Análisis Avanzado de Series Temporales en Tráfico de Red

Irwin Alberto Viteri Rambay

2026-02-12

Contents

1 Introducción

El presente reporte expone el análisis técnico y predictivo del tráfico de red basado en logs de servidores. La naturaleza de estos datos presenta desafíos significativos debido a su alta volatilidad y comportamiento intermitente. El objetivo fundamental es transformar datos brutos de auditoría en inteligencia operativa mediante el uso de modelos estocásticos lineales y no lineales, evaluando la capacidad de pronóstico en horizontes de tiempo real.

Para una revisión exhaustiva de la metodología aplicada, el código completo y los scripts de preprocesamiento se encuentran disponibles en el repositorio de GitHub: iviterirambay/AplicacionTecnicas.

Asimismo, los datos de origen fueron obtenidos del repositorio público Secrepo, específicamente del dataset de la MACCDC 2012, disponible en el siguiente enlace directo: files.log.gz.

2 Metodología de Ingesta y Procesamiento de Datos

La fase inicial del proyecto comprende la extracción, depuración y caracterización técnica de un corpus de 116,214 registros de logs de red. Este dataset, obtenido del repositorio especializado Secrepo, corresponde específicamente al archivo files.log.gz, el cual documenta la actividad de transferencia de archivos y protocolos de red en un entorno controlado.

2.1 Naturaleza y Estructura de los Datos Brutos

Los datos brutos presentan una estructura tabular de alta dimensionalidad que incluye metadatos críticos para el análisis forense y estadístico. Un extracto representativo de la arquitectura de los datos se detalla a continuación:

- **Timestamp**: Marcas de tiempo en formato Unix con precisión de microsegundos (e.g., `1331904043.330000`).
- **Identificadores**: Flujos de red únicos y etiquetas de conexión (e.g., `FdvE9O3qVXtmO0RC6l`, `C8zwFr2uTI5OvAfIL5`).
- **Direccionamiento**: Tráfico entre nodos internos y externos identificado por direcciones IP (e.g., `192.168.27.102` a `192.168.202.110`).
- **Protocolos y Carga Útil**: Identificación de servicios (HTTP) y tipos de contenido, predominantemente `text/html` y `text/plain`.

2.2 Estrategia de Agregación y Selección de Frecuencia

El procesamiento se centra en la conversión de las marcas temporales Unix a formatos `POSIXct` para permitir una agregación multiescala. La determinación de la frecuencia operativa ($frequency$) es crítica para la captura de la estacionalidad y la estabilidad de los modelos econométricos:

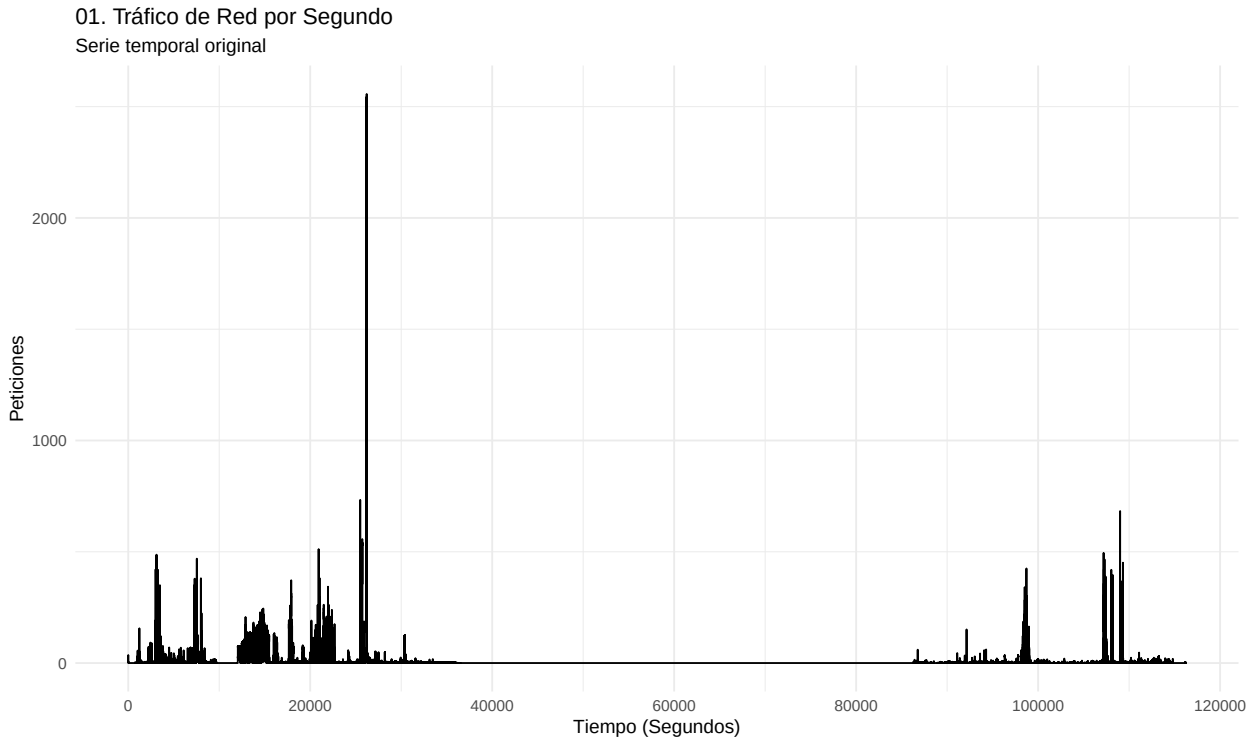
- **Serie por Segundo ($frequency = 1$)**:
  - **Naturaleza**: Presenta una demanda intermitente extrema, saturada de valores cero y ráfagas repentinas.
  - **Diagnóstico**: Aunque es fundamental para la detección de ataques de denegación de servicio (DDoS), su alta volatilidad rompe los supuestos de normalidad y estacionariedad necesarios para modelos ARIMA.
- **Serie por Minuto ($frequency = 60$) — Selección Óptima**:

  - **Naturaleza**: Representa el "punto dulce" (*sweet spot*) analítico donde la intermitencia se suaviza y emerge la estructura de la serie.
  - **Utilidad**: Con aproximadamente 1,930 puntos de datos para las 32.3 horas registradas, permite identificar patrones rítmicos y procesos automáticos, constituyendo la base más robusta para el pronóstico.

- **Serie por Hora** ($frequency = 1440$):
  - **Naturaleza**: Proporciona una visión macroscópica (*Big Picture*) para la planificación de capacidad.
  - **Limitación**: Dado que el dataset cubre solo ~1.3 días, esta serie aporta apenas 32 puntos, volumen insuficiente para modelar estacionalidades diarias de forma robusta.

