

Санкт-Петербургский государственный университет  
Прикладная математика и информатика

Учебная практика 2 (проектно-технологическая)

ИСПОЛЬЗОВАНИЕ МЕТОДА SSA В МАШИННОМ ОБУЧЕНИИ ДЛЯ  
ПРОГНОЗА ВРЕМЕННЫХ РЯДОВ

Выполнил:

Ежов Федор Валерьевич

группа 20.М03-мм

Научный руководитель:

к.физ.-мат.н., доцент

Голяндина Нина Эдуардовна

Кафедра Статистического Моделирования

Санкт-Петербург

2021

# Оглавление

<b>Введение</b>	4
<b>Глава 1. Алгоритм Singular Spectrum Analysis</b>	6
1.1. Этап 1. Построение траекторной матрицы (Вложение)	6
1.2. Этап 2. Singular Value Decomposition (SVD)	7
1.3. Этап 3. Группировка первых $r$ собственных троек	7
1.4. Этап 4. Диагональное усреднение	7
<b>Глава 2. Использование SSA в машинном обучении</b>	9
2.1. Задача	9
2.2. Подготовка данных	9
2.3. Модель	11
2.4. Прогнозирование	12
2.5. Метрики	12
<b>Глава 3. Модели</b>	13
3.1. Artificial Neural Network (ANN)	13
3.2. Recurrent neural network (RNN)	14
3.3. Long short-term memory(LSTM)	14
3.4. Gated Recurrent Unit (GRU)	15
<b>Глава 4. Синтетические данные</b>	16
4.1. Синус с шумом	16
4.1.1. Сравнение модели ANN и гибридной модели SSA-ANN	16
4.1.2. Подбор гипер-параметров моделей	19
4.1.3. Сравнение гипер-параметров SSA в гибридных моделях	25
4.2. Два синуса с близкими периодами	26
4.2.1. Подбор гипер-параметров моделей	27
4.2.2. Сравнение гипер-параметров SSA в гибридных моделях	32
<b>Глава 5. Реальные данные</b>	35
5.1. Среднемесячные осадки в Индии	35

5.1.1.	Различное использование SSA предобработки в гибридной модели SSA-ANN . . . . .	35
5.1.2.	Количество точек для предсказания . . . . .	37
5.1.3.	Вклад SSA в улучшение прогноза реального ряда . . . . .	40
<b>Заключение . . . . .</b>		<b>42</b>
<b>Список литературы . . . . .</b>		<b>43</b>

## Введение

В предыдущем семестре было показано, что вопреки утверждению из статьи [2], метод SSA в качестве препроцессинга улучшает точность прогноза методом ANN (полносвязная нейронная сеть с одним скрытым слоем). При этом использовались как данные из статьи, так и похожий по структуре модельный пример. В качестве модельного ряда для анализа был использован зашумленный синусоидальный сигнал (исходный ряд осадков в Индии был похож на зашумленную сумму гармонических рядов). Анализ реальных данных и анализ модельного ряда приводил к схожим выводам, в связи с чем возникла идея при анализе реальных данных сначала строить их модель и потом исследовать методы на модели. Модель удобна тем, что в ней можно менять параметры, можно проверять устойчивости результатов, и пр.

В этом семестре была поставлена задача:

1. Рассмотреть большее количество видов NN
2. Рассмотреть новые реальные данные и построить для них модель
3. Выработать методику выбора метода ML, его параметров и сравнения методов между собой, включая гибридные варианты с SSA препроцессингом.

По пункту 1 к ANN были добавлены рекуррентные методы: RNN, GRU и LSTM. По 2 пункту были рассмотрены новые данные для параметров вращения Земли (EOP). Они оказались гораздо сложнее для анализа. Так как после вычитания тренда данные похожи на модулированную сумму гармоник с близкими частотами, было решено начать с исследования модельного примера такого типа.

Таким образом, образовалось два модельных примера, зашумленный чистый синус с периодом 12 (данные сезонные) и зашумленная модулированная сумма гармоник с близкими периодами 12 и 14.

При сравнении методов машинного обучения между собой возникает проблема неизвестной точности оценок величин ошибок. Так как в рассматриваемых методах ML везде есть полносвязный скрытый слой, то оказался очень удачным график зависимости ошибок на валидационных данных от числа нейронов в этом слое. Во-первых, по нему можно определить, какое число нейронов лучше брать. Но, главное, он отражает точность оценок, так как содержит случайные колебания.

Общая схема исследования на модельных примерах получилась такая:

- По графикам зависимости ошибок от числа нейронов выбор одновременно разумно-оптимального числа нейронов и размера входных данных.
- Для выбранного числа нейронов и размера входных данных подбирать параметры метода SSA, используемого для препроцессинга. При этом варианты числа компонент для оценки сигнала методом SSA выбираются либо на основе теории метода SSA, либо на визуальном анализе результатов SSA разложения.

Структура работы следующая. В главе 1 описан алгоритм препроцессинга Singular Spectrum Analysis (SSA). В главе 2 описана основные методы и схемы, применяемые в экспериментах в области ML, описанных в главах 4 и 5. В главе 3 подробно описаны все методы прогнозирования, используемые в работе. В главе 4 описаны эксперименты на синтетических данных. В главе 5 описаны эксперименты на реальных данных.

## Глава 1

## Алгоритм Singular Spectrum Analysis

Метод SSA используется для разложение исходного ряда в сумму рядов, которые легко интерпретировать и понять их поведение. Обычно исходный ряд раскладывается в сумму трех рядов: тренд — медленно меняющаяся компонента, сезонность — циклическая компонента с фиксированным периодом и шум. Информацию про базовый алгоритм SSA и связанные с методом фундаментальные понятия можно найти в книге «Analysis of time series structure: SSA and related techniques» [1].

Алгоритм SSA состоит из четырех этапов:

1. Построение траекторной матрицы (Вложение).
2. SVD.
3. Группировка первых  $r$  собственных троек.
4. Диагональное усреднение.

Рассмотрим каждый этап подробнее.

Пусть  $X_N = (x_1, \dots, x_N)$  — временной ряд, где  $N > 3$ . Также будем предполагать, что найдется хоть одно  $x_i \neq 0$ , то есть ряд не нулевой. Обычно считается, что  $x_i = f(i\Delta)$  для некоторой функции  $f(t)$ , где  $t$  — время, а  $\Delta$  — некоторый временной интервал.

### 1.1. Этап 1. Построение траекторной матрицы (Вложение)

Выберем целое  $L$  — длина окна, такое что  $1 < L < N$ . Тогда  $K = N - L - 1$ . Построим вектора  $X_i = (x_i, \dots, x_{i+L-1})^T$ , для  $1 \leq i \leq K$ . Составим из векторов  $X_i$  траекторную матрицу:

$$\mathbf{X} = [X_1 : \dots : X_K] = \begin{pmatrix} x_1 & x_2 & x_3 & \cdots & x_K \\ x_2 & x_3 & x_4 & \cdots & x_{K+1} \\ x_3 & x_4 & x_5 & \cdots & x_{K+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & x_{L+2} & \cdots & x_N \end{pmatrix}.$$

Получили матрицу  $\mathbf{X}$  размерностью  $L \times K$ , составленную из пересекающихся частей исходного временного ряда. Можно заметить, что на побочных диагоналях стоят одинаковые числа, такая матрица называется ганкелевой. Существует взаимно-однозначное соответствие между ганкелевыми матрицами  $L \times K$  и рядами длиной  $N = L + K - 1$ .

Операцию получения из ряда  $\mathbf{X}_N$  траекторную матрицу  $\mathbf{X}$  обозначим:

$$\mathbf{X} = \mathcal{T}_L(\mathbf{X}_N),$$

соответственно обратная операция будет обозначаться:  $\mathcal{T}^{-1}$ .

## 1.2. Этап 2. Singular Value Decomposition (SVD)

На данном этапе применяется метод SVD к траекторной матрице  $\mathbf{X}$ . Пусть  $\mathbf{S} = \mathbf{X}\mathbf{X}^T$  и  $\lambda_1 > \dots > \lambda_L$  — собственные числа матрицы  $\mathbf{S}$ ,  $U_1, \dots, U_L$  — ортонормированная система базисных векторов, соответствующих собственным числам. Обозначим  $V_i = \frac{X^T U_i}{\sqrt{\lambda_i}}$  и  $d = \max\{i : \lambda_i > 0\}$ . Тогда сингулярное разложение матрицы  $\mathbf{X}$  запишется следующим образом:

$$\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d, \text{ где } \mathbf{X}_i = \sqrt{\lambda_i} U V_i^T,$$

Набор  $(\sqrt{\lambda_i}, U_i, V_i^T)$  будем называть  $i$ -й собственной тройкой.

## 1.3. Этап 3. Группировка первых $r$ собственных троек

На этапе группировки из всех значений  $\{1 \dots d\}$  берутся первые  $r$ . Пусть,  $I = \{1, \dots, r\}$ , тогда результирующая матрица соответствующая группе  $I$  имеет вид:  $\mathbf{X}_I = \mathbf{X}_1 + \dots + \mathbf{X}_r$ .

## 1.4. Этап 4. Диагональное усреднение

Пусть  $\mathbf{Y}$  — матрица  $L \times K$ ,  $L < K$ .  $y_{ij}$  — элементы матрицы, где  $1 \leq i \leq L, 1 \leq j \leq K$ . Также пусть  $N = L + K - 1$ . Диагональное усреднение преобразует матрицу  $\mathbf{Y}$  в ряд  $g_0, \dots, g_{N-1}$  по формуле:

$$g_k = \begin{cases} \frac{1}{k+1} \sum_{m=1}^{k+1} y_{m,k-m+2} & , \text{ для } 0 \leq k < L-1 \\ \frac{1}{L} \sum_{m=1}^L y_{m,k-m+2} & , \text{ для } L-1 \leq k < K \\ \frac{1}{N-k} \sum_{m=k-K+2}^{N-K+1} y_{m,k-m+2} & , \text{ для } K \leq k < N \end{cases}$$

Применяя диагональное усреднение к результирующей матрице группы  $I$ , получаем ряд  $\hat{F} = (f_1 \cdots f_{N-1})$ . Полученный ряд  $\hat{F}$  назовем оценкой сигнала, полученной с помощью SSA. Процедуру выделения сигнала с помощью SSA обозначим как:

$$\hat{F} = SSA_{L,r}(F),$$

где  $L$  — длина окна в SSA,  $r$  — количество первых собственных троек, участвующие в построении  $\hat{F}$ .



## Глава 2

# Использование SSA в машинном обучении

### 2.1. Задача

Рассмотрим  $\mathbf{Z}_N$  — временной ряд длины  $N$  и задачу: с помощью модели некоторой нейронной сети на основе  $T$  последовательных точек ряда  $\mathbf{Z}_N$ , предсказать следующие  $R$  точек ряда. Решение данной задачи, можно разбить на несколько частей: подготовка данных, обучение модели, прогнозирование.

### 2.2. Подготовка данных

$\mathbf{Z}_N$  — изначальный временной ряд длиной  $N$ . Мы можем представить ряд в виде траекторной матрицы для длины окна  $T + R$ , тогда получим:

$$\mathbf{Z} = \mathcal{T}_{T+R}(\mathbf{Z}_N) = \left( \begin{array}{cccc|cccc} z_1 & z_2 & \cdots & z_T & z_{T+1} & \cdots & z_{T+R-1} & z_{T+R} \\ z_2 & z_3 & \cdots & z_{T+1} & z_{T+2} & \cdots & z_{T+R} & z_{T+R+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ z_{N-T-R+1} & z_{N-T-R+2} & \cdots & z_{N-R} & z_{N-R+1} & \cdots & z_{N-1} & z_N \end{array} \right).$$

Матрица  $\mathbf{Z}$  будет размерности  $(N - T - R + 1) \times (T + R)$ . Левую часть матрицы  $\mathbf{Z}$  назовем  $\mathbf{Z}^x$ , а правую часть  $\mathbf{Z}^y$ . Также разобьем матрицу по строчкам на 3 части: train, val, test. Пусть  $\tau$ ,  $v$  и  $t$  номера последних строчек в каждой соответствующей части. Обозначим с помощью  $\mathbf{Z}_{a,b}^{(c,d)}$  часть матрицы  $\mathbf{Z}$  с  $a$  по  $b$  строчку и с  $c$  по  $d$  столбец. Тогда train, val, test части записываются как:  $\mathbf{Z}_{train} = \mathbf{Z}_{1,\tau}^{(1,T+R)}$ ,  $\mathbf{Z}_{val} = \mathbf{Z}_{\tau+T+R,v}^{(1,T+R)}$ ,  $\mathbf{Z}_{test} = \mathbf{Z}_{v+T+R,t}^{(1,T+R)}$ . В этих же обозначениях  $\mathbf{Z}^x = \mathbf{Z}_{1,t}^{(1,T)}$ ,  $\mathbf{Z}^y = \mathbf{Z}_{1,t}^{(T+1,T+R)}$ .

