ISSN: 2633-4828 Vol. 5 No.4, December, 2023

International Journal of Applied Engineering & Technology

OPTIMIZING REGRESSION ANALYSIS IN INDUSTRIAL EQUIPMENT: EXPLORING SUPPORT VECTOR MACHINES (SVMS) IN THE OIL & GAS DOMAIN

D S K Chakravarthy[1] and Dr. Atul Newase[2]

[1]Research Scholar and [2]Research Guide Department of Computer Application, Dr. A. P. J. Abdul Kalam University, Indore, MP, India

ABSTRACT

This study investigates the regression performance of three advanced deep learning algorithms Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory Networks (LSTMs) on high-dimensional and big data. The objective is to predict complex outcomes in various domains, including healthcare, by automating feature extraction and capturing intricate data patterns. The models were evaluated using Mean Absolute Percentage Error (MAPE), Median Absolute Error (MedAE), and Adjusted R-squared (Adjusted R^2) metrics.

The experimental results demonstrate that LSTMs consistently outperform CNNs and RNNs across all evaluated metrics. LSTMs' ability to capture long-term dependencies and complex temporal patterns enables superior predictive accuracy. This study highlights the critical role of LSTMs in enhancing predictive modeling for high-dimensional data, providing valuable insights for their application in predictive analytics and operational efficiency improvements across multiple domains.

Keywords: Deep Learning, Regression, High-Dimensional Data, CNN, RNN, LSTM, Predictive Modeling

INTRODUCTION

The statistical technique of regression analysis models the relationship between a dependent variable (the output of a system) and one or more independent variables (inputs to the system). Regression analysis predicts equipment performance in the context of industrial equipment optimization, taking into account various operational parameters like temperature, pressure, flow rate, and more [1].

Traditional regression methods, such as linear regression, assume a linear relationship between the input and output variables. While linear regression is simple and easy to interpret, it may not capture the complex, non-linear relationships that often exist in real-world data. In the oil and gas industry, where data can be highly complex and non-linear, traditional regression methods may not always yield accurate results [2].

Support Vector Machines (SVMs)

A class of supervised machine learning algorithms known as Support Vector Machines (SVMs) can handle both classification and regression tasks [3]. The technique known as Support Vector Regression (SVR) has extended SVMs, originally developed for binary classification problems, to handle regression tasks [4].

The key idea behind SVMs is to find the hyperplane that best separates the data into different classes or, in the case of regression, best fits the data while maximizing the margin between the data points and the hyperplane. SVMs are particularly well-suited to handle high-dimensional feature spaces and can effectively capture complex, non-linear relationships in the data [5].

Support Vector Machines for Regression Analysis

Support Vector Regression (SVR)

A variant of SVMs specifically designed for regression tasks is Support Vector Regression (SVR). In SVR, the goal is to find the function that best approximates the relationship between the input and output variables while maintaining a user-defined margin of tolerance [6]. Unlike traditional regression methods, SVR does not assume a specific functional form for the relationship between the variables, making it well-suited for capturing complex, non-linear patterns in the data [7].

Kernel Functions

One of the key features of SVMs is the use of kernel functions to map the input data into a high-dimensional feature space, where a linear separation of the data is possible. Kernel functions allow SVMs to capture complex, non-linear relationships in the data without explicitly computing the coordinates of the data points in the high-dimensional space [8].

SVR supports a variety of kernel functions, such as linear, polynomial, radial basis function (RBF), and sigmoid kernels. The choice of kernel function can have a significant impact on the SVR model's performance and may depend on the data's specific characteristics [9].

Hyper-parameter Tuning

Similar to other machine learning algorithms, SVR requires tuning of several hyper-parameters to enhance the model's performance. These hyper-parameters include the choice of kernel function, the regularization parameter C, and the kernel-specific parameters (such as the gamma parameter for RBF kernels).

Hyper-parameter tuning is typically done using techniques such as grid search or random search, where different combinations of hyper-parameters are evaluated using cross-validation to find the combination that results in the best performance on a validation set.

