Tanzania Journal of Engineering and Technology 2023, **42(3)**: 64-77

Websites: https://ajol.org/tjet; https://tjet.udsm.ac.tz

ISSN 1821-536X (**print**); ISSN 2619-8789 (**electronic**)
https://doi.org/10.52339/tjet.v42i3.802

*Regular Research Manuscript*

# Modelling Prediction of Cities Real Estate Price Trend Using Recurrent Neural Network: A Case of Dar es Salaam City

**Ellen A. Kalinga[1†], Ezekiel Nyamle[1] and Abdi T. Abdalla[2]**

[1]Department of Computer Science and Engineering,

College of Information and Communication Technologies, University of Dar es Salaam

[2]Department of Electronics and Telecommunications Engineering,

College of Information and Communication Technologies, University of Dar es Salaam

[†]Corresponding author: ellenakchale@gmail.com

ORCID: https://orcid.org/0000-0002-9262-0308

## ABSTRACT

*Real estate refers to a class of real property such as land and its associated infrastructure. The prediction of real estate prices in cities, which is affected by a number of parameters, is an open research problem. The lack of reliable and effective tools for price forecasting in real estate, especially in residential housing, can adversely affect investment flows and the growth of the real estate sector. Taking Tanzania as an example, the price prediction practices rely on human suggestions that are prone to personal bias and subjective to price hysteria for personal gain and impact consumer expectations. To address the challenge, this paper designed a real estate price trend prediction model for the cities using Recurrent Neural Networks (RNN) with a Long Short-Term Memory (LSTM). The study identified the factors influencing real estate property prices, including size, location, time, property quality, accompanied services, market nature, price of land, cost of building materials, and value for money. However, the study spotted the size, price, location, and time as key factors in predicting price trends when using RNN-LSTM. The results show that the proposed RNN-LSTM model performed better with 50% MSE less compared to the Convolutional Neural Network (CNN). In computing the price trend per location, the model prediction accuracy was 97.45%, 79.23%, and 53.8% for the high class, middle class, and low class, respectively, resulting in an average prediction accuracy of 76.8%.*

ARTICLE INFO

First submitted: **Oct. 7, 2022**

Revised: **Aug. 14, 2023**

Accepted: **Aug. 30, 2023**

Published: **Oct., 2023**

**Keywords:** *Modelling, Price Trend, Price Prediction, Real Estate, Recurrent Neural Network*

## INTRODUCTION

According to Jud and Frew (1986), real estate is a property consisting of land and building on it with its natural resources such as crops, minerals, or water bodies, and other immovable properties of this nature. Real estate is the business of buying, selling, or renting a building, land, or building, residential or commercial. The study by Martins *et al.*, (2016) revealed that the real estate market has a direct impact on the banking sector as real estate can be one of the main collaterals for loans. This implies that any increase in real estate value will result in increasing bank credit granting.

Forecasting real estate prices in fast-growing cities influences the development of the real estate sector as the power system faces a transition toward a more intelligent, flexible, and interactive system with higher penetration of renewable energy generation, load forecasting, especially short-term load forecasting for individual electric customers, plays an increasingly essential role in future grid planning and operation. Other than aggregated residential loads on a large scale, forecasting the electric load of a single energy user is fairly challenging due to the high volatility and uncertainty involved. In this paper, we propose a long short-term memory (LSTM) recurrent neural network-based framework, which is the latest and one of the most popular techniques of deep learning, to tackle this tricky issue. The proposed framework is tested on a publicly available set of real residential smart meter data, whose performance is comprehensively compared to various benchmarks, including the state-of-the-art in the field of load forecasting. As a result, the proposed LSTM approach outperforms the other listed rival algorithms in the task of short-term load forecasting for individual residential households.

This research seeks to demonstrate how machine learning, a branch of artificial intelligence, is able to deliver more accurate pricing predictions, using the real estate market as an example. Utilizing 24,936 housing transaction records, this paper employs Extra Trees (ET), k–Nearest Neighbors (KNN), and Random Forest (RF) to predict property prices and then compares their results with those of a hedonic price model. In particular, this paper uses a feature (property age x square footage) instead of property age in order to isolate the effect of land depreciation on property prices. Our results suggest that these three algorithms markedly outperform traditional statistical techniques in terms of explanatory power and error minimization. Machine learning is expected to play an increasing role in shaping our future. However, it may raise questions about privacy, fairness, and job displacement issues. It is therefore important to pay close attention to the ethical implications of machine learning and ensure that the technology is used responsibly and ethically. Researchers, legislators, and industry players must work together to create appropriate standards and legislation to govern the use of machine learning (Choy and Ho, 2023; Geng et al. 2023; Kong et al. 2019). According to Nilsson (2019), prediction surveys support real estate stakeholders such as developers, consumers, tenants, real estate agents, government institutions, and organizations dealing with land and housing in making optimal decisions and choices in real estate. As narrated by Jud and Frew (1986), the concept of appraising residential real estate is heavily reliant on the interpretation of human behaviour. This involves the prediction of human decision-making based on some factors, both objective and subjective. These factors form a complex union that influences the property's end value.

