

Санкт-Петербургский государственный университет
Прикладная математика и информатика

Производственная практика 1 (научно-исследовательская работа)

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

Введение	4
Глава 1. Singular Spectrum Analysis	5
1.1. Алгоритм SSA	5
1.1.1. Этап 1. Построение траекторной матрицы (Вложение)	5
1.1.2. Этап 2. Singular Value Decomposition (SVD)	6
1.1.3. Этап 3. Группировка первых r собственных троек	6
1.1.4. Этап 4. Диагональное усреднение	7
Глава 2. Использование SSA в машинном обучении	8
2.1. Задача	8
2.2. Подготовка данных	8
2.2.1. SSA-preprocessing	8
2.3. Модель	10
2.4. Прогнозирование	11
2.5. Метрики	12
Глава 3. SSA & ANN	13
3.1. Синус с шумом	13
3.1.1. Постановка задачи	13
3.1.2. Модели	14
3.1.3. Подготовка данных	14
3.1.4. Обучение и прогнозирование	15
3.1.5. Результаты	15
3.2. Реальный ряд	16
3.2.1. Постановка задачи	16
3.2.2. Модели	17
3.2.3. Данные, обучение и прогнозирование	17
3.2.4. Результаты	17
3.3. Итоги	18
Заключение	19

Список литературы	20
------------------------------------	----

Введение

Метод Singular Spectrum Analysis (SSA) — хорошо развитая методология анализа и прогнозирования временных рядов, которая включает в себя множество различных, но взаимосвязанных методов. Область применения SSA очень широка — от непараметрической декомпозиции и фильтрации временных рядов до оценки параметров и прогнозирования.

Artificial neural network (ANN) — математическая модель, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Нейронная сеть представляет собой частный случай методов распознавания образов, дискриминантного анализа, методов кластеризации и т.п.

В прошлом отчете обозревалась статья «The incorrect usage of singular spectral analysis and discrete wavelet transform in hybrid models to predict hydrological time series» [2], в которой говорилось, что использование гибридных моделей SSA-ANN ведет к значительным улучшениям в точности предсказания из-за заглядывания в будущее. Из-за чего преимущество гибридных моделей пропадает при корректном тестировании методов. В этой работе поставлена следующая задача: проверить возможность прироста точности при использовании гибридных моделей SSA-ANN в сравнении с обычными моделями ANN.

Глава 1

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].

1.1. Алгоритм SSA

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 Δ — некоторый временной интервал.

1.1.1. 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:

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

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 X_N траекторную матрицу \mathbf{X} обозначим:

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

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

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:

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

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

1.1.3. 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: $\mathbf{X}_I = \mathbf{X}_1 + \dots + \mathbf{X}_r$.

1.1.4. 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 $\hat{\mathbf{F}} = (f_1 \cdots f_{N-1})$. Полученный ряд $\hat{\mathbf{F}}$ назовем оценкой сигнала, полученной с помощью SSA. Процедуру выделения сигнала с помощью SSA обозначим как:

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

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

Глава 2

Использование SSA в машинном обучении

2.1. Задача

Рассмотрим Z_N — временной ряд длины N и задачу: с помощью Artificial Neural Network (ANN) модели на основе T последовательных точек ряда Z_N , предсказать следующие R точек ряда. Решение данной задачи, можно разбить на несколько частей: подготовка данных, обучение модели, прогнозирование.

2.2. Подготовка данных

Z_N — изначальный временной ряд длиной N . Мы можем представить ряд в виде траекторной матрицы для длины окна $T + R$, тогда получим:

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

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 τ , 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 $\mathbf{Z}^x = \mathbf{Z}_{1,t}^{(1,T)}$, $\mathbf{Z}^y = \mathbf{Z}_{1,t}^{(T+1,T+R)}$.