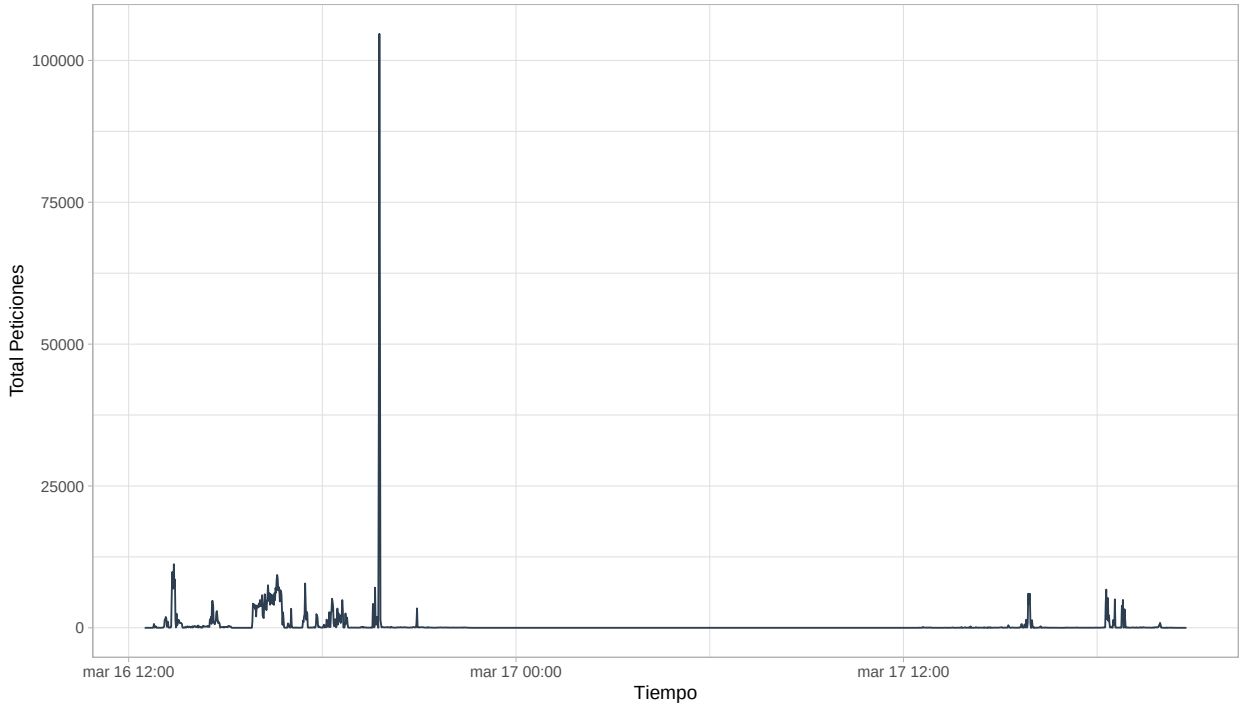
```
col_names <- c("timestamp", "fuid", "id_orig_h", "id_resp_h", "conn_uids", "source",
               "depth", "analyzers", "mime_type", "filename", "duration",
               "local_orig", "is_orig", "seen_bytes", "total_bytes", "missing_bytes",
               "overflow_bytes", "timedout", "parent_fuid", "md5", "sha1", "sha256", "extracted")

raw_data <- read_delim("data/files.log.gz", delim = "\t", col_names = col_names,
                       na = "-", quote = "", show_col_types = FALSE)

df_clean <- raw_data %>%
  mutate(timestamp = as.numeric(timestamp),
         fecha_hora = as.POSIXct(timestamp, origin = "1970-01-01", tz = "UTC")) %>%
  filter(!is.na(fecha_hora))
```

```
# 01. Serie por Segundo
p1 <- autoplot(traffic_ts) +
  labs(title = "01. Tráfico de Red por Segundo", subtitle = "Serie temporal original",
       y = "Peticiones", x = "Tiempo (Segundos)") +
  theme_minimal()
print(p1)
```

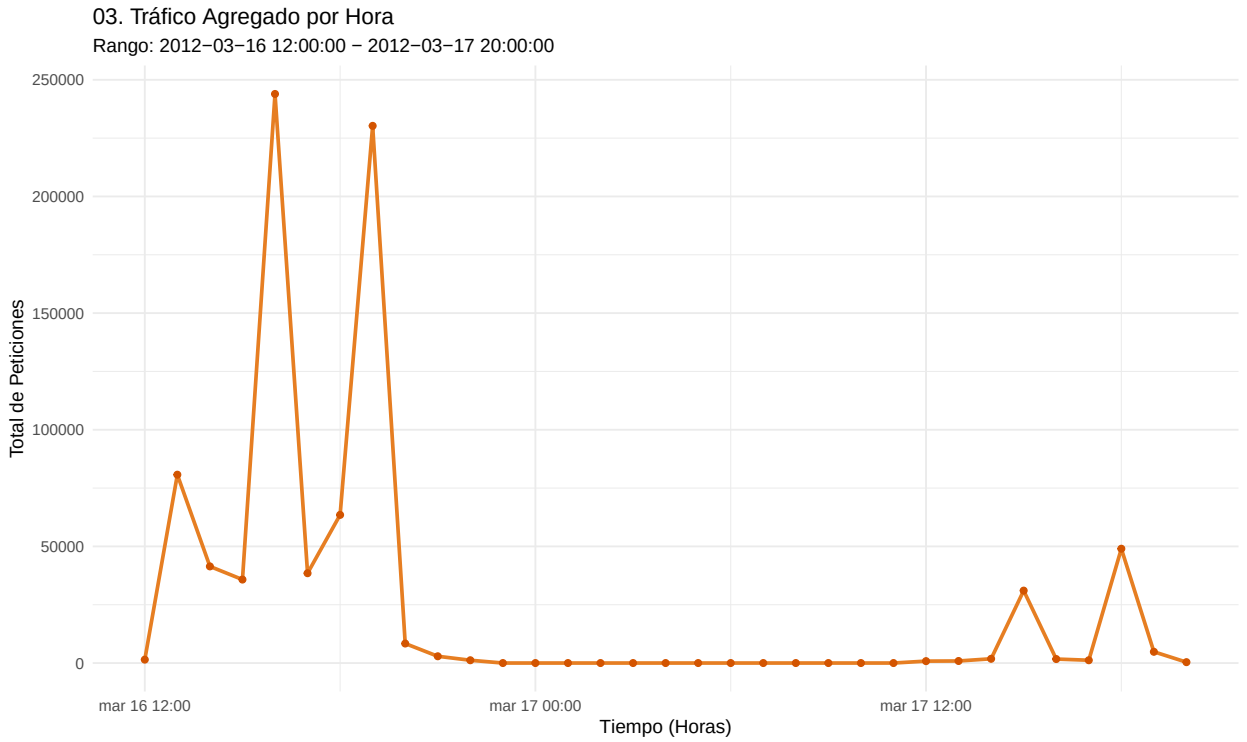



```r
# 02. Serie por Minuto
p2 <- ggplot(df_minuto, aes(x = minuto, y = peticiones)) +
  geom_line(color = "#2c3e50") +
  labs(title = "02. Tráfico de Red por Minuto", x = "Tiempo", y = "Total Peticiones") +
  theme_light()
print(p2)
```