### SSA-preprocessing

Предобработка SSA для тренировочной выборки описывается следующим алгоритмом.  $L, r$  — гипер-параметры, описанные в разделе «1.1. Алгоритм SSA»:

1. Преобразуем train часть матрицы  $\mathbf{Z}$  во временной ряд  $\tilde{\mathbf{Z}} = \mathcal{T}^{-1}(\mathbf{Z}_{train})$ .
2. Получим ряд  $\hat{\mathbf{Z}} = SSA_{L,r}(\tilde{\mathbf{Z}})$ .

3. Получаем траекторную матрицу  $\hat{\mathbf{Z}} = \mathcal{T}_{T+R}(\hat{\mathbf{Z}})$ .
4. Полученная траекторная матрица  $\hat{\mathbf{Z}}$  будет результатом работы предобработки SSA для тренировочной выборки.

Предобработка SSA для валидационной или тестовой выборки отличается от предыдущей, ввиду разных предназначений выборок. В отличие от тренировочной выборки о которой мы знаем все, считается, что о валидационной и тестовой выборках ничего не известно. В этих случаях SSA-обработку следует применять так, чтобы предыдущие значения ряда не получали информации от будущих («заглядывание в будущее»).

Пусть  $\mathbf{Z}_{b,e} = [z_b, z_{b+1}, \dots, z_e]$  подряд ряда  $\mathbf{Z}$ , где  $b$  — начальный индекс,  $e$  — конечный индекс. Пусть  $p$  — тоже индекс ряда, такой что  $b < p < e$ . Следующий алгоритм описывает процедуру получения ряда  $\mathbf{Z}_{p+1,e}$ , обработанного с помощью SSA без «заглядывание в будущее»:

1. Пусть есть ряд  $\mathbf{Z}_{b,e}$  и задано  $p$ . Тогда  $Q = e - p$  — размер ряда  $\mathbf{Z}_{p+1,e}$ . Пусть  $\hat{\mathbf{Z}}_Q = (\hat{z}_1, \dots, \hat{z}_Q)$  — ряд размера  $Q$ .
2. Для каждого  $i = [1, \dots, Q]$  получим  $\hat{z}'_{b+i-1,p+i} = \text{SSA}_{L,r}(\mathbf{Z}_{b+i-1,p+i})$ , присвоим значение последнего элемента полученного ряда  $\hat{z}'_{p+i}$  значению ряда  $\hat{\mathbf{Z}}_Q$  с соответствующим индексом, то есть  $\hat{z}_i = \hat{z}'_{p+i}$ .
3. Получили ряд  $\hat{\mathbf{Z}}_Q$  размера  $Q$ , значения которого являются значениям ряда  $\mathbf{Z}_{p+1,e}$ , обработанные с помощью SSA без «заглядывания в будущее».

Процедуру получения  $\hat{\mathbf{Z}}_Q$  обозначим:  $\hat{\mathbf{Z}}_Q = \text{SSA}^{(p)}(\mathbf{Z}_{b,e})$ . Тогда алгоритм предобработки для валидационной выборки запишется следующим образом:

1. Запишем  $\mathbf{Z}_{1,v}^{(1,T+R)}$  как  $\mathbf{Z}_{1,v+T+R}$ .
2. Выберем  $p = \tau + T + R$ .
3. Получим  $\hat{\mathbf{Z}}_Q = \text{SSA}^{(p)}(\mathbf{Z}_{1,v+T+R})$ .
4. Перейдем обратно к траекторной матрице  $\hat{\mathbf{Z}}_{val} = \mathcal{T}_{T+R}(\hat{\mathbf{Z}}_Q)$ .

$\hat{\mathbf{Z}}_{val}$  — будет результатом предобработки SSA для валидационной выборки. Размерность  $\hat{\mathbf{Z}}_{val}$  будет совпадать с размерностью  $\mathbf{Z}_{val}$ .

Запишем аналогичный алгоритм для тестовой выборки:

1. Запишем  $\mathbf{Z}_{1,t}^{(1,T+R)}$  как  $\mathbf{Z}_{1,t+T+R} = \mathbf{Z}_N$ .
2. Выберем  $p = v + T + R$ .
3. Получим  $\hat{\mathbf{Z}}_Q = \mathcal{SSA}^{(p)}(\mathbf{Z}_N)$ .
4. Перейдем обратно к траекторной матрице  $\hat{\mathbf{Z}}_{test} = \mathcal{T}_{T+R}(\hat{\mathbf{Z}}_Q)$ .

$\hat{\mathbf{Z}}_{test}$  — будет результатом предобработки SSA для тестовой выборки.

## 2.3. Модель

**Обучение** Оптимизация параметров модели проводится с помощью процедуры обратного распространения ошибки на тренировочной выборке. Модель учится по  $\mathbf{Z}_{train}^x$  предсказывать  $\mathbf{Z}_{train}^y$ . Эпоха — цикл прохода всех строчек из тренировочной выборки в обучении. Количество эпох для обучения является гипер-параметром. На валидационной выборке оценивается оптимальное количество эпох, нужных для модели (дабы избежать переобучения).

Перед началом обучения нужно выбрать гипер-параметры модели и количество эпох.

Алгоритм обучения модели после выбора архитектуры:

1. Инициализация модели со случайными весами.
2. На тренировочной выборке  $\mathbf{Z}_{train}$  оптимизируются веса  $w, \theta$  с заданным количеством эпох. Модель учится по данным строчкам  $\mathbf{Z}_{train}^x$  предсказывать соответствующие строчки  $\mathbf{Z}_{train}^y$ . Для каждой  $i$ -ой эпохи считается  $\epsilon_i$  — ошибка на валидационной выборке. Для валидационной выборки  $\mathbf{Z}_{val}^x$  строится прогноз  $\hat{\mathbf{Z}}_{val}^y$ . Ошибка  $\epsilon_i$  получается сравнением  $\hat{\mathbf{Z}}_{val}^y$  с  $\mathbf{Z}_{val}^y$  по какой-нибудь метрике (например MSE).
3. Находим  $i : \min(\epsilon_i)$ .
4. Веса модели сбрасываются (снова инициализация случайными весами).
5. Снова на тренировочной выборке оптимизируются веса  $w, \theta$  с количеством эпох равных  $i$ .

## 2.4. Прогнозирование

После того как модель обучена, можно перейти к прогнозированию точек ряда.

1. Возьмем  $\mathbf{Z}_{test}^x$  и  $\mathbf{Z}_{test}^y$ .
2. Представим  $\mathbf{Z}_{test}^x = [Z_{test}^{x,1} : \dots : Z_{test}^{x,Q}]^T$ , где  $Q$  — количество строчек в тестовой матрицы  $\mathbf{Z}_{test}$ .
3. Для каждой строчки матрицы  $\mathbf{Z}_{test}^x$  получаем прогноз с помощью обученной модели. Запишем результат прогноза как матрицу  $\hat{\mathbf{Z}}^y = [\hat{Z}^{y,1} : \dots : \hat{Z}^{y,Q}]^T$ .
4. Далее можно сравнить  $\hat{\mathbf{Z}}^y$  с  $\mathbf{Z}_{test}^y$  по какой-нибудь метрике.

## 2.5. Метрики

С помощью метрик MSE и RMSE можно мерить размер ошибки полученного прогноза.

$$MSE(\mathbf{Z}_{test}^y, \hat{\mathbf{Z}}^y) = \frac{1}{Q} \text{diag}((\mathbf{Z}_{test}^y - \hat{\mathbf{Z}}^y)(\mathbf{Z}_{test}^y - \hat{\mathbf{Z}}^y)^T)$$

$$RMSE(\mathbf{Z}_{test}^y, \hat{\mathbf{Z}}^y) = \sqrt{MSE(\mathbf{Z}_{test}^y, \hat{\mathbf{Z}}^y)}$$

## Глава 3

### Модели

В данной главе рассматриваются нейронные сети используемые в работе. Всего рассмотрено четыре архитектуры. Одна линейная, в дальнейшем называется ANN, и три рекуррентных нейронных сети: RNN, LSTM, GRU.

#### 3.1. Artificial Neural Network (ANN)

Artificial Neural Network (ANN) включает в себя входной слой, ряд скрытых слоев и выходной слой, каждый слой содержит несколько узлов. Считается, что ANN с одним скрытым слоем обеспечивает достаточную сложность для моделирования нелинейных взаимосвязей данных. ANN в этой работе формализуется следующим образом:

$$\hat{X}^T = [\hat{x}_1, \dots, \hat{x}_T]$$

– входные данные, на которых модели учиться делать предсказания.

$$\hat{Y}^T = [\hat{y}_1, \dots, \hat{y}_R]$$

– выходные данные, предсказания модели.

$$y_k = \phi_2 \left( \sum_{j=1}^h w_{jk}^{(2)} \phi_1 \left( \sum_{i=1}^T w_{ij}^{(1)} x_i + \theta_j^{(1)} \right) + \theta_k^{(2)} \right), k = [1, \dots, R],$$

где  $T$  — размер входного вектора на котором выполняется прогноз,  $h$  — размер скрытого слоя.  $w$  и  $\theta$  — веса модели.  $\phi$  — функция активации.  $R$  — размер выходного вектора-прогноза.

Ниже представлен список некоторых функций активаций:

1. Линейная функция активации:  $\phi(x) = x$ .

2. Сигмоида:  $\phi(x) = \frac{1}{1 + e^{-x}}$ .

3. ReLU:  $\phi(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$

### 3.2. Recurrent neural network (RNN)

Модель recurrent neural network (RNN), как следует из названия, используют рекурсию в своих архитектурах для решения задач, где данные содержатся в некоторой последовательности (например, временные ряды, текстовые задачи и др.). В RNN присутствует вектор скрытого слоя, служащий для «накопления» информации.

$\hat{X}^T = [\hat{x}_1, \dots, \hat{x}_T]$  – вектор входных данных.

$H_t^T = [h_1, \dots, h_J]$  – вектор скрытого слоя в момент  $t$ .

$\hat{Y}^T = [\hat{y}_1, \dots, \hat{y}_R]$  – вектор выходных данных в момент  $T$ .

Следующие формулы описывают модель:

$$H_t = f_h(Ux_t + \theta_1 + WH_{t-1} + \theta_2),$$

$$Y_t = f_y(VH_t + \theta_3),$$

где  $t = [1, \dots, T]$ ,  $U, V, W, \theta_i$  – веса модели, вычисляющиеся в процессе обучения,  $f_h, f_y$  – функции активации (в большинстве случаев сигмоиды).

### 3.3. Long short-term memory(LSTM)

Long short-term memory (LSTM) — разновидность рекуррентных моделей, с добавлением второго скрытого слоя, используемого для «долгосрочной» памяти.

$\hat{X}^T = [\hat{x}_1, \dots, \hat{x}_T]$  – вектор входных данных.

$H_t^T = [h_1, \dots, h_J]$  – вектор скрытого слоя в момент  $t$ .

$C_t^T = [c_1, \dots, c_J]$  – вектор скрытого слоя в момент  $t$ .

$\hat{Y}^T = [\hat{y}_1, \dots, \hat{y}_R]$  – вектор выходных данных в момент  $T$ .

Следующие формулы описывают модель:

$$f_t = \sigma(W_f \cdot [H_{t-1}, x_t] + b_f), i_t = \sigma(W_i \cdot [H_{t-1}, x_t] + b_i),$$

$$\tilde{C}_t = \tanh(W_c \cdot [H_{t-1}, x_t] + b_c), C_t = f_t * C_{t-1} + i_t * \tilde{C}_t,$$

$$o_t = \sigma(W_o \cdot [H_{t-1}, x_t] + b_o), H_t = o_t * \tanh(C_t),$$

$$Y_t = f(VH_t + b_0),$$

где  $t = [1, \dots, T]$ ,  $W, b$  – веса модели, вычисляющиеся в процессе обучения,  $f$  – функция активации. Оператор  $*$  – производит поэлементное умножение.