2.2.1. 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 $\hat{z}'_{b+i-1,p+i} = SSA_{L,r}(\mathbf{Z}_{b+i-1,p+i})$, 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: $\hat{\mathbf{Z}}_Q = SSA^{(p)}(\mathbf{Z}_{b,e})$. 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 $\hat{\mathbf{Z}}_Q = SSA^{(p)}(\mathbf{Z}_{1,v+T+R})$.
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

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:

$$\hat{X}^T = [x_1, \dots, \hat{x}_T]$$

— входные данные, на которых модели учиться делать предсказания.

$$\hat{Y}^T = [y_1, \dots, \hat{y}_R]$$

— 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 θ — веса модели. ϕ — 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: $\phi(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$

Обучение Оптимизация параметров модели ANN w, θ 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 ϕ_1, ϕ_2, h 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, θ 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 ϵ_i — ошибка на валидационной выборке. Для валидационной выборки \mathbf{Z}_{val}^x строится прогноз $\hat{\mathbf{Z}}_{val}^y$. Ошибка ϵ_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, θ 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

SSA & ANN

В этой главе описано два эксперимента, в которых сравнивается простая модель ANN и гибридная модель SSA-ANN на примере синуса с шумом и реальных данных.

3.1. Синус с шумом

Так как реальные данные, рассмотренные во втором эксперименте в разделе «3.2 Реальный ряд» похожи на сумму синусов с разной амплитудой, мы хотим показать, что SSA дает прирост в точности в задаче прогнозирования синуса с шумом.

3.1.1. Постановка задачи

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

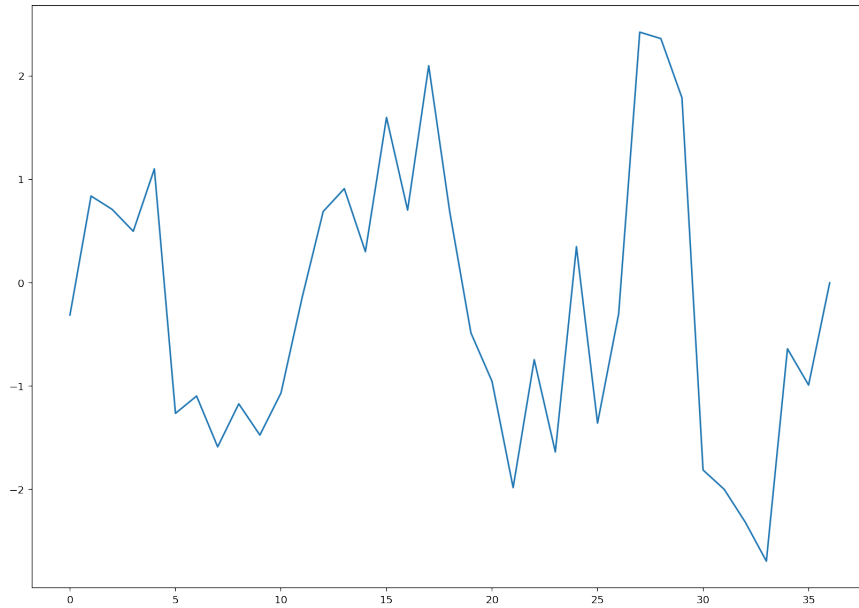


Рис. 3.1. Часть ряда Z_{1500} .

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.

3.1.2. Модели

Для предсказаний воспользуемся нейронной сетью, описанной в разделе «2.3 Модель». 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. ϕ_1 – является функцией активации ReLU, а ϕ_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 «3.1.1. Постановка задачи», для каждой обычной (например, SSA-X-ANN-1 или SSA-Y-ANN-2).

3.1.3. Подготовка данных

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

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

Разобъем 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 мат-

рицы \mathbf{Z} методом SSA, как описано в разделе «2.2.1. SSA-preprocessing». Для гибридных моделей SSA-Y-ANN-1 и SSA-Y-ANN-2 обрабатываем только правую часть матрицы \mathbf{Z} методом SSA. Таким образом получили для каждой модели свою матрицу \mathbf{Z} .

3.1.4. Обучение и прогнозирование

Теперь когда для каждой модели есть своя матрица \mathbf{Z} , их можно обучить как описано параграфе «обучение» раздела «2.3. Модель». После обучения модели тестируются на тестовой выборке по алгоритму, описанному в разделе «2.4. Прогнозирование».

3.1.5. Результаты

В ходе эксперимента полученные результаты для моделей ANN-1 и SSA-ANN-1 представлены в таблице 3.1. Результаты для моделей ANN-2 и SSA-ANN-2 представлены в таблице 3.2.