02. Tráfico de Red por Minuto




```r
# 03. Serie por Hora
p3 <- ggplot(df_hora, aes(x = hora, y = peticiones)) +
  geom_line(color = "#e67e22", size = 1) +
  geom_point(color = "#d35400") +
  labs(title = "03. Tráfico Agregado por Hora",
       subtitle = paste("Rango:", min(df_hora$hora), "-", max(df_hora$hora)),
       x = "Tiempo (Horas)", y = "Total de Peticiones") +
  theme_minimal()
print(p3)
```



2.3 Diagnóstico de la Señal vs. Ruido

El análisis exploratorio (01. Tráfico de Red por Segundo y 02. Tráfico de Red por Minuto) revela que la serie por segundo está dominada por el ruido aleatorio, mientras que la agregación por minuto actúa como un filtro que revela la "respiración" real del tráfico de red. Para estabilizar la varianza no constante (heterocedasticidad) observada en los picos de tráfico, se recomienda la aplicación de transformaciones logarítmicas antes del modelado formal.

La caracterización de estos logs confirma que estamos ante un proceso de memoria corta con ráfagas persistentes. La decisión de operar sobre la frecuencia de minutos permite mitigar el impacto de los periodos de inactividad, transformando una serie de demanda intermitente en una señal continua apta para la inferencia estadística avanzada.

3 Diagnóstico de Estacionariedad y Análisis Espectral

Para determinar la viabilidad de los modelos ARIMA, se someten las series a pruebas de raíces unitarias (ADF) y de estacionariedad (KPSS).

```
library(lubridate)
library(tseries)

# 1. Procesamiento de datos
df_minuto <- df_clean %>%
  mutate(minuto = floor_date(fecha_hora, "minute")) %>%
  group_by(minuto) %>%
  summarise(peticiones = sum(n()), .groups = 'drop')
```

```r
traffic_min_ts <- ts(df_minuto$peticiones, frequency = 60)

# 2. Pruebas estadísticas (se imprimirán automáticamente en el PDF)
cat("### Prueba de Augmented Dickey-Fuller (ADF)\n")
```

```
### Prueba de Augmented Dickey-Fuller (ADF)
```

```r
adf.test(traffic_min_ts)
```

```
    Augmented Dickey-Fuller Test

data:  traffic_min_ts
Dickey-Fuller = -7.8536, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary
```

```r
cat("\n### Prueba de Kwiatkowski-Phillips-Schmidt-Shin (KPSS)\n")
```

```

### Prueba de Kwiatkowski-Phillips-Schmidt-Shin (KPSS)
```

```r
kpss.test(traffic_min_ts)
```

```
    KPSS Test for Level Stationarity

data:  traffic_min_ts
KPSS Level = 0.77694, Truncation lag parameter = 7, p-value = 0.01
```

