



Санкт-Петербургский государственный университет  
Кафедра системного программирования

# Разработка прототипа системы определения тренда во временном ряде

Нафикова Лиана Ирековна

**Научный руководитель:** доктор физ.матем.наук, профессор Граничин Олег Николаевич

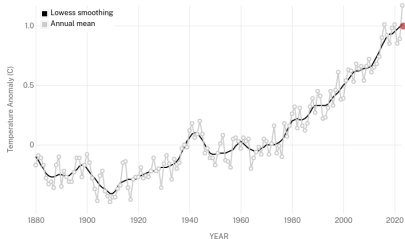
**Рецензент:** Deep Learning Engineer Brask Inc. Панков Викентий Дмитриевич

Санкт-Петербург  
2024

Временной ряд  $x_1, \dots, x_n$  – значения признака, измеренные через равные временные интервалы

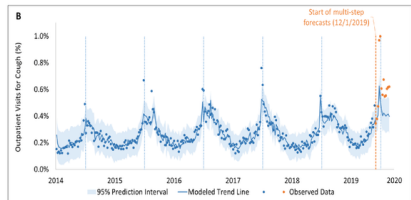
## GLOBAL LAND-OCEAN TEMPERATURE INDEX

Data source: NASA's Goddard Institute for Space Studies (GISS). Credit: NASA/GISS



## Tesla, Inc. Common Stock (TSLA)

1D 1M 8M YTD 1Y 5Y MAX



## Структура временного ряда

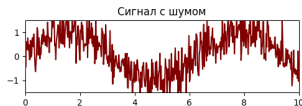
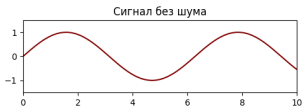
- Подбор методов анализа зависит от особенностей каждого временного ряда

## Причины ограничения объема данных

- Дорогостоящий мониторинг
- Быстрое изменение условий среды
- Невозможность проведения экспериментов

## Шум в данных

- Данные могут быть подвержены влиянию случайных факторов и погрешностей



# Постановка задачи

**Целью** работы является создание прототипа системы для анализа временных рядов, определения их трендов и моментов существенных изменений трендов

**Задачи:**

- сформулировать требования к системе
- провести обзор существующих решений и исследовать алгоритмы, позволяющие определять тренд по малому числу входных данных с шумом
- разработать архитектуру системы
- реализовать прототип системы
- провести эксперименты

## Для алгоритмов

- Возможность настраивать параметры алгоритмов
- Возможность задавать разные модели шума

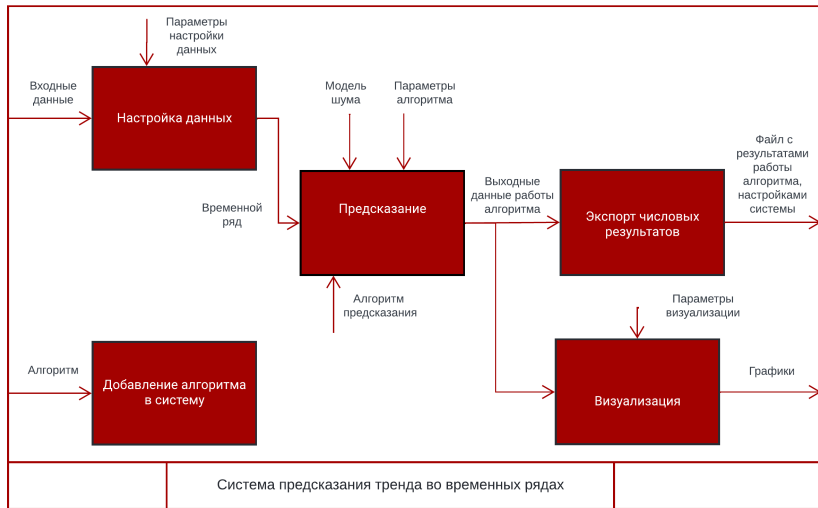
## Для прототипа системы

- Работа с разными режимами поступления данных
- Сохранение результатов работы
- Пользовательский интерфейс

