**Team Name: Data Rangers**

**Members: Bharath Raj Muppalla**

**Cade Phelan**

**Srinivas Pramodh Kotipalli**

**Tharun Reddy**

**arath Raj Muppalla**

**Cade Phelan**

**Srinivas Pramodh Kotipalli**

**Tharun Reddy**

**Classification of Home Network Users to Improve User Experience**

**Table of Contents**

Contents

[Problem Understanding 3](#_Toc117817436)

[Higher outliers in UBE assumption: 3](#_Toc117817437)

[Data Exploration 3](#_Toc117817438)

[Plotting all the indicators and comparing UGE and UBE users: 5](#_Toc117817439)

[Tentative Outlier definition and further exploration: 7](#_Toc117817440)

[Aggregated Data Exploration 8](#_Toc117817441)

[Grouping of data based on timestamp: 9](#_Toc117817442)

[Our Solution: Direct Classification Using Machine Learning methods 9](#_Toc117817443)

[MODEL DESIGN AND INTERPRETATION 9](#_Toc117817444)

[A fresh perspective on information from indicators: 9](#_Toc117817445)

[Discrete Wavelet Transformation(DWT) 10](#_Toc117817446)

[Preprocessing, Feature Extraction, and Experimentation: 11](#_Toc117817447)

[ML Classifiers 12](#_Toc117817448)

[SCHEMATIC DIAGRAM SHOWING THE ARCHITECTURE OF MODEL 11](#_Toc117817449)

[Machine learning algorithm parameters. 13](#_Toc117817450)

[Experimentation Phase: 13](#_Toc117817451)

[Hyperparameter tuning : 13](#_Toc117817452)

[Final Selected model parameters: 14](#_Toc117817453)

[Conclusion: 14](#_Toc117817454)

**TECHNICAL SUMMARY**

# Problem Understanding

The problem as presented can be broken down into two parts

1. Establish a threshold for indicators above which we consider an indicator value as an outlier and make sure that UGE users have a low proportion of outliers and UBE users have a higher proportion of outliers
2. Define a threshold for outlier proportion above which and user would be classified as UBE

## Higher outliers in UBE assumption:

The above breakdown was based on the explanation in the problem statement that UBE users would have high indicator values and a higher proportion of outliers. But the first step of the modeling is to verify how strongly the assumption hold.

# Data Exploration

It was essential to verify the above assumption visually and statistically if we were to make a single threshold that clearly separates UBE and UGE users. So, the first step we took was to load the multiple CSV files into a list of pandas dataframes so that we can loop through them to make plots and visually understand the data. Then, we plotted the indicators on the same scale for pairs of users side by side where one user is from the UBE group and the other is from the UGE group. This was needed to visually inspect the validity of the assumption that is seeing a clear pattern of UGE users having indicator values compared to UBE. It seemed to hold true for some indicators and users as in the following images

**UBE has more outliers 1st example**

A picture containing graphical user interface

Description automatically generated

**UBE has more outliers 2nd example**

A picture containing text, measuring stick, colorful, different

Description automatically generated

But it did not appear true for some as in the below image where UBE user seems to have lower values and fewer outliers

Timeline, histogram

Description automatically generated

**UGE has more outliers in this example**

In some cases, both groups seem to have similar patterns as in the below image

A picture containing text, colorful, measuring stick, colors

Description automatically generated

**UBE and UGE have similar outlier proportion**

## Plotting all the indicators and comparing UGE and UBE users:

To get a better perspective we plotted all eight indicators for each pair side by side for several users. The red is for a UBE user and the green is for a UGE user. We can again clearly see the variation. For the first pair in the next page, the UBE user has higher values and outlier proportions and for a different pair in the image on page 7, UGE user has higher. Upon inspection of several other users, it was clear that there might not be a very clear difference as assumed.

A picture containing text, building, screenshot

Description automatically generated

**UBE and UGE users plotted on same scale 1st example UBE has more outliers**

A picture containing text, building, screenshot

Description automatically generated

**UBE and UGE users plotted on same scale 2nd example, UGE has more outliers**

## Tentative Outlier definition and further exploration:

To see if it was possible to define a single threshold despite this violation of the assumption, we started with defining a tentative threshold based on 95% percentile values of the indicators for each of the users. Now if we choose one of these values as a threshold for that indicator then, users with a lesser 95% value will have less than 5% outliers, and users with a higher 95% percentile values, will have more outliers.

Again, we calculated these values for each of the users and plotted them for each group separately to get a perspective of the distribution. From the plots, it can see that.

