Enhancing Heart Disease Prediction:

A Comparative Analysis of ML and DL Approaches

1\*Rishi Mishra, 1Shubham Shukla ,1Rohitash Kumar and 2Dr. P. Karuppanan

*\*1, 2 Department of Electronics and Communication Engineering, Motilal Nehru National Institute of Technology, Prayagraj-211004*

Email: [12rishi2003@gmail.com](mailto:12rishi2003@gmail.com) , shubham20215123@gmail.com , [rohit77428@gmail.com](mailto:rohit77428@gmail.com) ;[pkaru@mnnit.ac.in](mailto:pkaru@mnnit.ac.in);

***Abstract*— The increase in cardiovascular disease worldwide necessitates the development of effective predictive tools to facilitate early detection and intervention. This research focuses on the use of ml to develop predictive models for heart diseases, focusing on the analysis of patient history data to identify event alerts and improve accurate diagnosis. The research used a variety of advanced models, including logistic regression, neural networks and random forest algorithm. The models are selected based on their performance in processing complex data and their ability to generate meaningful content. The importance of these methods is their ability to increase the accuracy of facts and help create personalized treatment strategies in the medical center. This study also includes various analyses to check the result of the model on various variables and data. The aim is to identify the most effective algorithm for predicting cardiovascular diseases and to provide insights into model optimization for future use.**

***Keywords—******neural network, machine learning, heart disease, deep learning***

1. INTRODUCTION

Heart related diseases continue to be a leading cause of death worldwide, spurring widespread research into predictive models to aid in early detection and personalized treatment strategies [1-2]. As the incidence of heart disease continues to increase worldwide due to an aging population, sedentary lifestyle, and increasing prevalence of obesity and diabetes, better diagnostic tools are needed. Traditional procedures such as physical exams, electrocardiograms (ECGs), and imaging often require significant resources and may not be completed in a timely or accurate manner. This has increased interest in developing predictive models that can better analyze patient data and provide recommendations to physicians. A new method for predicting cardiovascular disease, by analyzing big data, these methods can uncover patterns and relationships that traditional methods cannot. Machine learning models, especially those that use deep learning [3], hold great promise for improving the accuracy of medical diagnoses and aiding medical decisions. Researchers are using data-driven algorithms to improve patient outcomes through earlier detection and better treatment[4-5]. Keynote speaker Dr. William B. Kannell, 1960. The models uses factors such as sex, cholesterol levels, age and blood pressure to predict year heart disease [6]. Although the Framingham risk score is an important risk indicator, its predictive accuracy is limited by its reliance on linear assumptions. As logistic regression

[7] became more popular in the early 2000s, people began to turn to more sophisticated methods. For example, the ASCVD (atherosclerotic cardiovascular disease) risk calculator developed by DâAgostino et al. (2008) performed a more detailed analysis by incorporating additional risk factors based on the Framingham model. However, even this advanced model is inherently problematic and requires more clinical data [8-9]. Researchers have begun exploring algorithms such as SVM, decision trees and ensemble systems such as random forests [10]. Farias et al. (2016) demonstrated the effectiveness of random forests in classifying cardiac patients using multiple variables. Although these models are more accurate and robust, they also present issues related to interpretation and computational complexity. Neural Networks have gained importance due to their ability to process big data and diagnose comp lex patterns [10]. For example, Rajkumar et al. (2018) used deep learning to analyse health records (EHRs) and predict heart failure with high accuracy. Despite this progress, deep learning models still require significant computing resources, and their decision-making processes are often opaque. There are limitations. For example, Zhang et al. (2021) proposed a hybrid model that combines computational techniques with advanced machine learning to improve performance [11]. Similarly, Kumar et al. (2020) Research collaboration to develop an accurate and robust model. These efforts represent progress but need to address issues such as data quality, sample interpretation, and generalization. Many models suffer from poor data quality, including missing values, inconsistencies, and biases [12-14]. Interpreting machine learning models also remains challenging because doctors must trust and understand the body’s predictions. Getting models to work across populations and settings is another ongoing challenge, as training models on certain data may not perform well in clinical settings. To overcome these issues, machine learning techniques are being used to assess the quality of models for predicting heart disease. Research goals include improving prediction accuracy, improving interpretation, improving consistency, and making comparisons.

