**ENHANCING PROACTIVE RESOURCE ALLOCATION IN HIGH-DENSITY EDGE**

**COMPUTING USING EXPLAINABLE AI AND SHAP**

**KAVYA M** 1**, NAVEEN PRASATH J 2, MOHANAM K3**

[**e5222044@sret.edu.in**](mailto:e5222044@sret.edu.in)**,** [**e5222031@sret.edu.in**](mailto:e5222031@sret.edu.in)**,** [**e5222054@sret.edu.in**](mailto:e5222054@sret.edu.in)

**SRI RAMACHANDRA INSTITUTE OF HIGHER EDUCATION AND RESEARCH**

**ABSTRACT**

The study investigates efficient allocation of resources in high-density edge network computing through machine learning techniques. The ecological data set was used to evaluate four predictors' accuracy including Random Forest Regressor, LSTM, CNN, and Ridge Regression using R-Squared (R²), Mean Absolute Error (MAE) and Mean Squared Error (MSE) measures. The Random Forest Regressor model outperformed all other models because it demonstrated superior pattern recognition capabilities over time which made it the top choice for this particular application. The comparative study evaluates the strengths and weaknesses of each model while analysing their computational complexity and resource requirements. The research examines how well each model can be scaled up. Through SHAP values analysis researchers gained insight into feature importance and how ecological variables influenced prediction results. The SHAP analysis demonstrated the Random Forest model's effectiveness in detecting important relationships between features which confirmed its robustness. The study results demonstrate the importance of machine learning interpretability for resource allocation which helps improve decision-making processes in edge computing environments. By comprehending model behaviour alongside feature contributions stakeholders can improve both system adaptability and efficiency. This work highlights the potential for machine learning in dynamic settings, emphasizing the need for interpretable yet high-performing models for edge network resource allocation.

**KEYWORDS**: *Machine Learning Model, Random Forest, Ridge Regression, CNN, LSTM, SHAP, Performance Comparison, Scalability.*

**INTRODUCTION**

The automotive edge is also getting dense with more sensors and more connectivity with cloud providers connected via dense networks, making it not trivial to cost-effectively allocate resources in edge computing environments as dense networks scale. With a well-defined allocation strategy, it can significantly improve throughput, reduce latency and utilization of the available bandwidth. That problem can be addressed with the help of machine learning, in predicting future resource requirements. In this research, we compare four models Random Forest Regressor, Ridge Regression, LSTM, and CNN to find out which model would be most appropriate for edge computing workloads. Edge computing reduces the distance between computation and storage of data and where they are needed, lessening the dependency on central systems and minimizing delays. However, tight networks come with their own trade-offs, such as resource competition and variation in performance. An optimally distributed resource system can alleviate these problems and make networks more efficient, reliable, and robust. To gain a balanced view, we also look at a non-ML solution, Dynamic Hill Climbing. Although it is computationally light and simple to implement, it is weak in handling complicated, non-linear patterns in dynamic settings. Random Forest Regressor and LSTM machine learning algorithms demonstrate superior adaptability along with high accuracy and strong resilience. We utilize Shapley Additive explanations (SHAP) to explain model predictions and identify variable contributions toward impactful resource allocation determination. SHAP analysis provides additional evidence for the effectiveness of ML models especially Random Forest to adjust rapidly to changing environments.

