Forecasting of GSM Network Stress Load Using Machine Learning and Time Series Analysis*

Dmytro Petrov[1,†], Volodymyr Shymanskyi[1,*,†] and Hakan Kutucu[2,†]

[1] *Department of Artificial Intelligence, Lviv Polytechnic National University, Ukraine*
[2] *Faculty of Computer and Information Sciences, Department of Software Engineering, Karabük University, Turkey*

Abstract

Accurately forecasting network stress load in GSM infrastructure is essential for resource allocation and service stability. Currently, network parameters are typically configured statically based on engineers' prior experience, which may lead to suboptimal performance and limited adaptability in dynamic operational environments. Stress load prediction enables more informed or automated network adjustments for consistently stable service. This study evaluates LightGBM, LSTM, and TCN models for forecasting HR Usage and Blocking Rate. LightGBM was used as a baseline due to its proven effectiveness in the field, LSTM was chosen for its strength in adjacent tasks, and TCN was selected as a state-of-the-art alternative. While all models performed well on Blocking Rate (0.02%±0.07%), the error for HR Usage was significantly higher (6±4%). An in-depth analysis of the validation set revealed that 18% of base station cells consistently produced high errors across all models. Investigation using SHAP analysis and the Normalised Wasserstein test confirmed that these errors are driven by data drift and unusual feature behaviour rather than model limitations. Hence, increasing model complexity is not a sufficient solution for GSM network data. Instead, robust deployment requires a systemic approach integrating forecasting, OOD detection and human-in-the-loop notifications to manage unreliable forecasting during periods of data drift.

Keywords

ML, DML, time series, GSM, TCN, LightGBM, LSTM, ACF

1. Introduction

The increasing complexity and dynamic nature of modern telecommunication networks require intelligent automation solutions to ensure efficient resource allocation and optimal performance. In GSM networks, stress load can lead to service unavailability and network failures. To address these issues, predictive models capable of forecasting network stress load are essential for enabling proactive management and automation in mobile network infrastructure.

In recent years, machine learning (ML) and deep learning (DL) techniques have demonstrated strong potential in time series forecasting applications [1-4]. Deep learning architectures, including Temporal Convolutional Networks (TCN) [5] and Long Short-Term Memory (LSTM) [6], have shown promising results in capturing temporal dependencies. It is also worth paying attention to the use of neural networks with a fractal structure, which are used both for time series forecasting [7] and for processing images [8] with complex pattern dependencies. However, selecting the most suitable approach for GSM network optimisation remains an open challenge, especially without additional geo-location, which was leveraged in [9], and without the potential network load prediction during unexpected events such as power outages or social unrest, which deviate significantly from historical patterns. However, the last can be effectively predicted using time series [10].

---

* *SmartIndustry 2026: 3rd International Conference on Smart Automation & Robotics for Future Industry, March 26-27, 2026, Lviv, Ukraine*

[1*] Corresponding author.

[†] These authors contributed equally.

dmytro.d.petrov@lpnu.ua (D. Petrov); vshymanskiy@gmail.com (V. Shymanskyi); hakankutucu@karabuk.edu.tr (H. Kutucu)

0000-0001-9108-7903 (D. Petrov); 0000-0002-7100-3263 (V. Shymanskyi); 0000-0001-7144-7246 (H. Kutucu)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

Moreover, the most recent approaches [11] shift from centralised statistical models to Distributed Machine Learning and Federated Learning frameworks, which share the load of training a model on multiple nodes, requiring massive computational overhead and unnecessary software architecture complexity for spaced and complete time series [12, 13]. Furthermore, growing interest in next-generation mobile network technologies [14, 15] has contributed to a stagnation in research on the advancement of 2G infrastructure.

Therefore, this paper investigates the performance of machine learning and deep learning models, specifically Temporal Convolutional Networks (TCN) and LSTM, for a centralised forecasting network stress load. Using real-world data, we evaluate these models to determine the most reliable approach for industrial-scale automation in telecommunication networks for the 2G network. In addition, in-depth performance analysis was performed to evaluate the stability of the approach under irregular time series behaviour.

2. Methodology

In this study, we frame the research task as a regression problem to forecast future numerical load levels using historical patterns. Given a time series of past daily load and base station configurations, the goal is to develop a machine learning model that could forecast future values across different base stations. To achieve this, we preprocess the historical data, select a baseline and a specialised time series model, and evaluate their performance.