### 3.4. Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) — модель похожая на LSTM, но без дополнительного скрытого слоя.

$\hat{X}^T = [\hat{x}_1, \dots, \hat{x}_T]$  — вектор входных данных.

$H_t^T = [h_1, \dots, h_J]$  — вектор скрытого слоя в момент  $t$ .

$\hat{Y}^T = [\hat{y}_1, \dots, \hat{y}_R]$  — вектор выходных данных в момент  $T$ .

Следующие формулы описывают модель:

$$\begin{aligned} z_t &= \sigma(W_z \cdot [H_{t-1}, x_t] + b_z), \quad r_t = \sigma(W_r \cdot [H_{t-1}, x_t] + b_r), \\ \tilde{H}_t &= \tanh(W \cdot [r_t * H_{t-1}, x_t] + b), \quad H_t = (1 - z_t) * H_{t-1} + z_t * \tilde{H}_t, \\ Y_t &= f(VH_t + b_0), \end{aligned}$$

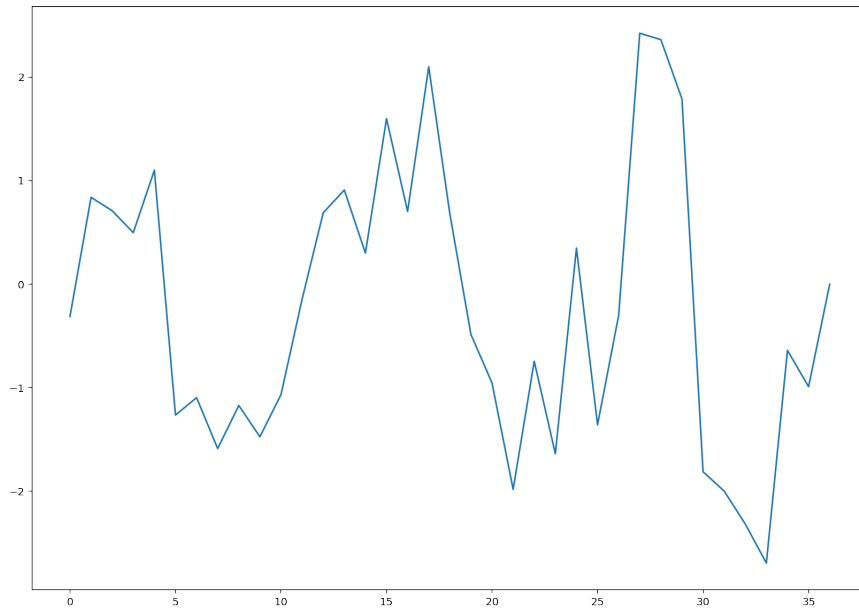
где  $t = [1, \dots, T]$ ,  $W, b$  — веса модели, вычисляющиеся в процессе обучения,  $f$  — функция активации. Оператор  $*$  — производит поэлементное умножение.

## Глава 4

## Синтетические данные

## 4.1. Синус с шумом

Рассмотрим следующий ряд  $Z_{1500}$ , состоящий из элементов  $z_i = \sin(2\pi\omega i) + \varepsilon_i$ , где  $\omega$  – частота, равная  $\frac{1}{12}$ ,  $\varepsilon_i$  – шум из стандартного нормального распределения  $N(0, 1)$ .

Рис. 4.1. Часть ряда  $Z_{1500}$ .

## 4.1.1. Сравнение модели ANN и гибридной модели SSA-ANN

Поставим задачу предсказывать следующую точку синуса по двенадцати предыдущим, то есть возьмем  $T = 12$  и  $R = 1$ . В рамках этой задачи хотим сравнить обычную модель ANN и гибридные модели SSA-ANN по метрике RMSE. Также мы будем сравнивать отклонение полученного прогноза от сигнала  $\sin(2\pi\omega i)$  и самого ряда  $Z_{1500}$ .

Мы будем рассматривать две версии гибридных моделей SSA-ANN. В первой версии обработка методом SSA будет применяться для левой части  $\mathbf{Z}^x$  траекторной матрицы. Таким образом мы хотим проверить гипотезу о том, что точность ANN повысится, если прогноз будет строиться на данных, очищенных от шума. Такие модели будем называть SSA-X-ANN. Во второй версии только правая часть  $\mathbf{Z}^y$  обрабатывается с помощью метода SSA. В этом случае мы хотим проверить гипотезу, что точность модели



ANN повысится, если модель будет обучаться прогнозировать данные без шума. Такие модели будем называть SSA-Y-ANN.

## Модели

Для предсказаний воспользуемся нейронной сетью ANN, описанной в разделе «3 Модели». Назовем ANN-1 – модель без скрытых слоев и обе функций активации являются линейными. Она соответствует обычной линейной регрессии. ANN-2 – назовем модель с одним скрытым слоем, размера 50.  $\phi_1$  – является функцией активации ReLU, а  $\phi_2$  – линейная функция активации. SSA-ANN-1 и SSA-ANN-2 назовем группы гибридных версии для каждой модели. Стоит отметить, что гибридные модели могут быть разными, мы будем рассматривать по две гибридные модели, которые были описаны в разделе ранее, для каждой обычной (например, SSA-X-ANN-1 или SSA-Y-ANN-2).

## Подготовка данных

Построим траекторную матрицу для ряда  $\mathbf{Z}_{1500}$  с окном длиной 13.

$$\mathbf{Z} = \mathcal{T}_{13}(\mathbf{Z}_{1500}) = \left( \begin{array}{cccc|c} z_1 & z_2 & \cdots & z_{12} & z_{13} \\ z_2 & z_3 & \cdots & z_{13} & z_{14} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ z_{1488} & z_{1489} & \cdots & z_{1499} & z_{1500} \end{array} \right).$$

Разобьем данные на тренировочную, валидационную и тестовую выборку следующим образом:  $\tau = 988$ ,  $v = 1238$  и  $t = 1488$ . Далее для моделей ANN-1 и ANN-2 будем использовать уже полученную матрицу  $\mathbf{Z}$  без каких-то обработок методом SSA. Параметры оператора  $SSA_{L,r}$  задаются следующим образом  $L = \frac{\tau + 12}{2}$ ,  $r = 2$ . Для гибридных моделей SSA-X-ANN-1 и SSA-X-ANN-2 обработаем только левую часть матрицы  $\mathbf{Z}$  методом SSA, как описано в разделе «2.2 SSA-preprocessing». Для гибридных моделей SSA-Y-ANN-1 и SSA-Y-ANN-2 обработаем только правую часть матрицы  $\mathbf{Z}$  методом SSA. Таким образом получили для каждой модели свою матрицу  $\mathbf{Z}$ .

## Результаты

В ходе эксперимента полученные результаты для моделей ANN-1 и SSA-ANN-1 представлены в таблице 4.1. Результаты для моделей ANN-2 и SSA-ANN-2 представле-

ны в таблице 4.2.

Таблица 4.1. RMSE для моделей ANN-1 и SSA-ANN-1.

	ANN-1	SSA-X-ANN-1	SSA-Y-ANN-1
Signal	0.414	0.086	0.409
Time series	1.043	0.998	1.037

Таблица 4.2. RMSE для моделей ANN-2 и SSA-ANN-2.

	ANN-2	SSA-X-ANN-2	SSA-Y-ANN-2
Signal	0.415	0.014	0.365
Time series	1.045	1.007	1.032

На результатах выше можно наблюдать, что гибридные модели показывают результаты лучше, чем обычная модель. Модели SSA-Y-ANN показывают результат сравнимый с обычной моделью, можно сказать, что в этом случае улучшение незначительное. В случае моделей SSA-X-ANN видно хорошее улучшение. Наблюдается снижение ошибки по RMSE на 0.4, если считать отклонение от сигнала. Это говорит о том, что гибридная модель SSA-X-ANN хорошо предсказывает сигнал ряда. Также посмотрим на ошибки для сигнала и ряда в случае SSA-X-ANN. Видно, что квадраты ошибок отличаются, примерно, на дисперсию шума  $\sigma^2$ , что говорит о том, что метод SSA в модели SSA-X-ANN удалил из ряда шумовые компоненты. В случае же моделей ANN таких результатов не наблюдается. Из этого следует, что модели SSA-X-ANN в отличии от ANN пытается прогнозировать чистый сигнал, что дает лучшие результаты.

## Выводы

Можно заключить, что для обеих архитектур моделей наблюдается одна тенденция: гибридные модели показывают результаты лучше, чем обычные. В частности, гибридные модели SSA-X-ANN показывает наилучшие результаты. Можно сказать, что правильное использование гибридных моделей SSA-ANN дает прирост в точности в задаче прогнозирования синуса с шумом.

### 4.1.2. Подбор гипер-параметров моделей

Поставим задачу подобрать гипер-параметры к моделям ANN, RNN, LSTM и GRU. В этот раз данные не будут предобработаны с помощью SSA.  $T = 12, 24$  и  $R = 1$ . Модели будем сравнивать с помощью метрики RMSE. Главной целью этого исследования стоит задача выяснить наилучшую конфигурацию модели для каждой архитектуры. Будем считать ошибку от ряда  $Z_{1500}$ , а также от его сигнала  $\sin(2\pi\omega i)$ .

#### Модели

Воспользуемся моделями, описанными в главе 3. Так как синус можно предсказывать по линейно рекуррентной формуле, то будет рассматривать функции активации линейную и ReLU. Рассмотрим модели поподробнее.

**ANN** Возьмем  $\phi_1$  – ReLU, а  $\phi_2$  – линейной функцией активации. Тогда в качестве настраиваемого гипер-параметра останется размер скрытого слоя. Его и будем подбирать.

**RNN** Возьмем  $f_h$  – ReLU, а  $f_y$  – линейной функцией активации. Будем подбирать размер скрытого слоя  $H$ .

**LSTM**  $f$  – линейная функция активации. Также подбираем размеры скрытых слоев  $H$  и  $C$  (в архитектуре LSTM они имеют одинаковый размер).

**GRU**  $f$  – линейная функция активации. Подбираем размер скрытого слоя  $H$ .

В итоге получаем, что для каждой модели подбираем размер скрытого слоя. Будем перебирать по сетке от 10 до 100 с шагом 5.

#### Подготовка данных

Построим траекторные матрицы для ряда  $Z_{1500}$  с окном длиной 13 и 25.

$$Z^{12} = \mathcal{T}_{13}^{12}(Z_{1500})$$

$$Z^{24} = \mathcal{T}_{25}(Z_{1500})$$

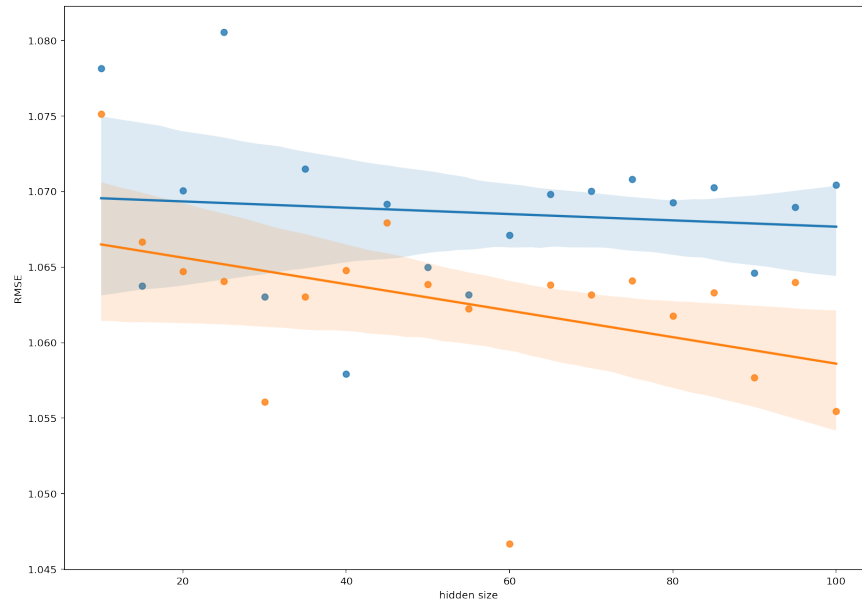
Разобъем данные на тренировочную, валидационную выборку следующим образом:  $\tau = 988$ ,  $v = 1488$  для  $T = 12$ , и  $\tau = 976$ ,  $v = 1476$  для  $T = 24$ . Далее для моделей, описанных выше, будем использовать полученные матрицы  $\mathbf{Z}^{12}$  и  $\mathbf{Z}^{24}$ . Так как мы подбираем гипер-параметры, тестовая выборка не нужна.