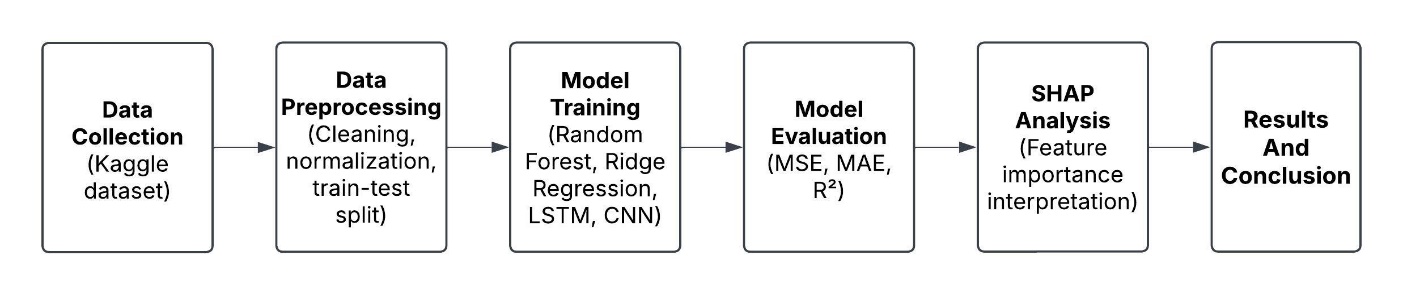
**LITERATURE REVIEW**

Edge computing progress has stimulated further research into proactive resource provisioning systems which aim to improve network efficiency. Aghazadeh et al. (2023) [1] investigated content caching methods within edge computing systems because predictive techniques play a crucial role in optimizing resource distribution. Similarly, Duc et al. Duc et al. (2019) [2] delivered an extensive literature review about machine learning applications for resource provisioning and confirmed their dependability in edge-cloud computing systems. Research efforts on resource allocation problems include studies by Sharkh et al. The paper from Sharkh et al. (2013) [3] examined network-based cloud computing environments and Chen et al. (2021) developed an adaptive prediction-based resource allocation strategy for cloud computing. Chen et al. (2021) [4] introduced an adaptive prediction-based resource allocation strategy for cloud computing environments. Edge computing optimization has benefited from the implementation of machine learning and AI-based technologies. Shakarami et al. (2022) [5] investigated resource provisioning for edge and fog computing systems through scalability and efficiency analysis whereas Alsadie (2024) [6] explored the use of AI techniques for resource management in fog computing and identified emerging trends and challenges. The research by Sirapangi et al. (2024) demonstrates that Explainable AI (XAI) is becoming more prevalent in resource allocation strategies. Sirapangi et al. (2024) [11] recommended improved decision-making through a multimodal strategy that integrates preemptive analysis with Explainable AI (XAI). The use of Shapley Additive Explanations (SHAP) to explain feature importance in prediction models has grown which showcases the need for model interpretability in applications like edge computing according to El Khatib et al. (2022) [7]. Dense networks need proactive resource scheduling to reduce latency and achieve high throughput. Elbamby et al. (2018) [8] investigated proactive edge computing for fog networks and showed methods to attain latency and reliability assurances. Further, Kim et al. The 2023 research by Kim et al. [9] examined joint service caching and resource assignment to improve computational efficiency. Rublein et al. concentrated their research on preemptive resource allocation, which brought in more effective task allocation methods for edge computing systems. Following these observations, the present work builds upon earlier research by adding SHAP and Explainable AI methods to machine learning-based resource allocation techniques. Drawing on these instruments, the study aims to improve model explainability and optimize resource allocation in high-density edge computing environments with the goal of optimizing both computational effectiveness and decision transparency.

**METHODOLOGY**

Used a Kaggle collection of a real network performance dataset containing throughput, bandwidth and latency values. The collection was appropriately pre-processed to ensure the quality, consistency and usability to machine learning algorithms. To maintain data integrity and not overfit the model with biased training data, the entries with Nan or infinite values were dropped. Further, all independent variables were Min-Max scaled to a range of zero to one to speed up convergence during neural network training. As this dataset exhibits ultra-dense networking scenarios, it is ideal for examining and optimizing resource allocations in edge computing networks. Various algorithms were used based on how well they would handle edge computing issues such as Random Forest Regressor, Ridge Regression, LSTM, and CNN. Random Forest is a strong ensemble technique that relies on a series of several decision trees to predict with high precision. It finds complex patterns precisely by taking an average of all the trees' predictions and keeps away from overfitting. Hyperparameter tuning was performed employing grid search so that optimal number of trees and depth were defined to ensure computationally efficient execution against predictability. Ridge Regression, a linear model with L2 regularization, is particularly beneficial for multicollinearity datasets. This regularization technique reduces coefficient values to prevent overfitting and improve generalization. The regularization parameter α was optimized to achieve the optimal trade-off between bias and variance. LSTM models, which are specifically for sequential and time-dependent data, were used to learn temporal relationships that exist in network performance metrics. The architecture included two LSTM layers of 50 units each, supported by dropout layers to avoid overfitting. The models were trained at a learning rate of 0.001 and optimized to make precise predictions about future network conditions. CNNs, which can handle grid-structured data, were used to process spatially correlated metrics. The CNN model had two convolutional layers with ReLU activation and subsequent max-pooling layers for reducing dimensions and fully connected layers for making final predictions. The Adam optimization algorithm was employed to speed up training and achieve convergence. The dataset was divided into two parts one for training (80%) and other for testing (20%) sets to assess model performance. Stratified sampling preserved class distribution across subsets, providing an unbiased assessment. Performance measures were Mean Squared Error (MSE), Mean Absolute Error (MAE) and R² score. MSE places greater importance on larger errors as it computes the average of the squared difference between actual and predicted values, while MAE offers the average magnitude of errors presents a balanced measure of error size, and R² measures the percentage of explained variance by the model, and higher values suggest better predictive fit. Shapley Additive Explanations (SHAP) served to investigate feature importance from models to enhance model explainability. SHAP demonstrated the primary impact of network parameters like latency and bandwidth on model predictions. Analysis confirmed that the Random Forest model detected important dependencies which enhanced its resource planning accuracy. SHAP provided detailed insights into model selection which helped demonstrate how key features affected predictions and enabled more informed and efficient resource management methods in edge computing.