2.1. Data

In this study proprietary time series data set was used, which consists of 16 feature columns and 2 target columns that contain daily statistics and settings of almost 3000 different GSM base stations over 3 years and a 10-month period. Overall, data was provided with 18 columns and 3 463 000 samples that capture the daily state and load of base nodes.

The data comes from real-world historical data. Feature space consists of the number of transreceivers, daily general processed traffic, traffic used in full rate connection and traffic used in half rate, daily volume of GPRS data, volume of EDGE data, control channel traffic and connection success rates statistics. The target columns include the daily blocking rate and half-rate usage rate.

Since all the traffic data is provided in the sum of Erlang over one day, these statistics can hardly provide insights into the load on the base station. The erlang may have the same value regardless weather the daily load was evenly distributed or had a spike in a busy hour. Thus, predicting blocking (when the system couldn’t process a new call) and the percentage of all load processed in half-rate mode (when the number of connections exceeded the threshold to process it in the full rate mode) were considered as target values.

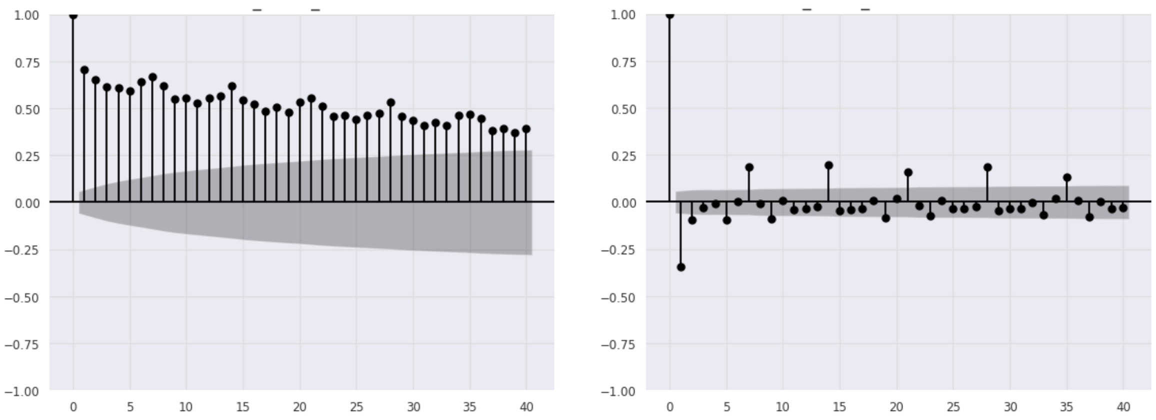

Figure 1: Averaged Autocorrelation Function for BS traffic (left) and its first derivative (right).

In order to avoid data leak cells were sorted by the first and the last date. So that after splitting cells on train and validation, the cells that were put into operation later were mainly or completely presented in the validation set. Next, samples were split by cells and a split date. Dates range from 2021-01-26 to 2024-10-09. The split date is 2024-05-01 in order to split the data into 40 months of training and 4 for validation and testing.

Each cell time series was lagged separately to create a prediction data set and used as a separate series to train or test on.

In order to determine the optimal look-back window for the models to use, we conducted a series of autocorrelation analyses of the data. To visualise more than 2500 separate time seriases each of them was separetely applyed to the autocorrelation function (ACF). Then the ACF results were averaged for each parameter and target that was analysed.

According to the average ACF results on Figure 1, all features that represents work load on a base station has 7 day long seasonality. Hence, it were desided to choose a look-back period in 7 records since it fully covers a season and will provide enough context.

2.2. Models

To predict the future stress load on a base station, several machine learning models were selected. LightGBM was chosen as the baseline due to the success of similar gradient boosting models in time series forecasting within the GSM domain, as demonstrated in [16]. For time series forecasting, two deep learning models known for their strong performance in this domain were chosen: LSTM [17, 18] and Temporal Convolutional Network (TCN).

LightGBM was configured as a multi-output regressor, where each target output was learned independently, and the depth of the trees was not limited. The LSTM and Temporal Convolutional Network models were set to predict pairs of blocking and half-rate usage.

All three models were designed to predict values one day ahead, as described in the data section.

2.3. Evaluation metrics

Mean Squared Error (MSE) was used as the loss function for training all three models. In the same way, early stopping was based on validation MSE [19].

Validation metrics were calculated over each series, as well as base station historical data, separately and then the mean value over all cells. In that way result can be analysed in detail for each node as well as correctly represent generalisation capabilities with its STD over validation and test cells.