Таблица 3.1. 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 3.2. 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 σ^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 отличие от ANN пытается прогнозировать чистый сигнал, что дает лучшие результаты.

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.

3.2. Реальный ряд

Теперь, когда мы убедились, что гибридные модели работают лучше, чем обычные в задаче прогнозирования синуса с шумом, перейдем к прогнозированию реального ряда.

3.2.1. Постановка задачи

Рассмотрим следующий ряд Z_{1500} (рис. 3.2) взятый из статьи [2]. На ряде отображены среднемесячное количество осадков в Индии.

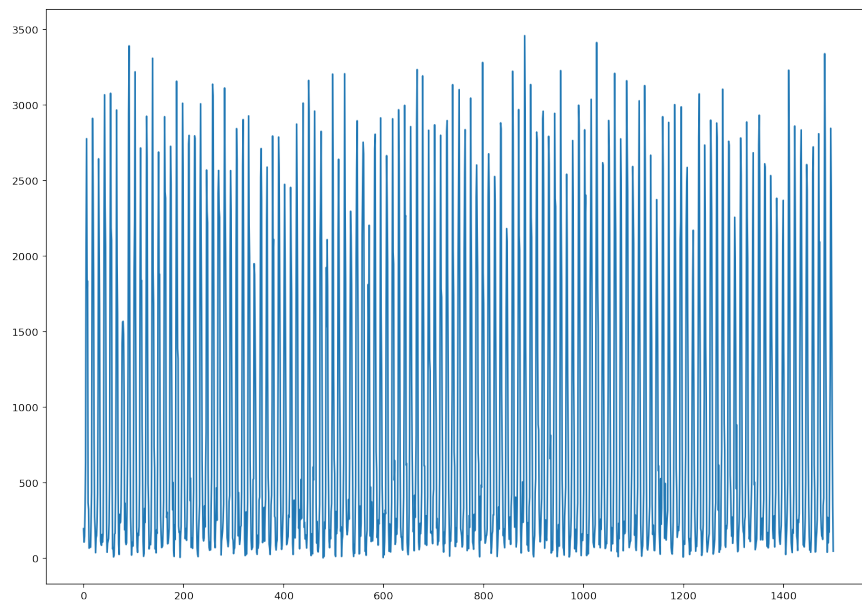


Рис. 3.2. Ряд Z_{1500} .

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 «3.1. 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 «3.1. Синус с шумом».

3.2.2. Модели

Для предсказаний воспользуемся теми же моделями, что и в задаче с синусом, описанными ранее в разделе «3.1.2 Модели».

3.2.3. Данные, обучение и прогнозирование

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

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

Разобьем данные на тренировочную, валидационную и тестовую выборку следующим образом: $\tau = 988$, $v = 1238$ 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 «2.2.1. 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} .

Теперь когда для каждой модели есть своя матрица \mathbf{Z} , их можно обучить как описано в параграфе «обучение» раздела «2.3. Модель». После обучения, модели тестируются на тестовой выборке по алгоритму, описанному в разделе «2.4. Прогнозирование».

3.2.4. Результаты

В ходе эксперимента получились для моделей ANN-1 и SSA-ANN-1 следующие результаты, представленные на таблице 3.3. А для моделей ANN-2 и SSA-ANN-2 на таблице 3.4.

Таблица 3.3. 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 3.4. 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.

3.3. Итоги

Исходя из результатов полученных в экспериментах, описанных ранее, можно заключить, что правильное использование гибридных моделей SSA-ANN приводит к весьма хорошим улучшениям в задаче прогнозирования рядов.

Заключение

В работе был рассмотрен алгоритм SSA, использующий в гибридных моделях SSA-ANN, как предобработка данных. Были описаны эксперименты в которых сравнивались обычные модели ANN и гибридные модели SSA-ANN в разных задачах. Для всех сравнений были приведены результаты в метрике RMSE. Также была описана математическая база, заложенная в основу всех экспериментов. В ходе всех экспериментов удалось достичь улучшения точности с помощью гибридных моделей SSA-ANN по сравнению с обычными моделями ANN.

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.