Поэтапное объяснение практической работы

Ваша работа — это полный цикл анализа и прогнозирования трафика IoT. Я объясню шаг за шагом, что происходит, с ссылками на коды. Это поможет понять логику и подготовиться к защите.

1. **Сбор и фильтрация данных (Wireshark, раздел 1 отчёта).**
   Открываете PCAP-файл. Фильтруете по MAC IoT (eth.addr OR). Меняете время на дельту (Seconds Since Previous Packet). Строите IO Graph (пакеты по времени). Экспортируете в CSV (Time — интервалы, Length — длина).
   Цель: Выделить IoT-трафик, получить данные для анализа.
   Код: Нет, это ручной процесс в Wireshark.
2. **Предобработка интервалов (`lizaMO.py` , подготовка для Хёрста).**
   Читаете Time из CSV. Логарифмируете разности (log*dif* = *ln(Time_t) - ln(Time*{t+1})). Сохраняете в logDifference.csv.
   Цель: Сделать ряд стационарным (убрать тренды).
   Код: Основной цикл — вычисление log_dif.
3. **Аппроксимация распределений и Хёрст (`lizaFirst.py` , разделы 2–3 отчёта).**
   - Гистограмма интервалов (bins по 0.03 с). Аппроксимация P_i = sum Pareto + Exp + 2 Gamma. Подбор K1–K4. RMSE/RMST.
   - Хёрст: Дисперсионный (log(var) vs log(m), H=1 - |slope|/2). R/S (R/S vs log(block), H=slope). Графики.
     Цель: Описать распределение интервалов, проверить самоподобие.
     Код: Блок с Pareto/Gamma и сегментами.
4. **Агрегация и ARIMA (`lizaSecond.py` , разделы 4–6 отчёта).**
   Агрегируете Length по 50с. Выбираете окошки. Проверяете стационарность (Дики-Фуллера на diff). Подбираете ARIMA по AIC. Прогноз, MAPE/SER. Диагностика остатков.
   Цель: Прогнозировать трафик.
   Код: Resample, adfuller, цикл pdq, fit().

Работа показывает, как IoT-трафик (нестационарный, самоподобный) моделировать и прогнозировать.

Вопросы, которые может задать преподаватель, и ответы

Я придумал 10 типичных вопросов по всей работе (на основе методички, твоего отчёта и кода). Они охватывают ключевые моменты. Ответы — краткие, но полные, чтобы ты мог повторить на защите.

1. **Почему именно этот фильтр в Wireshark? (Раздел 1)**
   Ответ: Фильтр по MAC-адресам выделяет трафик от IoT-устройств (камеры, датчики, розетки) из методички. Он отбрасывает не-IoT (ПК, роутеры), чтобы избежать шума. Адреса взяты из открытых датасетов IoT-трафика.
2. **Что показывает IO Graph в Wireshark?**
   Ответ: Распределение трафика по времени (пакеты/байты). В IoT пики активности часто ночью (синхронизация устройств), что подтверждает нестационарность трафика (изменение интенсивности).
3. **Почему смешанная модель для аппроксимации интервалов? (Раздел 2)**
   Ответ: Исходная гистограмма интервалов имеет пик в начале (короткие паузы — Gamma/Exp) и тяжёлый хвост (длинные паузы — Pareto). Смешанная модель (K1·Pareto + K2·Exp + K3·Gamma1 + K4·Gamma2) лучше описывает эту форму, чем одно распределение. Подбор K по минимизации RMSE/RMST.
4. **Что такое RMSE и RMST, как они считаются?**
   Ответ: RMSE — средняя локальная ошибка между эмпирической P_i и моделью sum (√(1/n Σ (P_i - sum_i)^2)). RMST — кумулятивная ошибка CDF (√(1/n Σ (P_C - sum_c)^2)). RMSE для отдельных интервалов, RMST для всего распределения. У меня RMSE=0.0426, RMST=0.0185 — модель хорошая.
5. **Что такое стационарность и как вы её проверяли? (Разделы 3–4)**
   Ответ: Стационарный ряд — среднее, дисперсия и автокорреляция не меняются во времени. IoT-трафик нестационарен (пики, тренды). Проверяли тестом Дики-Фуллера на diff() (p<0.05 — стационарен). Для Хёрста — лог-разности; для ARIMA — d=1.
6. **Как считается Хёрст по дисперсионному методу? (Раздел 3)**
   Ответ: Разбиваем лог-разности на сегменты m=[4,8,16,...]. Для каждого m считаем дисперсию сегментов, затем log(var) vs log(m). Наклон β = slope регрессии, H = 1 - |β|/2. У меня H=0.643 — персистентность.
7. **Как считается Хёрст по R/S-методу?**
   Ответ: Для блоков [500,1000,...]: В каждом блоке R = max - min накопленных отклонений, S = std. Avg R/S по блокам. Log(R/S) vs log(block), H = slope. У меня H=0.045 — низкий, возможно, маленькие блоки.
8. **Почему разные H в двух методах?**
   Ответ: Дисперсионный чувствителен к дисперсии на разных масштабах (H=0.643 — память есть). R/S — к размаху (H=0.045 — слабая зависимость на выбранных блоках). Разница может быть из-за размеров блоков или шума — рекомендую увеличить блоки для R/S.
9. **Что такое MAPE и SER, почему такие значения? (Раздел 5)**
   Ответ: MAPE = 100% * mean(|(data - forecast)/data|) — процентная ошибка прогноза (13–42%, нормально для IoT с пиками). SER = 10 log10 (сигнал^2 / ошибка^2) — отношение

сигнала к шуму в дБ (>10 дБ — хорошо). Высокий MAPE на сегменте с аномалией — ARIMA не идеальна для всплесков.

