ENHANCING USER EXPERIENCE IN HOME NETWORKS WITH MACHINE LEARNING-BASED CLASSIFICATION

Rushat Rai[1] and Thomas Basikolo[2]
[1]Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China, [2]Telecommunication Standardization Bureau (TSB), International Telecommunication Union, Switzerland

NOTE: Corresponding author: Rushat Rai, rushatrai@link.cuhk.edu.hk

Abstract – *With the rapid development of mobile Internet, home broadband has been integrated into people's daily lives, and the market has become increasingly saturated. User experience and broadband quality have become the key factors determining market competitiveness, and consequently, most operators currently are increasing attention to network quality issues and how to improve user experience. This paper proposes an efficient machine learning model to accurately evaluate home user network experiences. The dataset used encompasses network indicator data from 500 anonymized users, and presents a set of formidable challenges including a non-standard sampling rate and time range, an uneven distribution of observations, multiple recorded observations for identical timestamps, a constrained sample size, a subjective definition of Internet experience, and a lack of essential information regarding the data collection setup. Our novel time series characteristic-based method extracts thousands of descriptive statistics from the time series sequences which reveal that, even in the face of the dataset's inherent complexities, our proposed method excels, achieving an impressive 67% validation accuracy. This represents a substantial 3% enhancement over the performance of conventional models on this dataset. Furthermore, we explore the potential of a Recurrent Neural Network (RNN) model, which also yields promising results with a validation accuracy of 58%. It is important to underscore that the performance of the RNN model could be substantially enhanced with a larger dataset. By leveraging these findings, network operators can gain valuable insights towards developing effective machine learning models that proactively identify potential dissatisfied users. This capability will enable operators to implement timely corrective measures, ultimately enhancing the overall user experience. As a result, this proactive approach can significantly reduce customer churn, as users are more likely to remain loyal to a service provider that consistently offers a high-quality network experience.*

Keywords – Deep learning, DPI probe, network indicators, time series classification, user experience

1. INTRODUCTION

The rapid development of mobile Internet has resulted in widespread adoption of home broadband, leading to a highly competitive market [1]-[2]. In this environment, the quality of broadband and user experience plays a critical role in determining market competitiveness. Operators who fail to provide a high quality service risk losing customers to their competitors [3]. Consequently, operators are now focused on identifying network issues, predicting potential problems proactively, and promptly improving user experience. To achieve this, operators utilize a range of tools and techniques to monitor networks for problems such as network congestion, signal interference, and service outages [4]. However, with the increasing complexity and scale of networks, manually detecting and diagnosing all potential issues becomes impossible. As a result, many operators are turning to advanced analytics and machine learning algorithms to detect and predict potential issues before they arise [5].

Machine Learning (ML), a subfield of Artificial Intelligence (AI), focuses on developing algorithms and statistical models that enable computer systems to learn and improve from experience without being explicitly programmed [6]. ML finds numerous applications across various domains, including computer vision, natural language processing, speech recognition, signal processing, and others [7]. It has emerged as a powerful tool for solving complex problems, and its impact is expected to grow with the development of more applications and availability of data.

One such application is home user network classification, where ML can be used to classify a home broadband user's network experience based on factors like latency, packet loss, and throughput [4]. By identifying patterns and characteristics in these features, ML algorithms can make informed assessments of a user's online experience in real time.

A Deep Packet Inspection (DPI) probe is a pivotal component in network segmentation, effectively dividing it into uplink and downlink sides, as visually depicted in Fig. 1. Notably, a significant proportion of network issues can be traced back to the downlink network side. Therefore, there's an imperative need for proactive monitoring of changes in downlink-side indicators. Within the domain of the downlink side, a comprehensive set of fifteen indicators is available for evaluating network performance.

ZTE's [8] analysis of real-world user network data has identified eight of these indicators as pivotal for gaining insights into the quality of user experience. These eight indicators form the cornerstone of our dataset.

Our dataset comprises DPI device readings for various downlink-side indicators, collected from 500 anonymous individual users over a specific time period. Each user's experience is categorized as either a "good experience" (hereby referred to as "UGE") or a "bad experience" (hereby referred to as "UBE"). This data collection initiative was proposed as part of the 2022 ITU AI/ML in 5G Challenge, and it presents several unique challenges that our solution aims to overcome.

One prominent challenge relates to the subjective nature of determining whether a user's experience falls into the "good" or "bad" category. For instance, consistent packet loss [9] or high latency during video streaming or web browsing might be deemed as negative, while high throughput and low latency during online gaming [10] could be seen as positive. This inherent ambiguity in labeling UGE and UBE data points, lacking clear evaluation criteria, adds complexity to the problem. In this study, we will explore whether our machine learning model can outperform random sampling, even in the presence of this label ambiguity, while also overcoming the other challenges posed by the dataset.

In this paper, we conduct a comprehensive assessment of existing MTSC methods using the dataset introduced by ZTE, identifying their suboptimal performance due to the dataset's formidable challenges. Our study encompasses a comparison between conventional MTSC models and deep learning techniques, along with the introduction of a novel time series characteristic-based approach that involves extracting thousands of descriptive statistics from time series sequences to serve as input features for traditional classification models not explicitly designed for MTSC tasks.

The investigation highlights the remarkable performance of our proposed time series-characteristic models when applied to this unique dataset, achieving an impressive accuracy of 67%. This represents a notable 3% improvement over the closest competitor. Additionally, the study underscores the potential of deep learning models, notably Recurrent Neural Networks (RNNs), which exhibit commendable accuracy, achieving 58% on this relatively small dataset. Given a more extensive dataset, we can leverage the full capabilities of these deep learning models, and potentially surpass the performance of traditional statistical approaches.

The subsequent sections of this article delve deeper into the application of machine learning algorithms to classify home broadband user experiences. In Section 2, we conduct a comprehensive literature review to highlight previous research and findings concerning network performance and user experience. Section 3 provides an in-depth analysis of the dataset used in our study, along with the challenges that render it a unique and demanding problem in the networking domain. Additionally, Section 4 outlines our data preprocessing pipeline. In Section 5, we present and elucidate the various models employed in our study. Subsequently, Section 6 presents the results of our experiments, including accuracy and performance metrics, and Section 7 contextualizes them by providing a comparison to the existing state-of-the-art on this dataset. Lastly, in Section 8, we conclude the paper by summarizing our findings and suggesting potential future research directions.

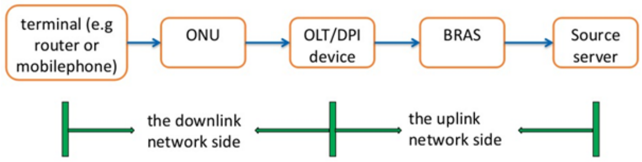

Fig. 1 – Downlink network side

2. LITERATURE REVIEW

While machine learning approaches for classifying network traffic and Quality of Experience (QoE) have been extensively studied, there is a notable research gap concerning the classification of network experience specifically for home broadband users. Additionally, the ZTE dataset, which we utilized for training our model, remains unexplored in the literature, and its challenges, particularly regarding ambiguous label definitions, have yet to be thoroughly investigated.