In Tanzania, there is inadequate information, such as location, size, quality of property, and inflation rate, about the housing market and supply dynamics, which influence the market price (Magina, 2016). Magina (2016) revealed that the access and information demand for residential housing and plots for development in the real estate sector is rapidly growing. This is due to the growth in the population and an increase in human economic activities. In Tanzania, for example, reliable house pricing varies due to numerous factors such as inflation rates, interest rates, demand, supply, and high costs in housing investments (Gardner *et al.*, 2019). However, there is no reliable price trend prediction model to bridge the information gap and uncover the hidden relationship between different residential real estate variables, allowing data-driven decision making in the real estate industry (Magina, 2016). The Tanzanian government has made direct and indirect efforts to take control of real estate matters, though the information on the prediction of residential housing and plot price trends in real estate is crucial for landlords, tenants, and buyers (Magina, 2016). The need to investigate the major players of the current world pricing economic determinants, namely inflation rate and interest rate, location, and other variables to determine house pricing is of great importance to establishing the housing pricing status in the cities.

Literature shows that the application of Artificial Intelligence (AI) based algorithms can significantly improve the performance of the prediction models as it enables the models to recognize their environment and learn to react independently to signals to make decisions (Chauhan *et al.*, 2018). In Tanzania, regardless of economic growth and periodical changes in government policies, there is no reliable model and well-defined parameters such as size, location, time, and documentation that can be employed in predicting real estate price trends in Tanzania (Magina, 2016). Therefore, to improve the accuracy of price trend predictions, this paper developed a real estate price trend prediction model using an AI-based algorithm, Recurrent Neural Networks with a Long Short-Term Memory, with Dar es Salaam city as a case study.

## RELATED WORKS

Bello *et al.* (2020) reviewed various techniques and methods adopted by researchers and housing experts in analysing the housing market and identified that the traditional multiple regression models and the estimated neural network (NN) models were useful in analysing housing markets. The proposed model has a good fit in both training and test results. The results provided by the NN can support investments in the transport system. The ANN model can also help appraisers make assessments and promote environmental regeneration. In Sampathkumar *et al.* *(*2015), the authors used multiple regression and neural network techniques to model the land price trend with the support of economic and social factors, in which all the input values are normalized using the MinMax method. The paper focused on the modelling and prediction of land prices in the Chennai Metropolitan Area. Even though both models were found to be well fitted with the data set of the land price in all locations, the model using NN (correlation 98%) showed better accuracy than the regression model (correlation 96%).

The study conducted by Wang *et al.* (2017) in Singapore identified 10 economic variables that may exhibit a strong correlation with housing prices, initially considered exogenous variables. A correlation test was performed to determine the most suitable variables for the input vectors before training the RNN. The identified variables are: Real Gross Domestic Product (GDP), Population, Unemployment Rate, Average Monthly Wages, Labour Costs, Straits Times Index (STI) for the stock market, Prime Lending Rate, Interbank Rate, and consumer spending, and Consumer Price Index (CPI). Wang *et al.* (2018) conducted a study which showed that the Recurrent Neural Network with a Long Short-Term Memory (RNN-LSTM) can learn from labelled samples and make predictions after online training.

In Tanzania, there is no scientifically proven means of predicting the future price of real estate products. Traditional practices make inferences based on varying experiences such as property quality, accompanied services, nature of the market, current price of land, cost of building materials, value for money, and internal policy in relation to the prediction of rental price trends in real estate. As a result, the price is determined by demand and supply, as well as any price fixation done by agents or brokers. Currently, there is no practical tool that can compute and predict price trends in Tanzania. Thus, given the widespread practice of predicting price trends, this paper seeks to design a real estate price trend prediction model using a RNN-LSTM and pre-test it using Dar es Salaam city real estate stakeholders' data. LSTM offers a number of advantages over traditional RNNs: they are much better at handling long-term dependencies, due to their ability to remember information for extended periods of time; and they are less susceptible to the vanishing gradient problem

## METHODS AND MATERIALS

The study adopted the RNN-LSTM design, following five stages: data collection, preprocessing of data, building a network, training a network, and testing a network, in (Mukhopadhayay *et al.*, 2014).

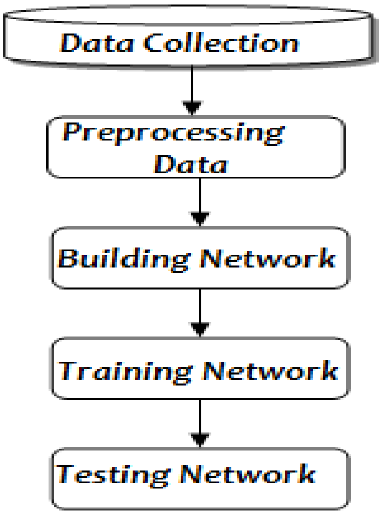



**Figure 1: Basic flow in neural network modelling (Source: Mukhopadhayay et al., 2014).**

The selected study area represented upper, middle, and lower-class residents, and hence, the results of the study can be fairly generalized. The study involved key informants from housing and plots such as investment officers from the National Housing Corporation (NHC), Tanzania Building Agency (TBA), private individual property owners, brokers, and the available mobile app-based broker. The informants were purposefully selected because of their experience in real estate investment, while customers operate as property renters and buyers.

In this study, the first step was to collect and prepare data. Through the Dalali App Mobile Android application, 10,406 raw data points were collected to relate to the contextual aspects of real estate in Dar es Salaam. To prepare the data in a manner that a deep learning model can accept, the data set was pre-processed and filtered to 4,000. This included data on plots for sale, houses for sale, and houses for rent. The collected data was transformed into a matrix vector with one row and many columns. Thereafter, the data was divided into 80% for training and 20% for testing purposes.

Through a literature review, the platform for identifying the pricing trend of residential real estate in Tanzania was established. By looking at the price of real estate, the basis for the pricing was established. The study then identified predictors to be adopted in the context of Tanzania. The number of input and output parameters per training sample was selected and altered based on the errors during training. This step also includes choosing training parameters (such as the type of membership function); training algorithms (such as back propagation and least squares); and training optimization methods (such as determining the initial step size increase and decrease rate).