## Результаты

Рассмотрим результаты для  $T = 12$  и  $T = 24$  вместе. Результаты представлены в виде графика, где на оси  $x$  отложена сетка по которой перебирался размер скрытого слоя, по оси  $y$  – ошибка RMSE. По точкам на графике нарисована линия регрессии, по ней можно понять общую тенденцию на графике, также в описания каждого графика указано  $p$ -value для линии регрессии. Так как хотелось бы лучше увидеть разницу между результатами, полученными при  $T = 12$  и  $T = 24$ , для каждой модели они отображаются на одном графике. Синим цветом –  $T = 12$ , оранжевым –  $T = 24$ .

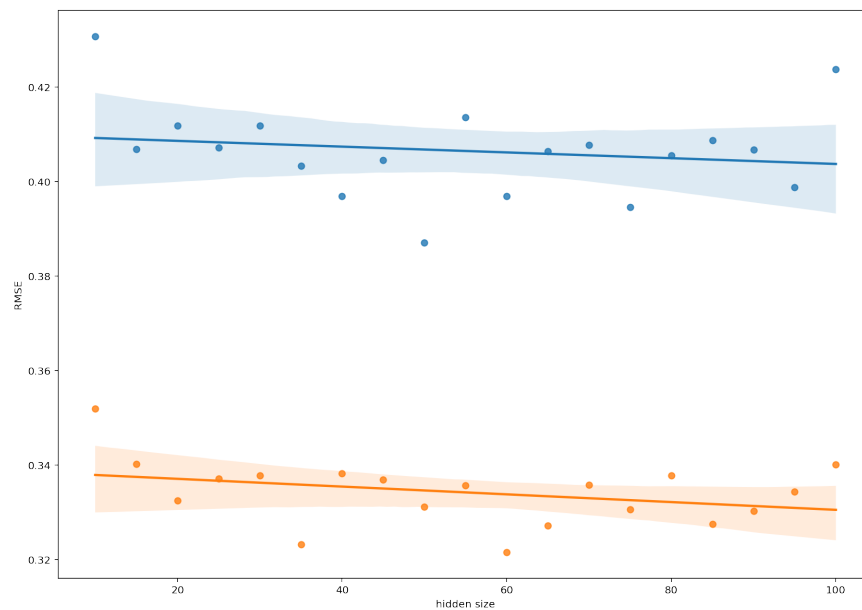
Для каждой линии регрессии будем проверять ее значимость с уровнем значимости 0.1. Если линия не значима, будем выбирать точки ближе к середине оси  $x$ . Если же линия значима, выбираем те точки с краю, на котором линия регрессии убывает. Также не будем брать точки, которые находятся далеко от линии регрессии. Выбирать размер скрытого слоя будем по линиям регрессии построенным на временном ряде.

**ANN** Посмотрим на рисунки 4.2, 4.3. Видно, что ошибки при  $T = 24$  ниже, чем при  $T = 12$  в обоих случаях. Смотрим на  $p$ -value, линии регрессии незначимые, кроме оранжевой на рис. 4.2 при уровне значимости 0.1. В этом случае выбираем размер скрытого слоя поближе к 100, в остальных будем выбирать размер ближе к 50.



синий –  $T = 12$ , оранжевый –  $T = 24$ .

Рис. 4.2. Time Series ANN: p-value для 12 точек: 0.6457, p-value для 24 точек: 0.0665.

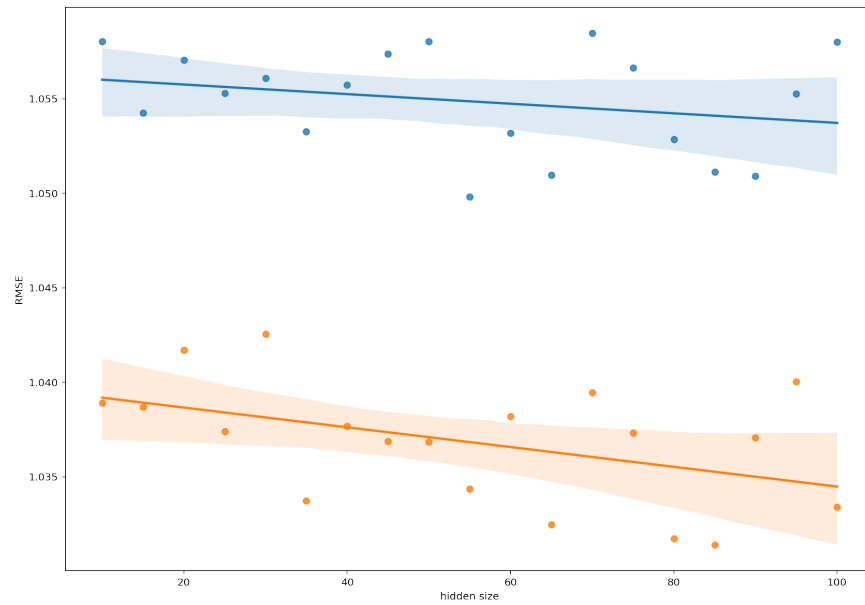


синий –  $T = 12$ , оранжевый –  $T = 24$ .

Рис. 4.3. Signal ANN: p-value для 12 точек: 0.4787, p-value для 24 точек: 0.1632.

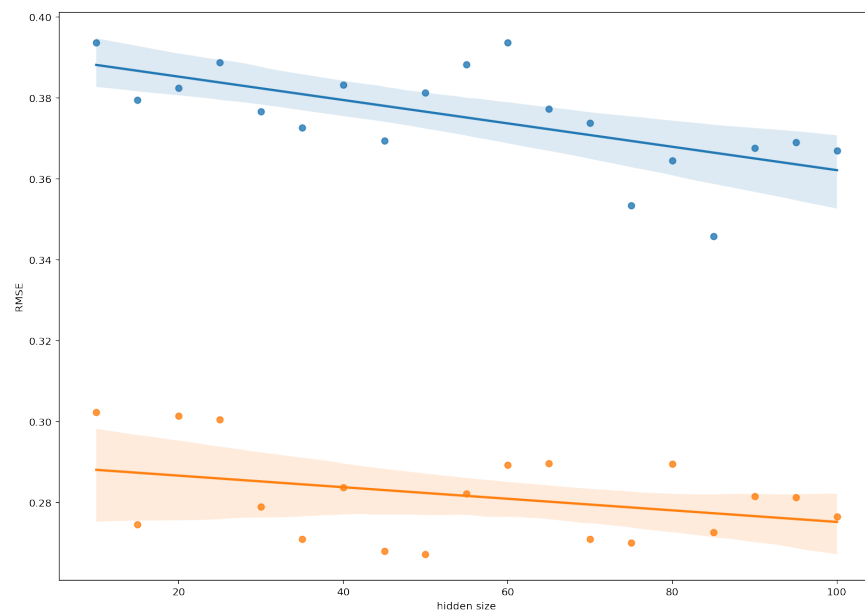
**RNN** Посмотрим на рисунки 4.4, 4.5. Наблюдаем такую же тенденцию как и в случае с ANN, результаты для  $T = 24$  лучше, чем для  $T = 12$ . На рисунке 4.4 линия регрессии для 12 точек незначимая, следовательно размер скрытого слоя выбираем ближе к середине, а для 24 – значимая, выбираем размер скрытого слоя ближе к 100. В случае

сигнала все меняется. На рисунке 4.5 видно, что в случае 12 точек линия – значимая, в случае 24 – незначимая.



синий –  $T = 12$ , оранжевый –  $T = 24$ .

Рис. 4.4. Time Series RNN: p-value для 12 точек: 0.2891, p-value для 24 точек: 0.0493.

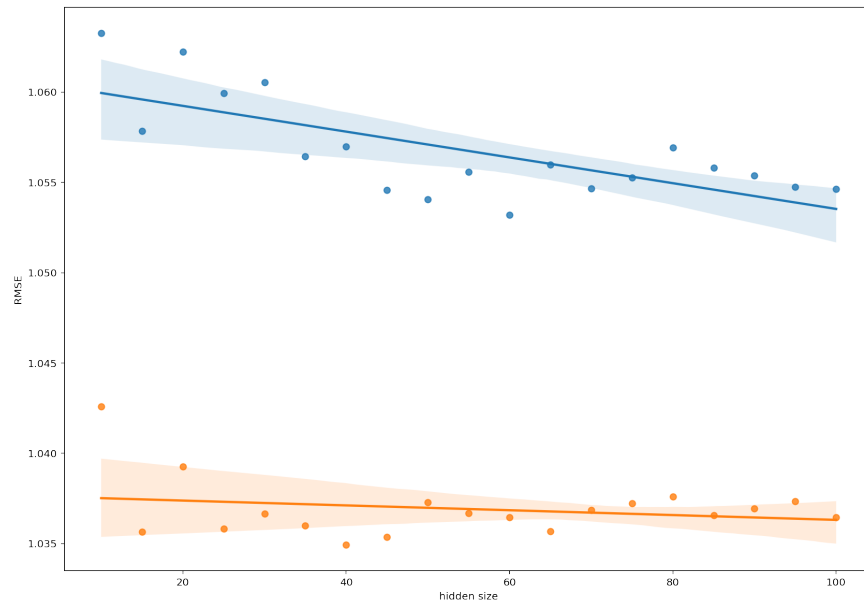


синий –  $T = 12$ , оранжевый –  $T = 24$ .

Рис. 4.5. Signal RNN: p-value для 12 точек: 0.0029, p-value для 24 точек: 0.1331.

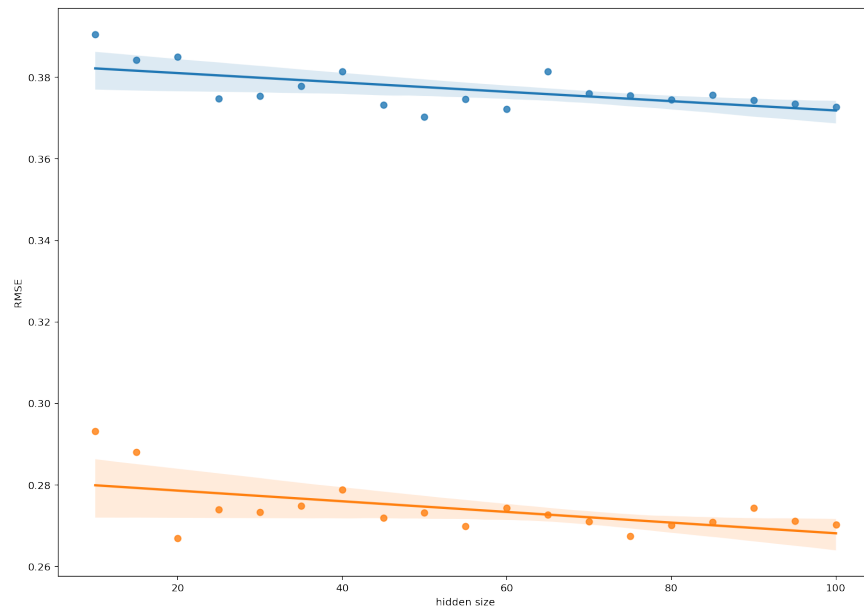
**LSTM** Посмотрим на рисунки 4.6, 4.7. Такая же тенденция, как и в случае с ANN и RNN. Смотрим на p-value, видно что только для случае сравнения с временным ряда

для  $T = 24$  линия регрессии незначимая, в этом случае стоит брать размер скрытого слоя ближе к 50. Для  $T = 12$  – ближе к 100.



синий –  $T = 12$ , оранжевый –  $T = 24$ .

Рис. 4.6. Time Series LSTM: p-value для 12 точек: 0.0006, p-value для 24 точек: 0.3559.

