Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6, 6625-6643
2024
Publisher: Learning Gate
DOI: 10.55214/25768484.v8i6.3415

Clustered temporal memory networks: A hybrid approach for signal strength prediction

Claude Mukatshung Nawej[1*], Pius Adewale Owolawi[2], Tom Walingo[3]
[1,2]Department of Computer Systems Engineering Tshwane University of Technology Pretoria, South Africa; nawejmc@tut.ac.za (C.M.N.) owolawipa@tut.ac.za (P.A.O)
[3]Department of Electrical, Electronics, and Computer Engineering University of Kwa-Zulu Natal Durban, South Africa; walingo@ukzn.ac.za (T.W.).

Abstract: The rapid expansion of 5G networks has revolutionized mobile communication by offering unprecedented speed, low-latency connections, and the ability to support vast numbers of connected devices. However, these advancements bring new challenges in maintaining consistent and reliable signal strength, critical for ensuring optimal Quality of Service (QoS). Traditional models, such as ARIMA, Random Forest (RF), and K-means clustering, struggle to capture the complex, nonlinear, and dynamic behaviour of 5G networks, leading to suboptimal prediction accuracy. In this study, we propose a novel hybrid model, Clustered Temporal Memory Networks (CTMN), which integrates DBSCAN clustering with Long Short-Term Memory (LSTM) networks to improve signal strength prediction in mobile networks. The CTMN model combines DBSCAN's ability to handle spatial variability and outliers in 5G data, combined with LSTM's capacity for modelling long-term dependencies and nonlinear time-series patterns. Our empirical analysis demonstrates that CTMN outperforms traditional methods, achieving up to a 20.82% improvement in prediction accuracy across key performance metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These findings indicate that CTMN provides a scalable, robust solution for enhancing signal strength prediction and optimizing network performance in next-generation mobile networks.

Keywords: *Clustering, Hybrid techniques, Mobile networks, Prediction model, Time-series analysis.*

1. Introduction

The continuous evolution of mobile networks has driven significant advancements in global communication and connectivity. Starting with the first-generation (1G) networks and progressing through to 4G, each new generation has introduced transformative improvements that reshape how people interact with information and each other. From basic voice services in 1G to high-speed data transfer in 4G, mobile networks have expanded their capabilities, enabling a wide range of applications and services. The unveiling of fifth generation (5G) networks represents the latest dive forward in this development, promising ultra-fast speeds, low-latency communication [1]-[3], and the capacity to connect vast numbers of devices simultaneously [4]. These advancements are expected to drive innovations in industries such as intelligent cities, manufacturing, and the Internet of Things (IoT) [5], [6]. However, with the rollout of 5G comes new challenges, particularly in maintaining consistent and reliable Quality of Service (QoS) across diverse and complex environments.

Accurate signal strength prediction is essential for ensuring reliable QoS in mobile networks. Effective prediction is critical for network planning, resource allocation, and optimizing user experiences [7]. However, the increasing complexity of mobile networks, especially with the dense and variable environments of 5G, presents challenges that traditional models struggle to address. Methods such as ARIMA, Support Vector Regression (SVR), Random Forest (RF), and even simpler clustering models like K-means have demonstrated some predictive power [8] - [10] but fall short in environments characterized by high variability, large data volumes, and intricate spatial dependencies.

* Correspondence: nawejmc@tut.ac.za

These traditional models are typically linear in nature and struggle with capturing the temporal and spatial dynamics inherent in 5G networks. ARIMA, while effective for linear time-series data, cannot adequately model the nonlinear and stochastic behaviour of modern network traffic [11]. Similarly, Random Forest and SVR, though successful in some scenarios, lack the ability to generalize well in highly dynamic and fluctuating environments like those of 5G. Current clustering approaches, such as K-means, often result in suboptimal clusters that fail to account for spatial dependencies [12], making the model less effective in heterogeneous mobile environments. As a result, the prediction accuracy and reliability of these models can be significantly reduced in real-world 5G applications.

To tackle these challenges, we suggest a novel hybrid framework called Clustered Temporal Memory Networks (CTMN). The CTMN model integrates clustering techniques with advanced time-series modelling to enhance both prediction accuracy and reliability. By segmenting the data into clusters that capture the diverse conditions of mobile environments and applying specialized time-series models (such as Long Short-Term Memory (LSTM)) to each cluster, CTMN aims to overcome the limitations of traditional methods. The objective of this study is to offer a more resilient approach for signal strength prediction in modern advanced mobile networks, particularly in the complex 5G environment.

Several existing hybrid methods have been explored to assess the effectiveness of machine learning systems in predicting QoS in mobile networks. For instance, in [13], a simple K-means clustering model was employed to group the data, followed by a decision tree to predict potential network anomalies. While this approach demonstrated high accuracy in simulated environments, it lacked the ability to handle real-world data, failing to account for actual traffic volumes and active users. In [14], logistic regression and Random Forest models have shown better performance in certain cases, but they still struggle with the nonlinearity and variability present in 5G networks. Deep learning (DL) has attested to be exceptionally successful in prediction jobs [15]-[17]. However, the study in [14] did not compare the results to deep learning techniques

Even more advanced models like Recurrent Convolutional Neural Networks (R-CNN) and Long Short-Term Memory (LSTM) architectures [18], [19] have demonstrated promise, but their effectiveness is limited when applied to real-world data without proper spatial clustering.

The literature suggests that LSTM-based techniques, particularly when combined with other machine learning models, show great potential for handling time-series data in mobile networks. However, studies have shown that clustering methods like K-means often lead to less effective models compared to more spatially aware approaches like DBSCAN. Moreover, while traditional models have achieved acceptable performance metrics such as RMSE and MSE, they have not been optimized for the large-scale and highly variable data sets characteristic of modern mobile networks.

This research introduces the Clustered Temporal Memory Networks (CTMN) model to address these limitations. CTMN integrates clustering with time-series analysis, improving the accuracy and reliability of signal strength prediction. By segmenting the data into meaningful clusters and applying tailored time-series models within each cluster, CTMN significantly outperforms traditional methods. Our empirical analysis shows that CTMN improves prediction accuracy by up to 20.82% across performance metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

In the following sections, we present the dataset and provide a detailed comparison of commonly used machine learning methods for signal strength prediction, followed by a thorough introduction to the CTMN model. These findings demonstrate the potential of CTMN as a robust solution to the complex challenges of signal strength prediction in modern mobile networks, especially in the 5G era.