TensorFlow version 1.6 and Python 3.5.2 with the Keras neural network library were used to develop the RNN-LSTM model. TensorFlow 1.6 was selected due to its simplicity and ease of use. The Python codes were written to enable all simulation experiments to be carried out on this toolbox.

## PRICE TREND PREDICTORS AND RNN MODEL DESIGN

### Analysis of Real Estate Price Trend Predictors

The real estate price trend predictors come from stakeholders, real estate agencies/organizations, and an online survey using a questionnaire. Interviews with stakeholders revealed that property quality, accompanied by services, nature of the market, price of land, cost of building materials, value for money, and internal policy variables (parameters) determine the real estate rental price. The answers provided showed that policy plays a major role in determining real estate rental prices. As such, property owners, particularly NHC, cannot rent a premise below the recommended price as guided by the policy.

In relation to the application of models by the real estate agencies to determine future prices of residential properties and plots for development, the study realized that there is no particular model used by the agencies in determining the prices. This is due to a lack of reliable and sufficient information about the market. It is therefore, challenging to predict the exact price due to challenges in the regulation of the real estate business, inflation rates, and economic changes that in turn affect demand for real estate regulatory policies. An online survey tool indicated that factors that determine price include size of land, number of rooms, current price, time and location, property ownership documents such as title deeds, and surveyed or unsurveyed areas. More of these are easy to access or experience quick growth, as is the size of the land, which determines the price of residential housing and plots for development. However, the study highlighted that a lack of proper documentation of properties and inconsistency of information affect price prediction. The study aligned with findings from the aforementioned studies to seal a decision to use time, price, location, and size in investigating future price predictors of real estate residential housing and plots for development in the context of Dar es Salaam city in Tanzania.

### Recurrent Neural Networks Model Design

The study divided RNN model implementation into three major parts: pre-training, training, and post-training. Each part of training depends on pre-training, while post-training depends on both training and pre-training.

#### (a) *RNN Model Pre-training*

Pre-training activities can be grouped into three stages: library importation, data preparation, and reading or uploading of data. For library importation, Chauhan *et al.* (2018) explain that different functions can easily perform complex computations and be represented in the form of graphs and by mapping the graph parts to the machine in the form of a cluster. Figure 2 shows the imported libraries required for training, visualization, and analysis of the model. Some of the features that are offered by the library include functions and APIs that are useful for: loading data into the model; training the model; making future predictions; and plotting graphs.

```
##Step 0: Importing various Libraries to be used to develope a model
import numpy as np
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense, Dropout
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.preprocessing import StandardScaler
import seaborn as sns
```

**Figure 2: Library imports.**

The preparation of data entails actions taken prior to loading data into the model. The data loaded into the model was categorized into three groups: plots for sale, houses for sale, and rentals. Considering key predictors such as price, location, size, and time, the data was split into three groups: high-, middle-, and low-class residents, as related to class "A", class "B", and class "C" classifications of residents, respectively, as suggested by Gower (2022). According to Gower (2022), location (well-located, easy access to major employers, hospitals, universities, etc.) is the biggest driving factor of residents' classification, followed by the age of a building (new or old), property condition (fully renovated and upgraded with high-end finishes), facilities, and occupancy. The predictors were arranged in columns in the data set, as shown in the sample data in Table 1. The issue of sorting data by time was a very crucial step in the time series model, as the model uses time to predict the future prices of real estate.

**Table 1: Data sorting**

| | Year | Price_IN_TZS | square_meter | title_deed | district_id | street_id | street_name | district_name |
|---|---|---|---|---|---|---|---|---|
| 0 | 09/03/2019 | 6000000 | 400 | 1 | 9 | 10 | KIGAMBONI | Kinondoni |
| 1 | 08/04/2019 | 20000000 | 400 | 0 | 9 | 185 | Goba | Kinondoni |
| 2 | 09/04/2019 | 7000000 | 400 | 0 | 8 | 223 | Goba Osterbay | Ilala |
| 3 | 17/04/2019 | 2800000 | 400 | 1 | 8 | 271 | Tabata | Ilala |
| 4 | 26/04/2019 | 9000000 | 400 | 0 | 9 | 185 | Goba | Kinondoni |

Uploading of data involves the loading of the prepared data set into the RNN model, as presented in Figure 3.

```
##Read Data from the Data Source
df = pd.read_csv('plots_for_sale_tegeta_size_m_400.csv')
```

This creates an understanding of the raw data pattern collected before filtering. Accurate data sampling and selection are crucial in the prediction of price trends. The data was visualized in graphs to prepare the data for the filtering process. Figure 2 indicates prediction results with errors before sampling and preprocessing of data. Unprocessed data was filtered to remove junk data before training the model. This process involves grouping data according to size, location, and, in rare cases, deleting data without required parameters, such as nil prices. This implies that the model needs to be trained with consistent data over a certain period of time to be able to predict future price trends. As well, it is ideal to have the same location, size of land, and price. As such, the model can be reproduced and retrained from different locations without confusion. Figure 3 shows the obtained filtered data. The RNN model designed in Figure 4**Error! Reference source not found.** uses LSTM and relies on an activation function using the Adam optimizer with an over fitting or, residential housing,.