	Orange	forecast	KNIME	Prophet
Настройки алгоритмов	+	+	+	+
Генерация данных	-	-	+	-
Загрузка датасетов	+	+	+	+
Чтение данных в режиме реального времени	-	-	+	-
Пользовательские алгоритмы	+	-	+	-
Модель шума	-	-	-	-
Экспорт результатов	+	+	+	+
GUI	+	-	+	-

Таблица: Сравнительная характеристика систем

# Архитектура системы



# Используемые технологии

Для реализации **серверной части**:



Python



Flask (фреймворк для создания веб-приложений)

Для реализации **клиентской части**:



JavaScript



React



Библиотека компонентов Material UI



Библиотека d3.js (для визуализации графиков)



# Математическая формулировка задачи

Задача определения тренда:

Найти  $T_t$ :

$$x_t = T_t + \varepsilon_t,$$

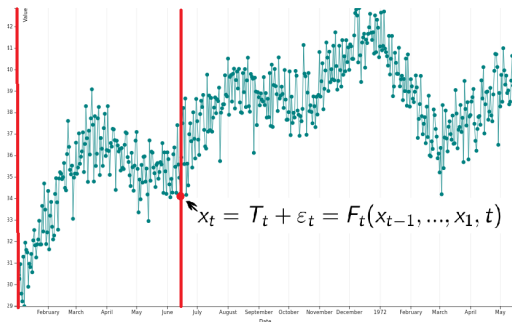
где  $T_t$  – тренд,  $\varepsilon_t$  – шум

Задача прогнозирования:

Найти функцию  $F_t$ :

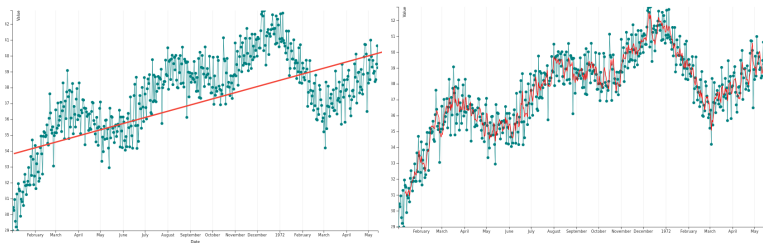
$$\hat{x}_{t|t-1} \approx F_t(x_{t-1}, \dots, x_1, t),$$

где  $\hat{x}_{t|t-1}$  – предсказание, соответствующее моменту времени  $t$  по  $t-1$  предыдущим измерениям

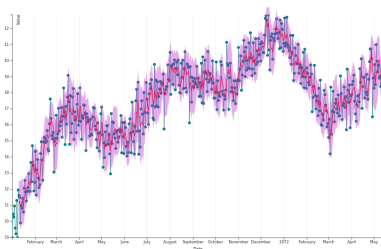


# Особенности реализации

## Использование режима скользящего окна



## Построение доверительного интервала

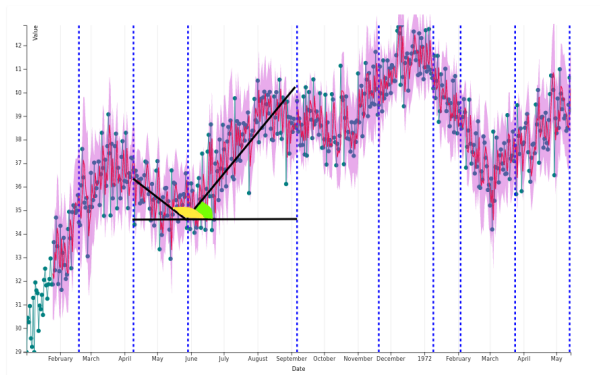


# Особенности реализации

## Вычисление моментов изменения направления тренда

- Угловые коэффициенты для каждого окна данных шириной  $W$  определяются как:  $m_i = \frac{x_{i+W} - x_i}{W}$
- Тренд считается измененным, если:

$$|\arctan m_{\text{previous}} - \arctan m_{\text{current}}| > \alpha$$



## Простые методы

- **Метод Хольта**

Экспоненциальное сглаживание с учетом локального изменения тренда

## Фильтры

- **Фильтр Калмана**

Оптимальная оценка состояния динамической системы

## ARIMA-методы

- **AR, MA, ARMA, ARIMA**

Модель авторегрессии и интегрированного скользящего среднего

## Time-based методы

- **Линейная регрессия**

- **SPS (Sign-Perturbed Sums)**

Метод оценки параметров

# SPS (Sign-Perturbed Sums)

## Модель наблюдений

$$Y = X^T \theta + E$$

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_T \end{pmatrix} \text{ — вектор значений тренда}$$

$$X = \begin{pmatrix} t_1 & t_2 & \dots & t_T \\ 1 & 1 & \dots & 1 \end{pmatrix} \text{ — матрица признаков}$$

$$\theta = \begin{pmatrix} k \\ b \end{pmatrix} \text{ — вектор параметров, которые необходимо оценить}$$

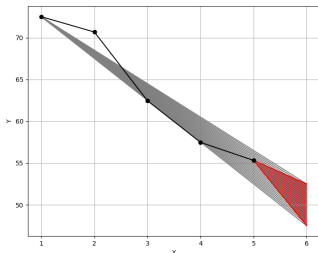
$$E = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_T \end{pmatrix} \text{ — вектор случайных ошибок}$$

# Инициализация SPS

- Выбор начальных параметров  $M$  и  $q$ , с помощью которых определяется доверительная вероятность  $p = 1 - \frac{q}{M}$ ,  $M > q > 0$
- Генерируется матрица знаков

$$SIGNS = \{\beta_{t,j}\} = \begin{pmatrix} 1 & \pm 1 & \cdots & \pm 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \pm 1 & \cdots & \pm 1 \end{pmatrix}_{T \times M}$$

- Построение доверительного множества  $\Theta = \{\theta \in \mathbb{R}^2 \mid SPS_{\text{indicator}}(\theta) = \text{True}\}$



- 1 Заполняется матрица для подсчёта знаково-возмущённых сумм

$$\Delta = \begin{pmatrix} \varepsilon_1(\theta) & \pm\varepsilon_1(\theta) & \cdots & \pm\varepsilon_1(\theta) \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon_T(\theta) & \pm\varepsilon_T(\theta) & \cdots & \pm\varepsilon_T(\theta) \end{pmatrix}$$

- 2 Генерируются знако-возмущённые суммы

$$H_0(x) = \sum_{j=1}^T (y_j - kt_j - b) = \sum_{j=1}^T \varepsilon_j(\theta)$$

$$H_i(x) = \sum_{j=1}^T \beta_{i,j} (y_j - kt_j - b) = \sum_{j=1}^T \beta_{i,j} \varepsilon_j(\theta)$$

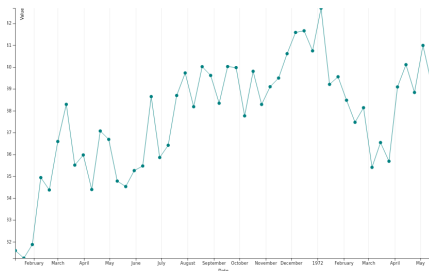
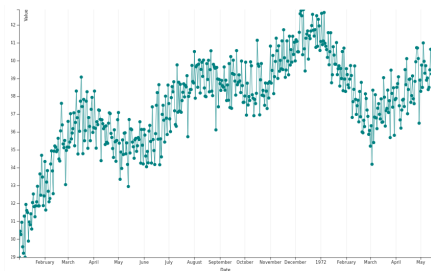
для  $i = 1 \dots M - 1$ .