This study aims to achieve these goals by developing an accurate and effective global heart disease assessment tool that will guide clinical decisions and improve patient pain. It provides important information for early intervention and improves the ability to diagnose heart disease by quickly and accurately analyzing patient data. The program is working to improve the accuracy of risk assessment and enable physicians to self-regulate using ml to predict cardio disease. A growing number of studies are using machine learning to improve health outcomes. Future studies will focus on integrating additional data, developing predictive methods, and evaluating the impact of the model on patient care in a real-life clinical setting.

1. STUDY OF ML ALGORITHMS

This is part of AI that specializes in developing models/algorithms and statistical models that permit computers to function without programming. Fig.1 shows the process of implementing the machine learning for the given problem. Within two steps first we are gathering and preprocessing the data and then choosing the one algo for training. As generation advances, machine learning is getting used in many regions, such as healthcare, finance, commercial enterprise, and private management, converting the way automated obligations are accomplished and improving decision-making procedures. This includes many numerous theories and algorithms, each suited to various types of data and problems.

Optimization of model



Collection

Preprocessing

Exploration

Training Machine

Fig. 1 Proposed Workflow for Models

1. *Logistic Regression (LR)*

Logistic regression is a type of supervised learning technique used to estimate likelihood of a specific outcome or event by modelling binary data distributions. LR examines the relationship between one or more independent variables and randomly partitions the data. It utilizes the sigmoid function to map predictor variables to probabilities. This function produces an S-shaped curve that represents values between 0 and 1 as shown in Fig. 2

Threshold Value = 0.5

0

1

Target Probability y

Input Parameter x

Fig. 2 Logistic Regression

The logistic function also said the Sigmoid function used in the algorithm is given as

Where z is the input parameter. For multiple independent variables function can be written as

Where βare the constants and z are the independent parameters.

1. *K-Nearest Neighbours (KNN)*

The k-NN method is a direct and intuitive method used for tasks like classification and regression. It can be adapted to handle data such as images by adjusting the parameters to suit the specific characteristics of the input.

It is very widely used in real life because it does not make any guesses about the data distribution (unlike other algorithms that assume a Gaussian distribution, such as GMM, which provides fixed data). Training data is given that divides the joint into groups defined by attributes. If we arrange these points on an image, we can find some groups or categories. To classify an unknown point, we can assign it to a group on the basis of classification of its neighbours that are most near to it. For example, if the closest points to the unclassified point belong to a group labelled as "red," the new point is more likely to be classified as "red.", as shown in Fig 3.

Parameter x1

Parameter x2

Cluster A

Cluster B

Fig. 2 LSTM Architecture

Fig. 2 A closer look at the LSTM architecture. The model cell (Ct ) of the LSTM records the state value of the neuron at time t, while selecting information from the previous process though the gate when a neural network with a sigmoid function is used. LSTM achieves this by using three gates, namely the forget fate (ft ), the input gate (it ) and the output gate mechanism. The memory gate allows the sigmoid function to be used to decide whether to keep or delete the previous solution in the state of previous cell. The output of this date is given as in eqn1, where σ is a random function (such as logistic sigmoid or hyperbolic tangent), ht−1 is the hidden state at the previous time, Xt is the input at time t, Wf represents the weight matrix, and bf represents the time bias.

