. The gradient descent approach is used to update the weights and biases at each iteration, while the back propagation technique is used to determine the network's error rate. In order for the network to have "learned" to do the assigned task, this procedure is repeated in order to lower the model's error on the training data set.

Support Vector Regression (SVR)

Support Vector Regression (SVR) is a potent statistical learning model that uses regression issues to apply the idea of Support Vector Machines (SVM). SVR attempts to locate a linear regression line, also known as a hyper-plane, in a high-dimensional feature space created by mapping data points (Qian et al., 2015). The SVR model's fundamental elements and methods of operation can be stated as follows (Hsu et al., 2009):

The margin value, epsilon (ε), establishes the maximum amount of inaccuracy that the model can produce. Errors inside the margin ε are accepted by SVR as zero. The plane that creates a connection between the independent

variables in the data set and the dependent variable that has to be forecasted is known as the hyperplane. SVR attempts to make the most accurate predictions of the data points by locating this hyperplane. In order to linearize a non-linear connection, the function that maps the input data into a higher-dimensional space is known as the feature space. The conventional method for doing this mapping is via a kernel function. The input data is converted into a high-dimensional feature space using the kernel function. Sigmoid, polynomial, linear, and radial basis function (RBF) functions are examples of common kernel functions. The coefficients known as the Langrange multipliers are used in the optimization process to calculate the impact of each data point on the hyperplane's location. The model weights the data points in order to calculate the hyperplane, and this is determined by these multipliers.

Lagrange multipliers are determined to be the most effective in determining the margin ε and the hyperplane's position throughout the SVR model's training process (Akay and Abasıkeleş, 2010). This is accomplished by resolving the subsequent optimization issue:

$$\min_{w,b} \frac{1}{2}\|\mathcal{W}\|^2 + C\sum_{i=1}^{n}(\xi_i + \tilde{\xi}_i)$$
$$\text{subject to} \begin{cases} y_i - \langle \mathcal{W}, \phi(x_i)\rangle - b \le \varepsilon + \xi_i \\ \langle \mathcal{W}, \phi(x_i)\rangle + b - y_i \le \varepsilon + \tilde{\xi}_i \\ \xi_i, \tilde{\xi}_i \ge 0 \end{cases} \quad (21)$$

where the slack variables are ξ and ξ , the normal vector of the hyperplane is , W the y-offset is , b the error penalization parameter is C , the mapping of the input data to the feature space is Ø(X_i), and the actual output values are y_i .

SVR is well-known for being resilient to data noise and can be applied to both linear and non-linear regression applications. SVR's robustness and generalization skills allow it to provide accurate predictions even with high-dimensional data sets.

K-Nearest Neighbors (kNN)

For classification and regression issues, the k-Nearest Neighbors (kNN) technique is a supervised learning model. The "learning" part of the model involves storing data points in a feature space and using the k nearest data points to make a prediction each time a new data point is received (Patwary et al., 2016).

Based on its attributes, every data point is represented as a point in an n-dimensional space. These characteristics include, for instance, the characteristics of the samples for a classification issue. The number of nearest neighbors to take into account when generating a classification or regression prediction is determined by the parameter k, which also gives the procedure its name. K is often selected using cross-validation. The "closeness" of a new data point to old ones is determined using the kNN method using a distance metric. Manhattan, Minkowski, and Euclidean distances are examples of common distance measures. The new data point is allocated to the class with the highest representation in the classification issue, which is decided by a majority vote based on the classes of the chosen k closest neighbors. The simple average or weighted average of the dependent variable values for the k nearest neighbors based on their distances is used to make a forecast for the regression issue (Yu et al., 2015).

The k-Nearest Neighbors (kNN) method finds the closest neighbors using a selected distance metric, then uses that knowledge to produce an output value. This is how it operates theoretically. The distance between a data point and a training sample is usually calculated as the Euclidean or Manhattan distance.

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2} \quad (22)$$

For the Euclidean distance:

For the distance to Manhattan:

$$d(x, x_i) = \sum_{j=1}^{n}|x_j - x_{ij}| \quad (23)$$

The jth features of x and xi are represented as xj and xij, respectively, where n is the number of features. The distance measure identifies the k neighbors who are closest to the new data point:

$$NN_k(x) = \{x_i \in X | d(x, x_i) \text{ between the smallest k values}\} \quad (24)$$

where X is the training data set and and $NN_k(x)$ is the k is the closest neighbors to the new data point .

In a regression, the average, or weighted average, of the output values of the closest neighbors is used to calculate the prediction:

$$\hat{y} = \frac{1}{k}\sum_{x_i \in NN_k(x)} y_i \quad (25)$$

or for the weighted average:

$$\hat{y} = \frac{\sum_{x_i \in NN_k(x)} w_i \cdot y_i}{\sum_{x_i \in NN_k(x)} w_i} \quad (26)$$

The weight of the inverse distance can be considered here as W_i :

$$w_i = \frac{1}{d(x,x_i)^2} \quad (27)$$

The actual output value of X_i is represented by Y_i, while the anticipated value is denoted by $\tilde{y}_i$.

The kNN algorithm's method for predicting a data point's categorization or regression is described in these stages. The ability of the kNN algorithm to provide data-driven predictions without requiring the modeling of feature space geometry is one of its benefits. However, the computational cost can be substantial for big data sets, thus selecting the right k number is crucial.

Findings of The Study

The LSTM, GRU, MLP, SVR, and kNN models are employed in this work to anticipate the output of cow cheese. The models employed a large number of independent variables, including the number of cultured dairy cows, the price of dairy feed, the amount of cow milk produced each month, and the amount of cheese exported. Using Python software, the MinMaxScaler technique was utilized to normalize the dependent and independent variables in the dataset. Next, the dataset was split into training (80%) and test (20%) groups. Then, to make sure that every model utilized the same data split, the "randomization" command was employed. This made it possible for the outcomes to be repeatable and comparable. In this regard, the models' predictions on the test data set also fared rather well. 42 was chosen as the random state rate.