Network traffic classification generally involves identifying the applications/protocols/services used in a monitored network [11]. Although this is a well-researched topic with several new studies emerging yearly, it differs from our specific objective of classifying the subjective network experience for home users. For instance, Lim et al. proposed two deep learning methodologies for network traffic classification: a multi-layer Long Short-Term Memory (LSTM) model and a combination of Convolutional Neural Networks (CNNs) and single-layer LSTM models [12]. The multi-layer LSTM model achieved a high F1-score in classifying network packets into eight different application label names. Another comprehensive study by Azab et al. reviewed various network classification techniques, including port-based identification, deep packet inspection, statistical features combined with machine learning, and deep learning algorithms, along with their implementations, advantages, and limitations [11].

In recent years, there has also been a growing interest in the classification of end users' perceived QoE when using online media apps, such as video streaming services [13]. A survey by Huang et al. provided an overview of the state-of-the-art data-driven approaches for QoE evaluation [14].

MTSC has been a thriving research area in machine learning and data mining, with numerous applications in domains such as healthcare [15], manufacturing [16], and image recognition [17]. Notably, Ruiz-Ortiz et al. conducted a comprehensive comparative study evaluating different MTSC algorithms on 26 out of the 30 University of East Anglia (UEA) [18] archive problems characterized by equal-length data [19]. The findings demonstrated that four classifiers outperformed the benchmark dynamic time warping algorithm significantly. Among these, a recently proposed classifier known as ROCKET exhibited remarkable improvements on the archive datasets, accomplishing the task in significantly less time compared to the other three classifiers.

With the existing scarcity of research on home user network experience classification and the specific dataset in question, the importance of this problem becomes increasingly evident. The growing dependence on home networks for various applications, such as online education and remote work, emphasizes the necessity for reliable classification methods. To address this research gap, our paper aims to evaluate a variety of machine learning models utilizing real-world DPI probe readings. Table 1 compares the work mentioned in this section and highlights the unique contribution of our paper to the field.

3. DATASET

The dataset used in this study was provided by ZTE [8] as part of the 2022 ITU AI/ML In 5G Challenge [20]. The dataset includes indicator data recordings for 500 individual anonymous users, each labeled as either UGE or UBE. For model development, 80% of the dataset (400 users) is used for training, while the remaining 20% (100 users) is used for validation. Both sets maintain a balanced class distribution, with 50% UGE and 50% UBE records.

The specific indicators used in the dataset include the following:

- Indicator 1: Time interval between the syn ack packet and the ack packet in the first step of the three-way handshake.
- Indicator 2: Time interval between the syn ack packet and the ack packet in the second step of the three-way handshake.
- Indicator 3: Time interval between the ack packet and the first payload packet in the three-way handshake.
- Indicator 4: Response delay of the first packet with payload after the establishment of TCP for multiple flows in the session.
- Indicator 5: Actual delay of transmission from the DPI position to the user terminal in TCP transmission.
- Indicator 6: Transmission delay from the DPI position to the website in TCP transmission.
- Indicator 7: Percentage of downlink retransmitted packets in the current session in TCP transmission.
- Indicator 8: Percentage of upstream retransmission packets of the current session in TCP transmission.