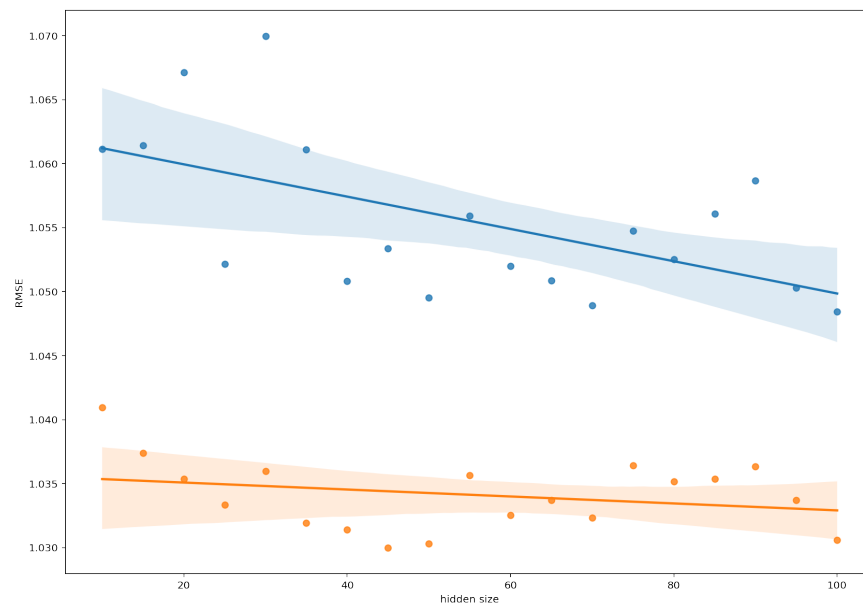


синий –  $T = 12$ , оранжевый –  $T = 24$ .

Рис. 4.7. Signal LSTM: p-value для 12 точек: 0.0043, p-value для 24 точек: 0.0115.

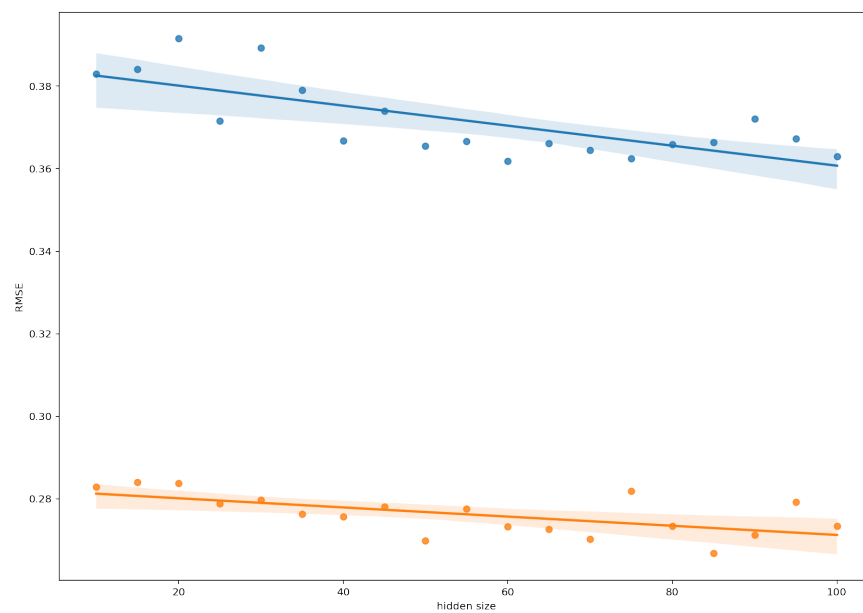
**GRU** Посмотрим на рисунки 4.8, 4.9. Сохраняется тенденция как и в случаях с ANN, RNN, LSTM. Значимость такая же как и в случае с LSTM, размер скрытого слоя вы-

бирается аналогично.



синий –  $T = 12$ , оранжевый –  $T = 24$ .

Рис. 4.8. Time Series GRU: p-value для 12 точек: 0.0102, p-value для 24 точек: 0.2568.



синий –  $T = 12$ , оранжевый –  $T = 24$ .

Рис. 4.9. Signal GRU: p-value для 12 точек: 0.0003, p-value для 24 точек: 0.0045.



## Выводы

После полученных результатов для каждой модели можно зафиксировать размер скрытого слоя и размер  $T$ .  $T = 24$ , ANN - 100, RNN - 65, LSTM - 55, GRU - 100. В дальнейших исследованиях для этого набора данных будем использовать эти модели с такими гипер-параметрами.

### 4.1.3. Сравнение гипер-параметров SSA в гибридных моделях

Поставим задачу сравнить параметр SSA  $r$  в моделях ANN, RNN, LSTM и GRU.  $T = 24$  и  $R = 1$ . Модели и гибридные модели будем сравнивать с помощью метрики RMSE. Будем считать ошибку от ряда  $Z_{1500}$ , а также от его сигнала  $\sin(2\pi\omega i)$ .

По теории для ряда  $Z_{1500}$   $r = 2$ . Также проверим случаи  $r = 1$  и  $r = 4$ .

## Модели

Будем использовать модели полученные в разделе 4.1.2.

## Подготовка данных

Построим траекторную матрицу для ряда  $Z_{1500}$  с окном длиной 25.

$$\mathbf{Z} = \mathcal{T}_{25}(Z_{1500})$$

Разобьем данные на тренировочную, валидационную и тестовую выборку следующим образом:  $\tau = 976$ ,  $v = 1226$  и  $t = 1476$ . Далее для обычных моделей будем использовать уже полученную матрицу  $\mathbf{Z}$  без каких-то обработок методом SSA. Положим  $L = \frac{\tau + 24}{2}$ . Для гибридных моделей обработаем только левую часть матрицы  $\mathbf{Z}$  методом SSA, как описано в разделе «2.2 SSA-preprocessing». Таким образом получили для каждой модели свою матрицу  $\mathbf{Z}$ .

## Результаты

Посмотрим на результаты в таблицах 4.3, 4.4. Видно, что среди обычных моделей лидирует GRU. Для случая  $r = 1$  результаты гибридных моделей хуже, чем у обычных моделей. Это объяснимо, так как по теории здесь нужно взять две компоненты, в случае  $r = 1$  мы не добрали еще одну компоненту. В случае  $r = 2$  гибридные модели

показывают наилучшие результаты. В частности SSA-RNN имеет наименьшую ошибку 0.99646 (в сравнении с сигналом данная модель также показывает лучшие результаты). Для  $r = 4$  ситуация выглядит немного хуже, так как в сигнал, выделенный SSA попал небольшой шум.

Таблица 4.3. Time Series: RMSE error.

	ANN	RNN	LSTM	GRU
raw	1.04703	1.04161	1.03374	1.02496
$r = 1$	1.27891	1.14262	1.08114	1.10521
$r = 2$	1.01494	0.99646	1.01255	1.01673
$r = 4$	1.01679	1.02385	1.03374	1.01549

Таблица 4.4. Signal: RMSE error.

	ANN	RNN	LSTM	GRU
raw	0.35654	0.29322	0.27239	0.28205
$r = 1$	0.73874	0.49316	0.35381	0.41770
$r = 2$	0.15457	0.11132	0.15909	0.16713
$r = 4$	0.1514	0.22734	0.16797	0.15862

## Выводы

Можно заключить, что правильное или избыточное количество компонент в предобработки SSA уменьшают ошибку для данного ряда. Наименьшую ошибку 1.02496 среди обычных моделей показала GRU. Среди гибридных моделей – SSA-RNN с ошибкой 0.99646.

## 4.2. Два синуса с близкими периодами

Рассмотрим следующий ряд  $Z_{1500}$ , состоящий из элементов

$$z_i = (\sin(2\pi \frac{i}{T}) + 1.6 \sin(2\pi \frac{i}{T+2})) \cdot (\cos(2\pi \frac{i}{432}) + 5) + 9.52\varepsilon_i,$$

где  $T = 12$ , а  $\varepsilon_i$  – шум из стандартного нормального распределения  $N(0, 1)$ .

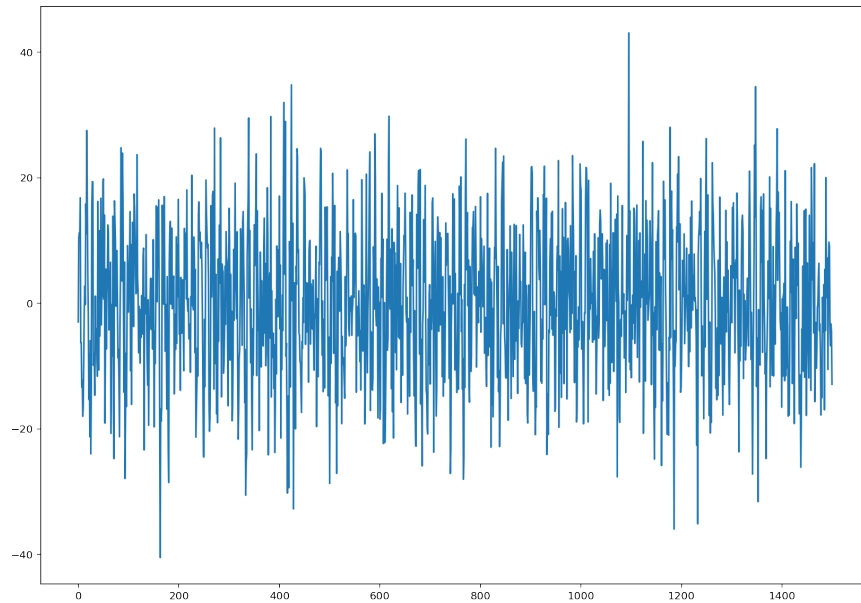


Рис. 4.10. Часть ряда  $Z_{1500}$ .

#### 4.2.1. Подбор гипер-параметров моделей

Проведем такой же эксперимент с подбором гипер-параметров, как и в разделе 4.1.2. В этот раз как средний период ряда равен 13, рассмотрим  $T = 13$  и  $T = 26$ . Цели эксперимента, модели и разбиение аналогичные, поэтому перейдем сразу к результатам.

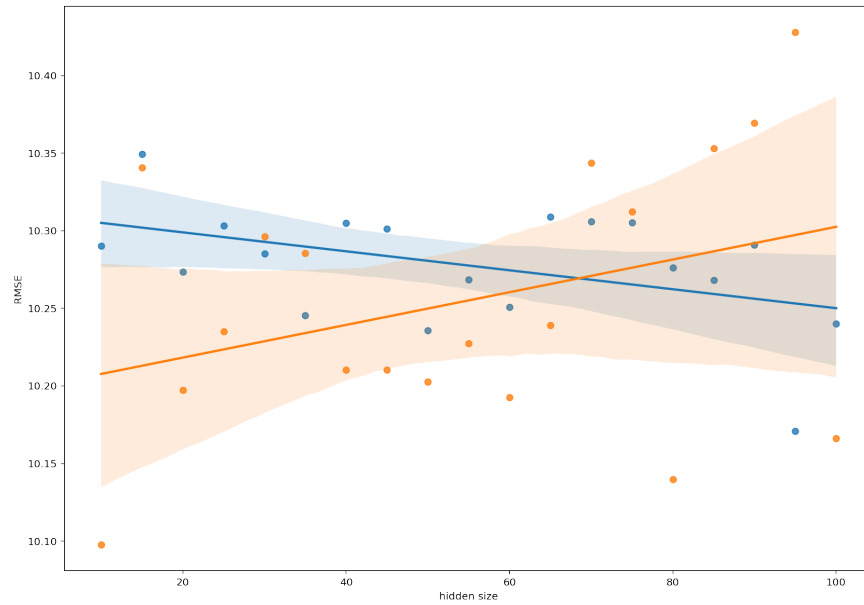
#### Результаты

Рассмотрим результаты для  $T = 13$  и  $T = 26$  вместе. Графики построены аналогично, как и в разделе 4.1.2. Синим цветом –  $T = 13$ , оранжевым –  $T = 26$ .

Для каждой линии регрессии будем проверять ее значимость с уровнем значимости 0.1. Если линия не значима, будем выбирать точки ближе к середине оси  $x$ . Если же линия значима, выбираем те точки с краю, на котором линия регрессии убывает. Также не будем брать точки, которые находятся далеко от линии регрессии. Выбирать размер скрытого слоя будем по линиям регрессии построенным на временном ряде. Стоит сразу отметить, что тенденция преимущества большего  $T$ , как и в разделе 4.1.2, здесь также сохраняется, кроме рис. 4.11 его рассмотрим отдельно. В остальных случаях будем сразу говорить о размере скрытого слоя.