2. Dataset

The dataset replicates the hardware configuration using a Valve Steam Deck gaming system running DragonOS Focal, a BB60C spectrum analyzer powered by an external USB3 hub, a srsRan software-defined radio (SDR) device, and a BladeRFxA9 software-defined radio (SDR) device. Data collection was performed across 20 Bihar's locations, with measurements taken every 3 minutes and 7 seconds, totaling 1926 time periods. The dataset encompasses various parameters, including timestamp,

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415

latitude, longitude, signal strength (dBm) taken from numerous devices of the like of BB60C, srsRAN, as well as the data throughput (Mbps), latency (ms), and network type. This dataset provides realistic signal metrics for LTE, 3G, 4G, and 5G, and it is publicly available on Kaggle.

3. Methodology

In this section, we begin by outlining the common machine learning (ML) methodologies used for modeling and predicting signal strength in mobile networks. Finally, we present a methodology for the hybrid model and conclude the study.

3.1. Models' Prediction

Predicting the mobile network communication traffic in terms of signal strength is a crucial element in maintaining optimal network performance and ensuring user satisfaction. Accurate signal strength predictions enable network operators to proactively manage and optimize their networks, ensuring that users enjoy reliable connectivity and high-quality service. Signal strength, typically measured in decibels (dBm), is a key indicator of the connection quality between a mobile device and the base station of the network. It directly affects data throughput, quality of communication, and user experience.

By accurately predicting the signal strength, mobile network specialists can further network capacity, increase user experience, plan network expansions, and improve capacity management. This process can be effectively accomplished through ML, a branch of artificial intelligence (AI) focuses on the use of algorithms and data analysis. Recently, ML has become more popular in mobile network applications [20], [21], proving to be more adept than traditional deterministic prediction models for handling complex data [22] - [26].

Signal strength prediction in mobile networks can be approached through three primary ML techniques: classical machine learning, time series methods, and hybrid models, each offering unique strengths for different types of data and prediction challenges.

3.1.1. Classical Machine Learning

Classical machine learning refers to a set of foundational algorithms and techniques that have been widely used in artificial intelligence (AI) and data computation for decades. These algorithms serve as the foundation for various predictive modeling tasks, including regression, classification, clustering, and dimensionality reduction. Classical machine learning techniques are often contrasted with more modern approaches such as deep learning, which relies on neural networks with many layers. These models are not fundamentally designed to handle time-based dependencies, which are key aspects of time series data. Many traditional machine learning methods have been employed, including logistic regression, random forests, decision trees, support vector machines (SVM), and linear regression.

3.1.1.1. Decision Tree

Decision trees are fundamental techniques used in classical ML for classification and regression assignments. They operate by sequentially dividing data into subgroups founded on feature values, creating a model of outcomes and their potential effects that resemble a tree. The construction of a decision tree involves choosing the best feature at each node to split the data, which is performed by minimizing the impurity or maximizing the information gain. The most common impurities used are Gini Impurity, and Entropy.

3.1.1.2. Logistic Regression

Logistic regression (LR) is a foundational machine learning algorithm commonly applied to binary classification applications. Regardless of the denomination suggesting regression, LR is a classification technique. It works by approximating the probability that a known input fits into a particular class, typically represented as binary outcomes such as 1 or 0, Yes or No, or True or False. The algorithm completes this by using a logistic (or sigmoid) code to a linear combination of the input features, converting the result into a probability score. The process starts by calculating a linear combination of

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415

the input features, which is then passed through a sigmoid function to predict the likelihood of an input being classified into a specific category.

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \quad (1)$$

where x_i is the feature values, β_i is the coefficient of the features, and β_0 is the intercept. The output z is transformed into a probability using the sigmoid function:

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (2)$$

The sigmoid function converts any real-valued number into a value between 0 and 1, which represents the probability that a given instance belongs to the positive class. When the output of the sigmoid function is 0.5, the instance is assigned to positive class (1). If the output is less than 0.5, the instance is assigned to the negative class (0).

3.1.1.3. Linear Regression

Linear regression was used to predict a continuous dependent variable using one or more independent variables. It assumes a linear correlation between the dependent variable z and the independent variable x.

3.1.1.4. Random Forest

Random Forest is a machine learning method used for both classification and regression tasks. It operates by building multiple decision trees during training and then providing either the mode of the classes for classification or the average prediction for regression from the individual trees. Random Forest is highly valued for its ability to improve prediction accuracy and reduce overfitting, making it more reliable than a single decision tree. In classification, the model predicts the class that receives the majority vote from all trees.

$$\hat{y} = mode\left(y_1, y_2, \ldots, y_n\right) \quad (3)$$

where y_i is the class predicted by the i^{th} tree.

For regression, Random Forest predicts the average of the outputs from all trees:

$$\hat{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \quad (4)$$

Where y_i is the output predicted by the i^{th}.

3.1.1.5. Support Vector Machine

The Support Vector Machine (SVM) is a highly effective and flexible supervised learning algorithm commonly used for classification, although it can also handle regression problems. The fundamental concept of SVM is to find the ideal hyperplane that divides distinct sets in the feature space, boosting the margin between them. This makes SVM particularly powerful in high-dimensional spaces, and it has applications in fields such as text classification, image recognition, and bioinformatics. SVM aims to solve the following optimization problem: For a linear SVM, the optimization problem is:

$$\min_{w,b} \frac{1}{2} \|W\|^2 \quad (5)$$

Subject to: $y_i(w \cdot x_i + b) \geq 1, \forall i$

where b is the bias term, w is the weight vector, and (x_i, y_i) are the data points and their labels.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415

For non-linearly separable data, slack variables ξ_i are introduced to allow for misclassifications, leading to a soft-margin SVM:

$$\min_{w,b,\xi,} \frac{1}{2} \|W\|^2 + C \sum_{i=1}^{n} \xi i \quad (6)$$

Subject to: $y_i(w \cdot x_i + b) \geq 1 - \xi i\ ,\ \xi i \geq 0, \forall i$

where C is a regularization constraint that controls the trade-off between maximizing the margin and minimizing classification error.