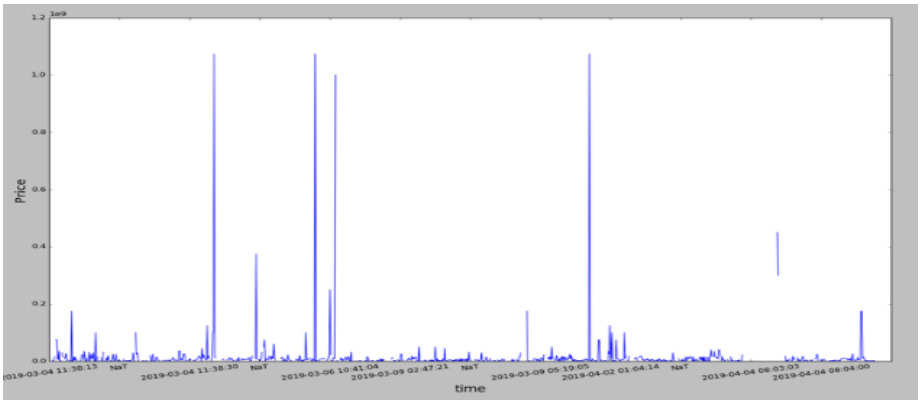



**Figure 3: Unfiltered data.**

Price

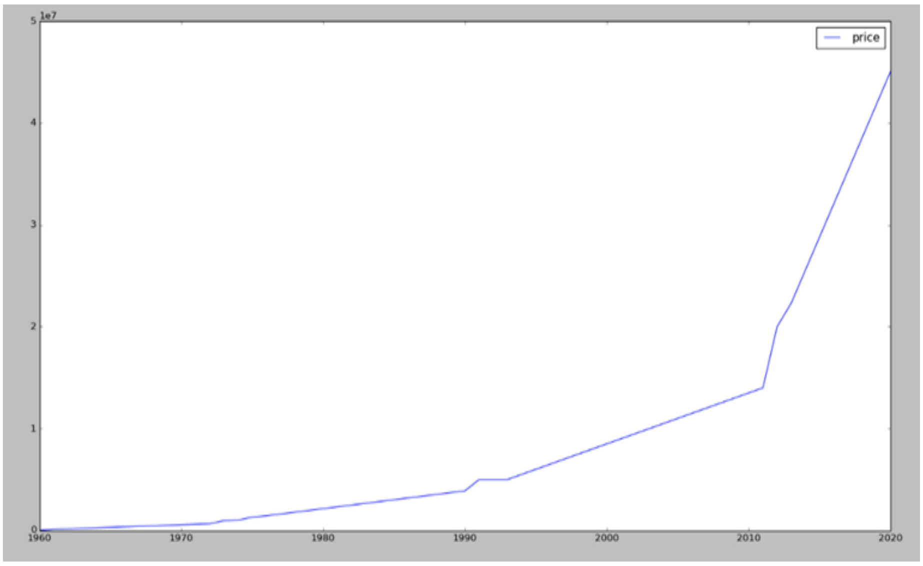



**Figure 4: Filtered data.**

Time (Year)

The RNN model designed in Figure 6 uses LSTM and relies on an activation function using the Adam optimizer with an over-fitting or dropout of 0.2 for fine-tuning the time series sequential data for real estate residential housing and plots for development

```
#Step 3 Building an LSTM -RNN model based on NN
#Definning the model
model = Sequential()

#RNN-LSTM PREDICTION

model.add(LSTM(64, activation='relu', input_shape=(trainX.shape[1], trainX.shape[2]), return_sequences=True))
model.add(LSTM(32, activation='relu', return_sequences=False))
model.add(Dropout(0.2))
model.add(Dense(trainY.shape[1]))
model.compile(optimizer='adam', loss='mse')
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 3, 64)             17408
_________________________________________________________________
lstm_2 (LSTM)                (None, 32)                12416
_________________________________________________________________
dropout_1 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 33
=================================================================
Total params: 29,857
Trainable params: 29,857
Non-trainable params: 0
_________________________________________________________________
```

**Figure 5: Data processing and selection.**

### (b) *RNN- LSTM Model Training*

The upload of prepared data into the model was accomplished by using Time Series data and the TensorFlow library, which is rich in computational and analysis functions, to produce various outputs such as graphs and tables. Next, layers for output were added before compiling the model using the Adam optimizer and MSE loss measure on epochs rounds of 50 cycles as in Figure 5. The numbers were changed based on the best practice of trial and error until the model could fit the data best. In this work, "n" _input refers to the number of steps to predict ahead, and the number is entered in run time as an input representing the number of years that the model should predict the price. "n" _feature is constant, as a model has only one output price, and Figure 6 is a sample code showing the parameters that are adjusted and fixed during training to best predict the number of epochs and MSE loss measure. After training, the model uses prepared data to predict future price trends. This input was set to "n", which means the model predicted the price. For example, plots for sale in particular locations and based on size. Where "n" represents the number of future years that can be predicted.