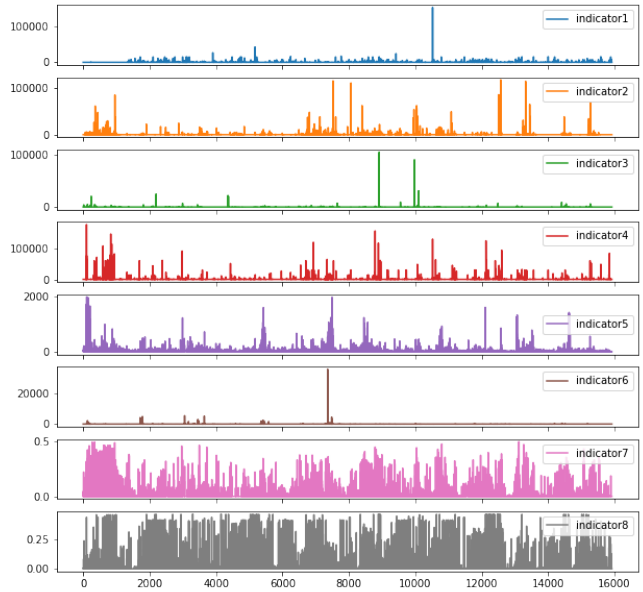

Fig. 2 – UBE indicators sample

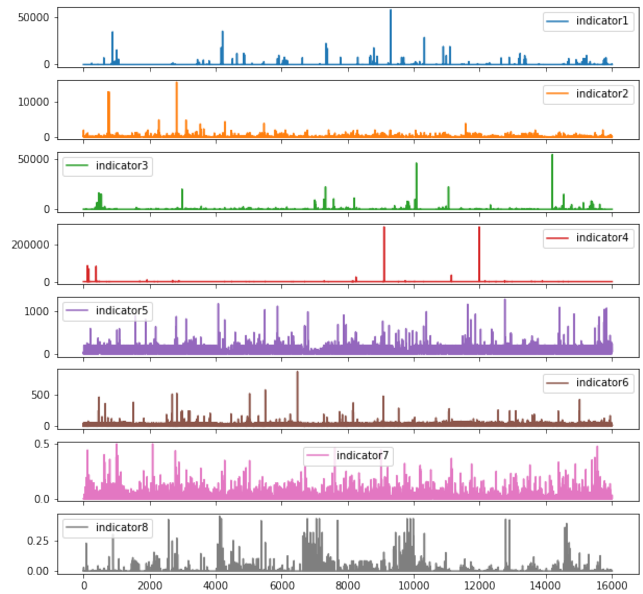

Fig. 3 – UGE indicators sample

Table 1 – Existing literature comparison

Study	Research Focus	Key Contribution
Lim et al. (2019)	Classifying network traffic from eight source applications using deep learning models with a custom preprocessing pipeline.	Highlights the effectiveness of deep learning models when applied to network traffic classification. The study showcases the benefits of employing a tailored preprocessing pipeline, leading to improved accuracy in identifying the source applications of network traffic.
Azab et al. (2022)	Comprehensive survey of network traffic classification techniques on eight unique datasets towards the identification of the used applications/protocols/services in a monitored network.	Surveys existing methods for network traffic classification, including port-based identification, deep packet inspection, statistical features with machine learning, and deep learning algorithms.
Ahmad et al. (2021)	QoE prediction for video streaming services using supervised learning ML models and their deployment in 5G/6G networks with SDN, NFV, and MEC.	Provides a tutorial on developing and deploying QoE prediction models for video streaming, introduces a reference architecture for ML model deployment in next-gen networks, and conducts a comparative study of supervised learning ML models for QoE prediction in video streaming applications.
Huang et al. (2018)	Survey of data-driven approaches for QoE evaluation, exploration of machine learning algorithms for QoE modeling and prediction, and research on QoE evaluation in imbalanced datasets.	Provides a comprehensive survey of data-driven methods for QoE evaluation, discusses the pros and cons of existing machine learning algorithms for QoE modeling, and addresses the challenges of QoE evaluation in imbalanced datasets.
Hammerla et al. (2016)	Human Activity Recognition using deep learning approaches on wearable sensor data, with focus on recurrent models and a novel regularization method.	Rigorously explores deep, convolutional, and recurrent approaches for HAR, introduces a new regularization technique, and demonstrates superior performance compared to the state-of-the-art on a benchmark dataset. Provides insights into model suitability for different HAR tasks and guidelines for applying deep learning in HAR settings.
Shojaee et al. (2021)	Modeling MTS data in smart manufacturing using deep neural networks and proposing the adaptive top-N linear generative-discriminative (AT-LinGD) method for efficient pipeline selection.	Investigates the use of DNN pipelines for MTS classification in smart manufacturing, introduces the AT-LinGD method for learning-to-rank top-N pipelines, and demonstrates its efficiency and accuracy in a real case study of aerosol jet printing process.
Zhao et al. (2021)	MTSC using image classification neural networks and transfer learning from ImageNet.	Proposes an MTSC method based on transforming time series into gray images and applying image classification neural networks. Explores two image classification networks on different imaging methods and assesses transfer learning performance. Shows competitive results in MTSC and demonstrates a 2.54% improvement in overall average accuracy with ResNet18 using fixed-sized images.
Ruiz et al. (2021)	Comparison of MTSC algorithms based on deep learning, shapelets, and bag of words approaches.	Reviews recently proposed MTSC algorithms, introduces ensembling for adapting univariate classifiers to MTSC, and demonstrates the superiority of certain classifiers, particularly ROCKET, on MTSC archive problems with significant improvements in computation time.
Bagnall et al. (2021)	Development of the first iteration of the MTSC archive for MTSC problems, containing 30 datasets with consistent length and train/test splits.	Addresses the lack of MTSC evaluation resources, collaboratively creates the MTSC archive, and facilitates standardized evaluations for MTSC algorithms with a diverse set of datasets.
This work	Classification of home user network experience using traditional MTSC, deep learning, and TS-Char models on the novel ZTE dataset.	Addresses the previously unexplored challenge of classifying home user network experience. Demonstrates the effectiveness of diverse machine learning models and introduces a novel TS-Char approach tailored to the ZTE dataset, contributing to the advancement of research in this domain.

The dataset comes with several issues that need to be addressed.

- Non-standard sampling rate and time range: The records in the dataset have a non-standard sampling rate and time range, making them challenging to analyze effectively.

- Uneven distribution of observations: The dataset exhibits an uneven distribution of observations across records, ranging from the largest containing 43,828 observations to the smallest with only 566 observations. This disparity makes the dataset incompatible with most state-of-the-art MTSC algorithms.

- Multiple observations for a single timestamp: Some records in the dataset contain multiple observations for a single timestamp. This situation could lead to inaccuracies in the analysis if not appropriately handled during data preprocessing.

- Limited sample size: The dataset has a relatively small sample size, consisting of only 500 unique users for training and validation. This limited size may impact the generalizability of results and model performance.

- Subjective definition of Internet experience: Defining a good or bad Internet experience is subjective, and user-reported labels may contain outliers. As a result, similar distributions of UGE and UBE indicators may arise, as depicted in 4. This subjectivity poses challenges for statistical models to accurately differentiate between UGE and UBE instances.

- Lack of data collection set-up information: The absence of information about the data collection setup/apparatus used prevents us from utilizing additional input/metadata to enhance the models.

In addition to the challenges posed by the dataset, it's important to note that this research was conducted as part of a competition framework. The rules of the competition explicitly prohibited any alterations to the provided dataset. This restriction was a critical factor in our research approach, influencing the methodologies employed and the nature of the challenges we faced. Underst anding these constraints is essential for a comprehensive evaluation of the study's methodology and outcomes. Despite these challenges, the dataset presents a valuable opportunity to develop models that can enhance our understanding of the factors that contribute to a good or bad Internet experience.

To preserve the integrity of the problem statement, outliers are not removed from the UGE data in all subsequent experiments. This rule is based on the fact that, in practical use, the indicators of users with good experiences are not completely free of outliers, and manual removal of outliers from UGE data would introduce bias.

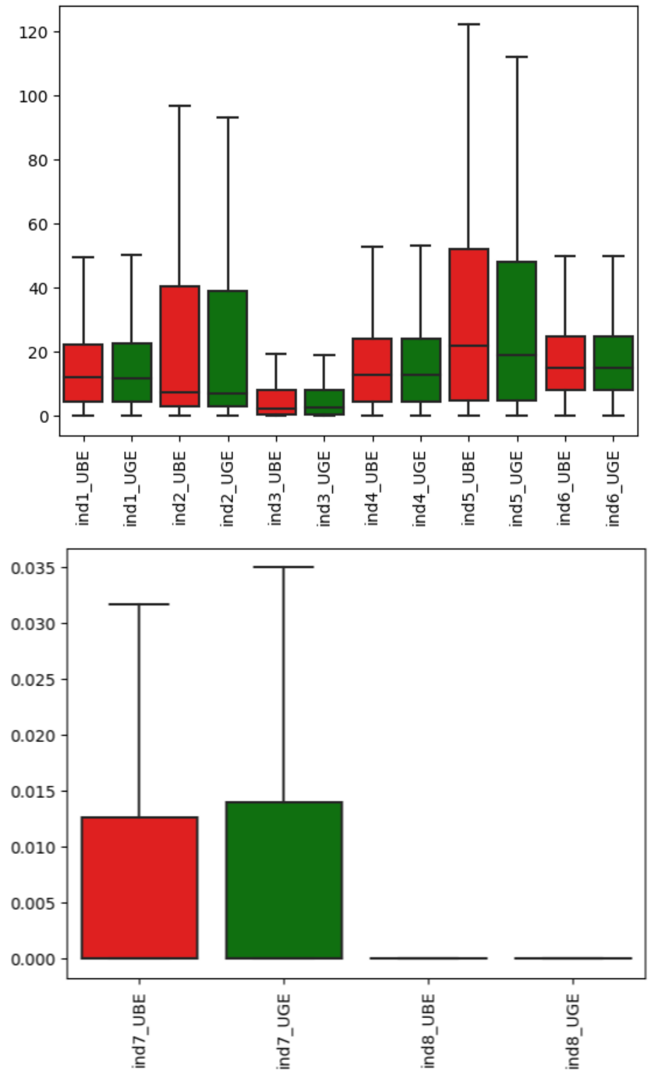

Fig. 4 – UGE/UBE indicator distributions

4. DATA PREPROCESSING PIPELINE

Prior to introducing the proposed methods, it is important to outline the standard data preprocessing pipeline that is employed. This pipeline includes several essential steps to address the issues with the data, such as linear interpolation, Z-normalization, and low-noise padding. These preprocessing steps enhance the effectiveness of the models by enabling them to better utilize the information present in the data, thereby improving classification accuracy.

4.1 Linear interpolation

Linear interpolation is a powerful regularization technique for time series data, as it resamples the data to a fixed interval [21]. This technique is particularly useful when dealing with dense time series or unevenly spaced sampling rates [22]. By resampling the data to standard intervals, such as 5-minute intervals, the resulting time series becomes more manageable and easier to work with. An example of linear interpolation is shown in Fig. 5, where a time series is resampled to 15-minute intervals. In the figure, the original data points are represented by gray marks, while the resampled points are depicted in orange.