3.1.2. Time Series Analysis

Time series models are specifically designed to analyze and forecast sequential and time dependent data. These models naturally consider the order and chronological structure of data points, making them particularly suited for tasks in which the timing and sequence of events are crucial. Suitable time series models, such as auto regressive integrated moving average (ARIMA), seasonal ARIMA (SARIMA), holt-winters (Exponential Smoothing), Prophet, and advanced models like Long Short-Term Memory (LSTM) networks, are used to capture long-term dependencies.

3.1.2.1. ARIMA

ARIMA, which stands for the Autoregressive Integrated Moving Average, is a commonly used statistical technique for forecasting time series data. It works particularly well with data that exhibit trends and can be transformed into a stationary series through differencing. The ARIMA model integrates three key elements: autoregressive (AR), integration (I), and Moving Average (MA). The final model equation incorporates the following components:

$$\text{ARIMA(p, d, q):}$$

$$\mathrm{Yt} = \mathrm{c} + \phi 1\mathrm{Yt} - 1 + \phi 2\mathrm{Yt} - 2 + \cdots + \phi \mathrm{pYt} - \mathrm{p} + \varepsilon \mathrm{t} + \theta 1\varepsilon \mathrm{t} - 1 + \theta 2\varepsilon \mathrm{t} - 2 + \cdots + \theta \mathrm{q}\varepsilon \mathrm{t} - \mathrm{q} \quad (7)$$

where Y_t represents the differenced data and incorporates both the autoregressive and moving average terms. Fitting an ARIMA model involves estimating the parameters ϕ and θ using techniques such as the maximum likelihood estimation. Once the model is fitted, it can be applied to forecasts based on historical data to provide insights into future trends.

3.1.2.2. Holt-Winters, and Prophet

The Holt-Winters model is a classical approach that is best suited for time series data with regular seasonal patterns and trends. This method is simple and effective for short-term forecasting. The Prophet model, on the other hand, is a flexible, modern approach that employs complex time series data with multiple seasonality's and holiday effects. It provides an intuitive interface for capturing and forecasting time series with robust handling of the missing data and outliers. Both models are valuable tools in time series forecasting toolkits, each with their strengths and applicable use cases. Prophet models the time-series data as an additive combination of the following components:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_{\mathrm{t}} \quad (8)$$

Where: g(t) is the piecewise linear for trend:

$$g(t) = \ g(t) = (k + a(t)^T \delta)t + (m + a(t)^T \gamma) \quad (9)$$

or logistic growth model for trend:

$$g(t) = \frac{C}{1 + \exp\left(-k(t - m)\right)} \quad (10)$$

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415

With C the carrying capacity.

The periodic component for seasonality modeled using Fourier series is:

$$s(t) = \sum_{n=1}^{N} \left(a_n cos\left(\frac{2\pi nt}{P}\right) + b_n sin\left(\frac{2\pi nt}{P}\right)\right) \quad (11)$$

where P is the period of seasonality, and a_n and b_n are coefficients learned from the data.

ε_t: $the\ error\ term$Forest performance.

a) LSTM:

LSTM networks offer a robust solution for learning and estimating the complex multi-dimensional characteristics of dataset. Their architecture, characterized by gating mechanisms, enables them to effectively maintain and update information over long sequences effectively [27], [28].
These gates, forget, input, update, and output gates or layers learn when provided with both new inputs, x_t and previous output connections h_{t-1} from multiple time intervals.

- Forget: controls which data from the memory cell (pipeline) must be chopped off.

$$f_t = \sigma\ (w_f h_{t-1} + w_f x_t) \quad (12)$$

- Input: decides what new data to retain in the memory pipe.

$$i_t = \sigma\ (w_i h_{t-1} + w_i x_t) \quad (13)$$

A tanh layer follows, creating a vector of fresh contender values, $\check{C}_t$, which may be included in the state.

$$\check{C}_t = tanh\ (w\check{c} h_{t-1} + w\check{c} x_t) \quad (14)$$

- Update: In this procedure, the forget vector is first multiplied pointwise by cell state. After the return of the input gate is acquired, a pointwise additive is executed to amend the cell state with fresh parameters that the machine learning algorithm judges important.

$$C_t = f_t C_{t-1} + i_t * \check{C}_t \quad (15)$$

- Output: Concludes what would make the next hidden state.

$$o_t = \sigma\ (w_o h_{t-1} + w_o x_t) \quad (16)$$

$$h_t = o_t * tanh(C_t) \quad (17)$$

This architecture makes LSTMs a fundamental tool in modern machine learning for tasks that need to understand and predict sequences and capture patterns that span across different time scales.

3.1.3. Hybrid Approaches

Classical machine learning models and traditional time-series approaches struggle to handle the complexities of advanced mobile networks of the likes of 5G networks due to their inability to capture nonlinear relationships, limited temporal understanding, inefficiency with high-dimensional data, and poor adaptability to real-time conditions In modern mobile network planning, hybrid-based prediction models are considered game changers because they often produce more accurate results than classical or time-series models alone. Hybrid approaches combine the strengths of different models while mitigating their limitations. For instance, time series models can extract features such as trends and seasonality, which are then fed into classical machine learning models for further analysis and prediction. Additionally, regression can be combined with classification techniques, and ensemble approaches can blend the predictions from both time series models and ML models to boost precision and robustness.

The hybrid methodology employed in this study follows several key stages: data processing and feature extraction, clustering technique development, preparation of data for LSTM, training of LSTM models for each cluster, and evaluation of model performance. The pseudocode below provides a detailed overview of the steps involved in the proposed method.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415

```
BEGIN

// 1_IMPORT REQUIRED LIBRARIES & DATASET

INPUT : GET LIBRARIES

        GET DATASET

            Load dataset '/path/data.excel' into DataFrame (df)

            Convert 'Timestamp' column to datetime

            Sort DataFrame (df) by 'Timestamp'

            Extract features 'hour', 'dayofweek', 'month' from 'Timestamp'

            Extract 'Signal Strength (dBm)' column into variable (signal_data)

        END

// 2_PRE-PROCESSING

For each feature in dataset

    IF missing_value_count (feature) is less than threshold

        Call fill_missing_value(mean)

    ELSE

        Call remove_records

ENDIF

ENDFOR

// 3_ SCALE

  Initialize StandardScaler

  Scale features 'Signal Strength (dBm)', 'hour', 'dayofweek', 'month'

  Store scaled features in variable (df_scaled)

// 4_DBSCAN CLUSTERING

  Initialize DBSCAN with eps and min_samples
```