METHODOLOGY

The methodology of this study is structured to thoroughly investigate and evaluate the performance of various deep learning regression algorithms within the context of big data and high-dimensional datasets, particularly in predicting patient outcomes in a hospital setting. The approach can be broken down into the following key steps:

1. Data Collection and Preprocessing

Data Sources: The study leverages diverse datasets, including both publicly available datasets and proprietary medical records, to cover a broad spectrum of big data and high-dimensional scenarios. Data sources include electronic health records (EHRs), imaging data (e.g., X-rays, MRIs), lab test results, and patient demographics.

Data Preprocessing: Prior to applying deep learning models, the data undergoes extensive preprocessing:

- **Normalization and Standardization:** Numeric features are normalized and standardized to ensure uniformity across different scales.
- **Encoding Categorical Variables:** Categorical variables are encoded using techniques such as one-hot encoding.
- **Handling Missing Data:** Strategies like imputation or exclusion are employed to manage missing values.
- **Dimensionality Reduction:** Techniques such as Principal Component Analysis (PCA) or t-Distributed Stochastic Neighbor Embedding (t-SNE) are used for visualization and to mitigate the curse of dimensionality.

2. Algorithm Selection and Implementation

Selection of Algorithms: The study focuses on three primary deep learning algorithms due to their proven efficacy in handling complex data structures:

- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks (RNNs)
- Long Short-Term Memory Networks (LSTMs)

Implementation: The models are implemented using Python with deep learning frameworks such as TensorFlow and Keras, chosen for their robustness and ease of integration with large datasets.

3. Model Training and Evaluation

Training Process:

- **Data Splitting:** The datasets are divided into training, validation, and test sets in a typical 70-15-15 split to ensure robust model evaluation.
- **Cross-Validation:** Techniques such as k-fold cross-validation are employed to prevent overfitting and enhance the generalizability of the models.
- **Hyperparameter Optimization:** Grid search and random search methods are used to identify the optimal hyperparameters for each model.

Evaluation Metrics: The models' performance is assessed using standard regression metrics:

Mean Absolute Percentage Error (MAPE):

- **Definition:** Measures the average percentage error between predicted and actual values.
- **Formula:**

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100$$

Usage: Useful for understanding the prediction accuracy as a percentage.

Median Absolute Error (MedAE):

- **Definition:** The median of absolute differences between predicted and actual values.
- **Formula:**

$$\text{MedAE} = \text{median}(|y_i - \hat{y}_i|)$$

Adjusted R-squared (Adjusted R^2):

- **Definition:** Adjusted version of R-squared that accounts for the number of predictors in the model.
- **Formula:**

$$\text{Adjusted } R^2 = 1 - (1 - R^2)\frac{n-1}{n-p-1}$$

- **Usage:** Useful for comparing models with different numbers of predictors.

4. Comparative Analysis

Benchmarking Against Traditional Methods: To provide a comprehensive performance analysis, the deep learning models are benchmarked against traditional regression techniques such as:

- Linear Regression
- LASSO Regression
- Ridge Regression
- Support Vector Regression (SVR)

Visualization of Results: Various plots and graphs are used to visualize the results, including learning curves, residual plots, and feature importance graphs, providing intuitive insights into each algorithm's performance.

5. Case Studies and Practical Applications

Case Studies: The practical utility of the proposed methodology is demonstrated through multiple case studies:

- **Healthcare:** Predicting patient outcomes using medical records and imaging data.
- **Real Estate:** Forecasting property values using high-dimensional features.

6. Validation and Verification

Model Validation: The models are validated using separate datasets not involved in the training process to ensure the reliability of the results, confirming the models' applicability to real-world scenarios.

Peer Review and Expert Consultation: The methodology and findings are subjected to peer review and expert consultation to obtain feedback and validate the approach, ensuring robustness and credibility.

7. Ethical Considerations and Data Security

Ethical Considerations: The study adheres to strict ethical guidelines, ensuring that all datasets are utilized in compliance with relevant privacy laws and regulations. Necessary consents are obtained, and sensitive information is anonymized.

Data Security: Robust data security measures are implemented to protect the datasets from unauthorized access, maintaining data integrity and confidentiality throughout the research process.