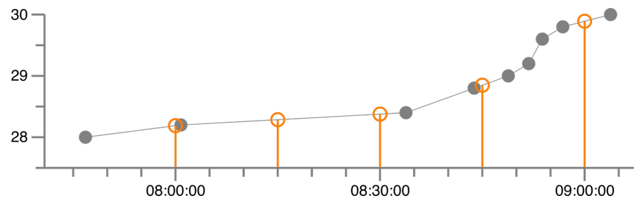

Fig. 5 – Illustration of linear interpolation process, source: [23]

4.2 Z-normalization

Z-normalization is a widely-used technique that transforms input vectors so that their mean is approximately zero and their standard deviation is close to one [24]. It is a powerful preprocessing step, particularly when analyzing the structural patterns of time series data. By normalizing the data in this way, a model can focus on similarities and dissimilarities in the shape and structure of the data, rather than being influenced by differences in amplitude [25]. In our specific problem, indicator columns have different units and scales, making z-normalization necessary to ensure accurate and meaningful comparisons between them. The formula for z-normalization is shown in Equation *(1)*, where x is the input vector, μ is its mean, and σ is its standard deviation.

$$Z = \frac{x - \mu}{\sigma} \tag{1}$$

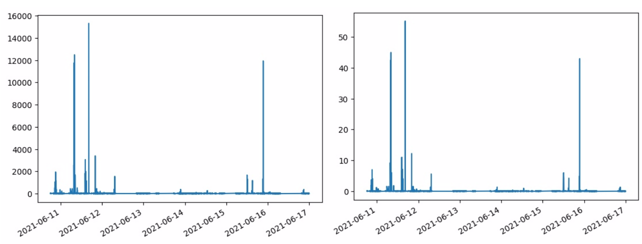

Fig. 6 – Time series before and after Z-normalization

4.3 Low-noise padding

Low-noise padding is a useful technique for handling unevenly-sized time series data. This method involves adding low-amplitude noise to the end of shorter time series, which results in sequences that are as long as the longest time series. Specifically, we added padding with an amplitude of 1e-6 to address the issue of uneven-length time series in the data.

Compared to other techniques such as uniform scaling, truncation, and ARIMA-based future value forecasting, low-noise padding is the easiest to implement and provides the best performance for our problem [26]. The main limitation of low-noise padding is that it may not be effective when the time series lengths differ significantly [27]. However, this issue has already been addressed by the use of linear interpolation, which makes this an ideal next step in the pipeline.

While some classifiers can directly process time series of varying lengths, they tend to underperform when compared to alternative approaches [19]. Research suggests that nearly any preprocessing strategy is better than none for time series with different fixed sampling frequencies [27], which is the case for our dataset. Therefore, we included low-noise padding in our preprocessing pipeline to ensure that all time series have uniform lengths, which can improve the performance of the classifiers.

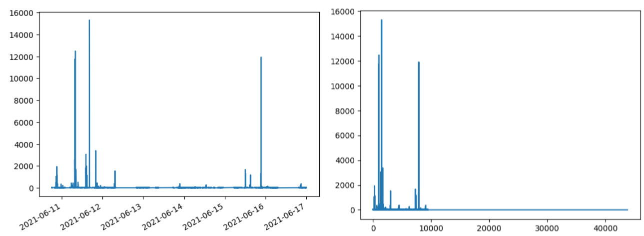

Fig. 7 – Time series before and after low-noise padding

5. TESTED METHODS

In this section, we test several models for this task, and categorize them into three groups: traditional MTSC models, deep learning-based MTSC models, and Time Series Characteristic (TS-Char) models. The sktime library [28] was used in our experimentation process, since the library implements most of the algorithms used for performance comparison in this work.

In this context, it is pertinent to note that extensive hyperparameter optimization attempts, including grid search techniques, did not yield significant enhancements over the default parameter configurations for the majority of the models implemented. This phenomenon appears to be attributable to the limited signal-to-noise ratio inherent in the dataset, a factor elaborately discussed in the dataset section of this paper.

5.1 Traditional MTSC models

Traditional MTSC models refer to machine learning models designed to classify multivariate time series signals directly. The methods tested in this category are ROCKET, DTW-KNN, and HIVE-COTE, which are all considered state-of-the-art in the field of MTSC.

5.1.1 ROCKET

The "Random Convolutional Kernel Transform" (ROCKET) model was introduced in [29] as a computationally efficient and highly accurate method for time series classification. The ROCKET model is based on the idea of generating random convolutional kernels and applying them to a time series to extract relevant features.

ROCKET has been shown to outperform many traditional models on a wide range of time series classification tasks [19]. Furthermore, the model is computationally efficient due to its use of random kernels and can handle variable length time series data without the need for padding or truncation. Thus, we chose to use a ROCKET model with 10 000 kernels as the baseline for our traditional MTSC models to establish a strong performance benchmark for other methods.

5.1.2 DTW-KNN

Another popular technique for time series classification is the "Dynamic Time Warping k-Nearest Neighbors" (DTW-KNN) algorithm, which has been widely used in the field of pattern recognition and machine learning. The DTW-KNN algorithm is based on the Dynamic Time Warping (DTW) distance measure, introduced in [30], which allows for the comparison of time series with varying lengths and time shifts. The DTW-KNN algorithm works by first computing the DTW distance between a query time series and all training time series, and then classifying the query time series based on the class labels of its k nearest neighbors in the training data.

DTW-KNN has shown promising results on a variety of time series classification benchmarks and is known for its robustness to noise and variability in the data [19]. Therefore, we have included a DTW-KNN with a brute force search method and n_neighbors set to 1 in our experiments to compare its performance with other proposed methods.

5.1.3 HIVE-COTE

HIVE-COTE (Hierarchical Vote Collective of Transformation-based Ensembles) is a state-of-the-art ensemble method for time series classification that has achieved outstanding results in a range of benchmark datasets [19]. HIVE-COTE leverages a hierarchical ensemble structure that is based on a range of transformation-based feature extraction methods and a collection of classifiers, including elastic ensemble, rotation forest, and random forest. This ensemble method (V2.0) was introduced in [31] as an extension of the Collective of Transformation-Based Ensembles (COTE) algorithm, which showed significant improvements over individual classifiers in terms of accuracy and speed. HIVE-COTE further enhances the effectiveness of COTE by introducing a hierarchical structure that involves multiple levels of feature extraction and classification, which improves the robustness of the method to noise and variations in the data.

The model has been shown to achieve remarkable performance on a variety of time series classification benchmarks, including those with multivariate data. Moreover, HIVE-COTE is designed to be robust to noise and variability in the data, making it suitable for real-world applications, and thus we chose to use the HIVE-COTE model as one of the traditional MTSC benchmarks in our experiments.