A price prediction visualization against time, in years, is presented from a training data set collected using the Dalali App. Notably, for future years, the model made price prediction trends. Figure 7 shows the process of plotting price prediction graphs, and Figure 8 shows price prediction results. The blue lines are actual prices, and the orange line is the price prediction from 2022 to 2026.

```
[552]: ##Step 4: Traing the LSTM-RNN Model
       # Fitting the model
       history = model.fit(trainX, trainY, epochs=50, batch_size=16, validation_split=0.3, verbose=1)

Train on 913 samples, validate on 392 samples
Epoch 1/50
913/913 [==============================] - 0s 461us/step - loss: 0.5498 - val_loss: 1.3193
Epoch 2/50
913/913 [==============================] - 0s 439us/step - loss: 0.3781 - val_loss: 1.3243
Epoch 3/50
913/913 [==============================] - 0s 453us/step - loss: 0.3565 - val_loss: 1.3421
Epoch 4/50
913/913 [==============================] - 0s 456us/step - loss: 0.4284 - val_loss: 1.3205
Epoch 5/50
913/913 [==============================] - 0s 454us/step - loss: 0.4718 - val_loss: 1.3235
Epoch 6/50
913/913 [==============================] - 0s 450us/step - loss: 0.3600 - val_loss: 1.3282
Epoch 7/50
913/913 [==============================] - 0s 463us/step - loss: 0.3549 - val_loss: 1.3275
Epoch 8/50
913/913 [==============================] - 0s 464us/step - loss: 0.3422 - val_loss: 1.3299
Epoch 9/50
913/913 [==============================] - 0s 447us/step - loss: 0.3749 - val_loss: 1.3218
```

**Figure 7: Training the model.**

```
##Step 5: Make Predictions or Forecasting the future data
#I am starting with the last year in training date and predict future...
n_future=5  #Redefining n_future to extend prediction dates beyond original n_future dates...
forecast_period_dates = pd.date_range(list(train_dates)[-1], periods=n_future, freq='1y').tolist()
#Start actual forecasting
forecast = model.predict(trainX[-n_future:])
```

**Figure 6: Prediction process.**

```
##Step 6: Plotting the Prediction Visualization
# Convert timestamp to date
forecast_dates = []
for time_i in forecast_period_dates:
    forecast_dates.append(time_i.date())

df_forecast = pd.DataFrame({'action_date':np.array(forecast_dates), 'price':y_pred_future})
df_forecast['action_date']=pd.to_datetime(df_forecast['action_date'])
original = df[['action_date', 'price']]
original['action_date']=pd.to_datetime(original['action_date'])
#original = original.loc[original['action_date'] >= '2019-6-1']
plt.ticklabel_format(useOffset=False)
plt.rcParams['figure.figsize']=16,5
sns.lineplot(original['action_date'], original['price'])
sns.lineplot(df_forecast['action_date'], df_forecast['price'])
plt.gcf().axes[0].yaxis.get_major_formatter().set_scientific(False)
```

**Figure 7. Printing predictions.**

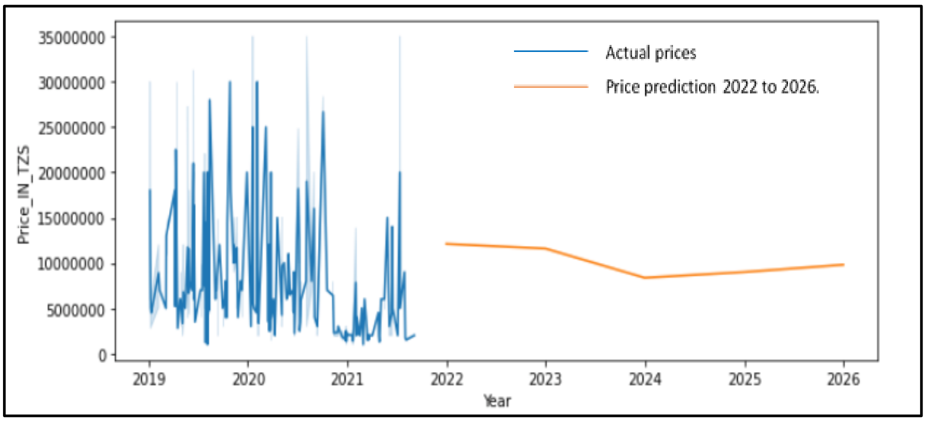



**Figure 8: Prediction results for middle class residential plots.**

## Compilation of RNN-LSTM Model

The next stage was the model compilation by running an efficient TensorFlow library numeric and establishing model computation readiness. The TensorFlow Library optimally represents the network for training and prediction on the CPU (Chauhan *et al.*, 2018). Training a network means finding the best set of weights to map inputs to outputs in the specified dataset. During compilation, some additional properties required for training the network were specified. However, the following loss function was used to evaluate a set of weights:

$$\mathbf{model.compile}\left(\mathbf{optimizer} = '\ \mathbf{adam}'\ , \mathbf{loss} = '\ \mathbf{mse}'\ \right),$$

the optimizer was used to search through different weights for the network and any optional metrics the model would like to collect and report during the training.