10. **Что показывают остатки ARIMA, почему не нормальные? (Раздел 6)**
    Ответ: Остатки = реальные - прогноз (ε_t). Должны быть белым шумом. Тесты: Ljung-Box (нет корреляции), ARCH (нет гетероскедастичности) — OK. Jarque-Bera p<0.05 — не нормальны (тяжёлые хвосты, типично для трафика с импульсами). Это не критично, модель всё равно адекватна.

Объяснение таблицы summary() модели ARIMA(3,1,3)

Это стандартный вывод `statsmodels` после `model.fit().summary()` . Он содержит всю информацию об обученной модели ARIMA(3,1,3). Разберём по блокам простыми словами.

1. Заголовок модели

- **Dep. Variable: Length** — зависимая переменная (что прогнозируем) — средняя длина пакетов.
- **No. Observations: 120** — количество точек в сегменте (120 агрегированных интервалов по 50 с).
- **Model: ARIMA(3, 1, 3)** — порядок модели: p=3 (AR), d=1 (дифференцирование), q=3 (MA).
- **Log Likelihood: -560.862** — логарифм правдоподобия (чем выше — лучше модель).
- **AIC: 1135.724** — критерий Акаике (чем меньше — лучше, используется для выбора модели).
- **BIC: 1155.177** — байесовский критерий (штрафует за сложность).
- **Sample: 09-25-2016 - 09-25-2016** — период данных (один день в твоём случае).
- **Covariance Type: opg** — метод расчёта ковариации (не важно для понимания).

2. Таблица коэффициентов (главная часть)

Параметр	coef	std err	z	P>	z		[0.025 0.975]
ar.L1	-1.1928	0.222	-5.362	0.000	-1.629 -0.757		
ar.L2	-1.1403	0.184	-6.194	0.000	-1.501 -0.779		
ar.L3	-0.2256	0.134	-1.686	0.092	-0.488 0.037		
ma.L1	-0.1154	0.337	-0.342	0.732	-0.776 0.545		

Параметр	coef	std err	z	P>	z		[0.025 0.975]
ma.L2	-0.1535	0.295	-0.520	0.603	-0.732 0.425		
ma.L3	-0.7246	0.258	-2.810	0.005	-1.230 -0.219		
sigma2	691.6655	176.282	3.924	0.000	346.158 1037.173		

Что значит каждая колонка:

- **coef** — коэффициент модели (вес влияния).
  - ar.L1, ar.L2, ar.L3 — коэффициенты авторегрессии (AR): насколько прошлые значения (с лагом 1,2,3) влияют на текущее после дифференцирования.
  - ma.L1, ma.L2, ma.L3 — коэффициенты скользящей средней (MA): влияние прошлых ошибок.
  - sigma2 — дисперсия шума (ε_t).
- **std err** — стандартная ошибка коэффициента (чем меньше — точнее оценка).
- **z** — z-статистика (coef / std err).
- **P>|z|** — p-value: если <0.05 — коэффициент значим (влияет на модель).
  - ar.L1 и ar.L2 — очень значимы (p=0.000).
  - ar.L3 — почти значим (p=0.092).
  - ma.L1 и ma.L2 — незначимы (p>0.05) → можно упростить модель.
  - ma.L3 — значим (p=0.005).
- **[0.025 0.975]** — 95% доверительный интервал (если не включает 0 — значим).

Вывод по коэффициентам: Модель в основном опирается на AR(2) и MA(3). Некоторые MA-коэффициенты незначимы — модель можно упростить (но AIC выбрал эту).

3. Диагностика остатков (нижняя часть)

- **Ljung-Box (Q): 0.01, Prob(Q): 0.93** — тест на автокорреляцию остатков. p=0.93 >0.05 → нет автокорреляции (хорошо!).
- **Jarque-Bera (JB): 1816.77, Prob(JB): 0.00** — тест на нормальность. p=0.00 <0.05 → остатки НЕ нормальны (тяжёлые хвосты, типично для трафика).
- **Heteroskedasticity (H): 3.47, Prob(H): 0.00** — тест на гетероскедастичность. p=0.00 → дисперсия остатков меняется (непостоянна) — плохо, но в IoT часто так.
- **Skew: 3.05** — асимметрия (положительная — правый хвост тяжёлый).
- **Kurtosis: 21.15** — эксцесс (очень высокий — острые пики и тяжёлые хвосты).

Общий вывод по модели:
Модель ARIMA(3,1,3) захватывает зависимость (нет автокорреляции в остатках), но остатки не нормальны и гетероскедастичны — типично для реального трафика с всплесками. Модель приемлема для прогноза, но не идеальна.

В отчёте пиши: "Коэффициенты AR значимы, остатки без автокорреляции, но с отклонением от нормальности (JB $p<0.05$), что характерно для IoT-трафика."