**IMPLEMENTATION**

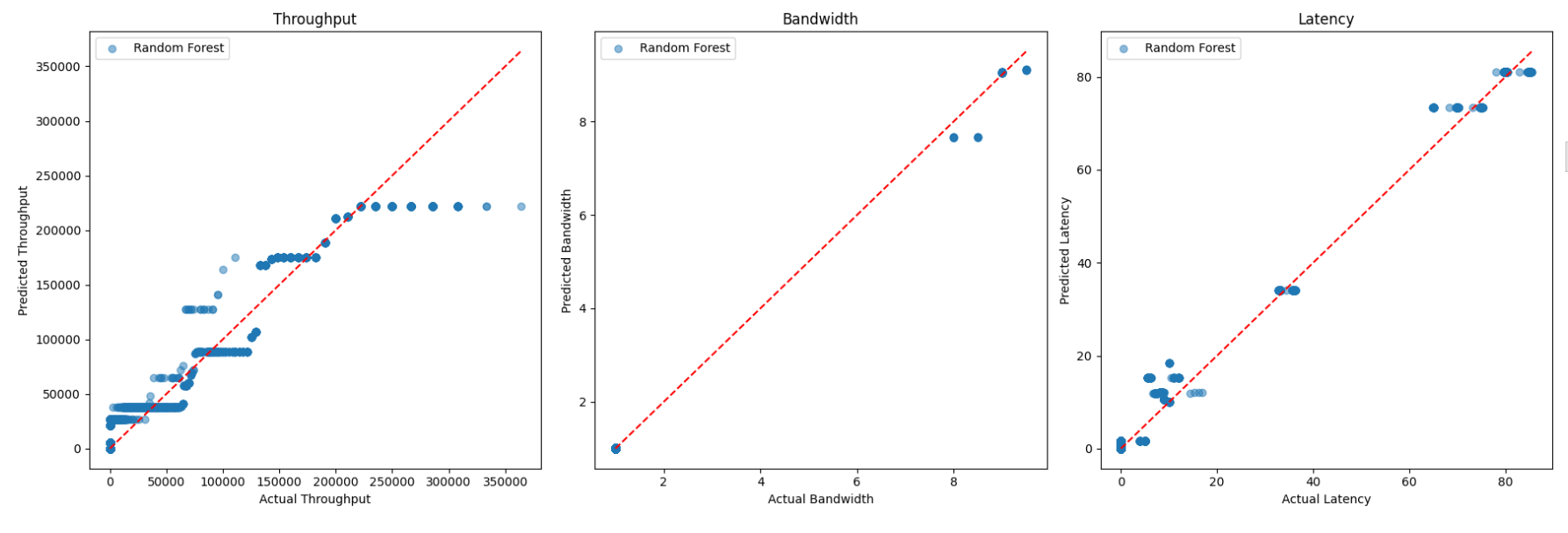


**FIGURE 1: IMPLEMENTATION FLOW CHART**

The first step to creating predictive models was to pre-processed the data. That was a key step for data quality, integrity and for training verification readiness. The dataset was acquired from Kaggle and was analyzed carefully to ensure proper structure and accuracy. Missing values were handled with removal of Nan’s or through imputation with appropriate estimates based on data distribution and relevancy. As a result, usage of incomplete data had to be handled carefully to avoid biased learning and in-model reliability. Importance metrics such as throughput, bandwidth and latency were included through feature engineering. These extracted features facilitate better insight into the network performance which is a good starting point for predictive modelling. Min-Max scaling was used to normalize feature scales and allow for model comparison, scaling values denoting values between a 0 and 1. This conversion allowed for quicker convergence, especially for deep learning models based on gradient-based optimization. Post-preprocessing, the data was divided into two which is training and testing subsets based on stratified sampling to maintain class distributions. This ensured that both sets had an equal proportion of data types to minimize biases and improve generalization. The research tested different deep learning and machine learning models, including Random Forest, Ridge Regression, CNN, and LSTM. Random Forest, an ensemble learning method, used many decision trees built with bagging in order to achieve maximum prediction accuracy and prevent overfitting and performed well in handling complex, non-linear patterns. Ridge Regression, which is a linear model that employs L2 regularization, prevented multicollinearity and generalized models by imposing a penalty on big coefficients. LSTMs, which were built for sequential data, learned long-term dependencies in network performance measures and were therefore particularly suitable for time-series forecasting. The model structure consisted of two LSTM layers with 50 units each and dropout layers to prevent overfitting. CNNs were used to process spatially correlated data using convolutional and pooling layers for hierarchical pattern detection. Both deep learning models were tuned with the Adam optimizer, adjusting learning rates dynamically to achieve stable convergence. Hyperparameter tuning also involved grid search for Random Forest and optimization of regularization parameter for Ridge Regression to further boost model performance. The evaluation stage measured model performance against important metrics: mean squared error (MSE), mean absolute error (MAE), and R² score. MSE calculated average squared errors, giving more weight to larger deviations, and MAE calculated the average absolute prediction error, and the R² score measured the fraction of variance explained by each model, with larger values representing better predictive performance. Continuous validation during training minimized overfitting and improved model performance. To improve interpretability, Shapley Additive Explanations (SHAP) were employed to examine feature importance between models. SHAP offered a decomposition of how single features, e.g., bandwidth and latency, impacted predictions. The examination reaffirmed that Random Forest successfully picked up on essential dependencies, affirming its application in resource allocation in edge computing. Through provision of insight into model choices, SHAP assisted in confirming the reliability of the chosen models and in offering network parameter influence insights, enabling better decision-making in the optimization of edge computing environments.

