

Report - Forecasting signal strength in LoRaWan networks

Renata Rojas Guerra - Thiago Tavares Lopes

Contents

1	Introduction	1
2	Data description	1
3	Results	3

1 Introduction

2 Data description

```
# summary(tinovi01_RSSI[,c(3:6)])
# summary(tinovi02_RSSI[,c(3:6)])
# summary(tinovi03_RSSI[,c(3:6)])
# summary(tinovi04_RSSI[,c(3:6)])
# summary(tinovi05_RSSI[,c(3:6)])
# summary(tinovi06_RSSI[,c(3:6)])
# summary(milesight01_RSSI[,c(3:6)])
# summary(milesight02_RSSI[,c(3:6)])

datasets <- list(
  tinovi01 = tinovi01_RSSI[, 3:6],
  tinovi02 = tinovi02_RSSI[, 3:6],
  tinovi03 = tinovi03_RSSI[, 3:6],
  tinovi04 = tinovi04_RSSI[, 3:6],
  tinovi05 = tinovi05_RSSI[, 3:6],
  tinovi06 = tinovi06_RSSI[, 3:6],
  milesight01 = milesight01_RSSI[, 3:6],
  milesight02 = milesight02_RSSI[, 3:6]
)

# Função corrigida para processar cada summary
process_summary <- function(data, name) {
  sum_df <- as.data.frame(summary(data))

  # Extrair estatísticas únicas
  stats <- unique(sum_df$Var1)

  sum_df %>%
    mutate(Dataset = name) %>%
    pivot_wider(
      names_from = Var2,
      values_from = Freq
    ) %>%
```

```

    rename(Statistic = Var1) %>%
    select(Dataset, Statistic, everything())
}

# Combinar todos os summaries
all_summaries <- bind_rows(
  lapply(names(datasets), function(x) process_summary(datasets[[x]], x))
)

## Warning: Values from `Freq` are not uniquely identified; output will contain list-cols.
## * Use `values_fn = list` to suppress this warning.
## * Use `values_fn = {summary_fun}` to summarise duplicates.
## * Use the following dplyr code to identify duplicates.
## {data} |>
## dplyr::summarise(n = dplyr::n(), .by = c(Var1, Dataset, Var2)) |>
## dplyr::filter(n > 1L)
## Values from `Freq` are not uniquely identified; output will contain list-cols.
## * Use `values_fn = list` to suppress this warning.
## * Use `values_fn = {summary_fun}` to summarise duplicates.
## * Use the following dplyr code to identify duplicates.
## {data} |>
## dplyr::summarise(n = dplyr::n(), .by = c(Var1, Dataset, Var2)) |>
## dplyr::filter(n > 1L)
## Values from `Freq` are not uniquely identified; output will contain list-cols.
## * Use `values_fn = list` to suppress this warning.
## * Use `values_fn = {summary_fun}` to summarise duplicates.
## * Use the following dplyr code to identify duplicates.
## {data} |>
## dplyr::summarise(n = dplyr::n(), .by = c(Var1, Dataset, Var2)) |>
## dplyr::filter(n > 1L)
## Values from `Freq` are not uniquely identified; output will contain list-cols.
## * Use `values_fn = list` to suppress this warning.
## * Use `values_fn = {summary_fun}` to summarise duplicates.
## * Use the following dplyr code to identify duplicates.
## {data} |>
## dplyr::summarise(n = dplyr::n(), .by = c(Var1, Dataset, Var2)) |>
## dplyr::filter(n > 1L)
## Values from `Freq` are not uniquely identified; output will contain list-cols.
## * Use `values_fn = list` to suppress this warning.
## * Use `values_fn = {summary_fun}` to summarise duplicates.
## * Use the following dplyr code to identify duplicates.
## {data} |>
## dplyr::summarise(n = dplyr::n(), .by = c(Var1, Dataset, Var2)) |>
## dplyr::filter(n > 1L)
## Values from `Freq` are not uniquely identified; output will contain list-cols.
## * Use `values_fn = list` to suppress this warning.
## * Use `values_fn = {summary_fun}` to summarise duplicates.
## * Use the following dplyr code to identify duplicates.
## {data} |>
## dplyr::summarise(n = dplyr::n(), .by = c(Var1, Dataset, Var2)) |>
## dplyr::filter(n > 1L)
## Values from `Freq` are not uniquely identified; output will contain list-cols.
## * Use `values_fn = list` to suppress this warning.
## * Use `values_fn = {summary_fun}` to summarise duplicates.
## * Use the following dplyr code to identify duplicates.
## {data} |>

```

```
##   dplyr::summarise(n = dplyr::n(), .by = c(Var1, Dataset, Var2)) |>
##   dplyr::filter(n > 1L)
## Values from `Freq` are not uniquely identified; output will contain list-cols.
## * Use `values_fn = list` to suppress this warning.
## * Use `values_fn = {summary_fun}` to summarise duplicates.
## * Use the following dplyr code to identify duplicates.
##   {data} |>
##   dplyr::summarise(n = dplyr::n(), .by = c(Var1, Dataset, Var2)) |>
##   dplyr::filter(n > 1L)
```

Table 1: Sumário Estatístico dos Conjuntos de Dados

3 Results