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415

```
    Apply DBSCAN clustering to (df_scaled)
    Assign cluster labels to DataFrame (df['Cluster'])
    Filter DataFrame to remove records where cluster
== -1 (noise points)
    Display cluster distribution in DataFrame
// 5_ SEQUENCE PREPARATION FOR LSTM
    Function create_sequences (data, time_steps):
        Initialize empty lists X, y
        For each index from 0 to len(data) - time_steps:
            Append time_step sequence to X
            Append next value to y
        Return X, y
END FUNCTION
    Set time_steps variable (e.g., 10)
    Initialize dictionary 'cluster_data' for sequences
    For each unique cluster in DataFrame
(df['Cluster']):
        Filter data points for the cluster
        Create sequences using create_sequences
        Store sequences in 'cluster_data'
    ENDFOR
// 6_ TRAIN – TEST SPLIT
    Initialize dictionary 'train_test_data'
    For each cluster in 'cluster_data':
        Split sequences into training (80%) and testing
(20%) sets
        Store training and test data in 'train_test_data'
    ENDFOR
// 7_ BUILD AND TRAIN LSTM MODEL
```

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415

```
    Function build_lstm_model (input_shape):
        Initialize Sequential model
        Add LSTM layer with 50 units, return_sequences=True
        Add Dropout layer with 20% dropout
        Add LSTM layer with 50 units, return_sequences=False
        Add Dropout layer with 20% dropout
        Add Dense layer with 25 units
        Add Dense output layer with 1 unit
        Compile model with 'adam' optimizer and 'mean_squared_error' loss
        Return model
    END FUNCTION
    Initialize dictionary 'models'
    For each cluster in 'train_test_data':
        Get X_train, X_test, y_train, y_test for the cluster
        Build LSTM model using input shape from training data
        Train LSTM model on training data for specified epochs (e.g., 10)
        Store trained model in 'models'
    ENDFOR
// 8_ MODEL EVALUATION
    For each cluster in 'models':
        Get X_test and y_test for the cluster
        Use the LSTM model to predict on X_test
        Calculate performance metrics:
            MAE, MSE, RMSE, R2 score
        Print performance metrics for the cluster
```

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415

Plot actual vs predicted signal strength for the cluster

ENDFOR

3.2. Performance Metrics

Regression analysis and classification analysis are two different techniques that can be used to predict the signal strength in mobile networks. In this study, the focus was on predicting the exact numerical value of the signal strength at a specific location or time, rather than categorizing it into discrete classes. Consequently, the problem was framed as a regression task, and the performance of the models was evaluated using metrics such as the Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2).

MAE computes the average of the absolute differences between anticipated and actual values. In this context, y^ represents the model's forecasts, y represents the actual values of the data points, and n is the total number of samples. MAE offers an intuitive measure of the error in the same units as the target variable, with lower MAE values indicating better model accuracy.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |(y_i - \hat{y}_i)| \quad (18)$$

Conversely, MSE calculates the average of the squared differences between the predicted and actual values. Similar to the MAE, lower MSE values indicate better model performance.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (19)$$

The RMSE is the square root of the MSE, which aligns the error metric with the unit of the target variable. Lower RMSE values suggest that the model fits the data more accurately.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \quad (20)$$

Finally, R-squared (R^2) quantifies the proportion of variance in the dependent variable that can be predicted based on the independent variables. The values span from 0 to 1, with higher values which implies that the model accounts for a larger share of the variance in the target variable. An R^2 value of 1 represents a perfect fit, whereas an R^2 value of 0 indicates that the model explains none of the variances. If R^2 is negative, it indicates that the model performs worse than the horizontal line representing the mean of the data.

4. Results and Analysis

An Acer TravelMate laptop equipped with an Intel® Core™ i7 8th Gen processor and 16 GB RAM was used to preprocess the data and train the models. Additionally, Jupyter Notebook via Anaconda, running on Windows 10, was employed to implement the machine learning framework using Python libraries such as NumPy [29], Pandas [30], Matplotlib [31], Seaborn [32], and Scikit-learn [33], [34]. The performance of the different models was compared using the evaluation metrics mentioned earlier. It is also important to ensure that the test data are representative of the training data, which can be visualized by mapping the distributions of the key features and comparing them between the training and test sets. For all models, 80% of the data were used for training, whereas the remaining 20% were reserved for testing.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415

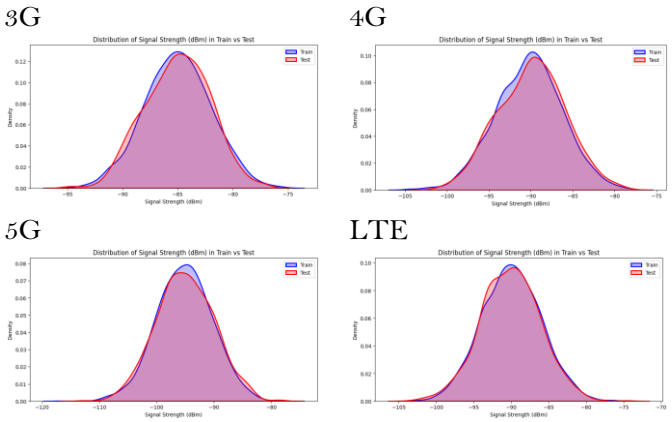

Figure 1.
Signal strength distribution.

When looking at the distributions in Fig. 1, the test set appears to be a subset of the training set, precisely what is needed for ML to work.

4.1. Classical Machine Learning Results and Analysis

In classical machine learning, the performances of various ML models (Decision Tree, Linear Regression, Random Forest, and Support Vector Machine) across various network technologies (3G, 4G, 5G, and LTE) can be interpreted based on the subsequent metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2).

Table 1.
3G.

Model	MAE	MSE	RMSE	R^2
Decision Tree	3.43944	18.44601	4.29488	-1.11043
Linear Regression	2.38735	8.73654	2.95576	0.00043
Random Forest	2.90180	13.25067	3.64015	-0.51603
Support V. Machine	2.38946	8.77671	2.96255	-0.00415

Table 2.
4G.

Model	MAE	MSE	RMSE	R^2
Decision tree	4.51068	31.48664	5.61130	-0.96177
Linear regression	3.23185	16.13102	4.01634	-0.00504
Random forest	3.91765	23.32881	4.82999	-0.45350
Support V. machine	3.22228	16.09559	4.01193	-0.00283

Table 3.
5G.