2.4. Evaluation metrics

Firstly, to determine the optimal input window (look-back period), we performed an Autocorrelation Function (ACF) analysis on the raw time series. As noted by [20], the ACF measures the correlation between a series and its lagged version. To identify the dominant seasonality across the entire dataset, we calculated the Average ACF. This was achieved by computing the ACF for each individual base station cell and then averaging the correlation coefficients at each lag across the N cells:

$$ACF = \frac{1}{N} \sum_{i=1}^{N} ACF_i(k), \tag{1}$$

where k is for time lag, N is the total number of time series, $ACF_i(k)$ is the autocorrelation coefficient for cell i at lag k.

The analysis revealed a significant spike at k multiple of 7, representing a strong 7-day periodicity at daily resolution. Consequently, it was fixed as the model context window of 7 days to capture these weekly cycles.

Secondly, SHAP (Shapley Additive Explanations) was utilised to interpret the decision-making process of the TCN and LSTM models, as was successfully done in [21]. SHAP assigns each feature an importance value for a specific prediction by calculating the contribution of that feature across all possible combinations of inputs. These values were used to compare feature importance between "normal" prediction segments and high-error segments to determine if the models adapted their logic during periods of high volatility.

Finally, to quantify the distribution shift between the training and validation sets, the Normalised Wasserstein Distance (also known as Earth Mover's Distance) [22] was employed. This metric measures the cost of transforming one probability distribution into another, which requires no prior knowledge for detecting anomalies, as applied in anomaly detection models [23].

For each feature, the distance was calculated to identify which variables drifted most significantly. A higher distance score indicates a more severe Out-of-Distribution (OOD) distribution, signalling that the data has changed beyond the “familiar” to the model set [24].

3. Experimental Results

LightGBM, LSTM and TCN were trained on time series of historical load and configurations of 2364 base stations to predict the next day blocking rate and half rate usage percentage.

Despite the fact that the LightGBM was selected as a baseline model, it suppressed the deep ML models. Table 1 shows the LGBM result over all test cells in comparison to the other models. LightGBM achieved the lowest MSE and RMSE. It indicated that this model provides the most accurate stress load prediction among the tested models, except for insignificantly less consistent HR Usage Rate prediction accuracy. This may be due to the fact that most of the base stations have similar configurations and load patterns that help the model to fit more precisely to the historical data. However, higher RMSE STD suggests that some new or unfamiliar base station traffic patterns make it more difficult to generalise compared to TCN.

LSTM presented the worst results among models. However, the biggest difference (Blocking RMSE std) in forecasting accuracy is only 0.4% less accurate compared to the LSTM.

TCN acquired similar errors with a difference of just over 0.5% (HR Usage RMSE) compared to the LightGBM. while producing more stable results with the lowest STD for HR Usage Rate. The results suggest that the convolutional model achieved better generalisation over data with 0.19% more stable predictions. However, the blocking rate shows 0.12% higher mean error and a difference in STD across the test data on 0.08% higher.

Table 1
Error Comparison for 7 Day Context Window

Model	HR Usage Rate				Blocking Rate			
	MSE	std	RMSE	std	MSE	std	RMSE	std
LightGBM	**0.0048**	**0.0070**	**0.0534**	0.0445	**0.0001**	**0.0003**	**0.0058**	**0.0085**
LSTM	0.0055	0.0073	0.0592	0.0443	0.0002	0.0007	0.0081	0.0125
TCN	0.0052	0.0072	0.0586	**0.0426**	**0.0001**	**0.0003**	0.0070	0.0093

3.1. Error analysis

While models performans mostly located under 2% error, HR Usage Rate forecasting shows a wide dispersion of quality. According to Figure 2, the error distribution for HR Usage is nearly identical for all three models.