(1)

The input gate checks whether the new data will be added to the LSTM memory. This gate gas two layers, a sigmoid layer that determines values to be changed, and a tanh layer that crates a vector of candid values that will be added to the LSTM Memory. The Equations mentioned are for the outputs of the two layers that are eq2 and eq3:

) (2)

(3)

The output is integrated as follows in eq4:

(4)

Output gate determines what information from the current cell should contribute for output and what is propagated to the next hidden layer. It is implemented using sigmoid layer to filter the information and a tanh function to scale the output values between -1 and 1. The output gate (Ot​) and the hidden cell (ht​) at time step t are calculated using equations eq5 and eq6:

(5)

(6)

The output from the LSTM cell is a refined version of the cell state, ensuring that only the most pertinent information is passed to the next layer in the network. The result of this operation varies between 0 and 1, where 0 means that all charges are removed and 1 means that they are retained. LSTM architecture can be improved by using a Bi- directional LSTM called Bi-LSTM as shown in Fig. 3, LSTM can also be paired with an Attention Model to create a LSTM-DL Model that perceives the human brain. Similarly, in AM, weighting is defined as prioritizing important information and ignoring irrelevant information.

h1

ht-1

ht

h’1

h’t-1

h’t

Output layer

Forward LSTM

Reverse LSTM

Hidden layer

Input layer

Output Layer

Fig. 3 Bi-LSTM Architecture

***2.2.*** ***Vector AutoRegression Model (VAR)***

Vector Autoregressive (VAR) models are powerful and flexible tools for analysing multivariate time series data. VAR models are an extension of univariate autoregressive models to capture the dynamic relationship among multiple time series data. They are readily used in economics and finance to forecast and analysing the interdependencies among time series variables. Fig. 4 illustrates the structure of a basic VAR model. Each variable in the system is modelled as a linear function of its own past value and the VAR model captures feedbacks and interactions at different points in time.

VAR models can be used for forecasting future values of the time series. Forecasts are computed using recursive substitution, and forecast errors are derived from the innovations process in eq11:

​ (7)

Y1,t-p

,t

Y1,t-2

Y1,t

,t

Y2,t-p

,t

Y2,t-2

,t

Y2,t-1

Y1,t-1

,t

Y3,t-2

Y3,t-p

Y2,t

Y3,t

Y3,t-1

Fig. 4 VAR Architecture

This is a form of a VAR model with p lags, denoted as VAR(p), for a vector of n series data Yt = (y1t, y2t, ..., ynt)′

is given by eqn7:

(8)

where t=1…T , Yt​ is a n×1 vector for multivariant time series variables. c is also a n×1 vector of intercepts. Πi are n×n coefficient matrices , ϵt​ is an n×1 vector of white noise errors with almost zero mean error and constant variance. VAR models are stable if all the roots given by the characteristic equation lie outside the unit circle. For a stationary VAR(p) model, the unconditional mean of the time series can be computed using the following formula in eqn8:

μ=(In​−Π1​−Π2​−⋯−Πp​)−1 c (9)

Estimation of a VAR model is usually performed using the Ordinary Least Squares (OLS) method equation by equation because each equation in the VAR model has the same set of regressors. Choosing the appropriate lag length p is crucial for model accuracy. selection criteria like the Akaike Information Criterion (AIC), Schwarz-Bayesian Criterion (BIC), and Hannan-Quinn Criterion (HQ) are often used: in eq9 and eq10, here ln ∣Σ(p)’∣ is the residual covariance matrix without the degrees of freedom correction from a VAR(p) model.

AIC(p)=ln ∣Σ(p)’∣+(2pn2​)/T, (10)

BIC(p)=ln ∣Σ(p)’∣+(lnT⋅pn2​)/2 (11)

The model for exchange rates, such as the US/Canadian dollar exchange rate and the forward premium, demonstrates the use of VAR models in analysing financial time series data. The model captures the dynamic relationship between spot returns and interest rate differentials, and the results are interpreted using the Granger causality tests, the impulse response functions, and for variance decompositions. They provide insights for the dynamic relationships among variables and are valuable tools for economic policy analysis, financial market studies, and various applications requiring multivariate time series analysis.