Model	MAE	MSE	RMSE	R^2
Decision Tree	5.52770	49.30279	7.02159	-0.93657
Linear Regression	4.04576	25.44547	5.04435	0.00052
Random Forest	4.78050	36.71026	6.05890	-0.44195
Support V. Machine	4.04390	25.43296	5.04311	0.00101

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415

Table 4.
LTE.

Model	MAE	MSE	RMSE	R^2
Decision Tree	4.49545	31.79257	5.63849	-1.01762
Linear Regression	3.15644	15.78044	3.97246	-0.00145
Random Forest	3.80503	22.42947	4.73598	-0.42342
SVM	3.16611	15.81356	3.97663	-0.00356

Across all the network technologies analyzed, 3G, 4G, 5G, and LTE, Linear Regression steadily emerged as the best performing model for predicting signal strength. It provided the most accurate predictions with the lowest error metrics and the least negative (or slightly positive) R^2 scores. In contrast, Decision Tree models demonstrate poor performance across the board, indicating that they are not well-suited for this particular task. Random Forest and Support Vector Machine models show moderate performance, falling somewhere between the effectiveness of Linear Regression and the inadequacy of the Decision Trees.

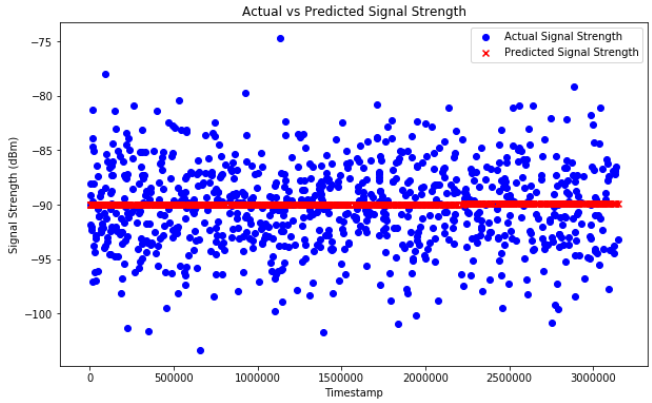

Figure 2.
LTE Linear Regression signal strength prediction.

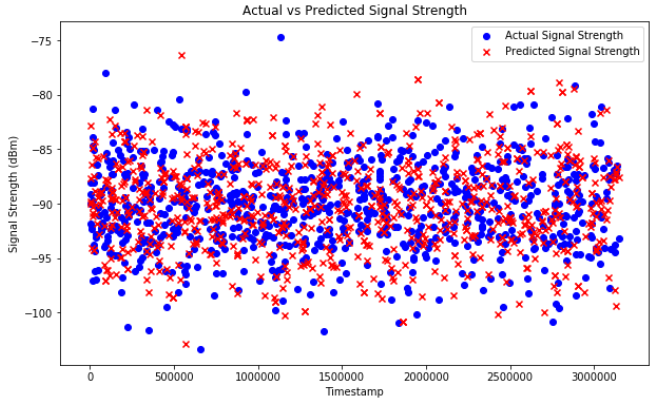

Figure 3.
LTE Decision Tree signal strength prediction.

This analysis indicates that for tasks involving the prediction of signal strength, Linear Regression is the most reliable model to use across different network types.

4.2. Time Series Results and Analysis

For the time-series, the results in Tables V through VIII present a comparative analysis of the performance of several time-series models: ARIMA, Holt-Winters, Prophet, and LSTM across various

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415

network technologies: 3G, 4G, 5G, and LTE. These tables provide insights into how well each model predicts signal strength in different types of mobile networks.

Table 5.
3G.

Model	MAE	MSE	RMSE	R^2
Arima	1.96226	7.81095	2.79481	-0.09629
Holt-Winters	2.74906	11.90074	3.44974	-0.30255
Prophet	5.1e+11	3.4e+23	5.8e+11	-3.7e+22
LSTM	2.47426	9.16039	3.02661	-0.01619

Table 6.
4G.

Model	MAE	MSE	RMSE	R^2
Arima	3.08521	16.88186	4.10875	-0.18446
Holt-Winters	3.52352	19.22481	4.38461	-0.17002
Prophet	3.25537	16.41435	4.05146	0.00102
LSTM	3.28256	16.91177	4.11239	-0.04093

Table 7.
5G.

Model	MAE	MSE	RMSE	R^2
Arima	4.73714	34.10862	5.84026	0.00045
Holt-Winters	4.18916	27.15697	5.21123	-0.15387
Prophet	3.87824	23.61668	4.85970	-0.00345
LSTM	3.93363	24.22849	4.92224	-0.00014

Table 8.
LTE.

Model	MAE	MSE	RMSE	R^2
Arima	3.69090	18.07970	4.25202	-0.03518
Holt-Winters	5.03822	39.69271	6.30021	-1.46259
Prophet	3.28218	16.29148	4.03627	-0.01074
LSTM	3.11807	15.65761	3.95697	-0.01598

Across all networks, no model consistently performs the best, but LSTM generally shows a more stable performance compared to the others, particularly in the LTE and 5G networks. Prophet performance is inconsistent, with very poor results in 3G networks and better performance in 4G and 5G networks. ARIMA shows a slightly better performance in 5G and LTE networks, with only a positive R^2 score in 5G. Holt-Winters consistently underperforms, particularly in LTE networks. The R^2 scores were predominantly negative across all models and networks, indicating that the models struggled to explain the variance in the signal strength data effectively. This implies potential overfitting or underfitting issues, depending on the model and the network. The MAE and RMSE values further highlight that the model predictions differ significantly from the actual signal strengths, particularly for more complex network technologies such as 5G and LTE.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415

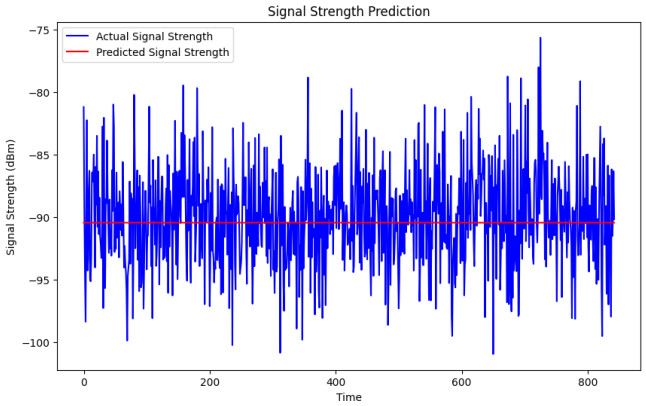

Figure 4.
LSTM signal strength prediction for LTE.