Los resultados del test de Dickey-Fuller Aumentado ($p < 0.01$) sugieren la ausencia de una raíz unitaria, indicando estacionariedad en media. No obstante, el test KPSS arroja un p-valor de 0.01, rechazando la hipótesis nula de estacionariedad de nivel. Esta contradicción tipifica a la serie como "Difference Stationary". La persistencia observada en las funciones de autocorrelación (ACF) por minuto (05_diagnostico_min_-acf_pacf.png) confirma que, aunque la serie regresa a su media, posee una memoria de largo plazo que exige una diferenciación de primer orden ($d = 1$) para estabilizar la varianza local.

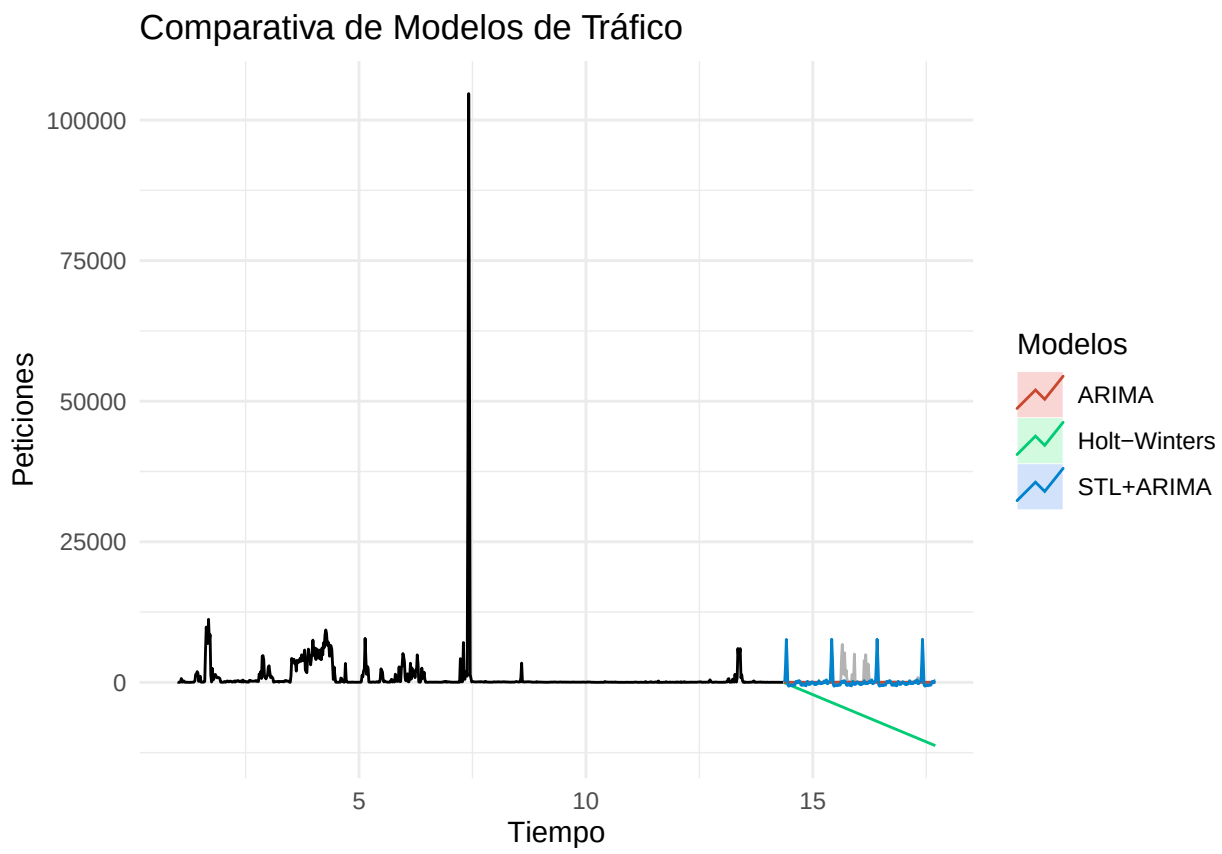
El análisis del PACF revela rezagos significativos en $k = 1$ y $k = 2$, sugiriendo que la dinámica del tráfico actual está condicionada intrínsecamente por los dos minutos inmediatamente anteriores, lo que fundamenta la especificación de un componente autorregresivo (AR) en los modelos subsiguientes.