***2.3.*** ***Block Resampling Method***

Forecasting presents a challenge when extreme events are infrequent, creating an imbalance in the dataset. In response, block resampling techniques have been introduced to address the imbalance problem by adjusting how data is sampled and presented to forecasting models. The approach leverages a novel method of block resampling, where both predictor variables and forecast values are grouped into "blocks." Instead of treating individual data points independently, the entire multivariate time series is divided into moving blocks of observations, with each block containing both input (predictor variables) and output (forecast values). These blocks are categorized into extreme event blocks (EE) and normal event blocks (NE) based on the nature of the forecast within each block. For example, an extreme event block contains a forecast value corresponding to an extreme observation, while a normal event block is associated with a regular observation. This categorization allows selective resampling, where extreme event blocks are oversampled to increase their representation in the training dataset, and normal event blocks may be under sampled to prevent bias toward normal observations.

To address data imbalance, the EE JTM blocks are oversampled, while the NE JTM blocks are under sampled, resulting in a balanced joint time model (JTM) that better represents both extreme and normal events in the dataset. This balanced training set is then used to train the forecasting model. Simultaneously, the regular portion of the data Z(100−σ)​ is processed through similar transformations, including normalization and block creation. The trained model is applied to these blocks, producing forecasts that are subsequently inverse normalized.

The performance of the model is evaluated using metrics such as Root Mean Square Error (RMSE) and Classification Accuracy (CA), ensuring that both the precision of extreme event prediction and overall forecasting accuracy are considered. This architecture effectively improves the model's ability to forecast extreme events by balancing the dataset and maintaining temporal dependencies within the multivariate time series.

The diagram in Fig. 5 illustrates the architecture of the Block Resampling Method for improving forecasting accuracy, especially in scenarios involving extreme events. The process begins with the multivariate time series data ZZZ, which is divided into two parts: Zσ​, representing the subset containing extreme events, and Z(100−σ)​, which contains regular observations. Both subsets undergo a Block Classification Transformation (BCT) to extract features relevant to their respective characteristics. These transformed blocks are then normalized and further split into "Predictor-Forecast" (PF) blocks, where δ+1 denotes the number of lagged observations considered in each block. In the joint predictor-forecast (JPF) space, the PF blocks are divided into extreme event joint time models (EE JTM) and normal event joint time models (NE JTM), based on the presence of extreme or normal observations in the forecast variable.

Split

(𝛿 +1)  
EEJTM

Oversample

Undersample

Normalize

𝛿   
Blocks

Trained   
Model

Inverse  
Normalise

Performance  
Measure

RSME

CA

Balanced  
JTM

(𝛿 +1)  
PF Blocks

Z0

Normalize

BCT

JTMM

(𝛿+1)  
NEJTM

Z

Split

Balanced  
Training Set

Trained  
Model

BCT

z̄

IBCT

Z(100 - 0)

Fig 5. Architecture of Block Resampling Method

To address data imbalance, the EE JTM blocks are oversampled, while the NE JTM blocks are under sampled, resulting in a balanced joint time model (JTM) that better represents both extreme and normal events in the dataset. This balanced training set is then used to train the forecasting model. Simultaneously, the regular portion of the data Z(100−σ)​ is processed through similar transformations, including normalization and block creation. The trained model is applied to these blocks, producing forecasts that are subsequently inverse normalized. The performance of the model is evaluated using metrics such as RMSE) and Classification Accuracy (CA), ensuring that both the precision of extreme event prediction and overall forecasting accuracy are considered. This architecture effectively improves the model's ability to forecast extreme events by balancing the dataset and maintaining temporal dependencies within the multivariate time series.

To address the imbalance in the dataset, extreme event blocks (denoted B) are oversampled. Mathematically, the oversampling process can be described as creating multiple copies of the extreme event blocks in eqn11 and 12:

(11)

(12)

Mathematically the goal is to predict the future values of the time series based on lagged predictors. Consider a multivariate time series ZT=(z1,z2,…,zT) where each zt represents a vector of observations at time t. The goal is to forecast the value ZT+h​, where h is the forecast horizon, based on the past δ lagged observations (zT,zT−1,…,zT−δ+1) This can be expressed as:

ZT+h​ = (ZT, ZT−1, …,ZT−δ+1 ) (13)