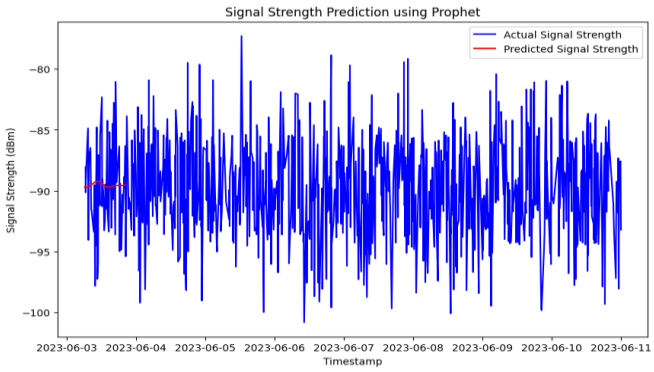

Figure 5.
Prophet signal strength prediction for LTE.

These results highlight the challenges of predicting the signal strength across different network technologies and underscore the need for careful model selection and tuning to achieve better performance.

4.3. Hybrid Approach Results and Analysis

In the context of mobile networks, signal strength is a critical parameter that reflects the quality of the connection between a mobile device and cell tower. Accurately predicting signal strength over time is essential for optimizing network performance, enhancing user experience, and efficiently managing network resources. This study's hybrid prediction approach combines clustering and a time-series model which make it better suited for the dynamic and variable environments of advanced mobile networks like 5G networks. This approach begins with clustering, where we group areas or time periods with similar signal strength patterns, then apply a time series model to each cluster.

4.3.1. Clustering

Clustering, as an unsupervised ML technique, is mostly used for discovering patterns in dataset rather than for making predictions as supervised ML algorithms do. However, clustering can still play a role in prediction in an indirect manner, often by enhancing other ML models or enabling certain types of predictions. K-means, DBSCAN, and agglomerative clustering are three commonly used clustering procedures, respectively suited to different types of data and applications. K-means is ideal for large datasets where the number of clusters is known in advance and the clusters are spherical in shape. DBSCAN is excellent for identifying clusters of arbitrary shapes and handling noise, making it useful

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415

for applications involving complex or noisy data. Agglomerative clustering provides a hierarchical approach, that offers flexibility and interpretability, particularly in smaller datasets or scenarios where the cluster hierarchy is of interest. Clustering facilitates the identification of various patterns and data segments. Figure 6 illustrates the distribution of clusters based on the latitudes and longitudes in this research.

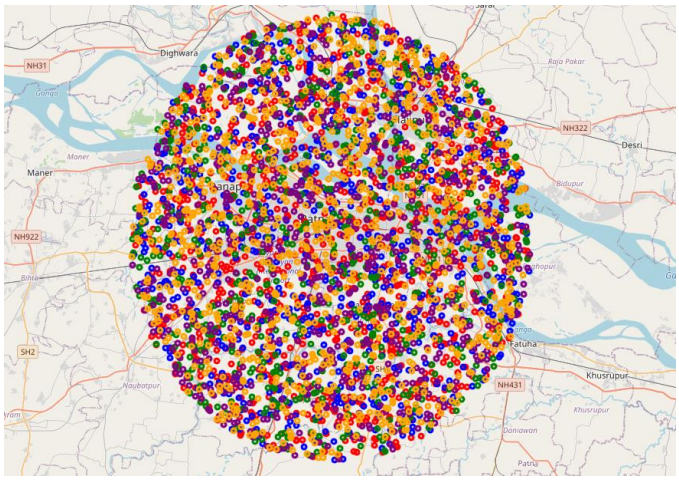

Figure 6.
Signal strength clusters distribution.

To assess the effectiveness of the cluster models, various evaluation metrics are employed, including the Silhouette score, Davies-Bouldin index, and Calinski-Harabasz Index. The Silhouette score quantifies the similarity of an object to its own cluster in relation to other clusters. A higher Silhouette score indicates well-defined clusters, meaning that the data points are well-matched within their assigned clusters and are clearly distinct from those in other clusters.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{21}$$

The Davies-Bouldin index metric evaluates the quality of clustering by assessing the average ratio of intra-cluster distance to inter-cluster separation. Equation (18) shows how to calculate it.

$$DB = \frac{1}{n}\sum_{i=1}^{n} \max_{i \neq j} \left(\frac{s_i + s_j}{d_{ij}}\right) \tag{22}$$

Lower values indicate better clustering performance. The Calinski-Harabasx indicator is employed to assess the clustering algorithms, particularly when the goal is to maximize the separation between clusters while minimizing the spread within clusters. This is particularly useful when comparing the performance of different clustering algorithms or the same algorithm with different numbers of clusters. A higher value implies that the clusters are well-separated and compact, indicating that the clustering solution is effective, and lower values indicate that the clusters may overlap significantly or that the data points within clusters are not well-grouped, indicating a less effective clustering solution.

TABLE IX contrasted the most used clustering models across all networks, and the result shows that DBSCAN outperforms the other models in terms of Silhouette Score across all network types (3G, 4G, 5G, and LTE), indicating it creates more distinct and well-separated clusters.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415

Table 9.
Silhouette score index.

Model	3G	4G	5G	LTE
K-Means	0.3599	0.3644	0.3600	0.3585
DBSCAN	0.4960	0.5128	0.5304	0.5080
Agglomerative	0.2855	0.3025	0.3068	0.3088

The increasing scores from 3G to 5G suggest that DBSCAN may be particularly well-suited for clustering data as network technology advances. Therefore, in this study, to address the temporal and spatial dynamics inherent in modern mobile networks, particularly in 5G networks, we used the DBSCAN clustering model to construct the hybrid model.

4.3.2. The Clustered Temporal Memory Networks (CTMN)

By applying a powerful time series to each cluster, the model can make accurate predictions based on the specific characteristics of each cluster. This research hybrid method, CTMN, combines the benefits of unsupervised learning, through DBSCAN clustering, and supervised learning, via LSTM networks, to produce more accurate and insightful predictions. The process began with clustering. Once clustering has grouped the data into these segments, each cluster can be considered a more homogeneous subset of the overall dataset. For each cluster, an LSTM model was trained. By training separate LSTM models for each cluster, the model learns the unique historical patterns that characterize energy consumption within that cluster. The output shown in TABLE X consists of LTE performance metrics for LSTM, and CTMN in its various clusters.