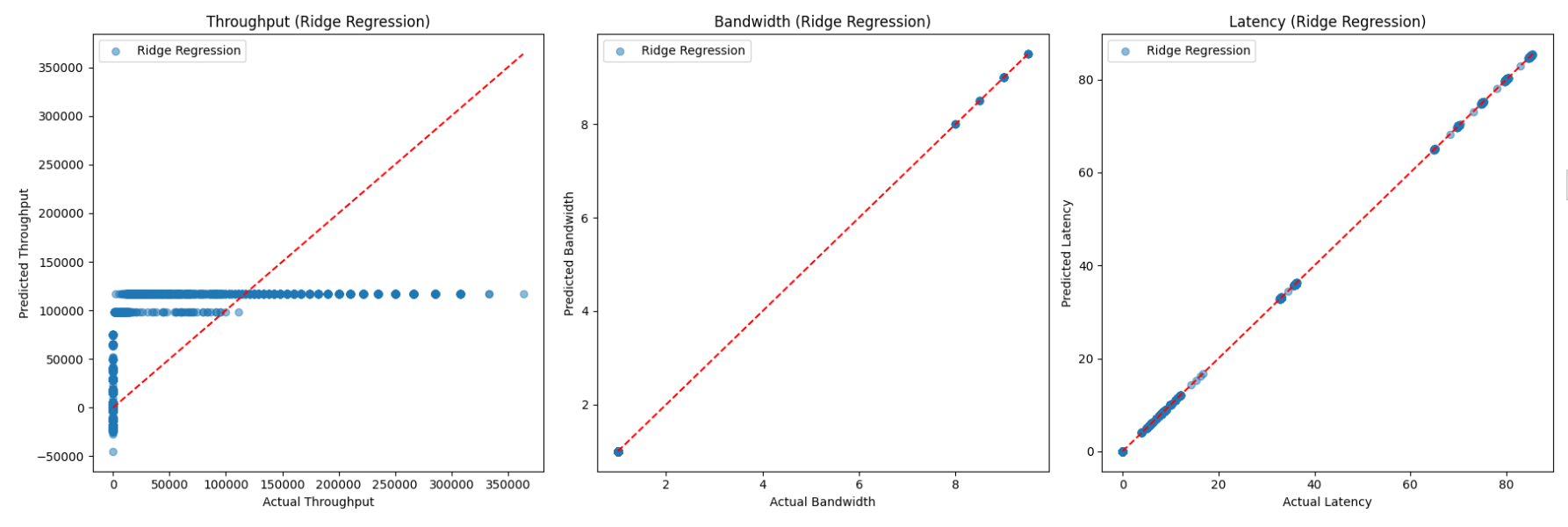
**RESULTS AND DISCUSSION**

Random Forest Regressor model performed well, with an MSE of 0.000832, an MAE of 0.013191, and R² score of 0.98. Its power lies in its ability to understand complex relationships between features and its high efficiency in dealing with high-dimensional data. It lacks the capacity to grasp the sequential nature or time dependencies of data and hence is less effective at time-series prediction. Even though it's an ensemble model, Random Forest is still fairly interpretable and is eminently suitable for feature importance-based assessments, especially in explainable-result scenarios. Its individual tree interpretability is somewhat compromised by its ensemble nature. The algorithm takes advantage of quick processing times and latency-reduced data processing, most notably when parallel computation is used to build the forest. Training times and memory usage do, however, escalate as dataset sizes increase. Whereas Random Forest has moderate resource needs, its scalability is significantly dependent on the number of trees and size of dataset.