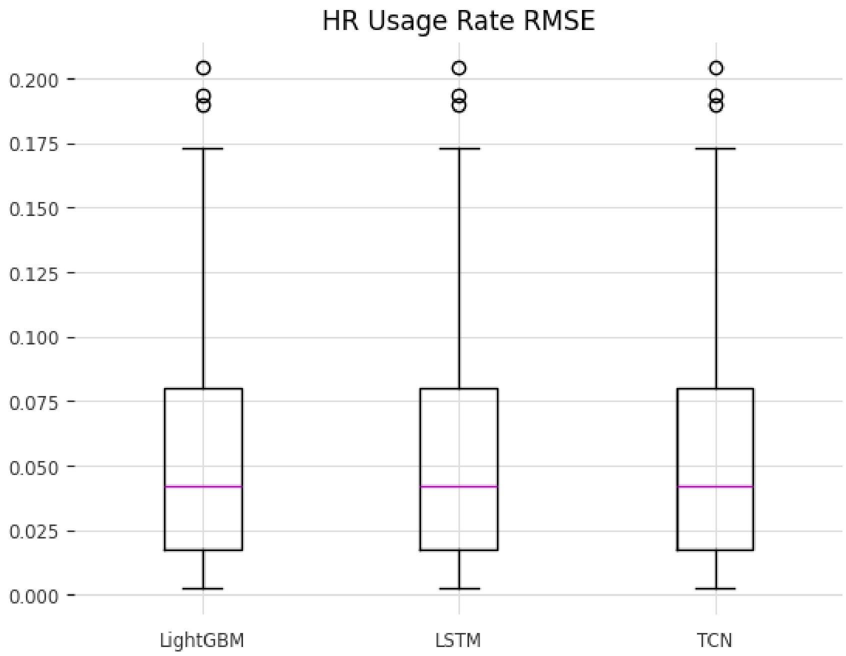

Figure 2: Distribution of RMSE Loss on Test Data Across LightGBM, LSTM, and TCN Models.

A deeper analysis discovered that for all the models, the series that brought errors in the 4th quartile and the upper outliers are mainly the same 57 cells. That 18.27% validation BS that models cannot reliably predict with more than 10% error.

SHAP analysis uncovered that models use the same features with minor rank differences for both high error cells and normal cells. Displated on Figure 3 presents insights that models mostly rely on information from past values of the target value (it's true for both targets), and the biggest impact on the prediction has the latest value of the target. Overall, concentrating on the values of the traffic seems a valuable strategy. Hence, it suggests that data itself has changed distribution with drifted values.

To quantitatively assess the possibility that data drift is present in the validation data, the Wassenstein was utilised. As shown in Table 2, almost all of the input columns were drifted except for the immediate assignment success rate and the temporary block flow rate. This suggests that the models are being tasked with generalising to "unseen" data.

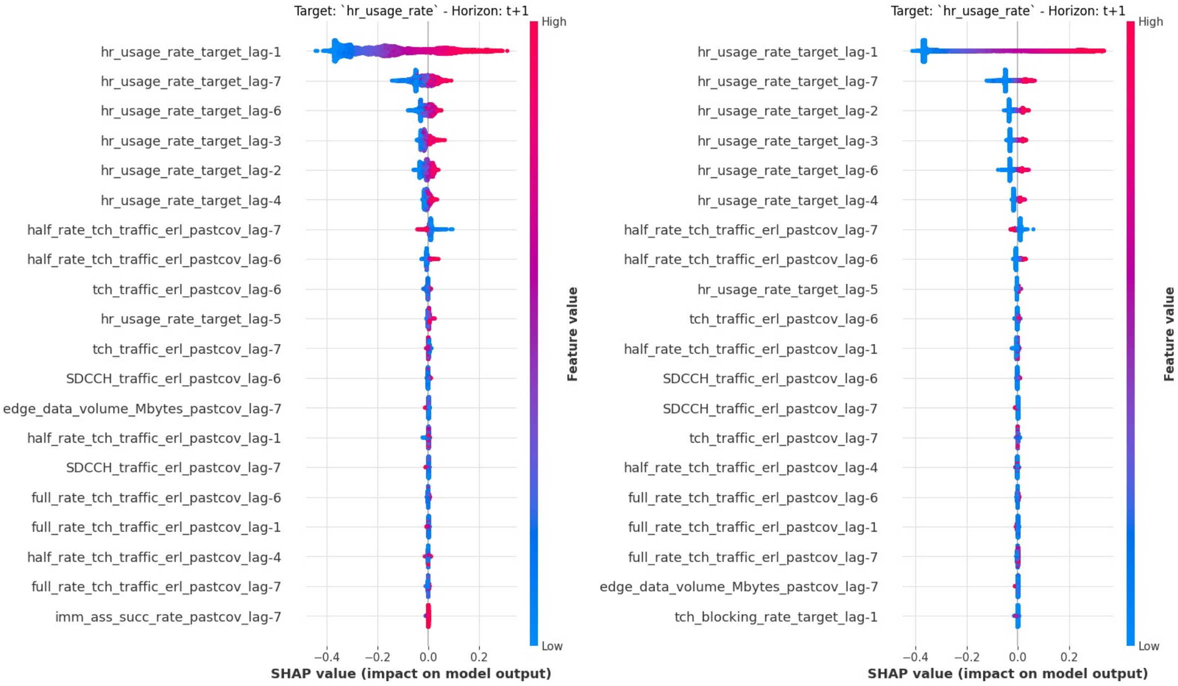

Figure 3: LightGBM SHAP analysis for drifted validation (left) and normal (right).