Findings Related to The MLP Model

The "Sequential" class was initially used to generate the MLP model once the data set had been prepared to assess the models' prediction skills. There are 300 neurons in the model's first hidden layer, 200 in the second, and 100 in the third. A dasigmoid activation function was employed in each layer. As this is a regression problem, the model's output layer has a single neuron and does not employ an activation function. On the other hand, a learning rate of 0.001 was employed with the "Adam" optimization method. The loss function that was employed was "mean_squared_error". The training loss of the model was recorded and shown on the screen for each of the 500 epochs that it was trained for. To track the model's performance, training and validation losses were recorded at the conclusion of each epoch. The test data served as the basis for the model's predictions. In order to assess the effectiveness of the training, predictions were also generated using the training data. The "MinMax Scaler" was then used to return the normalized forecasts and actual values to the original scale. On the test and training data sets, several error measures are computed, including , RMSE, MSE, MAE, and MAPE. Next, line graphs for the training and test sets are used to depict the actual and anticipated monthly output of cow cheese. Additionally, a graph illustrates how the loss varies throughout the training phase.

The training set was used to train the MLP model for 500 epochs after it was produced. Every epoch's training loss was recorded. The variable "history" was used to monitor the training loss at the conclusion of each epoch. The training loss values recorded throughout the training procedure are displayed on a graph. A graph of the MLP model's training loss values is displayed in Figure 2.

The training loss consistently drops over the course of epochs when the model's training outcomes are assessed using the supplied epoch-wise training loss numbers. From 0.1177 in the first epoch to 0.0037 in the 500th epoch, the loss is reduced. On the other hand, Table 2 displays the MLP model's training outputs' performance.

These results for the training data analysis of the data in Table 2 indicate that the MLP model performs rather well on the training data. Put otherwise, the model predicts the training set with a low error rate. These results, however, don't

show how well the model works with actual data. Its performance on test data is therefore more crucial. Furthermore, Figure 3 displays the MLP model's actual and predicted cow cheese production (CCP) values for the training data set.

Following the MLP model's training, predictions were produced using the trained model on test data. Then, the scales for these forecasts were adjusted to their initial values. Table 3 displays the results of the calculations made using RMSE, MSE, MAE, MAPE, and to determine the errors between the anticipated and actual values on the test data.

The MLP model's performance was evaluated with the test set of data using the error calculations displayed in Table 3. Test data often yields better results for the model than training data does. This shows that the model can reliably provide results in real-world applications and generalize effectively to new and unknown data. However, the actual and expected CCP values for the test data of the MLP model are shown in a graph. Figure 4 shows this graph.

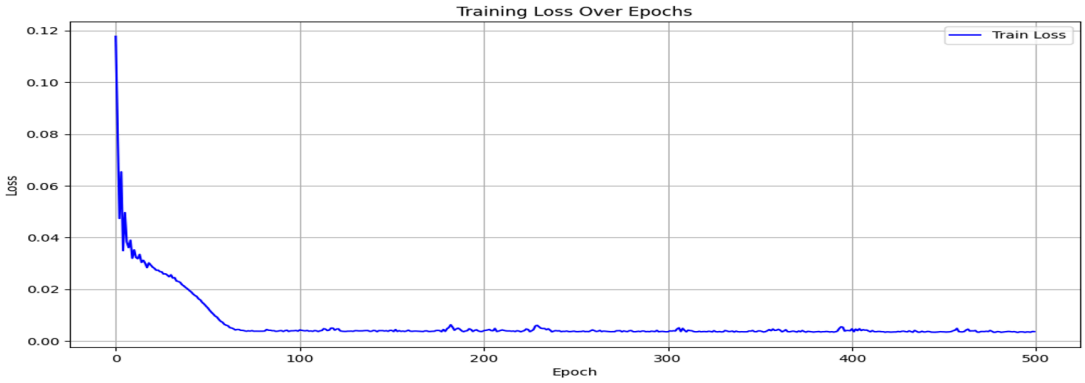

Figure 2. Loss Values of The MLP Model During Training

Table 2. Error Calculations of MLP Model Predictions Using Training Data.

	MLP
RMSE	2.1015
MSE	4.4167
MAE	1.2584
MAPE	0.0278
	0.8917

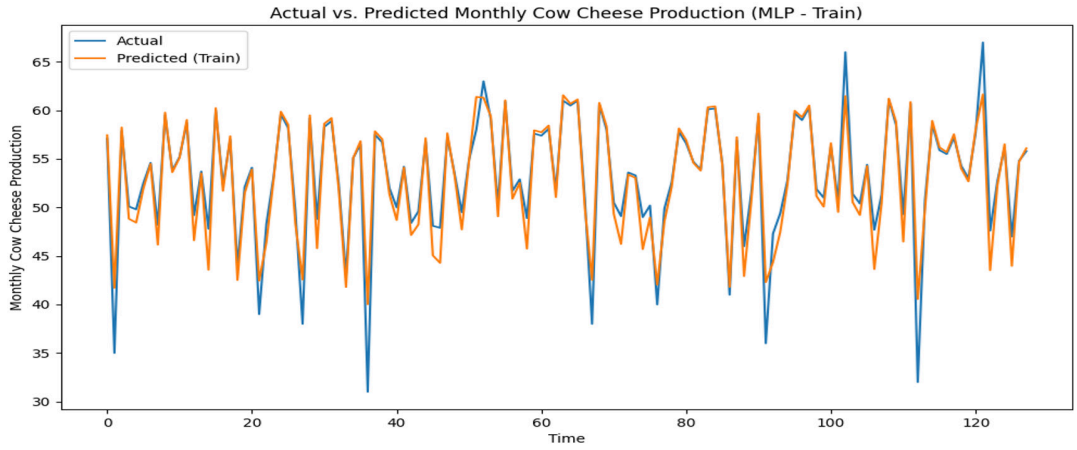

Figure 3. The Actual and Predicted CCP Values of The MLP model on The Training Data