5.2 Deep learning models

Deep learning models are a promising approach for automatically learning features from time series data, which could be especially useful for our dataset. In addition to potential outliers and frequency components, there may be complex patterns and relationships that are not readily apparent to humans or traditional statistical methods. To explore this potential, we tested two popular types of deep learning models for MTSC: Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). CNNs are well-suited for capturing local patterns in time series data, while RNNs are particularly effective at modeling sequential data and capturing long-term dependencies. Both types of models have shown impressive results in previous studies, making them promising candidates for our task. Simpler models like Fully Connected Networks (FCNs) don't naturally process sequences effectively, which limits their suitability for our time series data. Therefore, we did not include them in our study.

5.2.1 Convolutional neural networks

CNNs, originally developed for image recognition, have gained popularity in time series classification due to their ability to extract and learn hierarchical features from sequential data [32]. Unlike traditional feedforward neural networks that treat input data as a fixed-length vector, CNNs can handle variable-length sequences by sliding a filter over the input sequence and performing convolution operations to produce a feature map. The filters can capture local patterns in the input sequence, such as spikes, trends, and seasonality, and the pooling layers can aggregate and downsample the features to capture the most important information [33].

CNNs have shown promising results in various applications, and have even outperformed other traditional time series classification algorithms in some cases. For example, [34] found that CNNs achieve state-of-the-art performance in time series classification on a real-world dataset for human activity recognition. In this study, we used a CNN to classify the multivariate time series in our dataset. The hyperparameters used for training the CNN are outlined in Table 2.

Table 2 – CNN hyperparameters

Hyperparameter	Value
Convolutional Layers	2
Filter Size	6, 12
Kernel Size	7
Stride	1
Activation Function	Sigmoid
Pooling Layers	2
Pooling Type	Average Pooling
Pooling Size	3
Learning Rate	0.01
Batch Size	16
Loss	Mean Squared Error
Optimizer	Adam
Epochs	2000

5.2.2 Recurrent neural networks

RNNs have become a popular approach for time series classification due to their ability to model sequential data and capture temporal dependencies over extended periods [35]. This enables RNNs to learn intricate patterns in time series data and make accurate predictions on variable-length sequences.

In this study, we used a specific type of RNN, the Long Short-Term Memory (LSTM) network [36]. LSTMs have proven effective in addressing the vanishing gradient problem that plagues standard RNNs, by utilizing a memory cell and three gates: forget, input, and output. These components selectively update and store information in the memory cell, allowing LSTMs to retain information about past inputs and make more accurate predictions about future inputs [36]. The hyperparameters used for training the LSTM model are outlined in Table 3.

Table 3 – LTSM hyperparameters

Hyperparameter	Value
LSTM Layers	3
Dropout Layers	2
Layer Size	256
Dropout Rate	0.75
Learning Rate	0.001
Batch Size	10
Loss	Cross Entropy Loss
Optimizer	Adam
Epochs	136

Similar to our experience with XGBoost models, we found that hyperparameter optimization, learning rate adjustments, and changes in architectures had minimal impact on accuracy. Consequently, we chose to adopt standard values prevalent in widely-used implementations.
The training of the model was completed in approximately one hour utilizing a single RTX 3080 GPU to achieve the best checkpoint. Beyond this point, the model exhibited tendencies of overfitting, a phenomenon that we have elaborated upon in the results section of the paper.

5.3 Time series characteristic models

Time Series Characteristic (TS-Char) models rely on extracting descriptive statistics from time series data and using them as input features for traditional machine learning models not specifically designed for MTSC. The advantages to this approach lie in its robustness, since it is applicable to time series of any length or sampling frequency, with minimal preprocessing and low computational overhead. The TS-Char approach does not require the use of the data preprocessing pipeline, and thus it was not used when testing these models. However, the TS-Char approach may not capture the full complexity and dynamics of the underlying time series data, and may not perform as well as models designed specifically for MTSC tasks in certain contexts.

To evaluate the performance of the TS-Char approach, two methods were tested. The first method involved manual feature extraction, where a manually defined set of features was derived for each sequence. The second method used the library TSFresh [37] to automatically extract thousands of features from the data, providing new insights into the time series and their dynamics.

We used XGBoost with default hyperparameters [38] as the classifier for both manual feature extraction and TS-Fresh approaches. XGBoost is known for its robustness and ability to achieve good accuracy on most classification problems [39]. However, since we are extracting a large number of features for each sequence, we run into the Big-p, Little-n (p»n) problem, where the number of features is much greater than the number of observations. This violates the default assumptions of most machine learning algorithms and can lead to overfitting and poor generalization performance [40]. To address this issue, we applied a min-max normalization to scale the features to a common range and reduce the effect of outliers. This technique is based on the formula shown in Equation *(2)*, where X is the original data point, X_{min} is the minimum value in the dataset, X_{max} is the maximum value in the dataset, and X_{norm} is the normalized value between 0 and 1.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2}$$

We then applied Principal Component Analysis (PCA) for dimensionality reduction. PCA transforms the original features into a smaller set of uncorrelated variables, called principal components, that capture the most important information in the data [41]. The number of principal components retained was determined by setting the n_components parameter of PCA to 0.95, which keeps enough components to explain 95% of the variance in the data.

5.3.1 Manual feature extraction

For the manual feature extraction approach, the following statistics were extracted for each sequence:

- Mean: the arithmetic mean of the sequence
- Median: the median value of the sequence
- Standard deviation: a measure of the dispersion of the sequence values around the mean
- Quartiles: the values that divide the sequence into four equal parts (25th, 50th, and 75th quartiles)
- Maximum: the highest value in the sequence
- Skew: a measure of the symmetry of the distribution of values in the sequence
- Kurtosis: a measure of the "peakedness" of the distribution of values in the sequence
- Fourier coefficients: the top 10 Fourier coefficients of the sequence, which provide information about its frequency content
- Outlier percentage: the percentage of values in the sequence that are identified as outliers using the local outlier factor algorithm with a number of neighbors set to 5, and the Median Absolute Deviation (MAD) method

These statistics provide a summary of the time series sequence and can be used to represent its salient features for classification purposes.

5.3.2 TSFresh

For the TSFresh approach, we used an automatic feature extraction method provided by the TSFresh library. The specific set of features used in this study was "EfficientFCParameters," which includes all features except for those marked with the "high_comp_cost" attribute. This attribute indicates features that are computationally expensive to calculate and may not be suitable for real-time applications. By using the EfficientFCParameters set, we were able to extract 6246 features without incurring a high computational cost. The full list of features included in this set can be found in [42].

6. RESULTS

This section presents the results of our experiments with the proposed methods on the provided dataset. We evaluate the performance of each algorithm using accuracy, precision, and recall metrics, which provide insights into their strengths and weaknesses. Accuracy measures the overall proportion of correct predictions, precision measures the proportion of true positives among all positive

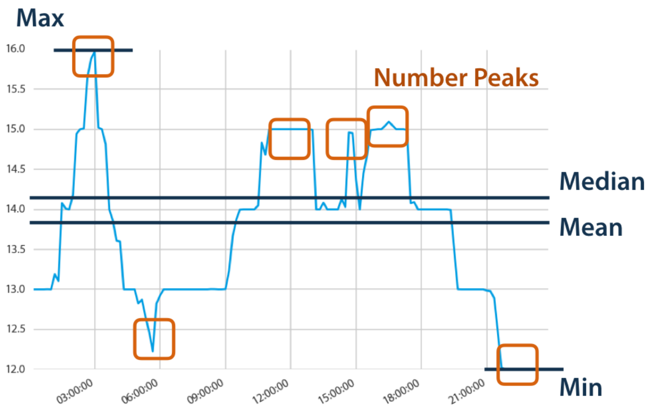

Fig. 8 – TSFresh feature extraction from signals, source: [42]

predictions, and recall measures the proportion of true positives among all actual positive cases. The analysis focuses on the validation set, which provides an unbiased assessment of the models' generalization performance.