Table 2
Wassershtein based data drift detection.

Feature	Drift Score	Data Drift	Feature	Drift Score	Status
hr_usage_rate	0.59422	Detected	edge_data_volume_Mbytes	0.41826	Detected
tch_blocking_rate	0.34897	Detected	full_rate_tch_traffic_erl	0.37305	Detected
tbf_drop_rate	0.78199	Detected	sss_proc_succ_rate	0.28629	Detected
tbf_success_ratio	0.78199	Detected	call_setup_success_rate	0.28450	Detected
tch_traffic_erl	0.70040	Detected	gprs_data_volume_Mbytes	0.25557	Detected
half_rate_tch_traffic_erl	0.58999	Detected	ass_succ_rate	0.21422	Detected
dropped_call_rate	0.45835	Detected	imm_ass_succ_rate	0.08939	Not Detected
SDCCH_traffic_erl	0.44026	Detected	tbf_blocking_rate	0.05147	Not Detected

A further analysis of the high-error segments revealed that performance degradation is not stochastically distributed but is instead localised within a specific temporal window, primarily between days 60 and 80 of the validation set. That segment has significantly higher mean values and high-amplitude oscillations in opposing directions.

These characteristics (shown in Figure 4) differ from the dynamics observed in the training set. Firstly, there is a mismatch in the rate of change: within the high-error validation window, the data underwent a 30% HR usage change over a 20-day period. In contrast, the most extreme transition in the training set was a 20% shift occurring over the span of 75 days. This suggests that because the models were trained on slower, more stable transitions, they cannot accurately predict these faster, "unnatural" jumps.