Table 3. Error Calculations of MLP Model Predictions on Test Data

	MLP
RMSE	1.4824
MSE	2.1977
MAE	0.9828
MAPE	0.0200
	0.9076

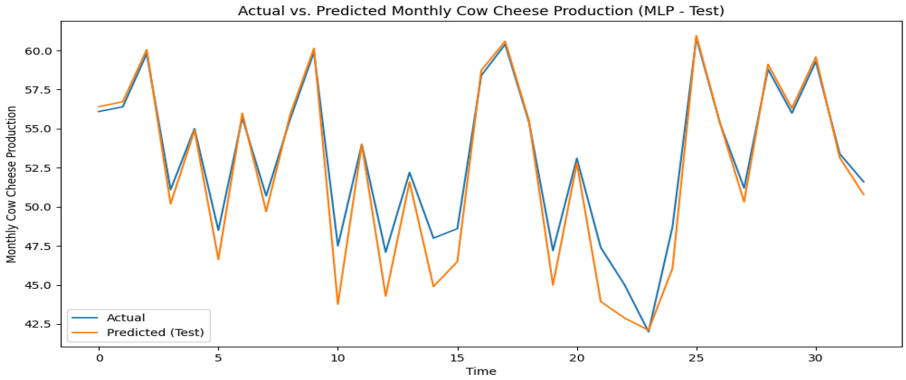

Figure 4. The Actual and Predicted CCP Values of The MLP model on The Test Data

Findings Related to The LSTM Model

The LSTM model is an additional technique employed in the study to estimate CCP values. A sigmoid activation function is applied to the 300 neurons that make up the first layer of the LSTM network. The amount of features in the training data set determines how the input shape is changed. Next, two dense layers with sigmoid activation functions—200 and 100 neurons, respectively—are added. There is just one neuron in the output layer, and regression makes no use of an activation function. Using a learning rate of 0.001, the "Adam" optimization technique was used to assemble the model. The loss function that was selected was "mean_squared_error".

When training machine learning and deep learning models, a hyperparameter known as learning rate is employed. A model's learning rate dictates how much weight is updated while it is being trained. More specifically, the amount by which the weights are changed is determined by the learning rate, which is set at each learning step. This method may require less data because of the enhanced generalization and less overfitting. These settings were designed to prevent overfitting.

The LSTM model was trained on the training set for 500 epochs after it was created. The training loss for each period was recorded. The variable "history" was used to track the training loss at the end of each epoch. The training loss values that were recorded during the training process were displayed on a graph. The training loss value graph of the LSTM model is shown in Figure 5.

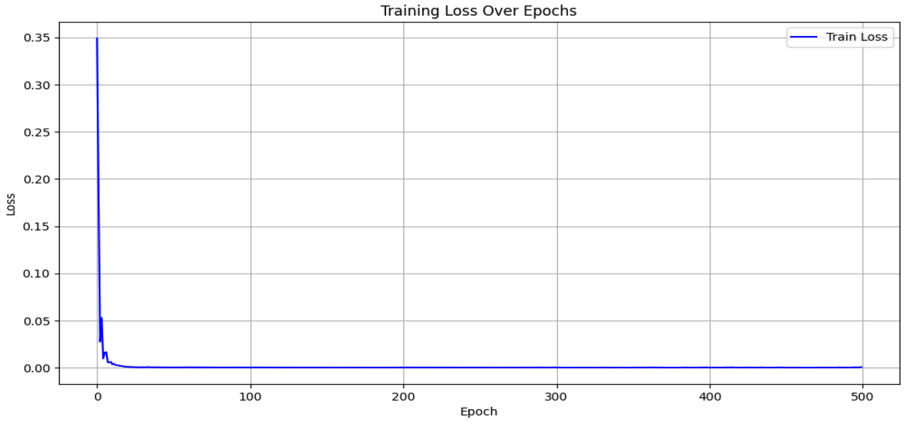

Figure 5. Loss Values of The LSTM Model During Training

The performance of an LSTM model during training is shown by the provided training losses. Given that the model's losses are minimal and often declining with time, its performance during training looks promising. But the model's performance on the test dataset must also be taken into account in order to evaluate its generalization ability. However, Table 4 displays the LSTM model's training outputs' performance.

Table 4. Error Calculations of LSTM Model Predictions Using Training Data

	LSTM
RMSE	0.7675
MSE	0.5890
MAE	0.4609
MAPE	0.0082
	0.9855

Upon analyzing the metrics shown in Table 4, it becomes evident that the model exhibits a strong fit to the training data, resulting in very accurate predictions. Nevertheless, it is crucial to acknowledge that in order to comprehend the model's capacity for generalization, it is imperative to assess its performance on the test data set as well. Furthermore, Figure 6 displays the LSTM model's actual and predicted cow cheese production (CCP) values for the training data set.

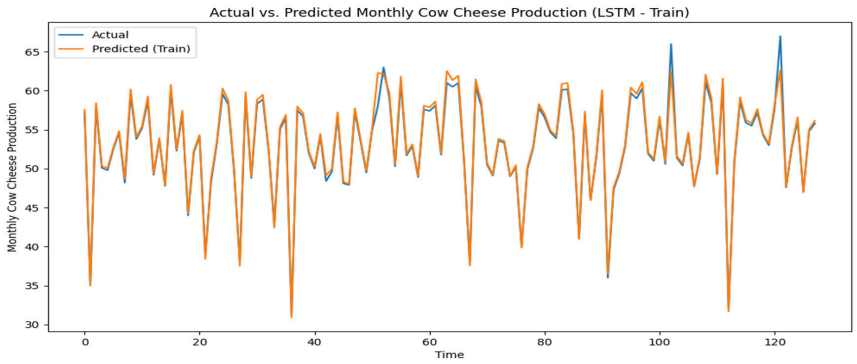

Figure 6. The Actual and Predicted CCP Values of The LSTM Model on The Training Data

Following the LSTM model's training, predictions were produced using the trained model on test data. Then, the scales for these forecasts were adjusted to their initial values. Table 5 displays the results of the calculations made using RMSE, MSE, MAE, MAPE, and to determine the errors between the predicted and actual values on the test data.