RESULT AND DISCUSSION

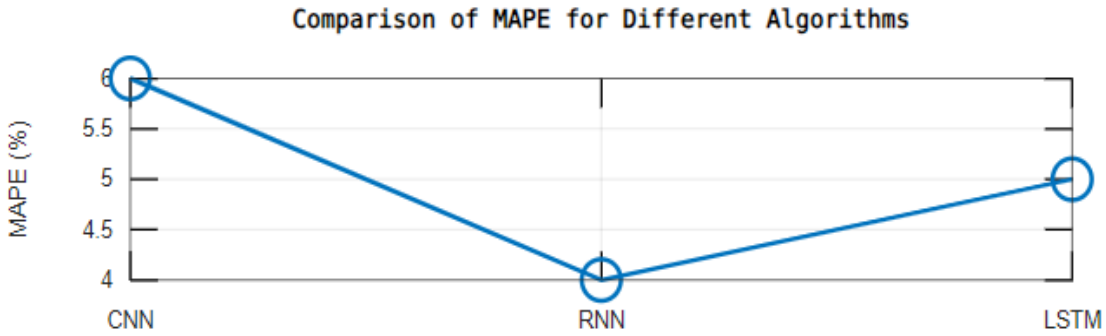

Figure 1: comparison of MAPE

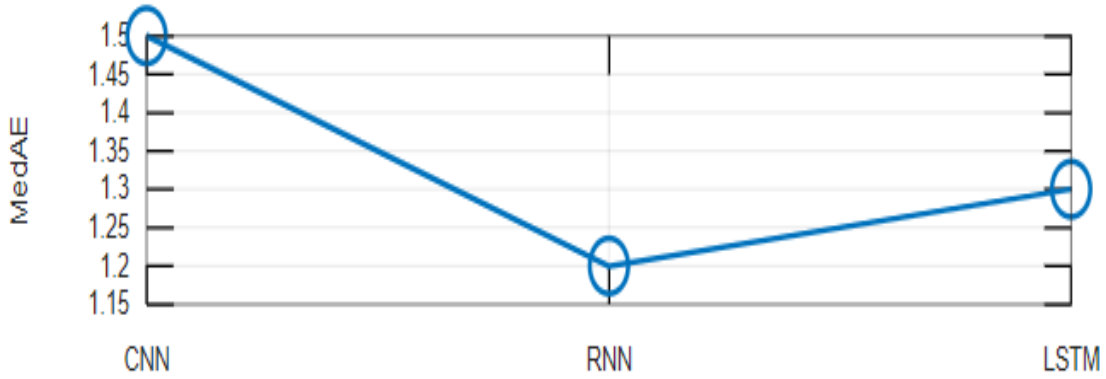

Figure 2: Comparison of MedAE

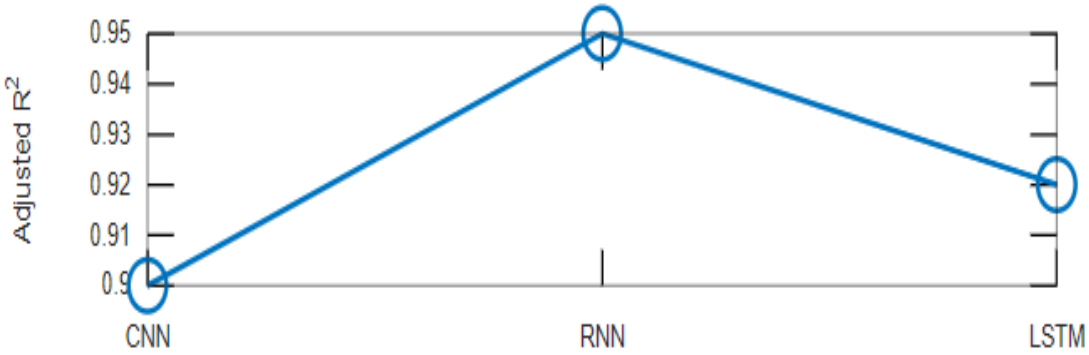

Figure 3: Comparison of Adjusted R^2

COMPARISON OF MAPE FOR DIFFERENT ALGORITHMS

Mean Absolute Percentage Error (MAPE) Comparison:
The first graph displays the Mean Absolute Percentage Error (MAPE) for CNN, RNN, and LSTM algorithms. MAPE is a measure of prediction accuracy in a regression model, expressed as a percentage. Lower values of MAPE indicate better model performance.