Table 4 summarizes the validation set performance of all the models that were evaluated.

Table 4 – Model metrics (validation set)

Model	Accuracy	Precision	Recall
ROCKET	0.40	0.43	0.30
DTW-KNN	0.37	0.29	0.20
HIVE-COTE	0.44	0.38	0.28
CNN	0.50	0.25	0.50
LSTM	0.58	0.73	0.58
Manual + XGBoost	0.46	0.46	0.46
TSFresh + XGBoost	0.67	0.67	0.68

The results show that the TSFresh + XGBoost model achieved the highest performance across all three metrics, with an accuracy of 0.67, precision of 0.67, and recall of 0.68. The LSTM model also achieved good performance, with an accuracy of 0.58, precision of 0.73, and recall of 0.58. These results suggest that these models are promising for accurately classifying the user type. The ROCKET, HIVE-COTE, and Manual + XGBoost models achieved moderate performance, with accuracies ranging from 0.40 to 0.46. The precision and recall scores were also moderate, with values ranging from 0.29 to 0.46. The DTW-KNN and CNN models achieved the lowest performance, with accuracies of 0.37 and 0.50, respectively, and low precision and recall scores. These results suggest that these models cannot deal with the challenges presented by this dataset.

Overall, our results provide practical guidance for selecting appropriate models for time series data, and demonstrate the potential of advanced feature extraction techniques for improving performance in this context. The TSFresh + XGBoost and LSTM models are promising options for accurately predicting the target variable, but further research is needed to explore their limitations and further applications.

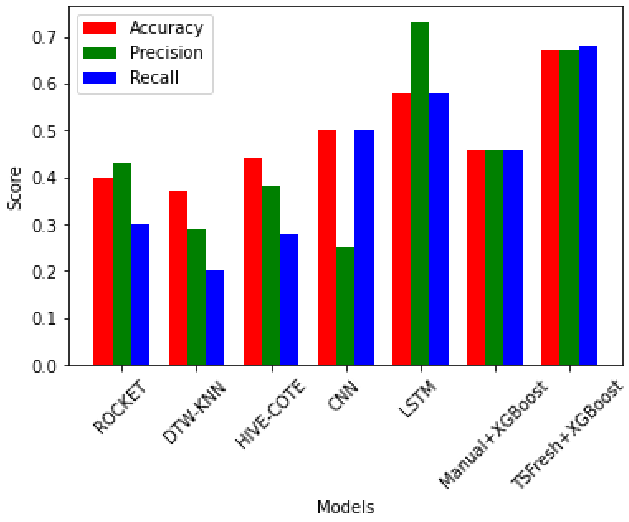

Fig. 9 – Model metrics (validation set)

Below we present a more in-depth analysis of the models we evaluated.

6.1 MTSC result analysis

ROCKET stands out as the top-rated and swiftest classifier, earning its place as the go-to choice for MTSC problems, as advocated by [19]. Consequently, we employed the sktime implementation [28] as a benchmark for MTSC models. Regrettably, this model exhibited a rather meager accuracy of 40% on the validation set.

One plausible explanation for this subpar performance lies in the presence of UGE outliers, which tend to blur the statistical distributions of UGE and UBE, making them notably similar. This hypothesis gains further credibility as similar lackluster outcomes were observed when experimenting with other MTSC models like DTW-KNN and HIVE-COTE.

Collectively, these findings underscore the unsuitability of traditional MTSC approaches for modeling this dataset. As such, we do not recommend relying on these methods to tackle the challenge at hand.

6.2 LSTM result analysis

The best LSTM checkpoint attains an impressive validation accuracy of 58%, marking a significant leap forward compared to traditional MTSC models. This substantial improvement can be attributed to the capacity of deep neural networks to capture a broader spectrum of underlying features compared to conventional statistical approaches.

Interestingly, a discernible pattern is observed in the model performance in which the training accuracy and loss stabilize at 4000 epochs whilst the validation loss starts to climb rapidly. This behavior suggests that the model has reached a saturation point, where it struggles to identify additional predictive features and consequently succumbs to overfitting.

This observation serves as a reminder of the limitations posed by the relatively small number of unique user samples in this dataset. Given access to a more extensive dataset, there's good reason to believe that this model could achieve substantially higher levels of accuracy.

6.3 TS-Char result analysis

The Manual feature extraction + XGBoost model demonstrates a 10-fold cross-validation accuracy of 0.51, accompanied by a test accuracy (on the manually selected validation set) of 0.46. This performance is likely impacted by the presence of UGE outliers, a challenge discussed in detail in the MTSC result analysis section. The manually extracted features struggle to provide distinct characteristics for the model to differentiate between these outliers and other data points.

In contrast, the TSFresh + XGBoost model seems to overcome these hurdles adeptly. It accomplishes this by extracting thousands of features, resulting in a validation accuracy of 67%, firmly establishing itself as the leading model in our experiments.

For a more nuanced understanding, we have provided class-wise metrics in Table 5 and a confusion matrix in Fig. 10, highlighting the model's consistent and reliable performance across both labels. This success is attributed to the extraction of descriptive features, offering deeper insights into the underlying data patterns and the variables that influence positive or negative experiences.

Unfortunately, the use of PCA has hindered our ability to pinpoint specific features as the most influential predictors. In forthcoming research, it may be worthwhile to explore alternative dimensionality reduction techniques that enable the extraction of feature importance, granting us a more detailed comprehension of the critical predictive elements at play.

Table 5 – TSFresh + XGBoost model detailed metrics

Label	Precision	Recall	F1 Score	Samples
UGE	0.67	0.68	0.67	50
UBE	0.67	0.66	0.67	50

7. BENCHMARKS

To facilitate a comprehensive comparison, we have integrated the findings of the leading teams in the 2022 ITU AI/ML 5G Challenge, and their respective results are presented in Table 5. This comparative analysis serves to

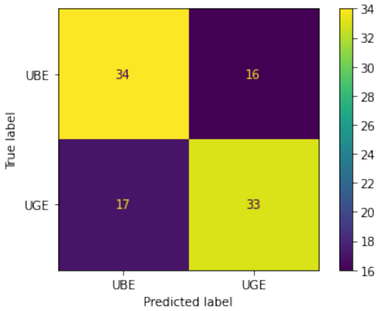

Fig. 10 – TSFresh + XGBoost model confusion matrix

contextualize our own findings and provides valuable insights into the current state-of-the-art in this domain.

Fig 11 presents the best models from the challenge, including our team's solution. "RF" denotes random forest and "LR" denotes logistic regression.