Table 5. Error Calculations of LSTM Model Predictions on Test Data

	LSTM
RMSE	0.4387
MSE	0.1924
MAE	0.3852
MAPE	0.0071
	0.9919

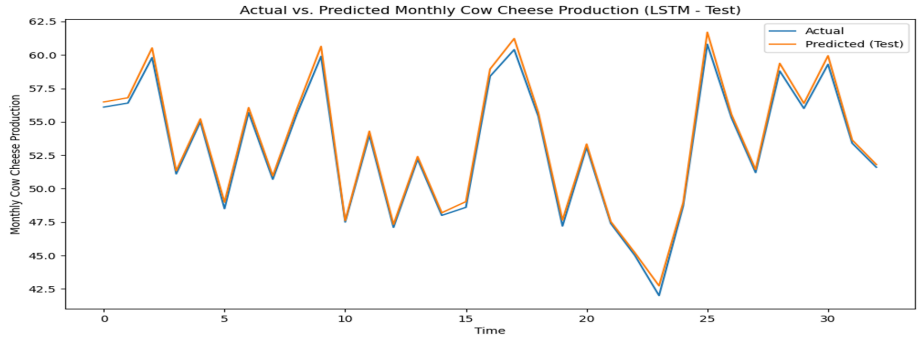

Figure 7. The Actual and Predicted CCP Values of The LSTM Model on The Test Data

Based on the aforementioned criteria, it can be concluded that the model has exceptional performance on the test data set and exhibits the ability to generate accurate predictions on both the test and training data sets. This suggests that the model possesses a strong capacity for generalization, enabling it to effectively extrapolate observed patterns to novel data. On the other hand, a graph displays the LSTM model's actual and predicted CCP values for the test data. In Figure 7, this graph is shown.

Findings Related to The GRU Model

An other method used in the study to estimate CCP values is the GRU model. The first layer of the GRU network consists of 300 neurons that are activated using a sigmoid activation function. The training data set's feature count dictates how the input shape is altered. Two dense layers with 200 and 100 neurons, respectively, having sigmoid activation functions are then added. The output layer consists of a single neuron, and regression does not employ an activation function. The model was put together using the "Adam" optimization strategy with a learning rate of 0.001. "mean_squared_error" was chosen as the loss function.

After it was developed, the GRU model was trained for 500 epochs on the training set. Every period's training loss was noted. At the conclusion of each epoch, the training loss was monitored using the variable "history." A graph showed the training loss values that were noted throughout the training procedure. Figure 8 displays the GRU model's training loss value graph.

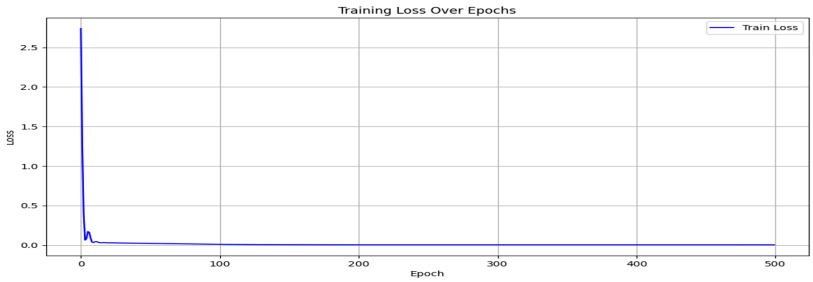

Figure 8. Loss Values of The GRU Model During Training

Figure 8 illustrates how the training loss seems to drop down quickly as training progresses. The training loss dramatically reduces at each epoch after the first, reaching a very low level by the 500th epoch. This demonstrates how effectively the model generalizes and learns the training set of data. Though it also raises the possibility of overfitting, a low training loss suggests that the model does a good job of fitting the training set. However, Table 6 displays the GRU model's training outputs' performance.

Table 6. Error Calculations of GRU Model Predictions Using Training Data

	GRU
RMSE	2.1812
MSE	4.7579
MAE	1.2936
MAPE	0.0286
	0.8833

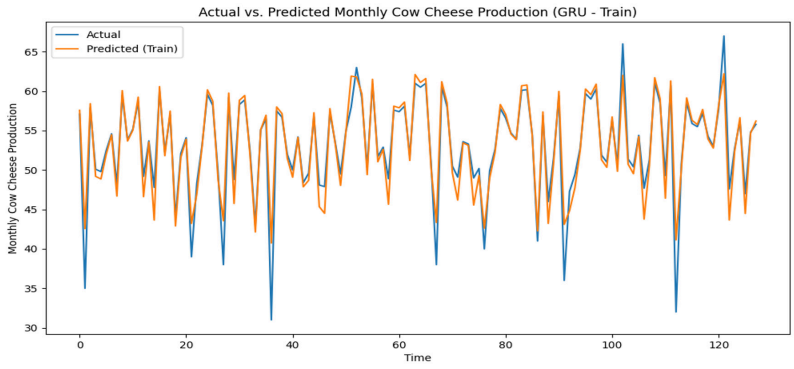

Figure 9. The Actual and Predicted CCP Values of The GRU Model on The Training Data

The table's error statistics suggest that the GRU model does well with training data. To find out how the model will function with real-world data, it is crucial to thoroughly verify and assess the test data. Furthermore, Figure 9 displays the GRU model's actual and predicted cow cheese production (CCP) values for the training data set.

Following the GRU model's training, predictions were produced using the trained model on test data. Then, the scales for these forecasts were adjusted to their initial values. Table 7 displays the results of the calculations made using RMSE, MSE, MAE, MAPE, and to determine the errors between the predicted and actual values on the test data.

Table 7. Error Calculations of GRU Model Predictions on Test Data

	GRU
RMSE	1.3121
MSE	1.7217
MAE	0.9410
MAPE	0.0189
	0.9276

These results demonstrate that the GRU model functions admirably on the test data as well. This indicates that the model can accurately forecast data from the actual world without overfitting the training set. These successes demonstrate that the model was properly trained using the issue context and available data. However, a graph displays the GRU model's actual and predicted CCP values for the test data. In Figure 10, this graph is shown.