**3. SIMULATION RESULTS AND ANALYSIS**

* 1. Evaluation Parameters

The performance of multivariate time series models is evaluated using several metrics to ensure robust predictions. Commonly used evaluation metrics include: Mean Absolute Error (MAE): Measures the average size of errors in a series of predictions, regardless their direction. It also provides insight into how far off predictions are from actual values. Root Mean Squared Error (RMSE): RMSE represents the square root mean error. It is used to evaluate a model’s performance where deviation from true value is given by large error values. Mean Absolute Percentage Error (MAPE): MAPE measures the percentage error for the predictions and actual values. It is used to track the relative accuracy in a model. R-squared (R²): R-squared provides the proportion of variance in the dependent variable that is predictable from the independent variables.

**3.2** **Deep Learning Algorithms Analysis:**

The dataset used in this study is sourced from Kaggle.com and focuses on the Superstore Sales Prediction dataset. The utilized libraries include NumPy, Pandas, Matplotlib, and Seaborn. Various exploratory data analysis techniques are employed, using plots such as distplot, histplot, boxplot, countplot, and violin plot to understand the data distribution and relationships. The model incorporates three different forecasting methods: LSTM, VAR, and Block Resampling. These methods are evaluated based on metrics such as RMSE, MAE, and MAPE to determine their effectiveness in predicting future sales trends and patterns in the dataset. This dataset represents sales data and provides multiple variables that can be used for different time periods. Key attributes include order ID, order date, delivery date, customer information, sales, quantity, discount, and revenue. These capture important details of each transaction, such as buyer behaviour, product details, and financial performance. For different forecast periods, a set of these attributes, specifically release date, sales, quantity, discount, and margin, can be used to model and predict future outcomes.

**3.2.1** **Long Short-Term Memory Analysis**:

The LSTM model applied to the Superstore Sales Prediction dataset shows a moderate predictive performance based on the evaluation metrics. The MSE is 30205.84, and the RMSE is 56.62, indicating a higher magnitude of error compared to the VAR model. The MAE is 47.24, reflecting a slightly greater average deviation from the actual sales values. The MAPE of 16.5% symbolises that the model’s predictions vary by 24.49% from actual value. The R² error value of 0.76 shows that 76% of the variance in sales data is explained by the model, indicating a lower degree of accuracy.

Fig. 6 (a) shows the residuals, giving insight into areas where the model's predictions differ more significantly from the actual data. These visualizations help better understand the model's performance in predicting sales trends.

The performance of the LSTM model can also be visualized through error distribution and time series prediction plots. Fig. 6 (b) compares actual and predicted sales values over time, highlighting how well the LSTM model captures sales patterns.

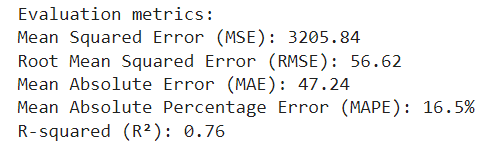


Fig. 6 (a) Simulation Result of LSTM Model

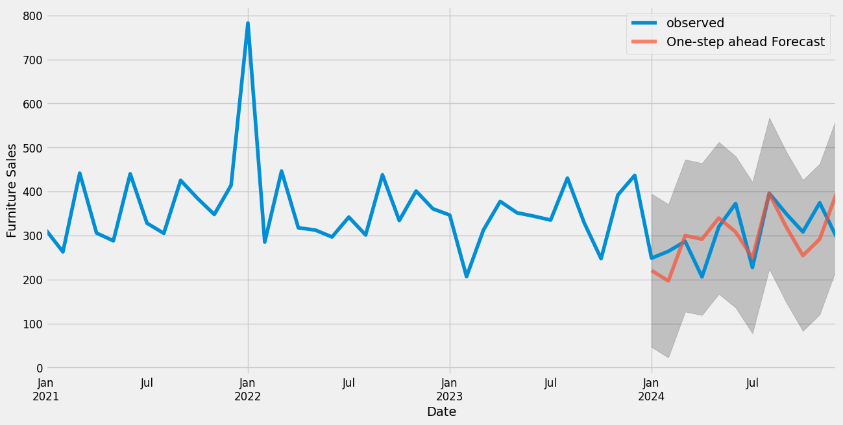


Fig. 6 (b) Output of LSTM Model

* + 1. VAR Auto-Regressive Model:

**VAR** model applied to the Superstore Sales Prediction dataset indicates a need for improvement in predictive performance. The **MSE** stands at **3438.98**, with a correspond in **RMSE** of **58.64**, showing a higher magnitude of error in the predictions. The **MAE** of **52.22** suggests a considerable average deviation from the actual sales values. **MAPE** of VAR model is **18.21%.** Lastly, the **R² error** value of **-0.01** implies that the model performs poorly, failing to explain the variance in the sales data effectively.

The residuals plot in Fig. 7 (a) illustrates the errors across the dataset, highlighting areas where the model tends to underperform or overestimate. These visualizations complement the evaluation metrics, offering deeper insights into the model's strengths and limitations in predicting sales patterns. The performance of the VAR model can be visualized through error distribution and time series prediction plots. Fig. 7 (b) shows the comparison between the actual Vs predicted sales values over time, providing a clear picture of how well the model tracks the observed data.

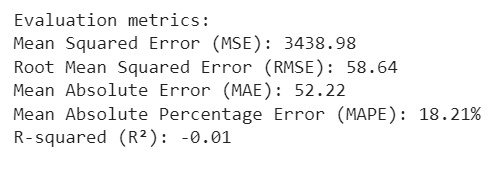


Fig. 7 (a) Simulation Result of VAR Model

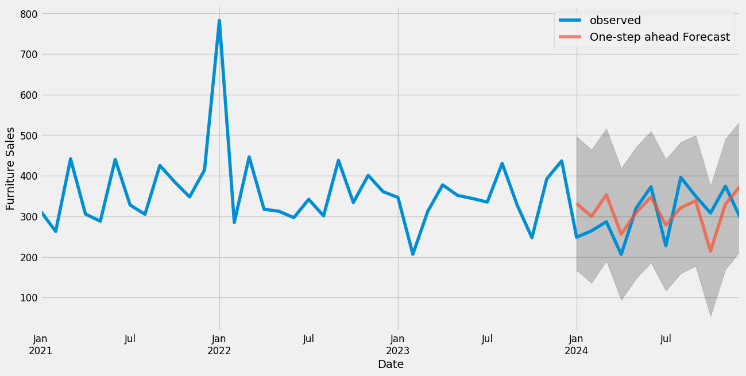


Fig. 7 (b) Outcome of VAR Model

* + 1. **Block Resampling Method Analysis:**

The **block resampling technique** applied to the Superstore Sales Prediction dataset indicates solid predictive performance. The **MSE** stands at **695.44**, with a corresponding **RMSE** of **26.37**, showing the magnitude of error in the predictions. The **MAE** of **23.69** suggests a relatively low average deviation from the actual sales values. The **MAPE** for block resampling method come out to be **8.38%.** Lastly, the **R² error** value of **0.8** implies that **80% of the variance** in the sales data is explained by the model, demonstrating a good level of accuracy.

The residuals plot in Fig. 8 (a) illustrates the errors across the dataset, highlighting areas where the model tends to underperform or overestimate. These visualizations complement the evaluation metrics, offering deeper insights into the model's strengths and limitations in predicting sales patterns. The performance of the block resampling technique can be visualized through error distribution and time series prediction plots. Fig. 8 (b) shows the comparison between the actual and predicted sales values over time, providing a clear picture of how well the model tracks the observed data.