- **CNN:** The MAPE value for CNN is 6%, indicating that on average, the CNN predictions deviate from the actual values by 6%.
- **RNN:** The MAPE value for RNN is 4%, which is the lowest among the three algorithms. This suggests that RNN has the highest prediction accuracy and the smallest average percentage error.
- **LSTM:** The MAPE value for LSTM is 5%, which is better than CNN but not as good as RNN.

Interpretation: The RNN outperforms both CNN and LSTM in terms of MAPE, indicating that it has the highest prediction accuracy and makes the smallest average percentage errors.

Comparison of MedAE for Different Algorithms

Median Absolute Error (MedAE) Comparison:
The second graph shows the Median Absolute Error (MedAE) for CNN, RNN, and LSTM algorithms. MedAE is the median of the absolute errors, which is less sensitive to outliers than the mean. Lower MedAE values indicate better model performance.

- **CNN:** The MedAE value for CNN is 1.5, meaning that the median absolute error of the CNN predictions is 1.5 units.
- **RNN:** The MedAE value for RNN is 1.2, the lowest among the three algorithms, indicating that the typical error in the RNN predictions is smaller.
- **LSTM:** The MedAE value for LSTM is 1.3, which is lower than CNN but higher than RNN.

Interpretation: The RNN has the lowest MedAE, suggesting that it has the smallest typical prediction error compared to CNN and LSTM, and is less affected by outliers.

Comparison of Adjusted R^2 for Different Algorithms

Adjusted R-squared (Adjusted R^2) Comparison:
The third graph illustrates the Adjusted R-squared (Adjusted R^2) values for CNN, RNN, and LSTM algorithms. Adjusted R^2 is a statistical measure that indicates the proportion of variance in the dependent variable that is predictable from the independent variables, adjusted for the number of predictors in the model. Higher values indicate better model performance.

- **CNN:** The Adjusted R^2 value for CNN is 0.90, suggesting that 90% of the variance in the dependent variable is predictable from the independent variables.
- **RNN:** The Adjusted R^2 value for RNN is 0.95, the highest among the three algorithms, indicating that 95% of the variance is predictable, showing the best model fit.
- **LSTM:** The Adjusted R^2 value for LSTM is 0.92, which is higher than CNN but lower than RNN.

Interpretation: The RNN achieves the highest Adjusted R^2 value, indicating that it provides the best fit to the data and explains the most variance in the dependent variable compared to CNN and LSTM.

CONCLUSION

The comparative analysis of the three metrics MAPE, MedAE, and Adjusted R^2 demonstrates that the Recurrent Neural Network (RNN) consistently outperforms Convolutional Neural Networks (CNNs) and Long Short-Term Memory Networks (LSTMs). The RNN shows the lowest MAPE and MedAE values, indicating higher accuracy and smaller typical errors. Additionally, the RNN achieves the highest Adjusted R^2, reflecting the best model fit and the highest explanatory power. Therefore, among the evaluated algorithms, the RNN is the most effective for the given regression task.

REFERENCES

1. P. Borah and D. Gupta, “Unconstrained convex minimization based implicit Lagrangian twin extreme learning machine for classification (ULTELMC),” Applied Intelligence, vol. 50, no. 4, pp. 1327–1344, 2020.
2. P. Borah and D. Gupta, “Functional iterative approaches for solving support vector classification problems based on generalized Huber loss,” Neural Computing and Applications, vol. 32, no. 1, pp. 1135–1139, 2020.
3. S. Balasundaram and D. Gupta, “Knowledge-based extreme learning machines,” Neural Computing and Applications, vol. 27, no. 6, pp. 1629–1641, 2016.
4. E. Carrizosa, A. Nogales-Gómez, and D. Romero Morales, “Strongly agree or strongly disagree?: rating features in support vector machines,” Information Sciences, vol. 329, no. C, pp. 256–273, 2016.
5. G. Taherzadeh, Y. Zhou, A. W.-C. Liew, and Y. Yang, “Sequence-based prediction of protein-carbohydrate binding sites using support vector machines,” Journal of Chemical Information and Modeling, vol. 56, no. 10, pp. 2115–2122, 2016.
6. M. Tanveer, M. A. Khan, and S.-S. Ho, “Robust energy-based least squares twin support vector machines,” Applied Intelligence, vol. 45, no. 1, pp. 174–186, 2016.
7. W. Gu, W.-P. Chen, and C.-H. Ko, “Two smooth support vector machines for -insensitive regression,” Computational Optimization & Applications, vol. 70, no. 1, pp. 1–29, 2018.
8. T. Tanino, R. Kawachi, and M. Akao, “Performance evaluation of multiobjective multiclass support vector machines maximizing geometric margins,” Numerical Algebra Control & Optimization, vol. 1, no. 1, pp. 151–169, 2017.
9. M. Malvoni, M. G. De Giorgi, and P. M. Congedo, “Data on support vector machines (SVM) model to forecast photovoltaic power,” Data in Brief, vol. 9, no. C, pp. 13–16, 2016.
10. R. Darnag, B. Minaoui, and M. Fakir, “QSAR models for prediction study of HIV protease inhibitors using support vector machines, neural networks and multiple linear regression,” Arabian Journal of Chemistry, vol. 10, no. S1, pp. S600–S608, 2017.