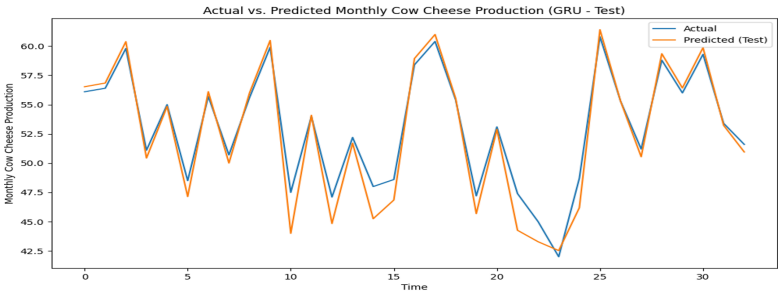

Figure 10. The Actual and Predicted CCP Values of The GRU Model on The Test Data

Findings Related to The SVR Model

An additional method for estimating CCP values in the study was the SVR model. The kernel used to create the SVR model was set to RBF (Radial Basis Function). Additionally, the model made use of the "C" and "Epsilon" parameters. One kind of regularization parameter that is used to manage overfitting is called "C". A margin is an error that is defined by epsilon. The margin determines how near the genuine values should be to the model's predictions. The margin's width is determined by the epsilon value. A larger epsilon number indicates a broader margin, whereas a smaller epsilon value indicates a narrower margin. The model's C value and Epsilon value are respectively set at 5.0 and 0.2. Following the creation of the SVR model, the training set was used to train the SVR model. Table 8 displays the performance of the training outcomes.

Table 8. Error Calculations of SVR Model Predictions on Training Data

	SVR
RMSE	3.6787
MSE	13.5332
MAE	3.3711
MAPE	0.0663
	0.6683

When these results are analyzed, it can be said that the SVR model may need some improvements. The RMSE and MAE values are quite high, and the value is low, indicating that the model cannot fully explain the data. However, the actual and predicted CCP values of the SVR model on the training set are shown in the graph in Figure 11.

Predictions were produced using the trained model on the test data after the training of the SVR model. Then, the scales for these forecasts were adjusted to their initial values. Table 9 displays the results of the calculations made using RMSE, MSE, MAE, MAPE, and to determine the errors between the anticipated and actual values on the test data.

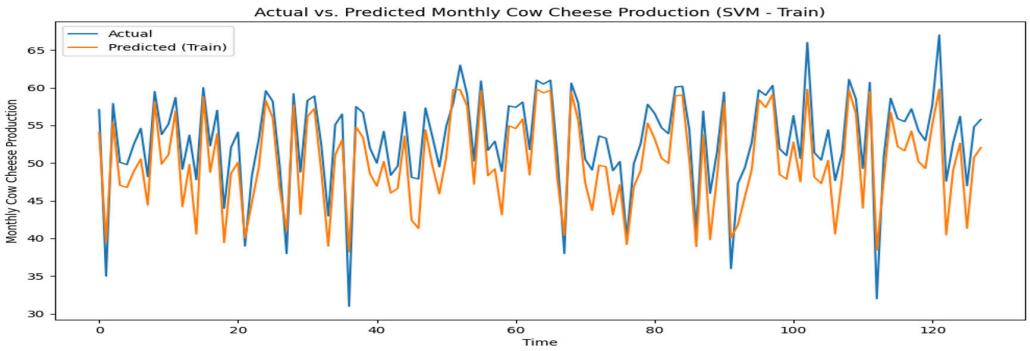

Figure 11. The Actual and Predicted CCP Values of The SVR Model on The Training Data

Table 9. Error Calculations of SVR Model Predictions on Test Data

	SVR
RMSE	3.8132
MSE	14.5406
MAE	3.5418
MAPE	0.0690
	0.3889

The results in Table 9 show that the SVR model does not perform well on the test data. Both RMSE and values are low, and MAE and MAPE values are not at acceptable levels. On the other hand, a graph displays the actual and predicted CCP values of the SVR model based on the test data. In Figure 12, this graph is shown.

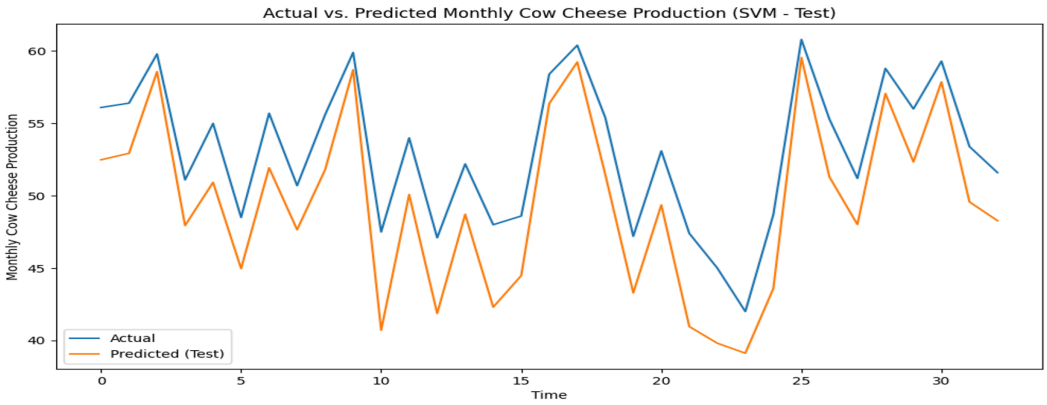

Figure 12. The Actual and Predicted CPI Values of The SVR Model on The Test Data

Findings Related to The kNN Model

The kNN model was another technique used in the research to estimate CCP values. The kNN regression model—the last machine learning model used in this study for CCP prediction—was created using the Python application "KNeighborsRegressor." This model uses the k-NN approach to find the relationship between the independent and dependent variables. The option "n_neighbors" (Number of Neighbors) determines how many neighbors to use in the kNN algorithm. In this model, N_neighbors is equal to 5. This suggests that the five nearest neighbors will be used in each prediction. Conversely, the "weights" parameter determines how the neighbors are weighted in the prediction. Two common choices are uniform and distance. Uniform weighting allocates the same weight to every neighbor, while distance weights are inversely proportional to neighbor distance. The model's weights are "uniform," which means that each neighbor has the same weight. The option "metric (distance metric)" specifies the distance metric to be used for calculating the distance between neighbors. Different metrics may be used, such as Euclidean, Manhatten, and Minkowski distances. In this paradigm, the metric is "euclidean," so Euclidean distance is used. The training set was used to train the k-NN model once the KNN model was created. The performance of the training results is shown in Table 10.