****

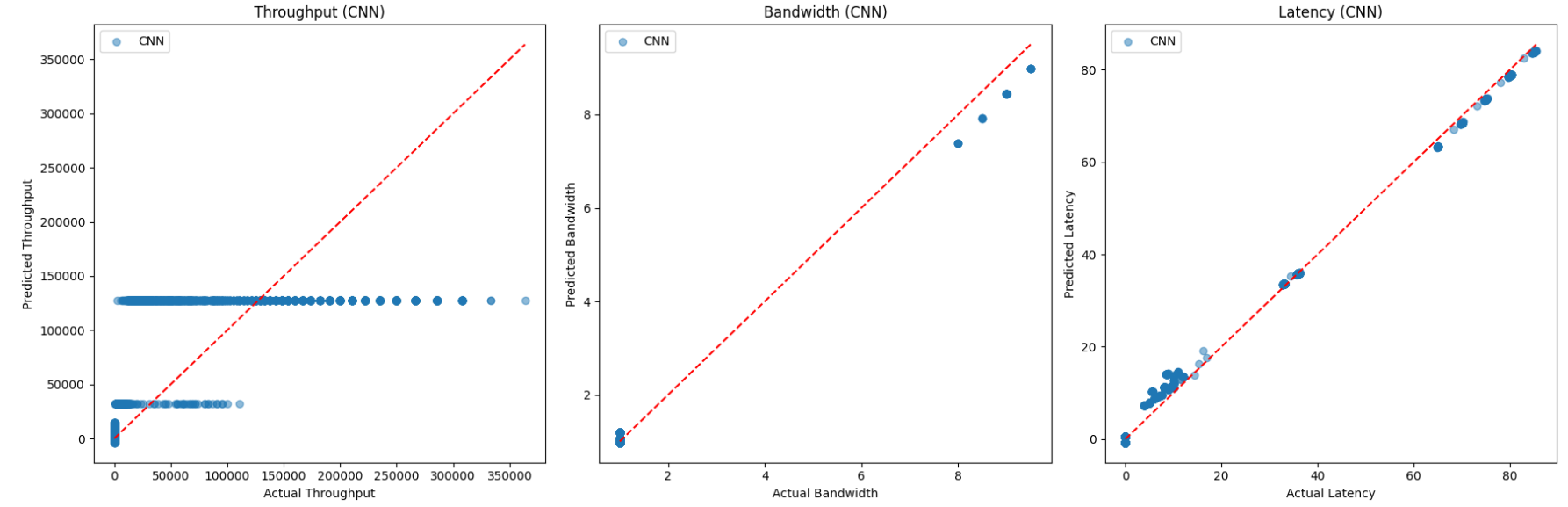
**FIGURE 2: RANDOM FOREST PREDICTION**

Ridge Regression performed well with an MSE of 0.006797, which is an MAE of 0.027790 and an R² score of 0.837995. Ridge Regression performed well in capturing linear relationships in the data but struggled with handling intricate non-linear relations. While its simplicity and interpretability make it a strong contender for basic regression problems, its linearity assumptions limit it from modelling advanced patterns. School-wise, Ridge Regression is computationally highly efficient with low processing demands and memory needs compared to other sophisticated models like CNN and LSTM. Ridge Regression is of low latency, which is suitable for real-time prediction, particularly where rapid deployment is needed. Ridge Regression also does not need large datasets to train successfully, and thus one can deploy it quickly. However, its lack of ability to handle non-linear relationships makes one consider data complexity when selecting a regression model.