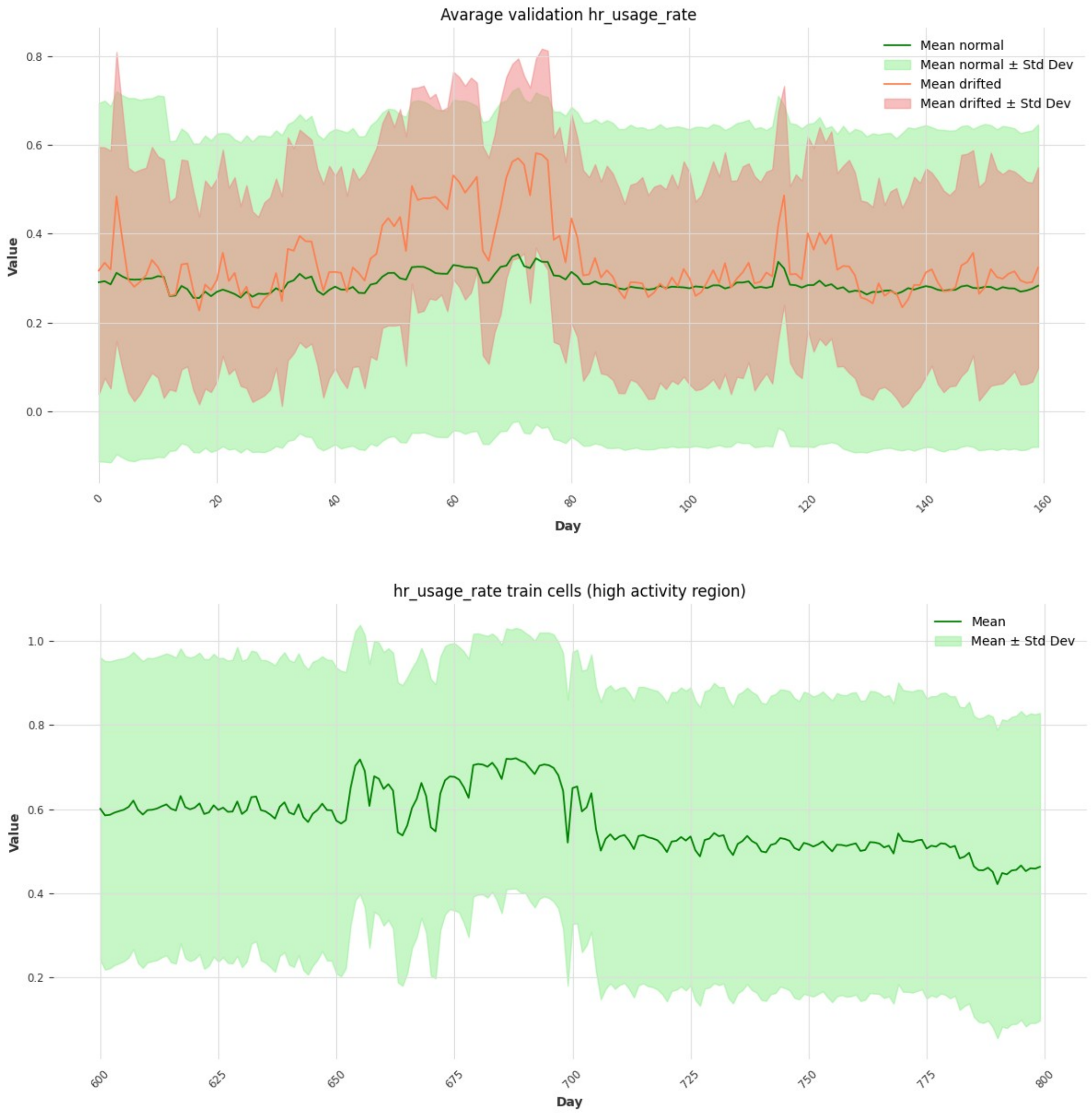

Figure 4: Average HR Usage Rate across cells on validation (up) and the most active region of the train set (down).

Multiple experiments varying the complexity of the LSTM and TCN architectures, including the number of layers and hidden units, were conducted. The results showed in Figure 5, no significant correlation between model complexity and forecast accuracy. Aside from very simple models that lacked the capacity to learn patterns, increasing the size of the models did not improve performance.

This plateau provides further evidence that the high error rates are not caused by architectural limitations. Instead, the errors are driven by the changing nature of the data distribution. Because the model failures are data-driven, simply increasing model parameters is an ineffective solution.

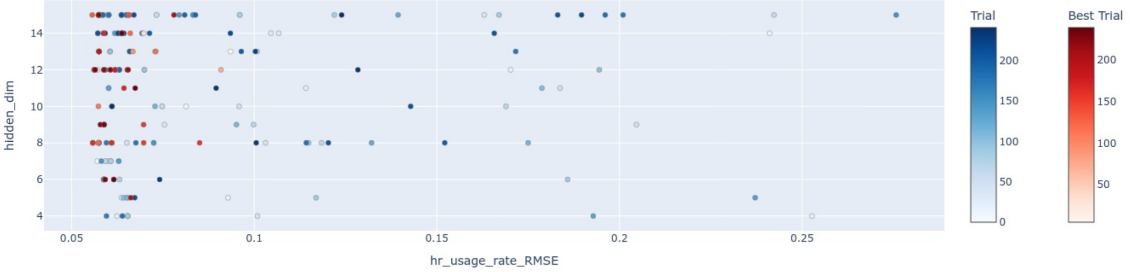

Figure 5: Relationship plot between the number of hidden dimensions of LSTM models and RMSE of the HR usage rate.

3.2. Discussion

The common way to deal with “unseen” data patterns is to add them to the training. While adding anomalous data to the training set might offer a temporary fix, a more systemic approach is needed for real-world deployment. Since increasing model complexity does not solve the data-level issues that were identified, future work should focus on a complex system that may include a "Human-in-the-Loop" (HITL) [25] monitoring framework.

This system would include an automated OOD detection layer to monitor data in real-time. If the system detects a significant distribution shift or a spike in volatility, it would trigger a confidence alert. This notifies human operators to intervene and simultaneously marks the anomalous data for model retraining. This approach shifts the focus from chasing small improvements in MSE to ensuring system reliability through active dataset enhancement.

The HITL approach for GSM management parallels the use of machine learning in Clinical Decision Support Systems (CDSS), where ML models augment rather than replace clinicians by providing real-time alerts and risk assessments [26]. Similarly, this framework allows telecommunication operators to ensure that automated systems support, rather than override, expert intervention during network instability.

Future research should focus on developing advanced techniques to minimise the impact of data drift on forecasting systems. This includes designing robust methods for accurate drift detection and creating effective tools for mitigation, thereby ensuring the forecasting system remains reliable and up-to-date.

4. Conclusions

In this study, we evaluated machine learning and deep learning models for predicting GSM network stress load. LightGBM, LSTM, and TCN were assessed based on their predictive performance using Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) as evaluation metrics.