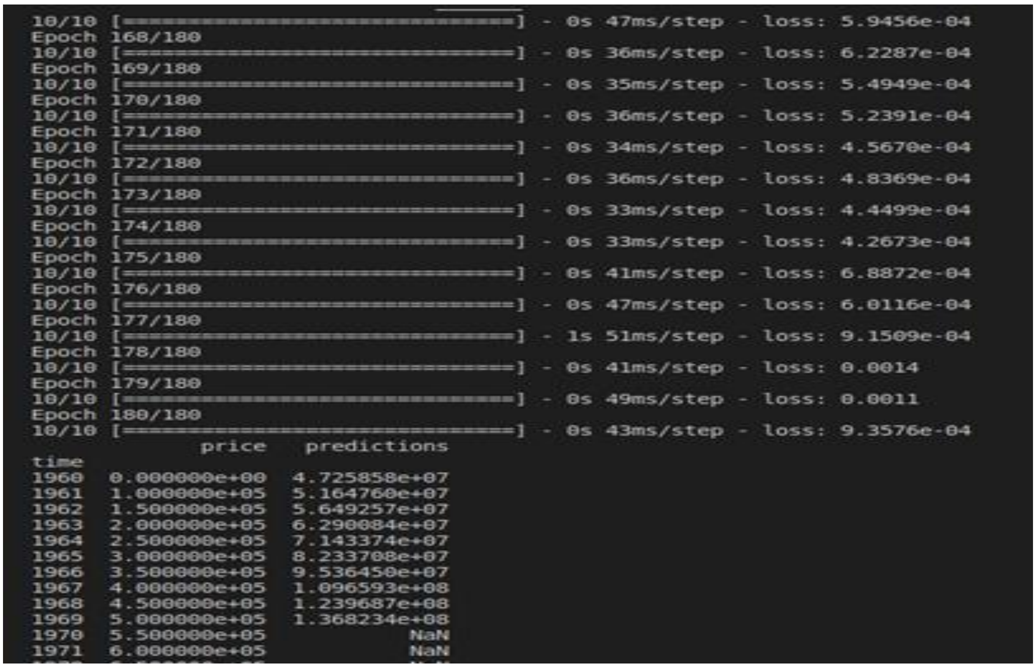
Model readiness for efficient computation was achieved using the TensorFlow library to enable the model execute specified data from the survey tool to allow training or fitting the model on loaded data by calling the fit function on the model. Model training occurs over epochs (one passes through all the rows in the training dataset), and each epoch is split into batches that contain one or more samples considered by the model within an epoch before weights are updated. Therefore, one epoch is composed of one or more batches, based on the chosen batch size, and the model is fit for many epochs.

## Creating model training abilities

The algorithm in the TensorFlow library automatically assigns abilities that facilitate the training process. The process runs for a fixed number of iterations through the dataset called epochs. The number of dataset rows was set at 18. This was considered before the model weights are updated within each epoch, called the batch size, and set using the batch size argument. This study applied 180 epochs and used a batch size of 100. These parameters can be chosen experimentally by trial and error. We want to train the model so that it learns a good (or good enough) mapping of rows of input data to the output classification with the minimum error possible after model convergence.

**Running the model and evaluating the loss**

After completion of model training, the process is run to see the results and evaluate the loss. However, the results of the model can be regulated by adjusting the values in the compiling function of the model. For instance, where the model runs at an MSE loss of 9.3576e-04, which is small and good for an LSTM model in training for 180 epochs to predict the price for plots in 10 years, the new prediction data frame prints developed to display a list of future prices in the next 10 years are shown in Figure 9.



```
10/10 [==============================] - 0s 47ms/step - loss: 5.9456e-04
Epoch 168/180
10/10 [==============================] - 0s 36ms/step - loss: 6.2287e-04
Epoch 169/180
10/10 [==============================] - 0s 35ms/step - loss: 5.4949e-04
Epoch 170/180
10/10 [==============================] - 0s 36ms/step - loss: 5.2391e-04
Epoch 171/180
10/10 [==============================] - 0s 34ms/step - loss: 4.5670e-04
Epoch 172/180
10/10 [==============================] - 0s 36ms/step - loss: 4.8369e-04
Epoch 173/180
10/10 [==============================] - 0s 33ms/step - loss: 4.4499e-04
Epoch 174/180
10/10 [==============================] - 0s 33ms/step - loss: 4.2673e-04
Epoch 175/180
10/10 [==============================] - 0s 41ms/step - loss: 6.8872e-04
Epoch 176/180
10/10 [==============================] - 0s 47ms/step - loss: 6.0116e-04
Epoch 177/180
10/10 [==============================] - 1s 51ms/step - loss: 9.1509e-04
Epoch 178/180
10/10 [==============================] - 0s 41ms/step - loss: 0.0014
Epoch 179/180
10/10 [==============================] - 0s 49ms/step - loss: 0.0011
Epoch 180/180
10/10 [==============================] - 0s 43ms/step - loss: 9.3576e-04
             price   predictions
time
1960  0.000000e+00  4.725858e+07
1961  1.000000e+05  5.164760e+07
1962  1.500000e+05  5.649257e+07
1963  2.000000e+05  6.290084e+07
1964  2.500000e+05  7.143374e+07
1965  3.000000e+05  8.233708e+07
1966  3.500000e+05  9.536450e+07
1967  4.000000e+05  1.096593e+08
1968  4.500000e+05  1.239687e+08
1969  5.000000e+05  1.368234e+08
1970  5.500000e+05           NaN
1971  6.000000e+05           NaN
```

**Figure 12: Model output.**

Figure 10 shows the procedures adhered to in building the user interface for training as well as data upload in the designed model. The diagram acts as a strategic design plan that gives a designer a clear picture of the process and how data flows from upload flow (user input) to model training and prediction.