- 3 Позиция  $H_0^2(x)$  в упорядоченном наборе сумм называется рангом  $r$ . В случае если ранг  $r \geq q$ , то прямая с данными коэффициентами лежит в доверительном множестве

## Экспериментальные данные

500 зашумленных измерений с 6 изменениями тренда

Прореженный набор с 50 измерениями и 4 изменениями тренда



Тестирование на наборах данных с окнами 50 и 10 измерений  
соответственно



## Метрики оценки определения тренда и расчётов доверительных интервалов

- Корень из среднеквадратичного отклонения (RMSE)
- Доля фактических значений в предсказанном доверительном интервале (CR)
- Средняя ширина доверительных интервалов (CW)

Ряд	Метрика	Holt	Регрессия	Kalman	ARIMA	SPS
1	RMSE	1.2310	0.8053	0.7088	0.7310	1.0651
	CR	0.8105	0.9103	0.9124	0.9674	0.9888
	CW	3.3396	2.6357	4.1720	3.2580	5.3217
2	RMSE	1.7676	1.2240	0.9139	1.4312	1.6950
	CR	0.9024	0.8536	0.9512	0.9756	1.0000
	CW	6.4080	3.3932	8.4067	5.8717	13.313

**Таблица:** Результаты расчёта метрик оценки определения тренда и расчётов доверительных интервалов для различных алгоритмов

## Оценка моментов изменения тренда

- Среднее время задержки (MDT)
- Точность (Precision) и полнота (Recall)

Ряд	Метрика	Holt	Регрессия	Kalman	ARIMA	SPS
1	MDT	15.0892	15.1734	-22.0143	22.3456	-5.7896
	Precision	0.8	0.8	0.75	1.0	1.0
	Recall	0.5714	0.7142	0.4285	0.7142	0.8571
2	MDT	-0.25	1.25	-1.25	1.25	-2.75
	Precision	0.50	0.75	0.75	0.75	1.00
	Recall	0.50	0.75	0.75	0.75	1.00

**Таблица:** Результаты расчёта метрик оценки моментов изменения трендов для различных алгоритмов

# Результаты

- исследованы алгоритмы: метод Хольта, линейная регрессия, ARIMA, фильтр Калмана, SPS
- определены требования к прототипу и настройке алгоритмов
- проведён обзор существующих решений: Orange, KNIME, forecast, Prophet
- разработана архитектура системы: определены основные компоненты, выбраны технологии
- реализован прототип системы — веб-приложение
- проведены эксперименты

Тезисы работы были представлены на конференции «**Современные технологии в теории и практике программирования**» и опубликованы в сборнике её материалов

**Исходный код** доступен по ссылке<sup>1</sup>

---

<sup>1</sup><https://github.com/Liana2707/TimeSeriesForecasting> — репозиторий проекта на Github

# Алгоритмы: метод Хольта, линейная регрессия и ARIMA

## Метод Хольта

Уравнение прогноза  $\hat{x}_{t+1|t} = l_t + b_t$

Уравнение уровня  $l_t = \alpha x_t + (1 - \alpha)(l_{t-1} + b_{t-1})$

Уравнение тренда  $b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}$

## ARIMA

$$(\Delta^d \hat{x}_{t|t-\min(p,q)}) = \mu + \varepsilon_t + \sum_{i=1}^q \theta_i (\Delta^d \varepsilon_{t-i}) + \sum_{i=1}^p \phi_i (\Delta^d x_{t-i})$$

Здесь  $\Delta^d$  — оператор разности порядка  $d$

$\mu$  — среднее значение ряда

$\theta_1, \dots, \theta_q$  и  $\phi_1, \dots, \phi_p$  — параметры модели

## Линейная регрессия

$$x_t = kt + b + \varepsilon_t,$$

где  $k, b$  — коэффициенты, показывающие, как зависит  $x_t$  от  $t$ .

# Алгоритмы: Фильтр Калмана

Модель движения  $x_t = F_t x_{t-1} + w_t$ ,

где  $F_t$  — матрица состояния,  $w_t$  — возмущение

Модель наблюдений  $z_t = H_t x_t + \varepsilon_t$ ,

$H_t$  — матрица наблюдений

Уравнения предсказания

$$\hat{x}_{t|t-1} = F_t \hat{x}_{t-1|t-1}$$

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t$$

где  $P_{t|t-1}$  — ковариация ошибки предсказания,  $Q_t$  — ковариация возмущения модели движения

Корректировка значений

$$K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t (z_t - H_t \hat{x}_{t|t-1})$$

$$P_{t|t} = (I - K_t H_t) P_{t|t-1},$$

где  $K_t$  — коэффициент Калмана,  $R_t$  — ковариация ошибки измерений