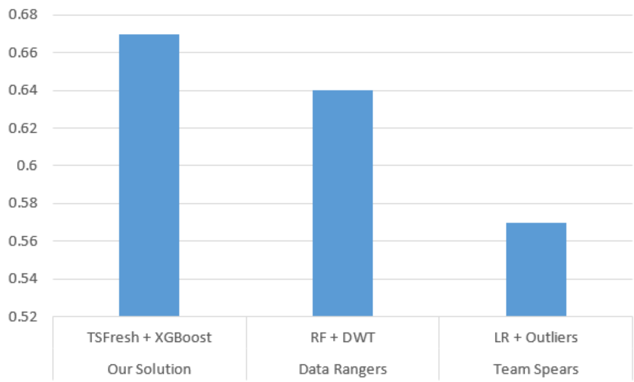

Fig. 11 – 2022 ITU AI/ML 5G Challenge best models

Our approach utilized the TSFresh feature extraction in combination with the XGBoost classifier, achieving a commendable accuracy of 67%. Upon examining the results, it becomes evident that our team's approach has outperformed the solutions proposed by other participating teams. This performance underscores the effectiveness of our chosen methodology in tackling the challenges posed by the ZTE dataset.

8. CONCLUSION

Our study sheds light on the effectiveness of various machine learning models for the task of classifying home network user data. The experiments demonstrate that the proposed TSFresh feature extraction method, coupled with an XGBoost classifier, achieves the highest accuracy on the dataset. While this model demonstrated the best performance, it is pertinent to note that the use of PCA in this context imposes limitations on the interpretability of the model, particularly with respect to conventional techniques like SHAP. This aspect highlights a valuable avenue for future research: developing methodologies that retain the predictive accuracy achieved by our approach while enhancing model transparency and interpretability, possibly through the avoidance of PCA or the integration of more explainable machine learning techniques. Additionally, the combination of a data preprocessing pipeline and an LSTM model shows promising results, with potential for improved generalization on larger datasets. However, it is worth noting that no single "best" model exists, as the choice of algorithm depends on the specific characteristics of the dataset. Our findings underscore the potential of machine learning in solving real-world problems and provide a solid foundation for future research in this area.

ACKNOWLEDGEMENT

This work was done as part of the 2022 ITU AI/ML in 5G Challenge. We gratefully acknowledge ITU for hosting the challenge and providing the computing resources that enabled us to conduct this research. Special thanks to ZTE for sponsoring the competition and providing the essential dataset and materials that facilitated our work.

REFERENCES

[1] Babette Boliek, Kim Makuch, Catherine Matraves, and Aleks Yankelevich. "Economics at the FCC 2018–2019: Competition, Broadband Deployment, and Transaction Review". In: *Review of Industrial Organization* 55.4 (2019), pp. 625–646. ISSN: 0889938X, 15737160. URL: https://www.jstor.org/stable/48703034.

[2] OECD. "OECD Digital Economy Outlook 2020". In: (2020). DOI: https://doi.org/10.1787/bb167041-en. URL: https://www.oecd-ilibrary.org/content/data/data-00170-en.

[3] J.D. Power. "U.S. Residential Internet Service Provider Satisfaction Study". In: (2020). URL: https://www.jdpower.com/sites/default/files/file/2020-09/US_ResISP20_MS_revFINAL_090720.pdf.

[4] Guang-Qian Peng, Guangtao Xue, and Yi-Chao Chen. "Network Measurement and Performance Analysis at Server Side". In: *Future Internet* 10.7 (2018), p. 67. DOI: 10.3390/fi10070067. URL: https://doi.org/10.3390/fi10070067.

[5] Hira Zahid, Tariq Mahmood, Ahsan Morshed, and Timos Sellis. "Big data analytics in telecommunications: literature review and architecture recommendations". In: *IEEE/CAA Journal of Automatica Sinica* PP (Dec. 2019), pp. 1–22. DOI: 10.1109/JAS.2019.1911795.

[6] Arthur L Samuel. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of Research and Development* 44 (1967), pp. 206–227. URL: https://api.semanticscholar.org/CorpusID:2126705.

[7] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and S. S. Iyengar. "A Survey on Deep Learning". In: *ACM Computing Surveys* 51.5 (Sept. 2018), pp. 1–36. DOI: 10.1145/3234150. URL: https://doi.org/10.1145/3234150.

[8] ZTE. *ML5G-PS-012: Classification of Home Network Users to Improve User Experience*. https://challenge.aiforgood.itu.int/match/matchitem/73. 2022.

[9] Juraj Bienik, Miroslav Uhrina, Lukas Sevcik, and Anna Holesova. "Impact of Packet Loss Rate on Quality of Compressed High Resolution Videos". In: *Sensors* 23.5 (Mar. 2023), p. 2744. DOI: 10.3390/s23052744. URL: https://doi.org/10.3390/s23052744.

[10] Mei Xintong Zha and Ying Zhang. *The Effects of Network Latency on Player Gaming Experience*. Tech. rep. Worcester Polytechnic Institute, 2019.

[11] Ahmad Azab, Mahmoud Khasawneh, Saed Alrabaee, Kim-Kwang Raymond Choo, and Maysa Sarsour. "Network traffic classification: Techniques, datasets, and challenges". In: *Digital Communications and Networks* (2022). ISSN: 2352-8648. DOI: https://doi.org/10.1016/j.dcan.2022.09.009. URL: https://www.sciencedirect.com/science/article/pii/S2352864822001845.

[12] Hyun-Kyo Lim, Ju-Bong Kim, Joo-Seong Heo, Kwihoon Kim, Yong-Geun Hong, and Youn-Hee Han. "Packet-based Network Traffic Classification Using Deep Learning". In: *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. 2019, pp. 046–051. DOI: 10.1109/ICAIIC.2019.8669045.

[13] Arslan Ahmad, Atif Bin Mansoor, Alcardo Alex Barakabitze, Andrew Hines, Luigi Atzori, and Ray Walshe. "Supervised-learning-Based QoE Prediction of Video Streaming in Future Networks: A Tutorial with Comparative Study". In: *IEEE Communications Magazine* 59.11 (2021), pp. 88–94. DOI: 10.1109/MCOM.001.2100109.

[14] R. Huang et al. "A survey of data-driven approach on multimedia QoE evaluation". In: *Frontiers of Computer Science* 12.6 (2018), pp. 1060–1075.

[15] Nils Y Hammerla, Shane Halloran, and Thomas Ploetz. "Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables". In: *arXiv preprint arXiv:1604.08880* (2016). URL: http://arxiv.org/abs/1604.08880.

[16] Parshin Shojaee, Yingyan Zeng, Xiaoyu Chen, Ran Jin, Xinwei Deng, and Chuck Zhang. "Deep Neural Network Pipelines for Multivariate Time Series Classification in Smart Manufacturing". In: *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*. 2021, pp. 98–103. DOI: 10.1109/ICPS49255.2021.9468245.

[17] Yuxuan Zhao and Zhongmin Cai. "Classify Multivariate Time Series by Deep Neural Network Image Classification". In: *2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)*. 2019, pp. 93–98. DOI: 10.1109/CCHI.2019.8901913.

[18] Anthony J. Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn J. Keogh. "The UEA multivariate time series classification archive, 2018". In: *CoRR* abs/1811.00075 (2018). eprint: 1811.00075. URL: http://arxiv.org/abs/1811.00075.