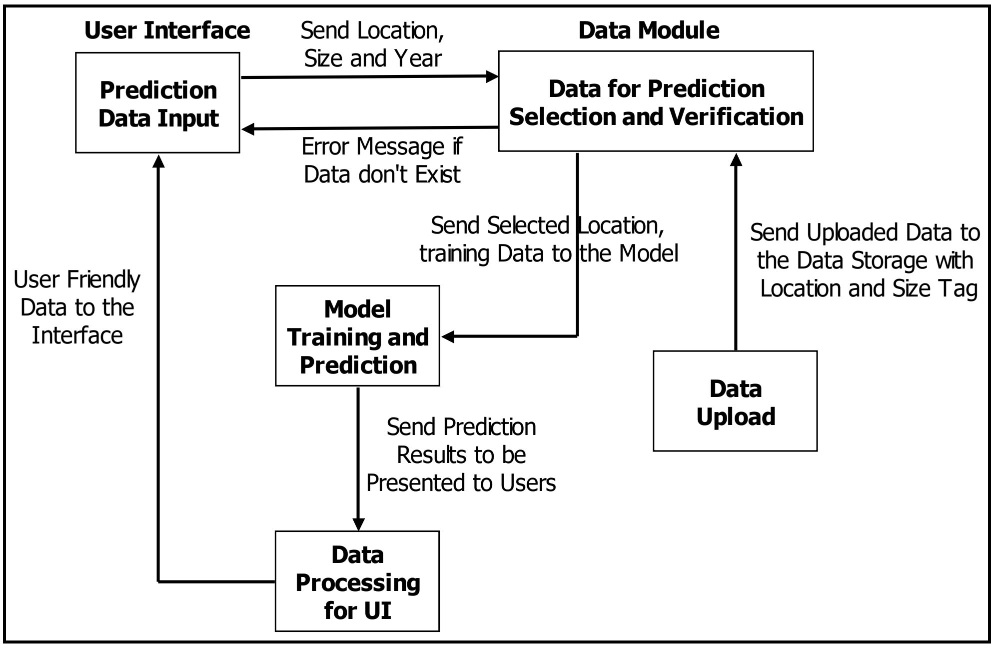



**Figure 10: Process diagram.**

## RESULTS AND DISCUSSION

The study evaluated the performance of the proposed RNN-LSTM model, and the results were compared to the CNN model in terms of training and prediction process and loss. Figure 15 and Figure 16 show prediction and the loss when using CNN and RNN-LSTM models, respectively. In this work, the term "loss" refers to the difference between the predicted price and the actual price. When both models are trained using the same data and number of epochs, the RNN attains a lower MSE compared to the CNN. That is, the MSE for the RNN-LSTM model is 50% less than that of the CNN model, as shown in Table 2. In addition, it is observed that the longer the time, the higher the price. The RNN-LSTM model requires more past data for training to provide a more accurate price trend prediction compared to the CNN model.

**Table 1: Mean square error between CNN and RNN-LSTM**

| Model | MSE | 10th Year Price (TZS) |
|---|---|---|
| CNN | 0.0026 | 3.270511e+08 |
| RNN-LSTM | 0.0013 | 1.258425e+08 |

(a)

(b)

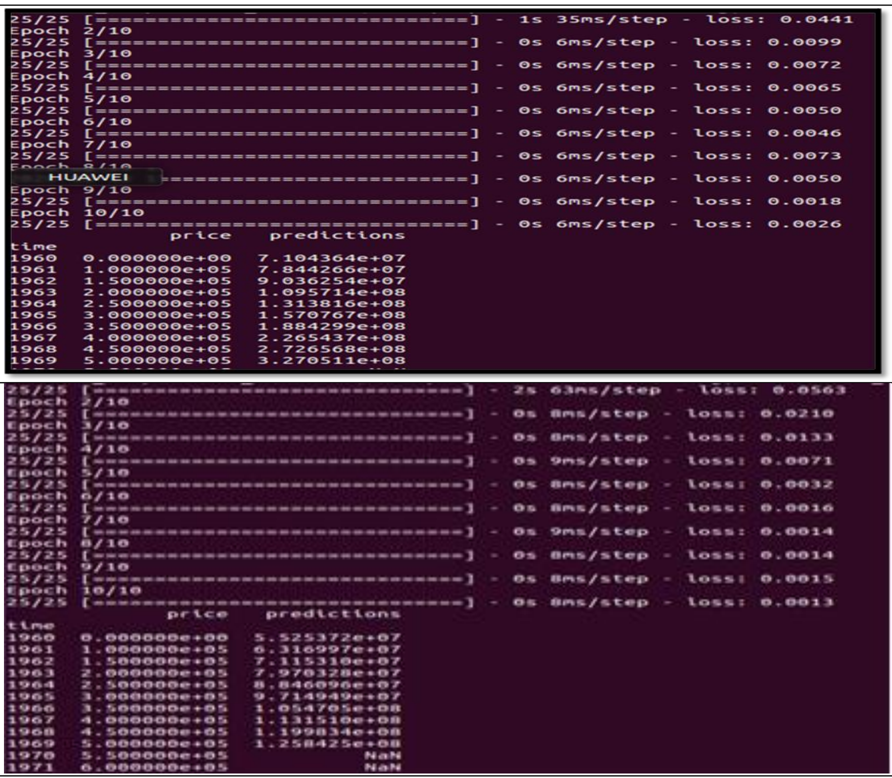



**Figure 11. Model training and prediction process (a) CNN, (b) RNN.**

The effectiveness of RNN-LSTM in the time series price trend prediction test considered high-class, middle-class, and low-class residents for a 400 sqm area from 2022 to 2026. Results show that the average prediction accuracy is 76.8% when extensive data in relation to coverage per year is used to train and validate the model. The model used four key parameters to compute price trends: current price, time, location status, and size. In computing the price trend per location, the model prediction accuracy was 97.45%, 79.23%, and 53.8% for the high, middle, and low classes, respectively. Essentially, the variation in percentage price prediction accuracy is due to the number of epochs used during the training of the model, the quality, and the amount of data set. According to Anani & Samarabandu (2018), RNN-LSTM with large datasets and a short time of training results in an accuracy of 98% to 99%.