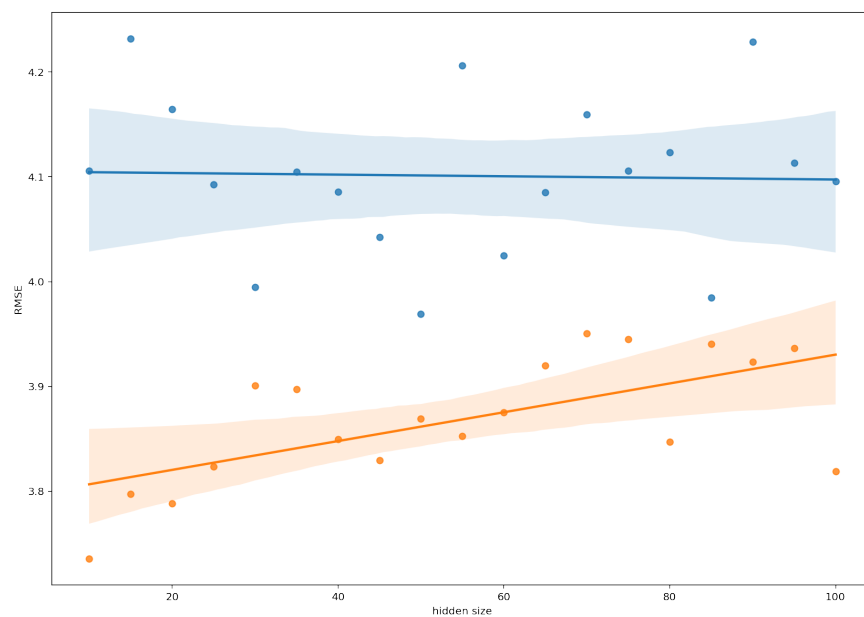
**ANN** Посмотрим на рисунки 4.11, 4.12. На первом рисунке не видим общей тенденции, но можно сказать что линия для 26 точек незначима для уровня значимости 0.1. Для

13 точек линия регрессии значимая, выбираем размер скрытого слоя поближе к 100, для 26 ближе к 50.



синий –  $T = 13$ , оранжевый –  $T = 26$ .

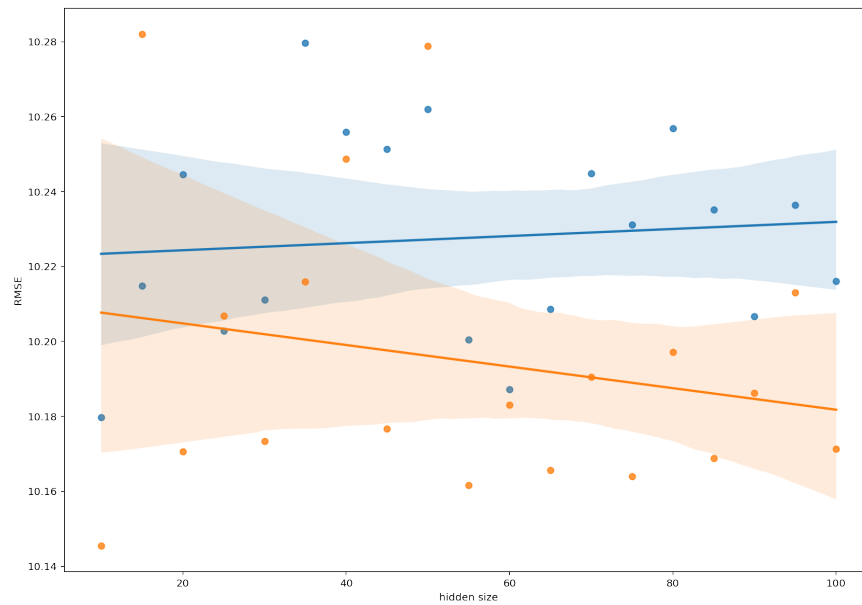
Рис. 4.11. Time Series ANN: p-value для 13 точек: 0.0548, p-value для 26 точек: 0.1516.



синий –  $T = 13$ , оранжевый –  $T = 26$ .

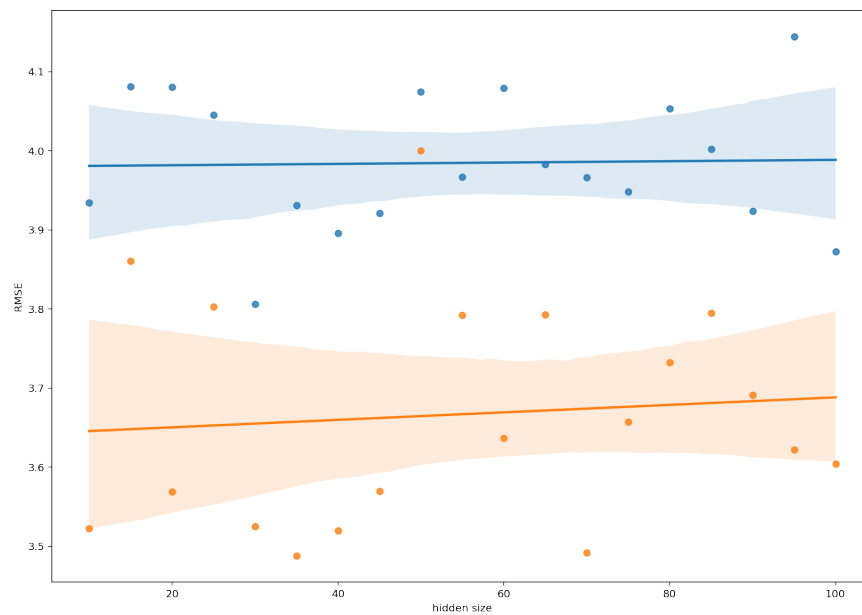
Рис. 4.12. Signal ANN: p-value для 13 точек: 0.9071, p-value для 26 точек: 0.0032.

**RNN** Посмотрим на рисунки 4.13, 4.14. Для 13 и 26 точек линии регрессии везде незначимые, выбираем размер скрытого слоя поближе к середине.



синий –  $T = 13$ , оранжевый –  $T = 26$ .

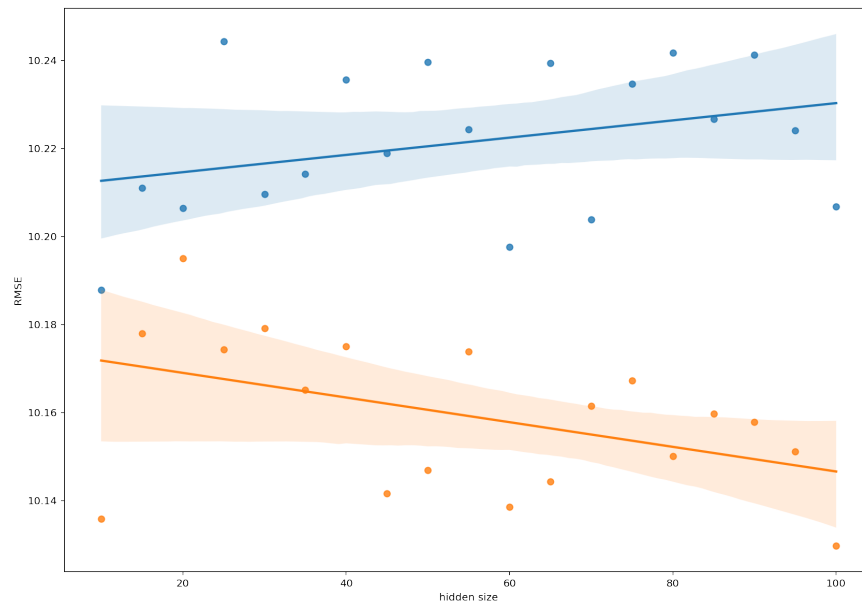
Рис. 4.13. Time Series RNN: p-value для 13 точек: 0.6919, p-value для 26 точек: 0.3872.



синий –  $T = 13$ , оранжевый –  $T = 26$ .

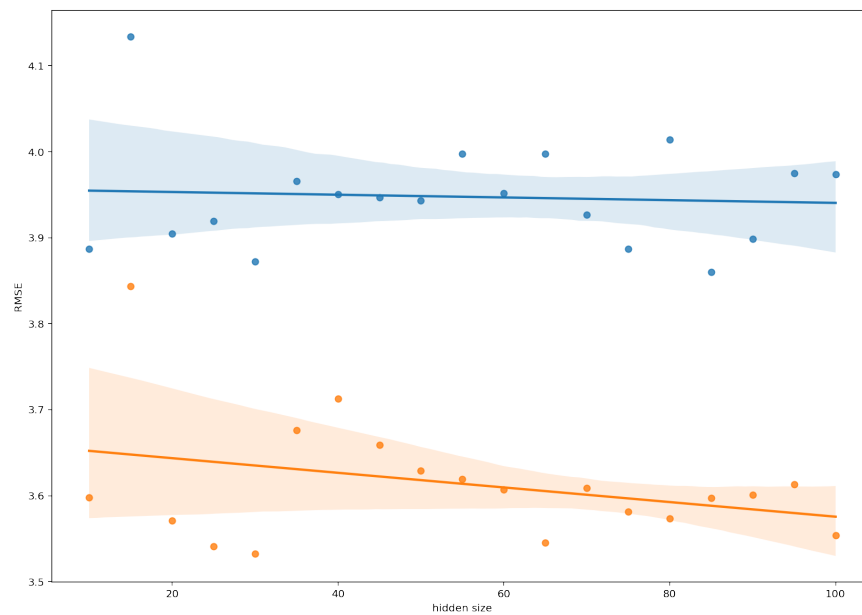
Рис. 4.14. Signal RNN: p-value для 13 точек: 0.9115, p-value для 26 точек: 0.7056.

**LSTM** Посмотрим на рисунки 4.15, 4.16. Для 13 точек линия регрессии незначимая, выбираем размер скрытого слоя поближе к 50, для 26 линии регрессии значима только на графике со временным рядом, выбираем размер ближе к 100.



синий –  $T = 13$ , оранжевый –  $T = 26$ .

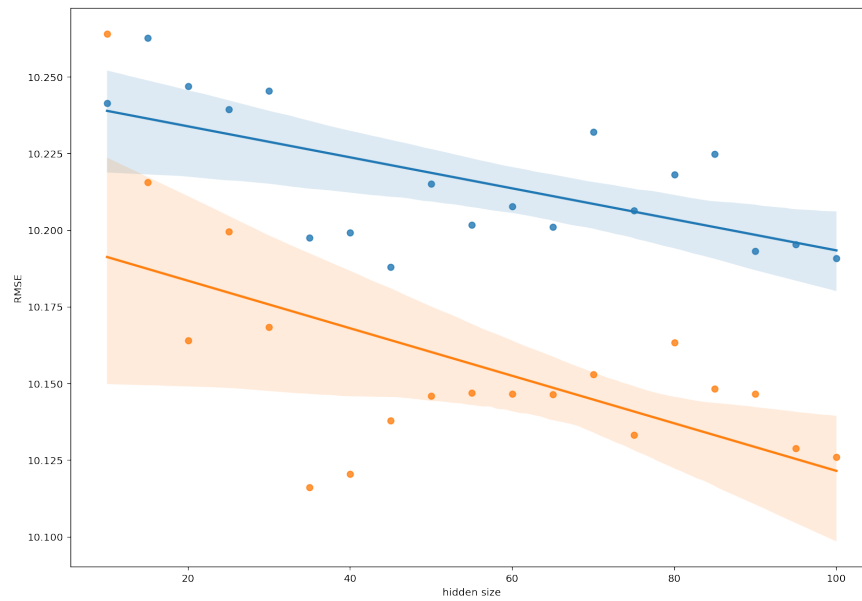
Рис. 4.15. Time Series LSTM: p-value для 13 точек: 0.1728, p-value для 26 точек: 0.0527.



синий –  $T = 13$ , оранжевый –  $T = 26$ .

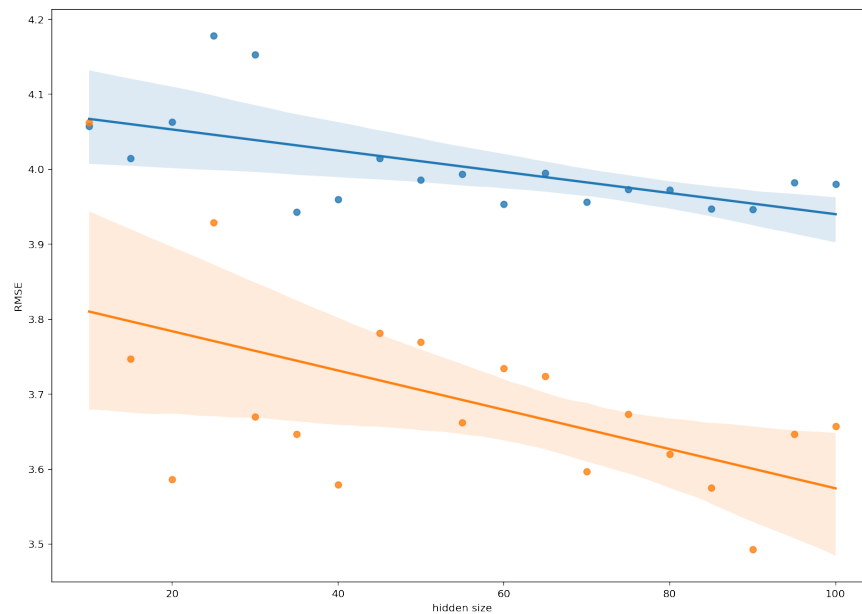
Рис. 4.16. Signal LSTM: p-value для 13 точек: 0.7746, p-value для 26 точек: 0.1665.