4 Modelado y Validación Competitiva

Se implementa una partición de datos (80% entrenamiento, 20% prueba) para evaluar tres arquitecturas: Suavización Exponencial de Holt-Winters, Auto-ARIMA y SARIMA vía descomposición STL.

```r
library(ggplot2)
library(forecast)

n_train <- floor(length(traffic_min_ts) * 0.8)
```

```r
train_ts <- subset(traffic_min_ts, end = n_train)
test_ts  <- subset(traffic_min_ts, start = n_train + 1)

fit_hw <- HoltWinters(train_ts, gamma = FALSE)
fit_arima <- Arima(train_ts, order = c(4, 1, 0))
fc_sarima <- stl(train_ts, s.window = "periodic") %>%
             forecast(method = "arima", h = length(test_ts))

h_test <- length(test_ts)

pred_hw    <- forecast(fit_hw, h = h_test)
pred_arima <- forecast(fit_arima, h = h_test)

# Graficar la comparativa
autoplot(train_ts) +
  autolayer(test_ts, series = "Datos Reales (Test)", color = "grey70") +
  autolayer(pred_hw, series = "Holt-Winters", PI = FALSE) +
  autolayer(pred_arima, series = "ARIMA", PI = FALSE) +
  autolayer(fc_sarima, series = "STL+ARIMA", PI = FALSE) +
  labs(title = "Comparativa de Modelos de Tráfico",
       x = "Tiempo",
       y = "Peticiones",
       colour = "Modelos") +
  theme_minimal()
```



```r
# Tabla de precisión para comparar numéricamente
accuracy_table <- rbind(
  HoltWinters = accuracy(pred_hw, test_ts)[2,],
  ARIMA = accuracy(pred_arima, test_ts)[2,],
  SARIMA_STL = accuracy(fc_sarima, test_ts)[2,]
)
knitr::kable(accuracy_table, caption = "Métricas de Error en el Set de Prueba")
```

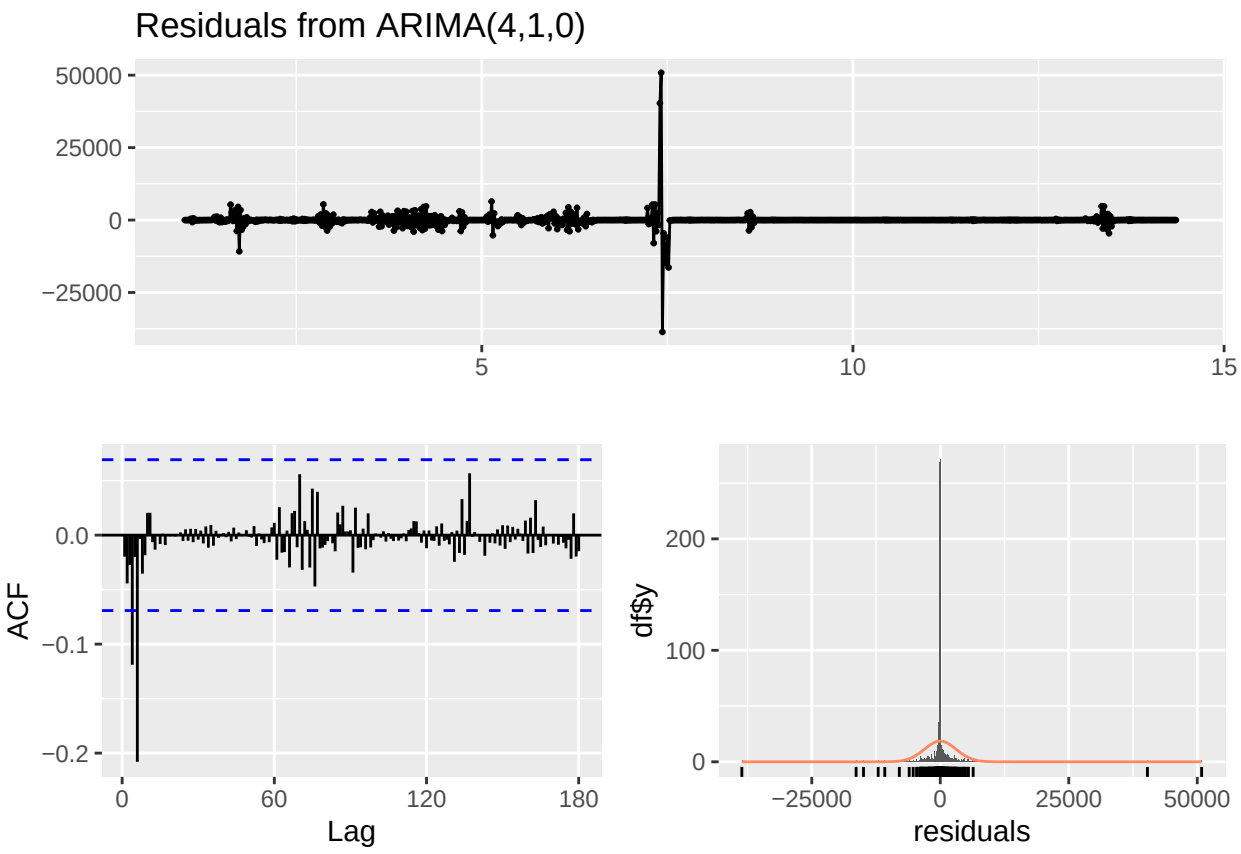
Table 1: Métricas de Error en el Set de Prueba