****

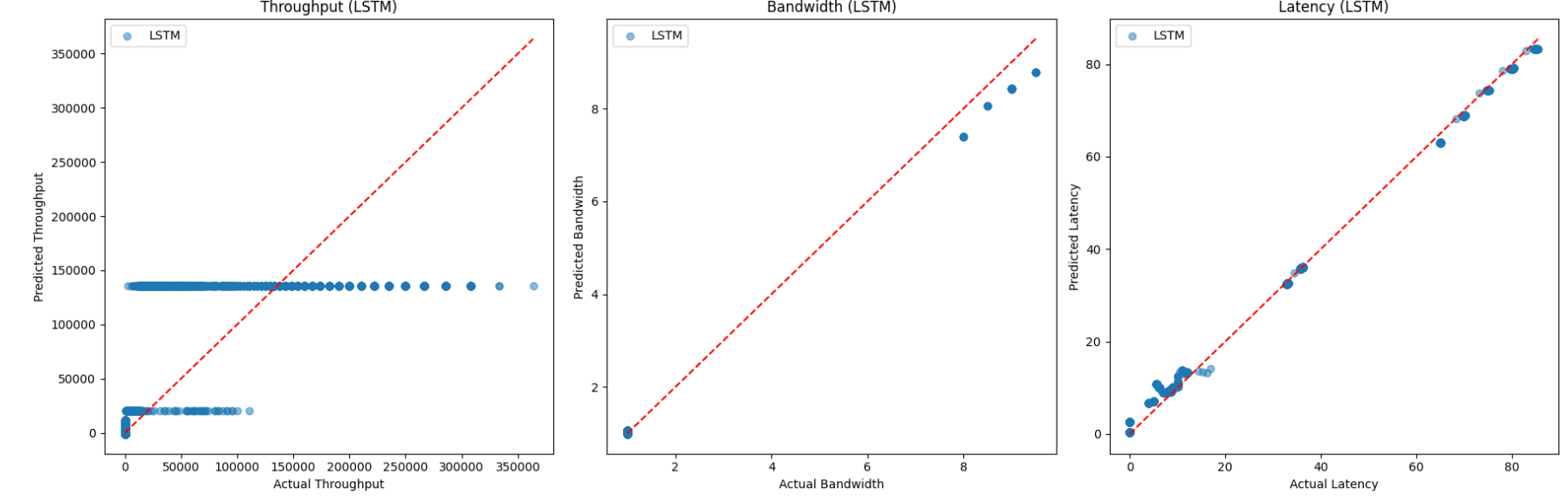
**FIGURE 3: RIDGE REGRESSION PREDICTION**

CNNs resulted in a learning performance of MSE 0.005458, MAE of 0.024325, and an R² score of 0.868913, strongly indicating that they fit the data exceptionally well. The hierarchical structure of CNNs helped them to learn non-linear, complex patterns, making feature learning more suitable for them. Nevertheless, this benefit is overshadowed by high computational requirements and the requirement for careful tuning of parameters to avoid overfitting. Despite these issues, CNNs were found to be very effective at detecting complex relationships in the dataset. Nevertheless, their requirement of big datasets and heavy computational resources might restrict their use in resource-poor settings. From the performance point of view, CNNs showcase excellent computational efficiency by using GPU acceleration for parallel processing. Nevertheless, training time is often long, particularly for deeper networks, and feature extraction involves many layers. Although CNNs are good at processing complicated, large-scale data, their memory and bandwidth needs can be substantial, so they are not as well-suited for real-time, low-resource applications.

****

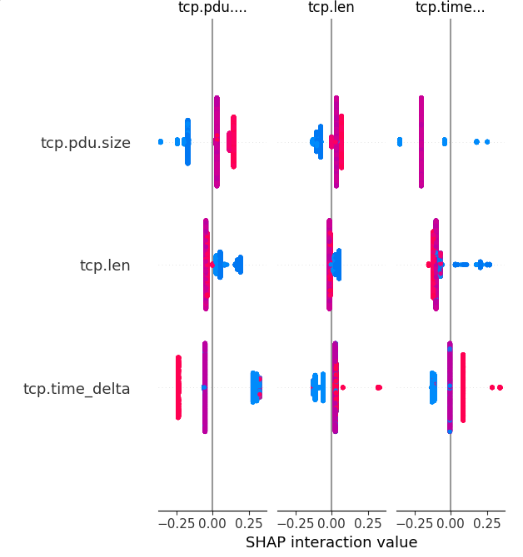
**FIGURE 4: CONVOLUTIONAL NEURAL NETWORK (CNN) PREDICTION**