Table 10.
CTMN clusters VS. LSTM.

Model	MAE	MSE	RMSE	R^2
CTMN_0	3.08798	13.97205	3.73792	-0.16728
CTMN_1	2.69463	10.71239	3.27298	-0.03094
CTMN_3	3.08843	14.43108	3.79882	-0.06563
CTMN_5	2.60127	10.40344	3.22544	-0.00596
LSTM	3.11807	15.65761	3.95697	-0.01598

This is further visualized in the chart in Figure 7, which offers a clearer understanding of the reader.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415

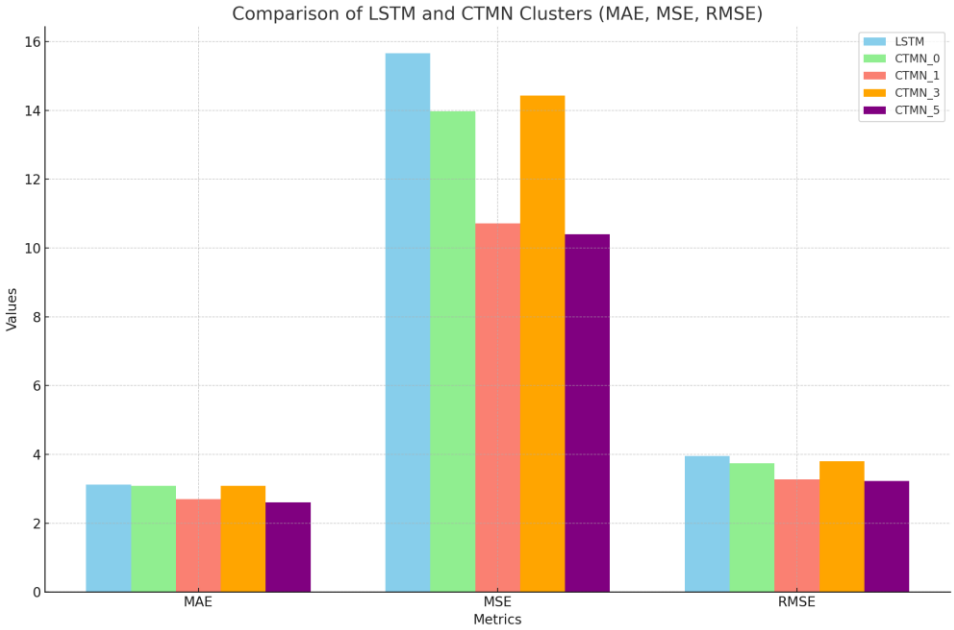

Figure 7.
Comparison of LSTM and CTMN clusters.

The chart visually demonstrates that the CTMN clusters, particularly CTMN_1 and CTMN_5, outperform the LSTM model across these metrics, highlighting the significant improvements achieved by the CTMN model.

5. Conclusion

In conclusion, this study effectively addressed the prediction shortcomings of traditional models by integrating clustering with time-series analysis. By combining DBSCAN's ability to manage spatial variability and outliers in 5G data with LSTM's strength in capturing long-term dependencies and nonlinear time-series patterns, the hybrid model significantly improved prediction accuracy compared to conventional methods, offering a scalable solution for next-generation mobile networks. The methodology can also be applied to other areas dealing with heterogeneous data, such as energy consumption forecasting, financial market analysis, and customer behaviour prediction.

However, while this approach has clear advantages, it also introduces challenges. The quality of the clustering process is crucial, as poor clustering can hinder LSTM performance. Additionally, the complexity of this method requires expertise in both clustering techniques and time-series modelling. LSTM models also rely on sufficient data within each cluster, often necessitating extensive data collection.

Future research should focus on addressing sequential bias in cluster averaging, mitigating the computational overhead of the hybrid DBSCAN-LSTM model, and optimizing LSTM hyperparameters to further improve prediction accuracy.

References

[1] *5G Services Market Size, Share & Trends Analysis Report*, Grand View Research, Pune, India, 2021. [12] H. Singh, S. Gupta, C. Dhawa

[2] C. X. Wang, J. Bian, J. Sun, W. Zhang, and M. Zhang, "A survey of 5G channel measurements and models," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3142–3168, Jan. 2018, doi: 10.1109/comst.2018.2862141.

[3] S. Alraih, R. Nordin, I. Shayea, N. F. Abdullah and A. Alhammadi, "Ping-Pong Handover Effect Reduction in 5G and Beyond Networks," *2021 IEEE Microwave Theory and Techniques in Wireless Communications (MTTW)*, Riga, Latvia, pp. 97-101, 2021 doi: 10.1109/MTTW53539.2021.9607205

[4] M. N. Mahdi, K. S. Mohamed, A. R. Ahmad, and M. A. Subhi, "The vision of 5G and cell-free communication networks in Malaysia," in *Proc. 8th Int. Conf. Inf. Technol. Multimedia (ICIMU)*, IEEE, Aug. 2020, pp. 156–161

[5] C. A. Ionescu, M. T. Fülöp, D. I. Topor, S. Căpuşneanu, T. O. Breaz, S. G. Stănescu, and M. D. Coman, "The new era of business digitization through the implementation of 5G technology in Romania," *Sustainability*, vol. 13, no. 23, pp. 13401, Dec. 2021, doi: 10.3390/su132313401.

[6] A. M. Ramly, N. F. Abdullah, and R. Nordin, "Cross-layer design and performance analysis for ultra-reliable factory of the future based on 5G mobile networks,' *IEEE Access*, vol. 9, pp. 68161–68175, 2021, doi: 10.1109/ACCESS.2021.3078165.

[7] Y. Xu, G. Gui & F. Adachi, "A Survey on Resource Allocation for 5G Heterogeneous Networks: Current Research, Future Trends and Challenges," in *2021 IEEE Communications Surveys & Tutorials*. 2021, pp. 1-1. 10.1109/COMST.2021.3059896.