**GRU** Посмотрим на рисунки 4.17, 4.18. Для 13 и 26 точек все линии регрессии значимы, выбираем размер скрытого слоя поближе к 100.



синий –  $T = 13$ , оранжевый –  $T = 26$ .

Рис. 4.17. Time Series GRU: p-value для 13 точек: 0.0036, p-value для 26 точек: 0.0059.



синий –  $T = 13$ , оранжевый –  $T = 26$ .

Рис. 4.18. Signal GRU: p-value для 13 точек: 0.0069, p-value для 26 точек: 0.0126.

## Выводы

После полученных результатов для каждой модели можно зафиксировать размер скрытого слоя и размер  $T$ .  $T = 26$ , ANN - 55, RNN - 70, LSTM - 80, GRU - 100. В

дальнейших исследованиях для этого набора данных будем использовать эти модели с такими гипер-параметрами.

#### 4.2.2. Сравнение гипер-параметров SSA в гибридных моделях

Проведем такой же эксперимент со сравнением гипер-параметров, как и в разделе 4.1.3.  $L = 130, \frac{\tau + 26}{2}, T = 26$ . Цели эксперимента такие же, используются модели из раздела 4.2.1, разбиение проводится аналогично, поэтому перейдем сразу к результатам.

#### Результаты

Сначала посмотрим на случай  $L = 130$ . Посмотрим на результаты в таблицах 4.5, 4.6. Видно, что среди обычных моделей лидирует GRU. Для случаев  $r = 4, 8, 12, 16$  результаты гибридных моделей оказываются, лучше обычных моделей. Только для  $r = 8$  и модели SSA-ANN,  $r = 16$  и модели SSA-RNN результаты хуже, чем у обычных моделей. Не смотря на то, что правильное число компонент в этом ряде 12, наилучший результат показывает SSA-RNN при  $r = 4$  в случае прогнозирования ряда. В случае же с сигналом лучшей моделью является SSA-LSTM при  $r = 4$ .

Таблица 4.5. Time Series:  $L = 130$ , RMSE error.

	ANN	RNN	LSTM	GRU
raw	10.0203	10.0113	9.9198	10.0682
$r = 4$	9.6720	9.6387	9.665	9.7527
$r = 8$	10.0507	9.7322	9.7252	9.7552
$r = 12$	9.7937	9.8797	9.8661	9.8187
$r = 16$	9.9824	10.1346	9.879	9.905



Таблица 4.6. Time Series:  $L = 130$ , RMSE error.

	ANN	RNN	LSTM	GRU
raw	4.2093	4.096	4.0024	4.503
$r = 4$	2.206	2.3964	2.1339	2.5055
$r = 8$	3.8976	2.8502	2.4311	2.6825
$r = 12$	3.1592	3.2036	2.5901	2.8049
$r = 16$	3.1775	3.5174	2.5237	2.7617

Теперь перейдем к случаю  $L = 130$ . Посмотрим на результаты в таблицах 4.7, 4.8. В этот раз все результаты гибридных моделей лучше, чем обычных. Наилучший результат показывает SSA-ANN при  $r = 12$  в случае прогнозирования ряда. В случае же с сигналом лучшей моделью является SSA-GRU при  $r = 8$ .

Таблица 4.7. Time Series:  $L = \frac{\tau + 26}{2}$ , RMSE error.

	ANN	RNN	LSTM	GRU
raw	10.0203	10.0113	9.9198	10.0682
$r = 4$	9.5322	9.7205	9.5465	9.5742
$r = 8$	9.5442	9.7658	9.6238	9.5965
$r = 12$	9.4907	9.7483	9.5908	9.6144
$r = 16$	9.5198	9.6957	9.6147	9.6166

Таблица 4.8. Time Series:  $L = \frac{\tau + 26}{2}$ , RMSE error.

	ANN	RNN	LSTM	GRU
raw	4.2093	4.096	4.0024	4.503
$r = 4$	2.0693	3.0939	2.0285	2.0219
$r = 8$	2.2793	3.1766	2.3245	1.9417
$r = 12$	2.3813	3.0823	2.1882	1.9889
$r = 16$	2.2503	2.3004	2.011	2.0127

## Выводы

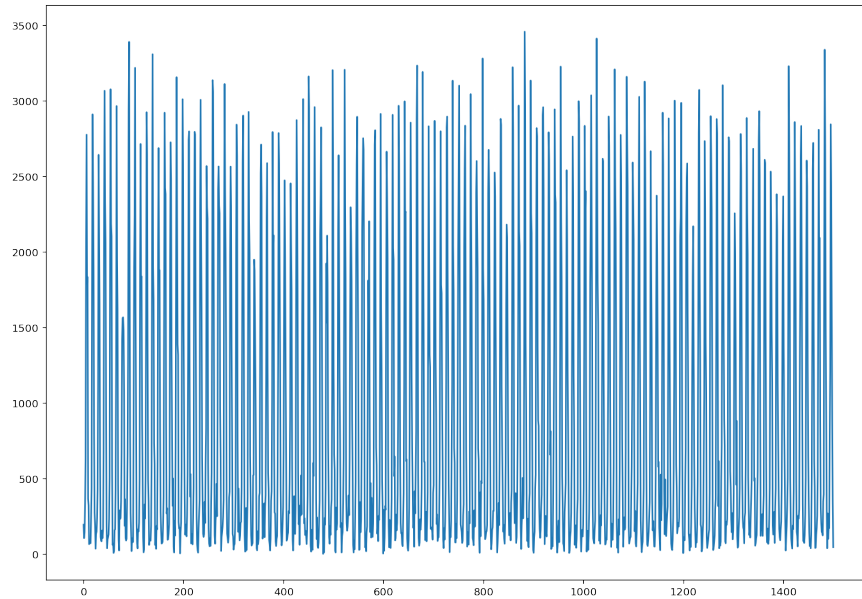
Также как и в аналогичном эксперименте, представленном в разделе 4.1.3, можно заключить, что правильное или избыточное количество компонент в предобработки SSA уменьшают ошибку для данного ряда. Наименьшую ошибку 9.9198 среди обычных моделей показала GRU. Среди гибридных моделей – SSA-RNN с ошибкой 9.4907. Можно зафиксировать для этого ряда  $L = \frac{\tau + 26}{2}$  и  $r = 12$ .

## Глава 5

## Реальные данные

## 5.1. Среднемесячные осадки в Индии

Рассмотрим следующий ряд  $Z_{1500}$  (рис. 5.1) взятый из статьи [2]. На ряде отображены среднемесячное количество осадков в Индии.

Рис. 5.1. Ряд  $Z_{1500}$ .

### 5.1.1. Различное использование SSA предобработки в гибридной модели SSA-ANN

Поставим задачу предсказывать следующую точку ряда  $Z_{1500}$  по двенадцати предыдущим, то есть возьмем  $T = 12$  и  $R = 1$ . В рамках этой задачи хотим сравнить обычную модель ANN и гибридные модели SSA-ANN по метрике RMSE. Также как и в главе «4.1 Синус с шумом» рассмотрим две версии гибридных моделей SSA-ANN. В первой версии обработка методом SSA будет применяться для левой части  $\mathbf{Z}^x$  траекторной матрицы. Во второй версии только правая часть  $\mathbf{Z}^y$  обрабатывается с помощью метода SSA. Будем использовать такие же обозначения для этих моделей как и в разделе «4.1 Синус с шумом».

## Модели

Для предсказаний воспользуемся теми же моделями, что и в задаче с синусом, описанными ранее в разделе 4.1.1.

## Подготовка данных

Построим траекторную матрицу для ряда  $Z_{1500}$  с окном длиной 13.

$$\mathbf{Z} = \mathcal{T}_{13}(Z_{1500}),$$

Разобьем данные на тренировочную, валидационную и тестовую выборку следующим образом:  $\tau = 988$ ,  $v = 1238$  и  $t = 1488$ .

Далее для модели ANN-2 будем использовать уже полученную матрицу  $\mathbf{Z}$  без каких-то обработок методом SSA. Параметры оператора  $SSA_{L,r}$  задаются следующим образом  $L = \frac{\tau + 12}{2}$ ,  $r = 7$ . Для гибридных моделей SSA-X-ANN обработаем только левую часть матрицы  $\mathbf{Z}$  методом SSA, как описано в разделе «2.2 SSA-preprocessing». Для гибридных моделей SSA-Y-ANN обработаем только правую часть матрицы  $\mathbf{Z}$  методом SSA. Таким образом получили для каждой модели свою матрицу  $\mathbf{Z}$ .

## Результаты

В ходе эксперимента получились для моделей ANN-1 и SSA-ANN-1 следующие результаты, представленные на таблице 5.1. А для моделей ANN-2 и SSA-ANN-2 на таблице 5.2.

Таблица 5.1. RMSE для моделей ANN-1 и SSA-ANN-1.

	ANN-1	SSA-X-ANN-1	SSA-Y-ANN-1
RMSE	267.21	224.35	269.23

Таблица 5.2. RMSE для моделей ANN-2 и SSA-ANN-2.

	ANN-2	SSA-X-ANN-2	SSA-Y-ANN-2
RMSE	239.84	218.57	242.72

На таблице видна такая же тенденция, как и в задаче с синусом с шумом. Модели ANN и SSA-Y-ANN показывают сравнительно похожие результаты. А вот модели

SSA-X-ANN показывают наилучшие результаты. Также видно значительное увеличение ошибки для моделей ANN-1 и ANN-2, это ожидаемый результат, так как сложность модели ниже ANN-1, чем у модели ANN-2. Но для моделей SSA-X-ANN-1 и SSA-X-ANN-2 разница в ошибке небольшая, что подчеркивает хорошую работу SSA в выделении сигнала.

Можно заключить, что с помощью гибридных моделей удалось улучшить точность прогнозирования реального ряда.

## Выводы

Исходя из результатов полученных в этом эксперименте, и в разделе 4.1.1, можно заключить, что правильное использование гибридных моделей SSA-ANN приводит к весьма хорошим улучшениям в задаче прогнозирования рядов.

### 5.1.2. Количество точек для предсказания

После выводов в экспериментах в разделах 4.1.2, 4.2.1 хотим проверить сохраняется ли тенденция, наблюдаемая в них. В этот раз данные не будут предобработаны с помощью SSA.  $T = 12, 24$  и  $R = 1$ . Модели будем сравнивать с помощью метрики RMSE. Все модели и разбиение данных проводится аналогично как в разделе 4.1.2.

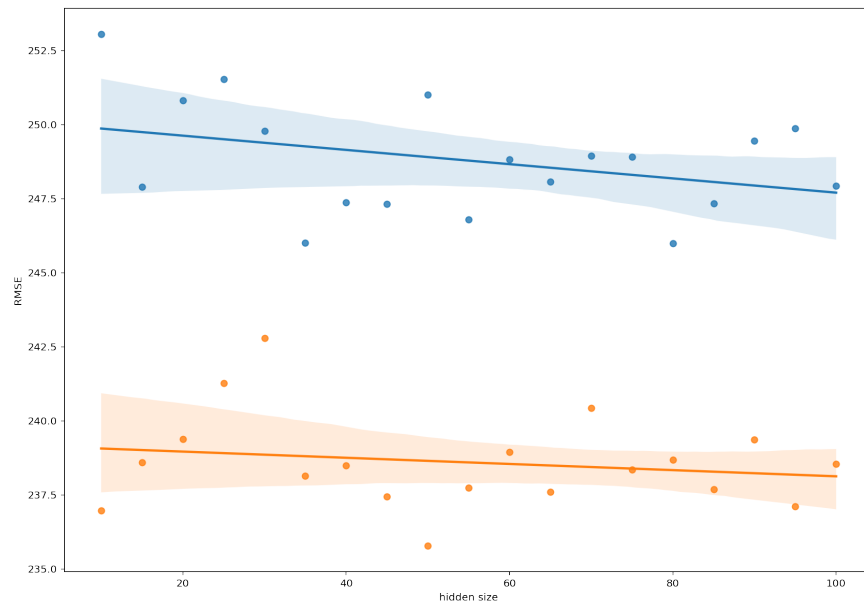
## Результаты

Рассмотрим результаты для  $T = 12$  и  $T = 24$  вместе. Графики построены аналогично, как и в разделе 4.1.2. Синим цветом –  $T = 12$ , оранжевым –  $T = 24$ . (Забегая вперед, скажем что тенденция будет сохраняться для всех моделей. Поэтому для экономии времени читателя будем рассматривать лучшие параметры только для случае  $T = 24$ )

Для каждой линии регрессии будем проверять ее значимость с уровнем значимости 0.1. Если линия не значима, будем выбирать точки ближе к середине оси  $x$ . Если же линия значима, выбираем те точки с краю, на котором линия регрессии убывает. Также не будем брать точки, которые находятся далеко от линии регрессии.