It is observed that CNN does not require massive previous data for prediction accuracy compared to the RNN-LSTM model. Nevertheless, RNN-LSTM can perform better in time series-related predictions with the lowest MSE of 0.0013 (50% less) compared to the lowest MSE of 0.0026 for CNN. It is also noted that the RNN-LSTM with a higher training time and a number of epochs above 100 gives lower accuracy predictions compared to the RNN-LSTM under identical conditions. However, the lack of reliable and sufficient information or limited information about the market has proved to be a major challenge in predicting price trends in real estate and plots for development in Dar es Salaam city. This was echoed by Anani et al. (2018), who urge that data and timeframe models such as RNN-LSTM can accurately predict price trends in real estate. This confirms the RNN model theory as presented by Hüsken & Stagge (2003), which shows the relationship between past and present data in decision-making. The study highlighted that RNN-LSTM has a lower MSE compared to CNN.

## CONCLUSION

According to the literature, forecasting real estate prices in growing cities has a direct impact on the development of the real estate sector. This study proposes an artificial intelligence-based prediction model for forecasting real estate prices in growing cities. The study uses price, location, size, and time as the main determinants in predicting the future of real estate, residential housing, and plots for development in Dar es Salaam, Tanzania. The used determinants are context-based and were derived using Dar es Salaam related datasets. The implementation, training, and validation of an RNN-LSTM model to predict prices were fairly conducted, and the results confirm that the RNN-LSTM-based model works on the principle of saving the output of a layer and feeding this back to the input to predict the outcome of the layer. However, unreliable and limited data affect the accuracy of the RNN-LSTM prediction model. The study used limited data collection from the Dalali App, which has been in use since 2018. As such, it was questionable to rely on data collected in six (2022-2026) years to train the designed model and make reliable predictions. This means that the ability of the model to generate accurate predictions was limited. However, the accuracy level of the RNN-based prediction model improves when using data spanning 10 years or more.

## REFERENCES


Anani W. and Samarabandu J. (2018, May). Comparison of recurrent neural network algorithms for intrusion detection based on predicting packet sequences. In 2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE) (pp. 1-4). IEEE.

Bello M.Z., Sanni L.M. and Mohammed J.K. (2020). Conventional methods in housing market analysis: a review of literature. Baltic Journal of Real Estate Economics and Construction Management. ISSN: 2255-9671 (online) 2020, **8**, 227–241. https://doi.org/10.2478/bjreecm-2020-0016

Chauhan N., Bhatt A.K., Dwivedi R.K. and Belwal R. (2018). Accuracy Testing of Data Classification using Tensor Flow a Python Framework in ANN Designing. 2018 International Conference on System Modeling & Advancement in Research Trends (SMART). doi:10.1109/sysmart.2018.8746945

Choy LHT and Ho WKO 2023 The Use of Machine Learning in Real Estate Research. *Land*. **12**(4).

Gardner D.A.V.I.D., Lockwood K.E.I.T.H. and Pienaar J.A.C.U.S. (2019). Assessing Rwanda's Affordable Housing Sector. Centre for Affordable Housing Finance in Africa (CAHF): Gauteng, South Africa, 58.

Geng L, Ran Q, Zhang G, and Zhou D 2023 Multi-feature house price trend prediction based on machine learning (January): **82**.

Gower A. (2022). Understanding Class A vs. B vs. C Multi Family Real Estate. Visited on December 2022 online from https://gowercrowd.com/real-estate-investing/class-a-b-c-properties

Hüsken M. and Stagge P. (2003). Recurrent neural networks for time series classification. Neurocomputing, **50**, 223–235. doi:10.1016/s0925-2312(01)00706-8

Jud G.D. and Frew J. (1986). Real Estate Brokers, Housing Prices, and the Demand for Housing. Urban Studies, **23**(1), 21–31. doi:10.1080/00420988620080031

Kong W, Dong ZY, Jia Y, Hill DJ, Xu Y, and Zhang Y 2019 Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network. *IEEE Transactions on Smart Grid*. **10**(1): 841–851.

Magina F.B. (2016). Housing market in redeveloping inner-city areas in Dar Es Salaam: Supply dynamics and their effects on the urban fabric (Doctoral dissertation, Faculty of Spatial Planning, TU Dortmund University, Germany).

Martins A.M., Serra A.P. and Martins F.V. (2016). Real estate market risk in bank stock returns: evidence for 15

European countries. International Journal of Strategic Property Management, **20**(2). https://doi.org/10.3846/1648715X.2015.1120790

Mukhopadhayay I., Polle S. and Naskar P. (2014). Simulation of denial of service (DoS) attack using matlab and xilinx. IOSR Journal of Computer Engineering (IOSR-JCE), e-ISSN: 2278-0661, p- ISSN: 2278-8727, **16**(3), Ver. IV (May-Jun), PP 119-125.

Nilsson P. (2019). Prediction of residential real estate selling prices using neural networks. Master in Computer Science Dissertation, KTH Royal Institute of Technology

Sampathkumar V., Santhi M.H., and Vanjinathan J. (2015). Forecasting the Land Price Using Statistical and Neural Network Software. Procedia Computer Science, **57**, 112–121. https://doi.org/10.1016/j.procs.2015.07.377

Wang L., Chan F.F., Wang Y. and Chang Q. (2017). Predicting public housing prices using delayed neural networks. IEEE Region 10 Annual International Conference, Proceedings/TENCON, 3589–3592. https://doi.org/10.1109/TENCON.2016.7848726

Wang J.J., Hu S.G., Zhan X.T., Luo Q., Yu Q., Liu Z., … Liu Y. (2018). Predicting House Price with a Memristor-Based Artificial Neural Network. IEEE Access, **6**, 16523–16528. https://doi.org/10.1109/ACCESS.2018.2814065