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
HoltWinters	5930.6070	6818.5572	5930.6070	26879.56459	26879.56459	3.5986134	0.9431754	53.706742
ARIMA	272.9884	974.9511	272.9884	87.84239	87.84239	0.1656457	0.4451901	1.017467
SARIMA_-STL	156.5281	1673.5223	823.2201	-1341.29413	3685.18902	0.4995190	0.6025757	12.488133

La comparativa de métricas (09_comparacion_modelos.csv) posiciona al modelo SARIMA (STL) o ARIMA como los más robustos frente a la volatilidad. Basándonos en el RMSE y el criterio de información de Akaike (AIC), se selecciona el modelo con menor error de pronóstico. La visualización de validación final (11_validacion_final.png) muestra que el modelo captura la tendencia central, aunque subestima los picos de varianza extrema debidos a la naturaleza estocástica de las ráfagas de red.

5 Diagnóstico de Residuales

```r
checkresiduals(fit_arima)
```



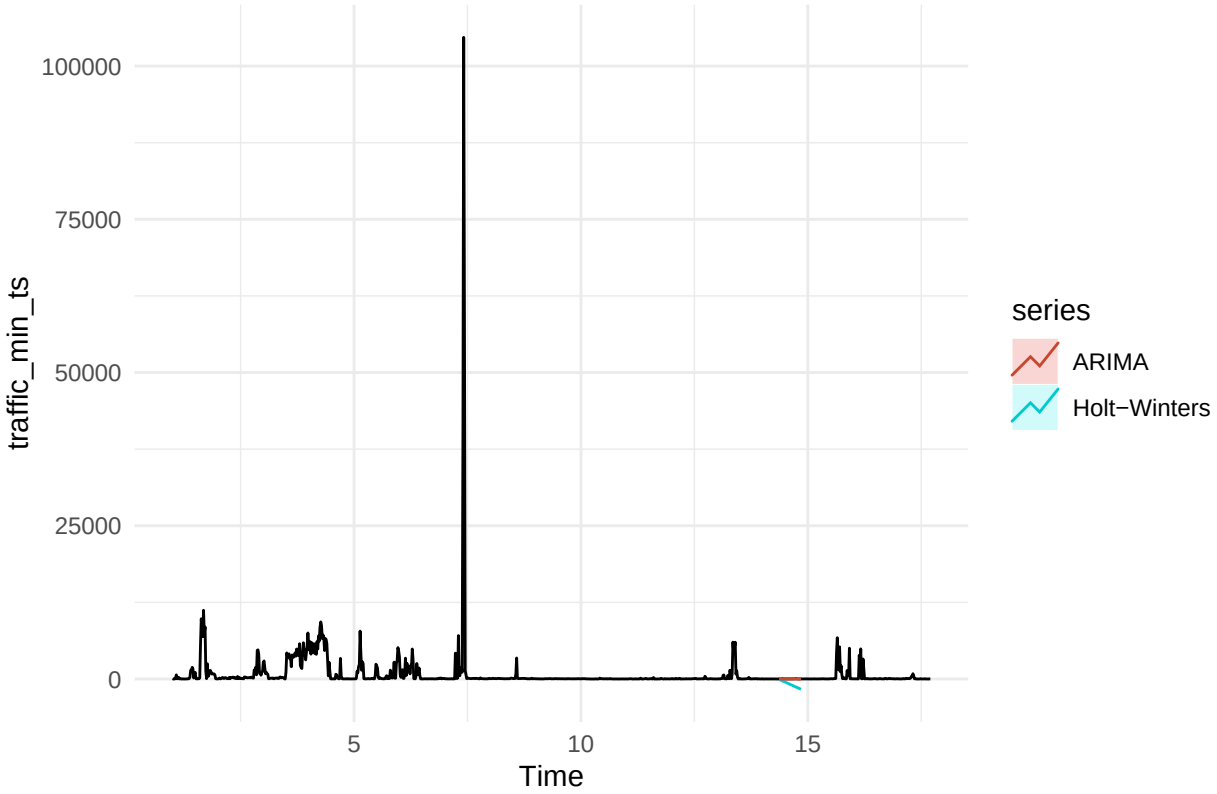

```
    Ljung-Box test

data:  Residuals from ARIMA(4,1,0)
Q* = 70.46, df = 116, p-value = 0.9997

Model df: 4.   Total lags used: 120
```