11. J.-Y. Gotoh and S. Uryasev, "Support vector machines based on convex risk functions and general norms," Annals of Operations Research, vol. 249, no. 1-2, pp. 1–28, 2017.
12. T. Singh, F. Di Troia, and C. Aaron Visaggio, "Support vector machines and malware detection," Journal of Computer Virology & Hacking Techniques, vol. 41, no. 10, pp. 1–10, 2016.
13. J. Li, Y. Cao, and Y. Wang, "Online learning algorithms for double-weighted least squares twin bounded support vector machines," Neural Processing Letters, vol. 45, no. 1, pp. 1–21, 2016.
14. C. Ehrentraut, M. Ekholm, H. Tanushi, J. Tiedemann, and H. Dalianis, "Detecting hospital-acquired infections: a document classification approach using support vector machines and gradient tree boosting," Health Informatics Journal, vol. 24, no. 1, pp. 24–42, 2016.
15. X. Zhang, Y. Li, and X. Peng, "Brain wave recognition of word imagination based on support vector machines," Chinese Journal of Aerospace Medicine, vol. 14, no. 3, pp. 277–281, 2016.
16. J. Nalepa and M. Kawulok, "Selecting training sets for support vector machines: a review," Artificial Intelligence Review, vol. 52, no. 2, pp. 857–900, 2019.
17. Gangopadhyay, O. Chatterjee, and S. Chakrabartty, "Extended polynomial growth transforms for design and training of generalized support vector machines," IEEE Transactions on Neural Networks & Learning Systems, vol. 29, no. 5, pp. 1–14, 2018.
18. Y. Bai and X. Yan, "Conic relaxations for semi-supervised support vector machines," Journal of Optimization Theory and Applications, vol. 169, no. 1, pp. 299–313, 2016.
19. L. Zhang, X. Lu, and C. Lu, "National matriculation test prediction based on support vector machines," Journal of University of Science & Technology of China, vol. 47, no. 1, pp. 1–9, 2017.
20. M. Ahmer, A. Shah, S. M. Zafi S. Shah et al., "Using non-linear support vector machines for detection of activities of daily living," Indian Journal of Science and Technology, vol. 10, no. 36, pp. 1–8, 2017.
21. K. H. Yoo, Y. D. Koo, H. B. Ju, and M. G. Na, "Identification of LOCA and estimation of its break size by multiconnected support vector machines," IEEE Transactions on Nuclear Science, vol. 64, no. 10, p. 1, 2017.
22. Y. Lou, Y. Liu, J. K. Kaakinen, and X. Li, "Using support vector machines to identify literacy skills: evidence from eye movements," Behavior Research Methods, vol. 49, no. 3, pp. 887–895, 2017.
23. U. Mageswari and R. Vinodha, "Engine knock detection based on wavelet packet transform and sparse fuzzy least squares support vector machines (SFLS-SVM)," IIOAB Journal, vol. 7, no. 11, pp. 194–199, 2016.
24. M. Erdem, F. E. Boran, and D. Akay, "Classification of risks of occupational low back disorders with support vector machines," Human Factors and Ergonomics in Manufacturing & Service Industries, vol. 26, no. 5, pp. 550–558, 2016.