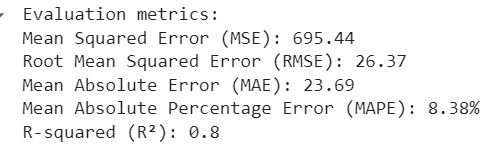


Fig. 8 (a). Simulation Result of Block Method

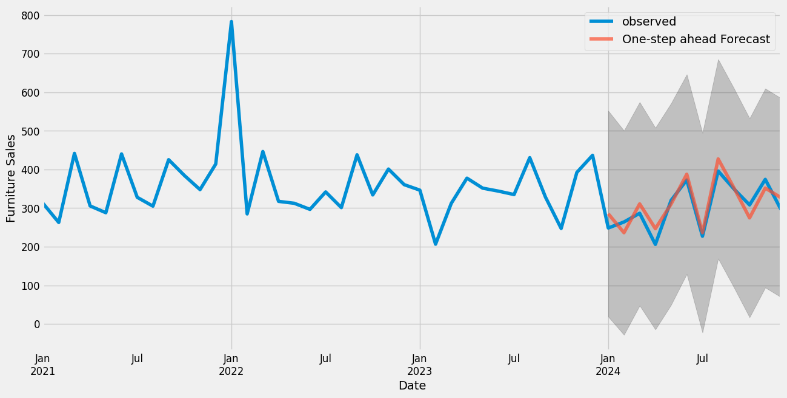


Fig. 8 (b) Outcome of Block Resampling Model

**3.3 Comparative Analysis**

Through the research, Table 1. shows three different multi-period forecasting models-LSTM, VAR and block regression are applied to a sales forecast data of a supermarket and the analysis using key performance indicators such as MSE, RMSE, MAE and MAPE and r-squared error. The performance of each model is analysed and compared to determine each model’s efficiency. The LSTM model performs well on time series data. It has a manageable error in its predictions which is determined by its MSE. MAE indicates that average predicted price of LSTM model deviates slightly. LSTM model captures good ability to capture patterns, while its higher error rare compared to other model suggests room for improvement. VAR Model for time estimation performed poorly for time estimation, high error rate for the predictions and very low r-squared value suggests high variance. This suggests it is not the best model for this data and forecasting task. While Block Resampling Method works best among all the models with a very high accuracy and very low variance makes it the best model to predict future sales in the comparison.

Table 1. Comparative Study Analysis of Models

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MODEL** | **MSE** | **RMSE** | **MAE** | **MAPE** | **R2** |
| LSTM | |  | | --- | |  |   3205.84 | |  | | --- | |  |   56.62 | |  | | --- | |  |   47.24 | |  | | --- | |  |   16.5% | |  | | --- | |  |   0.76 |
| VAR | 3438.98 | 58.64 | 52.22 | 18.21% | -0.01 |
| Block Resampling | 695.44 | 26.37 | 23.69 | 8.38% | 0.8 |

When comparing the models, it is clear that block resampling is the most effective and outperforms all other models in all important parameters. It shows that it provides the most accurate and reliable results by reaching the lowest MSE, RMSE, MAE and the highest R2 values. LSTM outperforms VAR it is slightly more efficient and can benefit from further improvements or refinements to increase its prediction accuracy. VAR model is not the best model for this situation and needs some improvement to perform better. The results show the importance of choosing the right time for the prediction model, because traditional methods such as VAR may not always be suitable for complex data , while advance techniques such as block repletion can provide robust and accurate results.

4. Conclusions

In conclusion, this study investigates advanced methods for multivariate time series forecasting, focusing on overcoming the challenges of complex dependencies, non-linearity, and imbalanced datasets, especially when predicting rare extreme events. By comparing three key approaches—LSTM networks, VAR models, and block resampling techniques—the study aims to enhance prediction accuracy for interconnected and dynamic systems. LSTM networks are pretty effective and addressing limitations of traditional RNNs. They excel in forecasting time-based data where past context plays a crucial role in future predictions. VAR models, on the other hand, offer a powerful framework for analysing multivariate time series, capturing the interdependencies among multiple variables. Their ability to handle feedback loops and relationships between different time series makes them valuable for fields like finance and economics. The novel block resampling technique significantly improves forecasting accuracy in scenarios involving extreme events, which are often underrepresented in datasets. By oversampling extreme event blocks and under sampling normal event blocks, this method creates a balanced dataset, ensuring that rare but important events are accurately predicted.