Long Short-Term Memory (LSTM) showed excellent performance with an MSE of 0.005361, an MAE of 0.023926, and R² score of 0.871390. LSTM performed exceptionally well in identifying sequential dependencies and intricate data relationships and thus proved especially good for tasks related to temporal or sequential data. The capacity of the LSTM to hold long-term dependencies helped it to better grasp contextual patterns within the dataset, giving it a huge advantage over conventional models. Though LSTMs are very demanding in terms of computation and training time, they are extremely flexible and scalable, hence easily applied across different fields where time-series modelling is essential, including finance, healthcare, and natural language processing. The performance-wise, LSTMs are relatively good during the training process, but inference time might be slower than non-recurrent models since it involves sequential processing. Latency is generally greater, especially with large sequences, because the model handles data incrementally. LSTMs also require large bandwidth and memory for weight and state storage, and so are more resource-hungry than more basic machine learning models.



**FIGURE 5:** **LONG SHORT-TERM MEMORY (LSTM) PREDICTION**

To further improve the interpretability of the models, Shapley Additive Explanations (SHAP) was used to examine feature importances in each model. In the SHAP analysis, detailed insights were received regarding how specific input features shaped the predictions by providing an unbiased understanding of decision-making by models. For Random Forest, it was shown via SHAP values that throughput and latency were primarily influential, attesting to how well the model was able to capture the relevant network performance metric. Conversely, Ridge Regression was more evenly weighted across features, signifying its linearity and inability to capture intricate interactions. LSTM and CNN models both displayed dynamic feature dependences, wherein SHAP values demonstrated that sequential patterns were important for LSTM's predictions, and CNN highlighted spatial relationships among features. SHAP analysis not only confirmed the robustness of machine learning models in resource allocation tasks but also reasserted the dominance of Random Forest in striking a balance between accuracy and interpretability. Interpretation of feature contributions using SHAP can be helpful in optimizing network resource allocation strategies so that the most effective parameters are given priority for decision-making in dense edge computing environments.



**FIGURE 6: SHAP ANALYSIS**

**COMPARISON AND INSIGHTS**

On comparison of the models, Random Forest stands out with the best performance, registering the lowest MSE (0.000832), MAE (0.013191), and an outstanding R² (0.98), reflecting high accuracy and variance capture. CNN and LSTM deliver similar outcomes, with a slightly better performance from CNN than LSTM in MSE (0.005458 vs 0.005361) and MAE (0.024325 vs 0.023926), while both models record similar R² values (0.87). On the other hand, Ridge Regression shows greater error measures, MSE (0.006797), MAE (0.027790), and an R² of 0.837995, making it the least effective of the machine learning models. Dynamic Hill Climbing, which is a non-machine learning model, is less effective than machine learning models because it has a limited decision-making strategy and fails to identify complex relationships. Generally, Random Forest is the most robust and accurate model, whereas CNN and LSTM are good substitutes, with Ridge Regression being least efficient for allocating resources in ultra-dense edge computing environments.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | |  | | --- | | **MSE** |  |  | | --- | |  | |  |  |  | | --- | |  |  |  | | --- | |  | | **MAE** | |  | | --- | | **R²** | |
| **Random Forest** | 0.000832 | 0.013191 | 0.983379 |
| **Ridge Regression** | 0.006797 | 0.027790 | 0.837995 |
| **LSTM** | |  | | --- | |  |  |  | | --- | | 0.005361 | | 0.023926 | 0.871390 |
| **CNN** | 0.005458 | 0.024325 | 0.868913 |
| |  | | --- | | **Dynamic Hill Climbing** |  |  | | --- | |  | | 4.478627 | 0.118209 | 0.178488 |

**TABLE 1: MODEL COMPARISON TABLE**

**CONCLUSION AND FUTURE SCOPE**

This study examines the strengths and limitations of various predictive models. For structured data, Random Forest and Ridge Regression demonstrate superior performance, with Random Forest particularly excelling in accuracy, error reduction, and variance explanation. It performs best for processes that include non-linear relationships and feature interactions. In the case of a complex pattern dataset, CNN and LSTM exhibit strong performance with CNN performing marginally better in terms of lower error rates. Ridge Regression is the least effective, failing to cope with data complexity. In order to improve model transparency, Shapley Additive Explanations (SHAP) was utilized to evaluate the importance of features, providing insights into factors affecting resource distribution. SHAP highlighted key network parameters impacting model predictions, offering unambiguous decision-making. The results emphasize feature selection as a key to optimizing performance, alleviating computational load, and enhancing real-world utility.