[8] S. Kassan, I. Hadj-Kacem, S. Ben Jemaa and S. Allio," A Hybrid machine learning based model for congestion prediction in mobile networks," in *2022 IEEE 33rd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Kyoto, Japan, 2022, pp. 583-588

[9] B. S. Shawel, T. T. Debella, G. Tesfaye, Y. Y. Tefera and D. H. Woldegebreal, "Hybrid Prediction Model for Mobile Data Traffic: A Cluster-level Approach," in *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, UK, 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9207655.

[10] N. Moraitis, L. Tsipi, and D. Vouyioukas, "Machine learning-based methods for path loss prediction in urban environment for LTE networks," in *Proc. 16th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2020, pp. 1–6, doi: 10.1109/WiMob50308.2020.9253369

[11] S. Jaffry and S. F. Hasan, "Cellular Traffic Prediction using Recurrent Neural Networks," in *2020 IEEE 5th International Symposium on Telecommunication Technologies (ISTT)*, 2020, pp. 94–98.

[12] B. S. Shawel, T. T. Debella, G. Tesfaye, Y. Y. Tefera and D. H. Woldegebreal, "Hybrid Prediction Model for Mobile Data Traffic: A Cluster-level Approach," in *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, UK, 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9207655.

[13] N. Koursioumpas, L. Magoula, S. Barmpounakis and I. Stavrakakis, "Network Traffic Anomaly Prediction for Beyond 5G Networks," *2022 IEEE 33rd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Kyoto, Japan, 2022, pp. 589-594, doi: 10.1109/PIMRC54779.2022.9977469

[14] I. Hadj-Kacem, S. Ben Jemaa, S. Allio and Y. Ben Slimen, "Anomaly prediction in mobile networks: A data driven approach for machine learning algorithm selection," in *2020 IEEE/IFIP Network Operations and Management Symposium (NOMS 2020)*, Budapest, Hungary, 2020, pp. 1–7

[15] C. Bolchini and A. Bosio, "Resilience of Deep Learning Applications: Where We are and Where We Want to Go," *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Valencia, Spain, 2024, pp. 1-1, doi: 10.23919/DATE58400.2024.10546613.

[16] Alzubaidi, L., Zhang, J., Humaidi, A.J. *et al.* Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data* **8**, 53 (2021). https://doi.org/10.1186/s40537-021-00444-8

[17] Wang Y, Liu L, Wang C. Trends in using deep learning algorithms in biomedical prediction systems. Front Neurosci. 2023 Nov 9;17:1256351. doi: 10.3389/fnins.2023.1256351. PMID: 38027475; PMCID: PMC10665494.

[18] M. R. Tanhatalab, H. Yousefi, H. M. Hosseini, M. M. Bonab, V. Fakharian and H. Abarghouei, "Deep RAN: A Scalable Data-driven platform to Detect Anomalies in Live Cellular Network Using Recurrent Convolutional Neural Network," in *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Herlany, Slovakia, 2020, pp.269–274.

[19] Y. Wang, Y. Guo, Z. Wei, Y. Huang and X. Liu, "Traffic Flow Prediction Based on Deep Neural Networks," in *2019 International Conference on Data Mining Workshops (ICDMW)*, 2019, pp. 210-215

[20] A. Zappone, M. Di Renzo, M. Debbah, T. T. Lam, and X. Qian, "Model-aided wireless artificial intelligence: Embedding expert knowledge in deep neural networks for wireless system optimization," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 60–69, Sep. 2019, doi: 10.1109/MVT.2019.2921627.

[21] N. Moraitis, L. Tsipi and D. Vouyioukas, "Machine Learning-Based Methods for Path Loss Prediction in Urban Environment for LTE Networks," in *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Thessaloniki, Greece, 2020, pp. 1-6, doi: 10.1109/WiMob50308.2020.9253369.

[22] H. Singh, S. Gupta, C. Dhawan and A. Mishra, "Path Loss Prediction in Smart Campus Environment: Machine Learning-based Approaches," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, Antwerp, Belgium, 2020, pp. 1-5, doi: 10.1109/VTC2020-Spring48590.2020.9129444.

[23] S. Ojo, A. Imoize, and D. Alienyi, "Radial basis function neural network path loss prediction model for LTE networks in multi-transmitter signal propagation environments," *Int. J. Commun. Syst.*, vol. 34, no. 3, pp. 1–26, Feb. 2021, doi: 10.1002/dac.4680.

[24] S. Aldossari and K.-C. Chen, "Predicting the path loss of wireless channel models using machine learning techniques in mmWave urban communications," in *Proc. 22nd Int. Symp. Wireless Pers. Multimedia Commun. (WPMC)*, Nov. 2019, pp. 1–6, doi: 10.1109/WPMC48795.2019. 9096057.

[25] I. S. Popool, S. Misra, and A. A. Atayero, "Outdoor path loss predictions based on extreme learning machine," *Wireless Pers. Commun.*, vol. 99, no. 1, pp. 441–460, Mar. 2018, doi: 10.1007/s11277-017-5119-x.

[26] R. He, Y. Gong, W. Bai, Y. Li, and X. Wang, "Random forests-based path loss prediction in mobile communication systems," in *Proc. IEEE 6th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2020, pp. 1246–1250, doi: 10.1109/ICCC51575.2020.9344905.
[27] C. M. Nawej, P. A. Owolawi, T. Walingo, "Towards a realistic comparative analysis of recurrent neural network's methods via long-term memory approaches," in *Book Title*, edition (if not first), Editor's initials. Editor's Surname, Ed. Place of publication: Publisher, Year, page numbers.
[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997
[29] T. E. Oliphant, *A guide to NumPy (Vol. 1)*. Trelgol Publishing, USA, 2006.
[30] W. McKinney, "Data structures for statistical computing in python," *Proc. of the 9th Python in Science Conf.*, vol. 445, pp. 51–56., 2010.
[31] J. D. Hunter, "Matplotlib: A 2D graphics environment," in *Computing in Science & Engineering*, vol. 9, no.3, pp. 90–95, May-June 2007.
[32] A. Lavanya, S. Sindhuja, L. Gaurav and W. Ali, "A comprehensive review of data visualization tools: features, strengths, and weaknesses," *Int. J. Comput. Eng. Res. Trends*, vol. *10, no.* 01, pp.10-20, 2023
[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion and O. Grisel, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, pp. 2825–2830, 2011.
[34] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly Media, Inc., 2022.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 6: 6625-6643, 2024
DOI: 10.55214/25768484.v8i6.3415