Table 10. Error Calculations of kNN Model Predictions on Training Data

	KNN
RMSE	2.5715
MSE	6.6128
MAE	1.1426
MAPE	0.0269
	0.8379

The results in Table 10 show that the kNN model performs well on the training data. Low RMSE, MSE, and MAE values indicate that the model's predictions on the training data are generally close to their true values. Also, the high value indicates that the model is well fitted to the training data. However, these results are only obtained for training data and do not provide information about the model's performance on test data. In addition to this, the actual and predicted CCP values of the kNN model on the training set are shown in the graph in Figure 13.

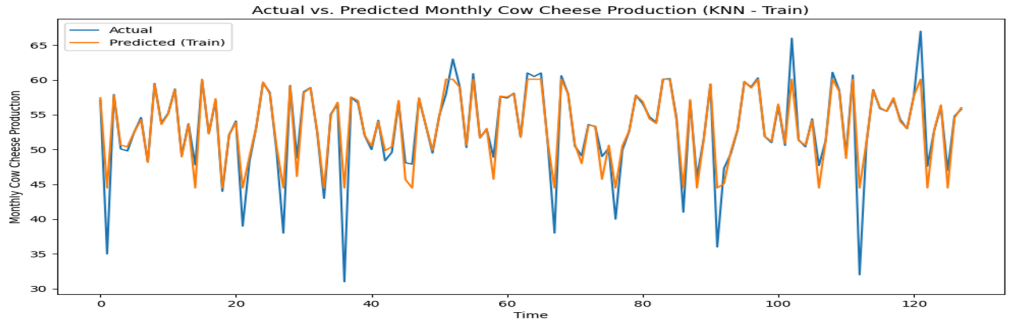

Figure 13. The Actual and Predicted CCP Values of The kNN Model on The Training Data

Following the kNN model's training, predictions were generated using the trained model on the test data. Following that, the predictions' scales were returned to their starting points. The computations of the errors between the expected and actual values on the test data using RMSE, MSE, MAE, MAPE, and are shown in Table 11.

Table 11. Error Calculations of kNN Model Predictions on Test Data

	kNN
RMSE	1.0633
MSE	1.1307
MAE	0.6235
MAPE	0.0129
	0.9524

These outcomes demonstrate the kNN model's excellent performance on the test set. A graph displays the actual and anticipated CCP values of the kNN model based on the test data. In Figure 14, this graph is shown.

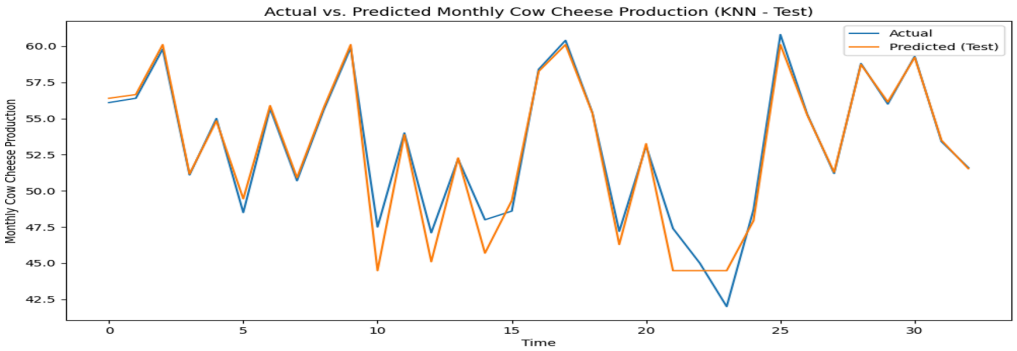

Figure 14. The Actual and Predicted CCP Values of The kNN Model on The Test Data

Table 12 summarizes the test results for each of the forecasting models used in this study.

Table 12. Error Calculations of All Models Predictions on Test Data

	LSTM	GRU	MLP	SVR	KNN
RMSE	0.4387	1.3121	1.4824	3.8132	1.0633
MSE	0.1924	1.7217	2.1977	14.5406	1.1307
MAE	0.3852	0.9410	0.9828	3.5418	0.6235
MAPE	0.0071	0.0189	0.0200	0.0690	0.0129
	0.9919	0.9276	0.9076	0.3889	0.9524

The error estimates of the predictions produced by five different models on the test data are shown in Table 12's findings. From various angles, each error measure aids in assessing a model's performance. The LSTM model performs the best and has the lowest RMSE when it comes to this data. Next in line is the kNN model. The model's predictions are said to be quite accurate if the MSE value is low. Once again, the model with the lowest MSE is LSTM. The model's predictions are said to be reasonably accurate if the MAE value is low. The LSTM model has the lowest MAE once again. The model's predictions are in near percentage agreement with the real values when the MAPE value is low. Once again, the LSTM model has the lowest MAPE. A high value means that a significant amount of the dependent variable's variation can be explained by the model. As the model that most closely matches the test data in this situation, the LSTM model has the greatest .

In light of this, the comparison indicates that, when it comes to test data performance, the LSTM model outperforms the other models. SVR and MLP models perform worse than GRU and kNN models, which nevertheless perform well.

Based on these results, LSTM is selected as the most successful model in terms of both training and testing performance in the forecasting process of CCP data, and then the forecasting process for the next 12 months is performed using this model. The 12-month prediction was produced using a pre-trained LSTM model with the same hyperparameter parameters as previously mentioned for future forecasting. The model predicted the most recent test data point "X_test[-1]" for each forecast. The anticipated values were added to the "future_predictions" list. A new data point, the final predicted value, was added to the "X_test" dataset. This was applied to the subsequent forecast. Following Min-Max normalization, the projected values were transferred back to the original scales. After that, Python software was used to print the forecasts. Table 13 shows CCP values for the next 12 months.