Indicator 1(2000 vs 6000 with very few peaks), indicator 3 ( 10000 vs 25000), and indicator 4 ( 400 vs 4000) seem to have lesser values in the UGE group. This means theoretically if we choose the threshold carefully for these indicators say 500, most of the UGE users would have their 95% percentile value below this and thus get classified as UGE due. At the same time, most of the UBE users would have higher 95% percentile values and get classified as UBE

Rest of the indicators 2,3,5,6,7 and 8 have a very similar distribution of 95% values in both groups. This again means that it is very difficult to set a threshold for these indicators which clearly separates out UGE and UBE indicators, no matter what we choose there would be many misclassifications and overall accuracy would be less.

**95% percentile values of each indicator with UGE users on x - axis**

Graphical user interface

Description automatically generated

**95% percentile values of each indicator with UBE users on x - axis**

Graphical user interface

Description automatically generated

## Aggregated Data Exploration

The above observations leave us with a choice to ignore the high variability indicators and proceed with classification using only Indicators 1, 3, and 4. But doing so might solve the problem just for this dataset but would leave out a lot of valuable information from the other indicators and not a robust enough solution to handle any other set of data. This understanding has led us to the next step of combining the indicators based on functional understanding.

## Grouping of data based on timestamp:

We have observed that several of the rows for each user have the same timestamp and we understood this as data related to parallel connections being established for the same user activity as clicking or opening a new website

Summing the first 6 indicators and averaging indicators 7 and 8: This was done based on the understanding that all these events occur in a sequence and together determine the total time taken. But this was done just to use all the indicators together

But even this yielded similar results where there is a lot of misclassification resulting from UBE and UGE having a similar pattern of outlier

## Our Solution: Direct Classification Using Machine Learning methods

To make use of all the information we have in the indicators and discover hidden patterns in the data we went ahead with exploring various methods for longitudinal data and time series data modeling and classification. We have tried approaches like

1. Isolation Forest Anomaly Detection
2. VAR based Anomaly Detection
3. DTW-based time series classification
4. CWT and CNN-based time series classification
5. DWT and a combination of ML classifiers like Random Forest, XGBoost , KNN and so on

All these methods are well-researched and established methods capable of using all the information in the data and discovering useful patterns. But in the current case DWT was the method successful in discovering the pattern classifying with good accuracy along with good precision and recall. Thus, this was selected as the final method and further tuned to make the final model which achieves 64% accuracy. The detailed schema and design are provided in the next section

# MODEL DESIGN AND INTERPRETATION

## A fresh perspective on information from indicators:

The outlier-based data exploration so far was focused mainly on making conclusions solely based on the absolute values of time delay indicators and retransmitted packets. This is because it’s well-known that more the time it takes to establish a TCP connection, to transfer the data and more the retransmissions worse will be the user experience. This is true but the user experience may depend on several factors :

* Scenario 1: A user with a high-speed connection would have low time delays i.e., there are very few outliers in indicators, yet his experience would be bad if the network fluctuates a lot and there are high retransmitted packets.
* Scenario 2: If a user’s activity involves many high-volume downloads or uploads, which often go smoothly, his experience would be good even if the time delay indicators are large or have many outliers.
* Scenario 3: Consider a user with a high-speed connection but bad performance in terms of package retransmission and connection failures/delays. But he/she uses the network minimally and the speed compensates usually compensates for the connection issues, the user may report a good experience.

From scenarios like the above, we can recognize that there might be several hidden factors apart from the absolute value of indicators at play to determine the overall experience of a user. Only when we use statistical and machine learning approaches to systematically extract these factors, we can discover clear patterns which define and user’s experience. This understanding has led us to explore several data transformation methods and machine learning methods and to build a final model based on DWT and Random Forest

## Discrete Wavelet Transformation(DWT)

Since the data provided is collected at various timestamps, we went ahead and looked at this data from a time-series analysis perspective and specifically from a signal-processing perspective. Signals are a type of time series. More specifically, signals are time-varying quantities that represent physical events. Thus, we can consider our data as a combination of 8 signals which are produced parallelly due to a user’s activity and together contain the information that can tell us about the user’s experience. The purpose of our analysis is

1. To extract this information using signal processing methods.
2. Discover a pattern clearly differentiating UGE from UBE using machine learning algorithms and train a model
3. Use the model to make predictions on the unseen or test data

For step 1 there are several methods available and transformations that change the time domain to a different one like the frequency domain had been applied to succeed in several use cases. Thus, we explored such methods and chose DWT as the one suitable for this use case. There are two main aspects of DWT:

1. It captures both frequency and location information (location in time) information as compared to other transformations like Fourier transformation.
2. Consecutive values of a time series are usually not independent but highly correlated, thus there is a lot of redundancy. DWT is very efficient at representing the signal while removing this redundancy.

Thus, DWT is used for Feature extraction to compress the time series, keeping only the important information while discarding noise and removing correlations.