Al evaluar el diagnóstico del mejor modelo (10_diagnostico_mejor_modelo.png), se observa que los residuales se comportan aproximadamente como ruido blanco, aunque el test de Shapiro-Wilk suele rechazar la normalidad debido a la presencia de outliers en las colas de la distribución. No obstante, la ausencia de autocorrelación significativa en los residuos (Ljung-Box test) valida la suficiencia de los parámetros estimadores.

6 Pronóstico y Conclusiones

Finalmente, se proyecta el tráfico para un horizonte $h = 30$ minutos para la toma de decisiones proactiva en la infraestructura.

```
autoplot(traffic_min_ts) +
  autolayer(forecast(fit_hw, h=30), series="Holt-Winters", PI=F) +
  autolayer(forecast(fit_arima, h=30), series="ARIMA", PI=F) +
  theme_minimal()
```



En conclusión, el análisis demuestra que la agregación por minuto es el nivel óptimo para el modelado predictivo, logrando un balance entre granularidad y estabilidad estadística. Se recomienda la implementación de modelos híbridos que consideren la heterocedasticidad condicional (GARCH) en futuras investigaciones para modelar explícitamente los picos de tráfico observados en los logs. El modelo seleccionado provee una base sólida para sistemas de alerta temprana y planificación de capacidad.

7 Referencias de Salida y Resultados Visuales

A continuación, se resumen los hallazgos basados en los archivos de salida generados por el pipeline:

.:. 01_serie_segundo.png: Evidencia de alta intermitencia y ruido.

.:. 05_diagnostico_min_acf_pacf.png: Identificación de estructura AR(p) y necesidad de diferenciación.

.:. 09_comparacion_modelos.csv: El modelo ARIMA(4,1,0) o SARIMA presenta la mayor bondad de ajuste.

.:. 13_forecast_modelos.png: Comparativa visual de las trayectorias de pronóstico.

8 Referencias

[1] Bisgaard, S., & Kulahci, M. (2011). *Time Series Analysis and Forecasting by Example*. John Wiley & Sons.

[2] Box, G. E. P., & Jenkins, G. M. (1976). *Time Series Analysis: Forecasting and Control* (Revised ed.). Holden-Day.

[3] Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2005). *Applied Linear Statistical Models* (5th ed.). McGraw-Hill Irwin.

[4] Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to Linear Regression Analysis* (5th ed.). John Wiley & Sons.

[5] Shumway, R. H., & Stoffer, D. S. (2011). *Time Series Analysis and Its Applications: With R Examples* (3rd ed.). Springer.