Table 13. CCP Values for the Next 12 Months According to the LSTM Model

Months	Monthly CCP (Thousand Ton)
2023-10	62.09
2023-11	61.80
2023-12	61.36
2024-1	60.79
2024-2	60.07
2024-3	62.01
2024-4	62.72
2024-5	62.39
2024-6	63.05
2024-7	63.73
2024-8	64.48
2024-9	65.33

The monthly output of cow cheese is trending increasing, according to an examination of the projections. This suggests that the output of cheese will probably continue to increase in the next months. Although there is an overall rising tendency, there is monthly uncertainty amongst projections. This variation can be a reflection of variations in output levels over several months. The LSTM model's predictions seem to have a good overall performance in terms of model performance since they are quite close to the actual values. Our understanding of future output and demand for economic sectors like the dairy industry may be improved with further study and improvement of these forecasting models. Ultimately, your LSTM model's monthly cow cheese output projections seem to be a helpful resource for predicting production patterns and variability in the future. However, the graph of the CCP forecast for the next 12 months is shown in Figure 15.

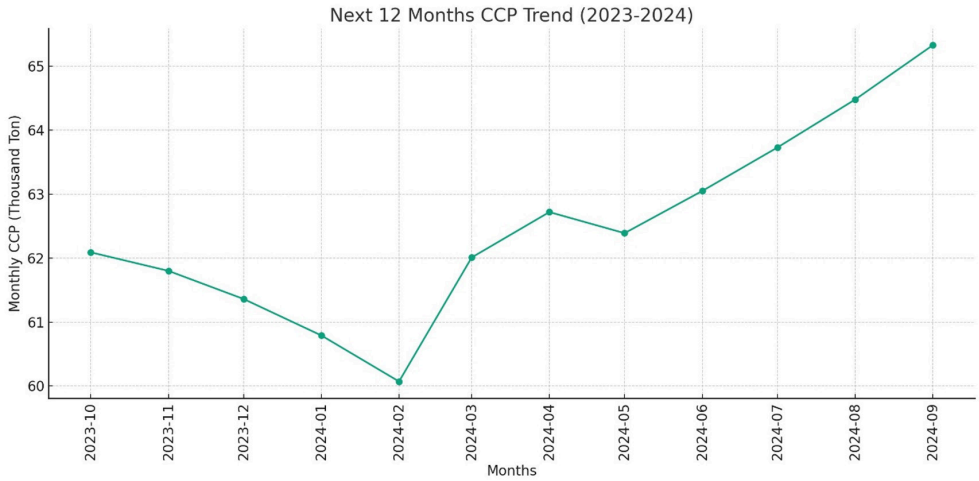

Figure 15. Next 12 Months CCP Trend (2023-2024)

The trend of the Monthly CCP (in thousand tons) from October 2023 to September 2024 is shown in this graphical representation of the data that was given. The CCP value for a given month is represented by each point on the graph, and the line that links these points shows the trend over time.

Conclusion and Recommendations

This study emphasizes how crucial forecasting is to the dairy sector, especially when it comes to predicting the amount of cheese produced from cow's milk (CCP). Precise predictions are essential for effective management of the supply chain, achieving a balance between production and demand, and developing plans to lessen the impact of future market swings.

Because machine learning and deep learning techniques have shown efficacy in handling complicated, non-linear data with several unexpected elements influencing agricultural yield, they were used in this investigation. The goal was to determine which model—LSTM, GRU, MLP, SVR, and kNN—was the most useful for CCP prediction by comparing them.

The results demonstrate that in terms of CCP prediction, the LSTM (long short-term memory) model performs noticeably better than the other models. This advantage is shown in a number of error measures, showing the model's resilience and dependability, including RMSE, MSE, MAE, MAPE, and . The LSTM model seems to be especially useful for forecasting tasks like CCP forecasting, where previous patterns play a significant role in creating future trends, because of its past patterns and its resilience to gap length in the time series.

Nonetheless, there should be a lot of advantages to utilizing the LSTM model to predict cow cheese output for the next year. Production process management is made possible by forecasts. This makes it easier to calculate the precise amount of raw materials needed and to modify production capacity in response to demand. Reducing stock shortages and surpluses is facilitated by knowing future production volumes. This stops waste and lowers expenses. It guarantees that the market will have enough supply to fulfill demand. Customers are happier and brand loyalty is strengthened as a result. Supply-demand balance-based pricing methods may be developed more successfully with the use of production quantity forecasting. It makes it possible to anticipate hazards associated with unforeseen changes or production bottlenecks. This makes it possible to create and put into practice risk management plans. gives details on potential investment possibilities, general market dynamics in the industry, and future production patterns. provide the knowledge required to decrease food waste and make better use of resources, both of which promote environmental sustainability. lays the groundwork for a deeper comprehension of shifts in customer preferences and market demand, which promotes the creation of new goods and the enhancement of current ones.

To further highlight the uniqueness and significance of the research, it would be helpful to compare the cow cheese production forecasting findings achieved in this work using the LSTM model with comparable studies in the literature. When it came to error measures like RMSE, MSE, MAE, MAPE, and , the LSTM model fared better in the research than the other models. This implies that, in comparison to other research approaches, including the ANN model of Goyal and Goyal (2013) and the ARIMA model of Yildirim and Altunc (2020), the LSTM model offers an edge in predicting accuracy. Specifically, the LSTM model exhibits more accuracy when juxtaposed with the values of the ANN models in the Goyal and Goyal research. Nonetheless, this research closes a vacuum in the literature since the majority of earlier studies in the subject of cheese production forecasting concentrated on the forecasting of milk, dairy, and other agricultural

goods. This makes the research more significant in the literature and offers a special contribution. The models used in research works like Li and Liu (2023) and Keskinbıçak (2023) are capable of managing intricate data sets and accounting for diverse data kinds. According to this research, the LSTM model's capacity to handle such intricate and varied data presents a big benefit for predictions that are sector-specific. Research like those conducted by Ma et al. (2021) and Gandotra et al. (2023) shows how machine learning and deep learning models affect real-world data. The findings of this study might have a significant influence on future demand and production projections, particularly in industries like the dairy sector. The research conducted by Liseune et al. (2021) demonstrates the advanced methodological advances and advancements of the LSTM model utilized in this investigation, as shown by the usage of complicated machine learning and deep learning models. Consequently, this comparison demonstrates that our work is among the first to anticipate cheese output using the LSTM model and is a noteworthy methodological advancement in this area. Its unique application and accurate forecasts also constitute a significant addition to the literature.