The non-machine learning Dynamic Hill Climbing algorithm performed worse than ML models, supporting the need for machine learning in edge computing resource allocation. Follow-up studies may examine hybrid approaches where Random Forest is combined with CNNs or LSTMs in order to take advantage of both structured and sequential data. Better computational efficiency by using methods such as model pruning, quantization, and optimal architectures can also improve scalability on resource-limited systems. Subsequent research would leverage SHAP and other Explainable AI (XAI) methods to better understand machine learning model decisions, revealing hidden dependencies and optimizing feature selection. Merging real-time data processing with adaptive learning capabilities might improve model robustness, allowing for dynamic adaptation of allocation strategies to changing network environments. Such developments would further solidify AI use in edge computing, increasing trust and usability in fields such as healthcare, finance, and smart cities.

**REFERENCE**

1. Aghazadeh R, Shahidinejad A, Ghobaei‐Arani M. Proactive content caching in edge computing environment: A review. Software: Practice and Experience. 2023 Mar;53(3):811-55.
2. Duc TL, Leiva RG, Casari P, Östberg PO. Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey. ACM Computing Surveys (CSUR). 2019 Sep 13;52(5):1-39.
3. Sharkh MA, Jammal M, Shami A, Ouda A. Resource allocation in a network-based cloud computing environment: design challenges. IEEE Communications Magazine. 2013 Nov 11;51(11):46-52.
4. Chen J, Wang Y, Liu T. A proactive resource allocation method based on adaptive prediction of resource requests in cloud computing. EURASIP Journal on Wireless Communications and Networking. 2021 Feb 6;2021(1):24.
5. Shakarami A, Shakarami H, Ghobaei-Arani M, Nikougoftar E, Faraji-Mehmandar M. Resource provisioning in edge/fog computing: A comprehensive and systematic review. Journal of Systems Architecture. 2022 Jan 1;122:102362.
6. Alsadie D. A Comprehensive Review of AI Techniques for Resource Management in Fog Computing: Trends, Challenges and Future Directions. IEEE Access. 2024 Aug 21.
7. El Khatib RF, Elsayed SA, Zorba N, Hassanein HS. Optimal proactive resource allocation at the extreme edge. InICC 2022-IEEE International Conference on Communications 2022 May 16 (pp. 5657-5662). IEEE.
8. Elbamby MS, Bennis M, Saad W, Latva-Aho M, Hong CS. Proactive edge computing in fog networks with latency and reliability guarantees. EURASIP Journal on Wireless Communications and Networking. 2018 Dec;2018:1-3.
9. Kim M, Cho H, Cui Y, Lee J. Joint service caching and computing resource allocation for edge computing-enabled networks. IEEE Transactions on Wireless Communications. 2023 Jun 13;22(12):9029-44.
10. Rublein C, Mehmeti F, Mahon M, La Porta TF. Improved Methods of Task Assignment and Resource Allocation with Preemption in Edge Computing Systems. arXiv preprint arXiv:2403.15665. 2024 Mar 23.
11. Sirapangi MD, Gopikrishnan S. MAIPFE: An Efficient Multimodal Approach Integrating Pre-Emptive Analysis, Personalized Feature Selection, and Explainable AI. Computers, Materials & Continua. 2024 May 1;79(2).
12. Seng JK, Ang KL, Peter E, Mmonyi A. Artificial intelligence (AI) and machine learning for multimedia and edge information processing. Electronics. 2022 Jul 18;11(14):2239.
13. Tahir HA, Alayed W, Hassan WU, Haider A. Proposed explainable interference control technique in 6G networks using large language models (LLMs). Electronics. 2024 Nov 8;13(22):4375.
14. El-Hajj M. Enhancing Communication Networks in the New Era with Artificial Intelligence: Techniques, Applications, and Future Directions. Network. 2025 Jan 6;5(1):1.
15. Sefati SS, Haq AU, Craciunescu R, Halunga S, Mihovska A, Fratu O. A Comprehensive Survey on Resource management in 6G network based on internet of things. IEEE Access. 2024 Aug 15.