The results indicate that LightGBM achieved the lowest MSE and RMSE for both HR Usage Rate and Blocking Rate targets. However, TCN and LSTM performed competitively. All three models showed an error for Blocking Rate 0.02±0.07%, while for HR Usage Rate, 6±4% error. Higher HR Usage error led to in depth analysis of the problem of the higher error timeseriases that consists of just over 18% of all BS cells in the validation set.

Further analysis of RMSE distributions revealed that all three models reached a performance plateau due to data characteristics. This was enforced with SHAP analysis that demonstrated that models in prediction for each target mainly look at that target's previous values and workload. In addition to HR usage itself having unusual behaviour, most of the features were drifted according to the normalised Wasserstein test.

In conclusion, increasing model complexity or training data span is not a complete solution for real-life time series. Instead, robust forecasting systems must integrate automatic OOD detection and human-in-the-loop notifications. This approach ensures that when the data distribution shifts beyond the model's learned experience, the system can alert operators and initiate retraining rather than providing unreliable forecasts.

Declaration on Generative AI

During the preparation of this work, the author used Gemini and Grammarly in order to perform Grammar and spelling check. After using these services, the author reviewed and edited the content as needed and takes full responsibility for the publication’s content.

References

[1] W. Joko Hastyo, M. Suryanegara, D. Gunawan, and F. H. Juwono, “A QoE-Driven Framework for Mobile Broadband Spectrum Planning Using Time Series Forecasting,” IEEE Access, vol. 13, pp. 197191–197205, 2025, doi: 10.1109/ACCESS.2025.3633723.

[2] C. Nguyen, A. A. Cheema, C. Kurnaz, A. Rahimian, C. Brennan, and T. Q. Duong, “Deep Learning Models for Time-Series Forecasting of RF-EMF in Wireless Networks,” IEEE open J. Commun. Soc., vol. 5, pp. 1399–1414, 2024, doi: 10.1109/OJCOMS.2024.3365708.

[3] S. Vakaruk, A. Karamchandani, J. E. Sierra-García, A. Mozo, S. Gómez-Canaval, and A. Pastor, “Transformers for Multi-Horizon Forecasting in an Industry 4.0 Use Case,” Sensors, vol. 23, no. 7, 2023, doi: 10.3390/s23073516.

[4] W. A. Aziz, I. I. Ioannou, M. Lestas, H. K. Qureshi, A. Iqbal, and V. Vassiliou, “Content-Aware Network Traffic Prediction Framework for Quality of Service-Aware Dynamic Network Resource Management,” IEEE Access, vol. 11, pp. 99716–99733, 2023, doi: 10.1109/ACCESS.2023.3309002.

[5] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, “Temporal Convolutional Networks for Action Segmentation and Detection,” Nov. 16, 2016, arXiv: arXiv:1611.05267. doi: 10.48550/arXiv.1611.05267.

[6] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[7] V. Shymanskyi, O. Ratinskiy, and N. Shakhovska, “Fractal Neural Network Approach for Analyzing Satellite Images,” Applied Artificial Intelligence, vol. 39, no. 1, p. 2440839, Dec. 2025, doi: 10.1080/08839514.2024.2440839.

[8] N. Shakhovska, V. Shymanskyi, and M. Prymachenko, “FractalNet-LSTM Model for Time Series Forecasting,” CMC, vol. 82, no. 3, pp. 4469–4484, 2025, doi: 10.32604/cmc.2025.062675.

[9] T. Dkhar, C. Pandey, S. A. J. Francis, D. Sinha Roy, and A. Kr Luhach, “NeuroSync: A Novel Neural Network Architecture for Time Series Forecasting of Vehicle Traffic Data Over 5G and Beyond,” Int J Commun Syst, vol. 38, no. 6, 2025, doi: 10.1002/dac.70035.

[10] J. Zheng, D. Feng, Z. Yang, Y. Xiang, H. Zhang, and S. Li, “TransKS: An Anomaly Detection Method for Telecommunication Networks Based on Deep Learning,” IEEE Access, vol. 11, pp. 118048–118060, 2023, doi: 10.1109/ACCESS.2023.3326815.

[11] O. Nassef, W. Sun, H. Purmehdi, M. Tatipamula, and T. Mahmoodi, “A survey: Distributed Machine Learning for 5G and beyond,” Computer Networks, vol. 207, p. 108820, Apr. 2022, doi: 10.1016/j.comnet.2022.108820.