Future research may benefit from comparison evaluations of various deep learning and machine learning models (such as CNN and GRU) to determine which models are more appropriate for certain circumstances. The general validity and robustness of the model may be strengthened by experimenting with bigger and more varied datasets. The model's potential uses might be increased by using it to predict the production of milk and cheese as well as other food and agricultural products. Real-time data may be included into the modeling to increase prediction accuracy and timeliness. Forecasts may be made more thorough by include external aspects like market movements, climate change, and socioeconomic issues in the modeling. Forecast success may be increased by fine-tuning the LSTM model's parameters and settings. It is feasible to get more detailed information about potential future trends and changes by creating longer-term projections and conducting analysis under various scenarios. These suggestions may improve understanding and applications in this area and be helpful for both academic and industry uses.

Compliance with Ethical Standards

Peer-review

Externally peer-reviewed.

Conflict of interest

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Author contribution

The author read and approved the final manuscript. The author verifies that the Text, Figures, and Tables are original and that they have not been published before.

Funding

No financial support was received for this study.

Data availability

Not applicable.

Consent to participate

Not applicable.

Consent for publication

Not applicable.

REFERENCES

Agarap, A. F. M. (2018). A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data. In Proceedings of the 2018 10th international conference on machine learning and computing (pp. 26-30).

Akay, M. F. & Abasıkeleş, I. (2010). Predicting the performance measures of an optical distributed shared memory multiprocessor by using support vector regression. Expert Systems with Applications, 37(9), 6293-6301.

Athiwaratkun, B. & Stokes, J. W. (2017). Malware classification with LSTM and GRU language models and a character-level CNN. In 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 2482-2486).

Bulut, E. (2024). Market Volatility and Models for Forecasting Volatility. In *Business Continuity Management and Resilience: Theories, Models, and Processes* (pp. 220-248). IGI Global.

Desai, M. & Shah, M. (2021). An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN). Clinical eHealth, 4, 1-11.

Durlu-Özkaya, F. & Gün, İ. (2007). Cheese culture in Anatolia. ICANAS, International Congress of Asian and North African Studies, 10-15.

Gandotra, S., Chhikara, R. & Dhull, A. (2023). Wheat, Rice and Corn Yield Prediction for Jammu District Using Machine

Learning Techniques. In International Conference on Information and Communication Technology for Intelligent Systems (pp. 499-512). Singapore: Springer Nature Singapore.

Goyal, S. & Goyal, G. (2013). Machine learning models for predicting shelf life of processed cheese. International Journal of Open Information Technologies, 1(7), 28-31.

Güllü, M. (2022). The Chemistry Of Milk: Cheese And Cheese Varieties In Turkey. Thematic Researches in the Field of Gastronomy I, 47.

Hsu, S. H., Hsieh, J. P. A., Chih, T. C. & Hsu, K. C. (2009). A two-stage architecture for stock price forecasting by integrating self-organizing map and support vector regression. Expert Systems with Applications, 36(4), 7947-7951.

Kahraman, E. M. (2012). Milk and Dairy Products Production and Consumption Comparative Analysis of Turkey and World Data. Agricultural Engineering, (359), 48-52.

Li, B., Lin, Y., Yu, W., Wilson, D. I. & Young, B. R. (2021a). Application of mechanistic modelling and machine learning for cream cheese fermentation pH prediction. Journal of Chemical Technology & Biotechnology, 96(1), 125-133.

Li, W., Kiaghadi, A. & Dawson, C. (2021b). Exploring the best sequence LSTM modeling architecture for flood prediction. Neural Computing and Applications, 33, 5571-5580.

Li, X. & Liu, J. (2023), Hyperspectral Imaging Combined with Long-Short Term Memory Network for Accurately Detecting Adulteration in Milk. Available at SSRN 4597631.

Liseune A, Poel V, Hut P, Eerdenburg F & Hostens M (2021). Leveraging sequential information from multivariate behavioral sensor data to predict the moment of calving in dairy cattle using deep learning, Computers and Electronics in Agriculture, no. 191.

Ma, Y., Zhang, Z., Kang, Y. & Özdoğan, M. (2021). Corn yield prediction and uncertainty analysis based on remotely sensed variables using a Bayesian neural network approach. Remote Sensing of Environment, 259, 112408.

Nosouhian, S., Nosouhian, F. & Kazemi Khoshouei, A. (2021). A Review of Recurrent Neural Network Architecture for Sequence Learning: Comparison between LSTM and GRU. Preprints. https://doi.org/10.20944/preprints202107.0252.v1

Patwary, M. M. A., Satish, N. R., Sundaram, N., Liu, J., Sadowski, P. Racah, E., ... & Dubey, P. (2016). Panda: Extreme scale parallel k-nearest neighbor on distributed architectures. In 2016 IEEE international parallel and distributed processing symposium (IPDPS) (pp. 494-503). IEEE.

Qian, Z. L., Juan, D. C., Bogdan, P., Tsui, C. Y., Marculescu, D. & Marculescu, R. (2015). A support vector regression (SVR)-based latency model for network-on-chip (NoC) architectures. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 35(3), 471-484.

Ramchoun, H., Idrissi, M. J., Ghanou, Y. & Ettaouil, M. (2017). Multilayer Perceptron: Architecture Optimization and training with mixed activation functions. In Proceedings of the 2nd international Conference on Big Data, Cloud and Applications (pp. 1-6).

Smagulova, K. & James, A. P. (2019). A survey on LSTM memristive neural network architectures and applications. The European Physical Journal Special Topics, 228(10), 2313-2324.

Şimşek, A. I. (2024). Using Machine Learning and Deep Learning Methods in Predicting the Islamic Index Price. *Fintech Applications in Islamic Finance: AI, Machine Learning, and Blockchain Techniques*). IGI Global. Doi:10.4018/979-8-3693-1038-0.ch018

Turkish Statistical Institute, https:// data.tuik.gov.tr/ Access Date: 21.09.2023

Yıldırım, A. & Altunç, Ö. F. (2020). Estimation of milk production in Muş province with arima model. Anemon Muş Alparslan University Journal of Social Sciences, 8(UMS'20), 137-146.

Yu, C. D., Huang, J., Austin, W., Xiao, B. & Biros, G. (2015, November). Performance optimization for the k-nearest neighbors kernel on x86 architectures. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 1-12).