[19] Alejandro Pasos Ruiz, Michael Flynn, James Large, Matthew Middlehurst, and Anthony Bagnall. "The great multivariate time series Classification bake off: A review and experimental evaluation of recent algorithmic advances". In: *Data Mining and Knowledge Discovery* 35.2 (2020), pp. 401–449. DOI: 10.1007/s10618-020-00727-3.

[20] ITU-T. *ITU AI/ML in 5G Challenge*. https://aiforgood.itu.int/about-ai-for-good/aiml-in-5g-challenge/. 2022.

[21] Venkat N Gudivada, Dhana Rao, and Vijay V Raghavan. "Chapter 9 - Big Data Driven Natural Language Processing Research and Applications". In: *Big Data Analytics*. Ed. by Venu Govindaraju, Vijay V Raghavan, and C R Rao. Vol. 33. Handbook of Statistics. Elsevier, 2015, pp. 203–238. DOI: https://doi.org/10.1016/B978-0-444-63492-4.00009-5. URL: https://www.sciencedirect.com/science/article/pii/B9780444634924000095.

[22] Mohamed Noor, Mohd Mustafa Al Bakri Abdullah, Ahmad Shukri Yahaya, and Nor Ramli. "Comparison of Linear Interpolation Method and Mean Method to Replace the Missing Values in Environmental Data Set". In: *Materials Science Forum* 803 (2014), pp. 278–281. DOI: 10.4028/www.scientific.net/MSF.803.278.

[23] Mstringer. *Stack Overflow*. https://i.stack.imgur.com/Wxkcp.png. Accessed on October 14, 2022. 2016.

[24] Gökhan Aksu, Cem Oktay Güzeller, and Mehmet Taha Eser. "The Effect of the Normalization Method Used in Different Sample Sizes on the Success of Artificial Neural Network Model". In: *International Journal of Assessment Tools in Education* 6.2 (2019), pp. 170–192. DOI: 10.21449/ijate.479404.

[25] P. Sinin. *SAX-VSM*. https://jmotif.github.io/sax-vsm_site/morea/algorithm/znorm.html. Accessed on October 17, 2022. 2016.

[26] Agnieszka Bier, Agnieszka Jastrzębska, and Paweł Olszewski. "Variable-Length Multivariate Time Series Classification Using ROCKET: A Case Study of Incident Detection". In: *IEEE Access* 10 (2022), pp. 95701–95715. DOI: 10.1109/ACCESS.2022.3203523.

[27] Chang Wei Tan, François Petitjean, Eamonn J. Keogh, and Geoffrey I. Webb. "Time series classification for varying length series". In: *CoRR* abs/1910.04341 (2019). eprint: 1910.04341. URL: http://arxiv.org/abs/1910.04341.

[28] Markus Löning, Anthony J. Bagnall, Sajaysurya Ganesh, Viktor Kazakov, Jason Lines, and Franz J. Király. "sktime: A Unified Interface for Machine Learning with Time Series". In: *CoRR* abs/1909.07872 (2019). eprint: 1909.07872. URL: http://arxiv.org/abs/1909.07872.

[29] Angus Dempster, François Petitjean, and Geoffrey I. Webb. "ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels". In: *CoRR* abs/1910.13051 (2019). eprint: 1910.13051. URL: http://arxiv.org/abs/1910.13051.

[30] Eamonn Keogh and Chotirat Ratanamahatana. "Exact indexing of dynamic time warping". In: *Knowledge and Information Systems* 7 (2005), pp. 358–386. DOI: 10.1007/s10115-004-0154-9.

[31] Matthew Middlehurst, James Large, Michael Flynn, Jason Lines, Aaron Bostrom, and Anthony J. Bagnall. "HIVE-COTE 2.0: a new meta ensemble for time series classification". In: *CoRR* abs/2104.07551 (2021). eprint: 2104.07551. URL: https://arxiv.org/abs/2104.07551.

[32] Aji Prasetya Wibawa, Agung Bella Putra Utama, Hakkun Elmunsyah, Utomo Pujianto, Felix Andika Dwiyanto, and Leonel Hernandez. "Time-series analysis with smoothed Convolutional Neural Network". In: *Journal of Big Data* 9.1 (2022). DOI: 10.1186/s40537-022-00599-y.

[33] Kunihiko Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological Cybernetics* 36.4 (1980), pp. 193–202. DOI: 10.1007/bf00344251.

[34] Jian Bo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiao Li Li, and Shonali Krishnaswamy. "Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition". In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI'15. Buenos Aires, Argentina: AAAI Press, 2015, pp. 3995–4001. ISBN: 9781577357384.

[35] Benjamin Lindemann, Timo Müller, Hannes Vietz, Nasser Jazdi, and Michael Weyrich. "A survey on long short-term memory networks for time series prediction". In: *Procedia CIRP* 99 (2021). 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020, pp. 650–655. DOI: https://doi.org/10.1016/j.procir.2021.03.088. URL: https://www.sciencedirect.com/science/article/pii/S2212827121003796.

[36] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[37] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W. Kempa-Liehr. "Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)". In: *Neurocomputing* 307 (2018), pp. 72–77. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2018.03.067. URL: https://www.sciencedirect.com/science/article/pii/S0925231218304843.

[38] *XGBoost Parameters*. Accessed on September 29, 2023. URL: https://xgboost.readthedocs.io/en/stable/parameter.html#parameters-for-tree-booster.

[39] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *arXiv preprint arXiv:1603.02754* (2016). URL: http://arxiv.org/abs/1603.02754.

[40] Gareth James. *An introduction to statistical learning: With applications in R*. 1st ed. Springer Texts in Statistics. New York: Springer, 2013.

[41] Ian T Jolliffe and Jorge Cadima. "Principal component analysis: a review and recent developments". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150202. DOI: 10.1098/rsta.2015.0202.

[42] Maximilian Christ. *Overview on extracted features*. https://tsfresh.readthedocs.io/en/latest/text/list_of_features.html. Accessed on September 20, 2022.

AUTHORS

Rushat Rai is currently pursuing a bachelor of engineering degree in information engineering at The Chinese University of Hong Kong (Hong Kong, China). His research pursuits revolve around the realms of machine learning, deep learning and signal processing, with a keen focus on their versatile applications across various domains. In 2022, he received the Best Student Solution Award at the ITU AI/ML in 5G Challenge for his work on classifying home user network experiences.

Thomas Basikolo is a programme officer with the ITU Telecommunication Standardization Bureau (TSB). He coordinates and manages the ITU AI/ML in 5G Challenge and is an advisor to the ITU-T Focus Group on Autonomous Networks. Prior to joining ITU, he worked as a research engineer in the Engineering Department of Microwave Factory Co., Ltd, Tokyo, Japan. He received a PhD in electrical and computer engineering from Yokohama National University, Japan. He is a recipient of multiple best paper awards, the IEEE AP–S Japan Student Award and the Young Engineer of the year award by IEEE AP–S Japan in 2018. He has co-authored peer-reviewed journal and conference papers, predominantly in the areas of wireless communications and antenna engineering. He serves as a reviewer of IEEE and IEICE journals. His interests include machine learning, deep learning and network science, and their applications in wireless networks.