**ANN** Посмотрим на рисунок 5.2. Видно, что тенденция наблюдаемая в разделе 4.1.2 сохраняется и на реальном ряде. В данном случае размер скрытого слоя стоит выбрать

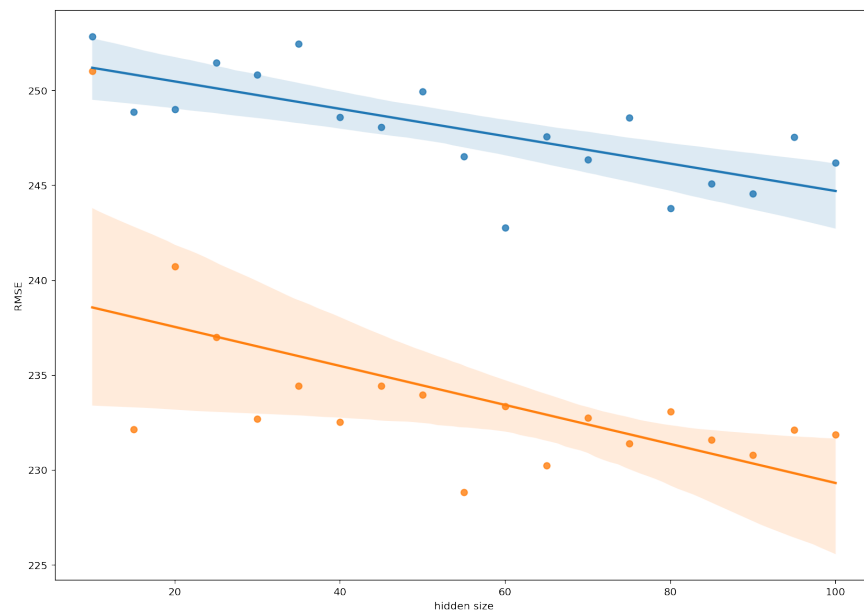
ближе к 50, так как линия регрессии незначимая.



синий –  $T = 12$ , оранжевый –  $T = 24$ .

Рис. 5.2. Indian Rain ANN: p-value для 12 точек: 0.1356, p-value для 24 точек: 0.4538.

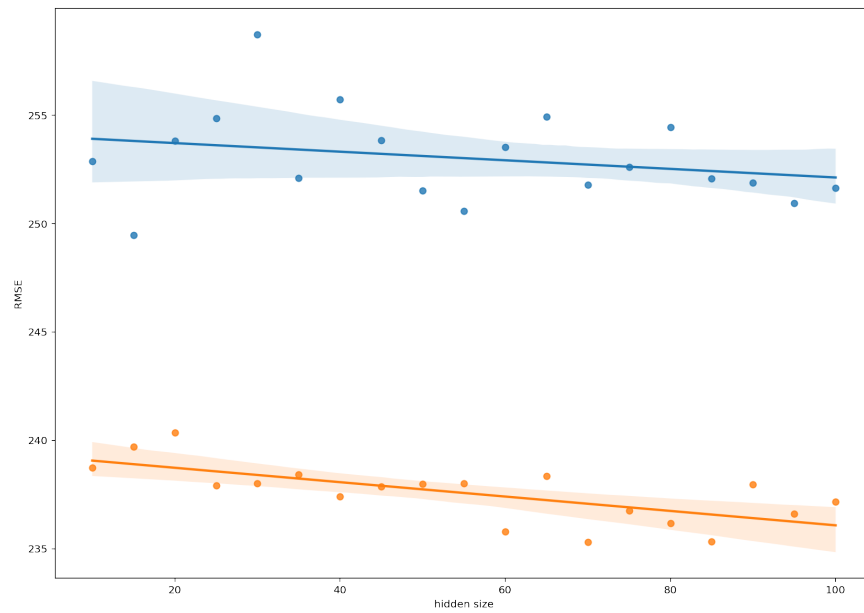
**RNN** Посмотрим на рисунок 5.3. В данном случае размер скрытого слоя стоит выбрать ближе к 100, так как линия регрессии значимая, в этот раз.



синий –  $T = 12$ , оранжевый –  $T = 24$

Рис. 5.3. Indian Rain RNN: p-value для 12 точек: 0.0005, p-value для 24 точек: 0.0072.

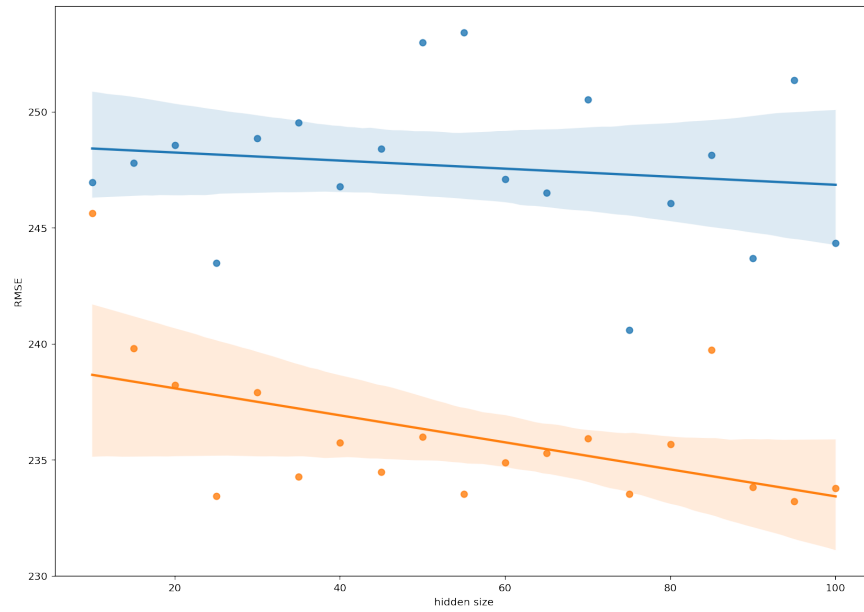
**LSTM** Посмотрим на рисунок 5.4. В данном случае размер скрытого слоя стоит брать ближе к 100.



синий –  $T = 12$ , оранжевый –  $T = 24$

Рис. 5.4. Indian Rain LSTM: p-value для 12 точек: 0.2802, p-value для 24 точек: 0.001.

**GRU** Посмотрим на рисунок 5.5. Также как и в случаях LSTM, RNN линия регрессии значима, но последние значения слишком сильно выходят из доверительного интервала, поэтому стоит взять размер скрытого слоя до 80.



синий –  $T = 12$ , оранжевый –  $T = 24$

Рис. 5.5. Indian Rain GRU: p-value для 12 точек: 0.5381, p-value для 24 точек: 0.0202.

## Выводы

После полученных результатов для каждой модели можно зафиксировать размер скрытого слоя и размер  $T$ .  $T = 24$ , ANN - 60, RNN - 90, LSTM - 95, GRU - 65. В дальнейших исследованиях для этого набора данных будем использовать эти модели с такими гипер-параметрами.

### 5.1.3. Вклад SSA в улучшение прогноза реального ряда

После выводов в экспериментах в разделах 4.1.3, 4.2.2 хотим проверить какой вклад внесет SSA на реальных данных. В этот раз возьмем  $T = 24$  и  $R = 1$ . Возьмем модель GRU, которая показала себя наилучшей в экспериментах 4.1.3, 4.2.2. Размер скрытого слоя будем перебирать по сетке от 10 до 100 с шагом 5. Модели GRU и SSA-GRU будем сравнивать с помощью метрики RMSE. Все модели и разбиение данных проводится аналогично как в разделе 4.1.2.

## Подготовка данных

Построим траекторную матрицу для ряда  $Z_{1500}$  с окном длиной 13.



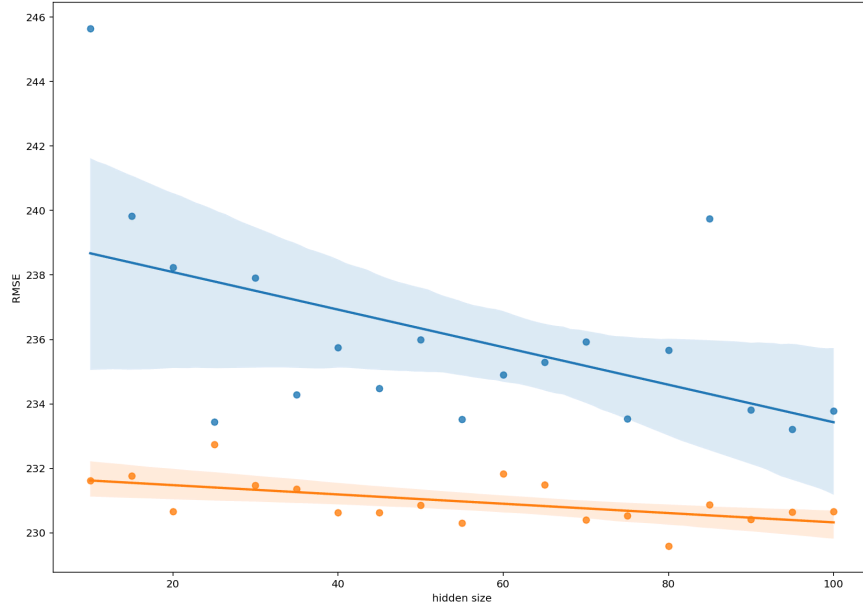
$$\mathbf{Z} = \mathcal{T}_{13}(\mathbf{Z}_{1500}),$$

Разобьем данные на тренировочную, валидационную и тестовую выборку следующим образом:  $\tau = 976$ ,  $v = 1226$  и  $t = 1476$ .

Далее для обычной модели будем использовать уже полученную матрицу  $\mathbf{Z}$  без каких-то обработок методом SSA. Параметры оператора  $SSA_{L,r}$  задаются следующим образом  $L = \frac{\tau + 24}{2}$ ,  $r = 7$ . Для гибридной модели обработаем только левую часть матрицы  $\mathbf{Z}$  методом SSA, как описано в разделе «2.2 SSA-preprocessing». Таким образом получили для каждой модели свою матрицу  $\mathbf{Z}$ .

## Результаты

Посмотрим на картинку 5.6. На ней видно, что SSA имеет весомый вклад на этих данных. Также можно видеть, что с ростом скрытого слоя ошибка гибридной модели SSA-GRU падает. То есть SSA не ограничивает модель.



синий – GRU, оранжевый – SSA-GRU

Рис. 5.6. Indian Rain GRU: p-value для GRU: 0.0202, p-value для SSA-GRU: 0.0118.

## Выводы

Из результатов становится понятно, что использование предобработки SSA приводит к уменьшению ошибки на реальных данных, рассматриваемых в этой главе.

## Заключение

В работе была рассмотрена методика исследования сравнения методов, которая показалась весьма успешной, особенно на реальных данных среднемесячных осадков в Индии. В дальнейшем планируется ее развивать, в частности попробовать перенести все полученные результаты на данные параметров вращения Земли (ЕОР). Возможно придется усложнить модельные данные для данных ЕОР.

## Список литературы

1. Golyandina, N., Nekrutkin, V., & Zhigljavsky, A. (2001). Analysis of time series structure: SSA and related techniques. Chapman & Hall/CRC.
2. Kongchang Du, Ying Zhao, Jiaqiang Lei (2017). The incorrect usage of singular spectral analysis and discrete wavelet transform in hybrid models to predict hydrological time series.