[12] M. U. Afzal, A. A. Abdellatif, M. Zubair, M. Q. Mehmood, and Y. Massoud, “Privacy and Security in Distributed Learning: A Review of Challenges, Solutions, and Open Research Issues,” IEEE Access, vol. 11, pp. 114562–114581, 2023, doi: 10.1109/ACCESS.2023.3323932.

[13] V. Perifanis, N. Pavlidis, R.-A. Koutsiamanis, and P. S. Efraimidis, “Federated learning for 5G base station traffic forecasting,” Computer Networks, vol. 235, p. 109950, Nov. 2023, doi: 10.1016/j.comnet.2023.109950.

[14] M. G. Sumithra and M. Suriya, “Improved spectrum prediction model for cognitive radio networks using hybrid deep learning technique,” Int. J. Intell. Netw., vol. 5, pp. 286–292, 2024, doi: 10.1016/j.ijin.2024.05.003.

[15] I. Tommy, T. Akinola, X. Li, and L. Qian, “Spatio-temporal beam-level traffic forecasting in 5G wireless systems using multi-task learning,” Front. Commun. Netw., vol. 6, 2025, doi: 10.3389/frcmn.2025.1658461.

[16] P. Swetha and R. B. Dayananda, “Improvised_XgBoost Machine learning Algorithm for Customer Churn Prediction,” EAI Endorsed Transactions on Energy Web, vol. 7, no. 30, pp. 1–7, 2020, doi: 10.4108/eai.13-7-2018.164854.

[17] V. Shymanskyi, V. Liaskovets “Cascade model for price and time of car sales prediction”, CEUR Workshop Proceedings, ProfIT AI 2023, Waterloo, Canada, 2024, Vol. 3641, pp. 152–167.

[18] V. Shymanskyi, V. Sabov “An RNN-Based Stock Price Forecasting Model Enhanced by Sentiment Analysis of the Daily Financial News”, Lecture Notes on Data Engineering and Communications Technologies, 2025, Vol. 242, pp. 261 – 270, doi: 10.1007/978-3-031-84228-3_22

[19] J. Heaton, “Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618,” Genetic Programming and Evolvable Machines, vol. 19, Oct. 2017, doi: 10.1007/s10710-017-9314-z.

[20] G. Tunnicliffe Wilson, “Time Series Analysis: Forecasting and Control,5th Edition, by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, 2015. Published by John Wiley and Sons Inc., Hoboken, New Jersey, pp. 712. ISBN: 978-1-118-67502-1,” Journal of Time Series Analysis, vol. 37, p. n/a-n/a, Mar. 2016, doi: 10.1111/jtsa.12194.

[21] D. Sen, “Explainable Deep Learning for Time Series Analysis: Integrating SHAP and LIME in LSTM-Based Models,” Journal of Information Systems Engineering and Management, vol. 10, no. 16s, pp. 412–423, Mar. 2025, doi: 10.52783/jisem.v10i16s.2627.

[22] R. Xie and W. Wei, “Wasserstein-Distance Distributionally Robust Optimization,” in Distributionally Robust Optimization and its Applications in Power System Energy Storage Sizing, R. Xie and W. Wei, Eds., Singapore: Springer Nature, 2024, pp. 213–271. doi: 10.1007/978-981-97-2566-3_6.

[23] S. Liaskovska, I. Izonin, and Y. Martyn, “Investigation of Anomalous Situations in the Machine-Building Industry Using Phase Trajectories Method,” in Advances in Computer Science for Engineering and Manufacturing, Z. Hu, S. Petoukhov, F. Yanovsky, and M. He, Eds., Cham: Springer International Publishing, 2022, pp. 49–59. doi: 10.1007/978-3-031-03877-8_5.

[24] X. Wu, F. Teng, X. Li, J. Zhang, T. Li, and Q. Duan, Out-of-Distribution Generalization in Time Series: A Survey. 2025. doi: 10.48550/arXiv.2503.13868.

[25] F. M. Zanzotto, “Viewpoint: Human-in-the-loop Artificial Intelligence,” Journal of Artificial Intelligence Research, vol. 64, pp. 243–252, Feb. 2019, doi: 10.1613/jair.1.11345.

[26] N. Melnykova, Y. Patereha, S. Skopivskyi, M. Farion, S. Fedushko, and K. Drohomyretska, “Machine learning for stroke prediction using imbalanced data,” Sci Rep, vol. 15, no. 1, p. 33773, Sep. 2025, doi: 10.1038/s41598-025-01855-w.