Using a dataset like the Supermarket sales Forecasting Dataset, the study validates these approaches, demonstrating that a combination of deep learning models (LSTM), statistical deep learning techniques (VAR), and block resampling can yield superior forecasting performance. This multi-model approach offers a robust solution for time series prediction in fields where accurate forecasts are critical for decision-making, resource management, and strategic planning.

5. References

*[1] Xuanbing Zhu; Dunbin Shen; Hongyu Wang; Yingguang Hao (2024),”* *FCNet: Fully Complex Network for Time Series Forecasting”, IEEE* ***1*** *1-5****,*** *DOI 10.1109/JIOT.2024.3449903*

*[2] Kai Wu; Jing Liu; Penghui Liu; Shanchao Yang (2020) ,"* *Kai Wu; Jing Liu; Penghui Liu; Shanchao Yang", IEEE Transactions 28 (12), 3110-3115, doi: 10.1109/TFUZZ.2019.2956904*

*[3] T. Li, M. Hua and X. Wu, (2020) "A Hybrid CNN-LSTM Model for Forecasting Particulate Matter (PM2.5)," in IEEE Access,* ***8****, 26933-26940, doi: 10.1109/ACCESS.2020.2971348*

*[4] Xianyun Wen, Weibang Li (2022) "Advanced Time Series Forecasting Techniques: A Case Study on Air Quality Prediction." Southwest Minzu University for Nationalities* ***13****(5), 788,*[*https://doi.org/10.3390/atmos13050788*](https://doi.org/10.3390/atmos13050788)

*[5] P. -Y. Hao and J. -H. Chiang, (2008) "Fuzzy Regression Analysis by Support Vector Learning Approach," in IEEE Transactions on Fuzzy Systems,* ***16*** *(2), 428-441, doi: 10.1109/TFUZZ.2007.896359*

*[4] Mehmet Ozgur Turkoglu; Stefano D'Aronco; Jan Dirk Wegner; Konrad Schindle "A Comprehensive Study on Improving Time Series Forecasting Models Using LSTM." IEEE Explore, 2021.*

*[5] Wang Xue Song, Dong Chao, Bo Cui Mei "A Hybrid Modelling Approach for Batch Process Based on LSTM-RNN." IEEE Explore, 2021.*

*[6] Wang, Lin & Wang, Zhigang & Liu, Shan. (2016). An effective multivariate time series classification approach using echo state network and adaptive differential evolution algorithm. Expert Systems with Applications,* ***43****, 237–249. 10.1016/j.eswa.2015.08.055.*

*[7] Salim Lahmiri,(2016) ”A variational mode decomposition approach for analysis and forecasting of economic and financial time series,*

*Expert Systems with Applications,” ISSN 0957-4174,* ***55 268-273,*** *https://doi.org/10.1016/j.eswa.2016.02.025.*

*[8] Q. Ma, Z. Chen, S. Tian and W. W. Y. Ng,(2022) "Difference-Guided Representation Learning Network for Multivariate Time-Series Classification," in IEEE Transactions on Cybernetics,* ***52****(6), 4717-4727,doi: 10.1109/TCYB.2020.3034755*

*[9] Z. Wang, J. Fan, H. Wu, D. Sun and J. Wu, (2024) "Representing Multiview Time-Series Graph Structures for Multivariate Long-Term Time-Series Forecasting," in IEEE Transactions on Artificial Intelligence,* ***5****(6),2651-2662, doi: 10.1109/TAI.2023.3326796.*

*[11] X. Chen, L. Gupta and S. Tragoudas, (2022) "Improving the Forecasting and Classification of Extreme Events in Imbalanced Time Series Through Block Resampling in the Joint Predictor-Forecast Space," in IEEE Access,****10****, 121048-121079, 2022, doi: 10.1109/ACCESS.2022.3219832.*

*[12] W. Wang, K. Lin and J. Zhao,(2020) "Network-Based Layered Architecture for Long-Term Prediction," in IEEE Access,* ***8****, 18252-18257, doi: 10.1109/ACCESS.2020.2968473.*