## Preprocessing, Feature Extraction, and Experimentation:

* **DWT on Grouped Data** :Since we do not know which wavelet type can represent the data in the most efficient way, we developed 6 different sets of features using 6 types of wavelets
* **DWT on PCA on Grouped Data :** Its well-known transforming a multivariate dataset into orthogonal components helps in more informative useful feature generation. Thus, using PCA we created new set of variables. But it should be noted that that we had to choose 8 components to explain maximum variability and thus feature reduction was not achieved. DWT was performed over it
* **DWT on PCA of derivative of Grouped Data :** As discussed earlier in data exploration section in some scenarios instead of absolute values of the indicators, the fluctuation, or the rate of change of them can be more informative about the experience of the user. Thus, we created another set of features using PCA over estimated derivatives of the indicators

### ML Classifiers

For each of these sets, 10 different ML classifiers are trained.

1. XGBoost Classifier
2. Gradient Boosting Classifier
3. Random Forest
4. Nearest Neighbors
5. Decision Tree
6. Linear SVM
7. Neural Net
8. Naive Bayes
9. AdaBoost
10. Gaussian Process

Thus, a total of 180 experiments were performed for each combination of wavelet type, preprocessing type, and ML classifier with 150 UBE and 150 UGE users. These trained models’ performance was tested on the test dataset of 50 UBE and 50 UGE users and the 2 models with accuracies were selected for further hyperparameter tuning. A notable point here is that when the 50 validation users were added along with the 150 training users for each class, the performance was reduced, so we used only the training users provided.

# SCHEMATIC DIAGRAM SHOWING THE ARCHITECTURE OF MODEL

Diagram

Description automatically generatedBelow Image represents the data reading and pre-processing part

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

Text

Description automatically generated

# Machine learning algorithm parameters.

## Experimentation Phase:

During experimentations phase all the models are imported from sklearn python package and are run with the default parameters.

## Hyperparameter tuning :

1. In this phase initially a wide range of parameters grids was chosen for both Random forest and Nearest Neighbors Classifiers to run RandomSearchCV from sklearn with 3-fold cross validations.
2. When the best parameters are found a smaller range near the best parameters are used in GridsearchCV with 3-fold cross validation.
3. For Random Forest classifier below parameter grid was used

**{**'bootstrap'**:** **[True,** **False],**

'max\_features'**:** **[**'log2'**,** 'sqrt'**],**

'min\_samples\_leaf'**:** **[**1**,** 2**,** 4**],**

'min\_samples\_split'**:** **[**2**,** 3**,** 4**,** 5**,** 6**],**

'n\_estimators'**:** **[**90**,**91**,**92**,**93**,**94**,**95**,**96**,**97**,**

98**,**99**,**100**,**101**,**102**,**103**,**104**,**105**,**106**,**

107**,**108**,**109**,**110**]}**

1. For Nearest Neighbors below parameter grid was used

**{** 'n\_neighbors'**:** **range(**1**,**10**),**

'leaf\_size'**:** **range(**1**,**40**),**

'p'**:** **(**1**,**2**,**3**,**4**),**

'weights'**:** **(**'uniform'**,** 'distance'**),**

'metric'**:** **(**'minkowski'**,** 'chebyshev'**)** **}**

1. Below are the test accuracies :

Table

Description automatically generated

Graphical user interface, application, table

Description automatically generated

## Final Selected model parameters:

Since 64% test accuracy is the highest achieved in all experiments and tuning, we selected below model parameters as the final one.

tuned\_RF**=**RandomForestClassifier**(**n\_estimators**=**90**,**

min\_samples\_split **=** 2**,**

min\_samples\_leaf **=** 1**,**

max\_features **=**'sqrt'**,**

max\_depth **=** 20**,**

bootstrap **=** **True,**

random\_state**=**27**)**

# Accuracy metrics and ROC curve

Below is the training confusion matrix and ROC curve

precision recall f1-score support

0 1.00 1.00 1.00 150

1 1.00 1.00 1.00 150

accuracy 1.00 300

macro avg 1.00 1.00 1.00 300

weighted avg 1.00 1.00 1.00 300

Square

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Below is the test metrics and roc curve

precision recall f1-score support

0 0.63 0.66 0.65 50

1 0.65 0.62 0.63 50

accuracy 0.64 100

macro avg 0.64 0.64 0.64 100

weighted avg 0.64 0.64 0.64 100

Square

Description automatically generatedChart, line chart

Description automatically generated

# Conclusion:

1. It was demonstrated that the best accuracy is achieved using transformation and Machine Learning instead of solely relying on outliers.
2. Even though 64 % accuracy is decent using such efficient methods we can achieve higher accuracy if we have more informative variables. Thus, adding more indicators is also recommended