

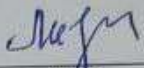
Министерство науки и высшего образования Российской Федерации

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК

Руководитель ООП

д-р тех наук, профессор


_____ К.И. Лившиц
подпись

« _____ » _____ 20 ____ г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

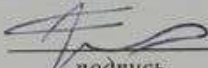
Алгоритм обнаружения момента разладки стохастического
процесса с неизвестными параметрами и неизвестной дисперсией

по направлению подготовки 01.03.02 Прикладная математика и информатика,
направленность (профиль) «Математические методы в экономике»

Лихота Константин Евгеньевич

Руководитель ВКР

д-р физ.-мат. наук, доцент


_____ С.Э. Воробейчиков
подпись

« _____ » _____ 20 ____ г.

Автор работы

студент группы № 931823

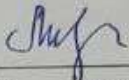

_____ К.Е. Лихота
подпись

« _____ » _____ 20 ____ г.

Томск – 2022

Министерство науки и высшего образования Российской Федерации.
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ
Руководитель ООП
д-р тех наук, профессор

 К.И.Лившиц
«25» октября 2021 г.

ЗАДАНИЕ

по выполнению выпускной квалификационной работы бакалавра обучающегося

Лихота Константин Евгеньевич

Фамилия Имя Отчество обучающегося

по направлению подготовки 01.03.02 Прикладная математика и информатика,
направленность (профиль) «Математические методы в экономике»

1 Тема выпускной квалификационной работы

Алгоритм обнаружения момента разладки стохастического процесса с неизвестными параметрами и неизвестной дисперсией.

2 Срок сдачи обучающимся выполненной выпускной квалификационной работы:

а) в учебный офис / деканат – 27.05.2022 б) в ГЭК – 09.06.2022

3 Исходные данные к работе:

Объект исследования – Решение проблемы обнаружения разладки случайного процесса

Предмет исследования – Алгоритмы обнаружения разладки случайного процесса при неизвестной дисперсии

Цель исследования – Построение процедуры обнаружения разладки авторегрессионного процесса с неизвестной дисперсии шума.

Задачи: Построить процедуры для обнаружения разладки случайного процесса, оценки множителя неизвестного шума. Провести модулирование на основе сгенерированных и реальных данных.

Методы исследования: Численные методы

Организация или отрасль, по тематике которой выполняется работа, –

НИ ТГУ, Институт прикладной математики и компьютерных наук, кафедра САиММ

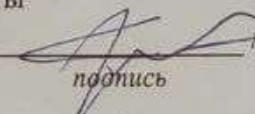
4 Краткое содержание работы

Описание построения процедуры обнаружения разладки авторегрессионного процесса произвольного порядка. Описание построения процедуры оценки неизвестной дисперсии. Результаты моделирования на сгенерированных и реальных данных

Руководитель выпускной квалификационной работы

Профессор кафедры САиММ

должность, место работы


подпись

С.Э.Воробейчиков

И.О. Фамилия

Задание принял к исполнению

Студент группы 931823 ТГУ

должность, место работы


подпись

К. Е. Лихота

И.О. Фамилия

АННОТАЦИЯ

В работе рассматривается задача обнаружения моментов разладки случайного процесса авторегрессии n -го порядка с гауссовыми шумами, для которых значение параметров авторегрессии и дисперсия шума считаются не известными. Предложенный алгоритм основан на оценке неизвестных параметров и дисперсии на последовательных участках стохастического процесса.

На начальном этапе находятся оценки параметров по методу наименьших квадратов на интервале заданной длины. Полученные результаты используются для построения оценки неизвестной дисперсии на последующем интервале. После этого используется последовательная процедура оценивания параметров с гарантированной среднеквадратической точностью. Знание точности полученных оценок позволяет использовать их для определения моментов изменения статистических свойств наблюдаемого процесса. Обнаружение момента изменения параметров процесса осуществляется с помощью сравнения оценок параметров, полученных на соседних интервалах наблюдения процесса. Для этого используется сравнение нормы разности оценок параметров с заданным порогом.

Проведено численное моделирование построенной процедуры обнаружения разладки. Результаты применяются для анализа поведения финансовых индексов

ОГЛАВЛЕНИЕ

Введение.....	5
1 Постановка задачи.....	8
2 Построение модели	10
2.1 Построение решающей процедуры	10
2.2 Построение оценки неизвестной дисперсии	15
3 Результаты моделирования	17
3.1 Результаты моделирования на сгенерированных данных	17
3.2 Результаты моделирования на реальных данных	26
Заключение	34
Список использованной литературы.....	35
Приложение А Код имитационной модели	37

ВВЕДЕНИЕ

Актуальность темы исследования. При работе с экспериментальными данными часто случается, что параметры математических моделей на разных участках могут значительно отличаться друг от друга. Такие изменения характеристик или даже самого вида случайного процесса называются разладкой. Задача обнаружения момента разладки случайного процесса является одной из широко распространенных задач анализа и обработки информации. Многие прикладные задачи анализа данных в различных областях можно свести к задаче обнаружения разладки. Приведем несколько примеров.

В экономике - анализ финансовых рынков и контроль качества выпускаемой продукции.

В метеорологии – ученые получают множество данных при исследовании земной коры. И для предсказания землетрясений или других бедствий на основе анализа показателей датчиков сейсмических станций, необходимо определять моменты изменения свойств фиксируемых процессов.

В медицине – при мониторинге жизненных показателей тяжелобольных пациентов также возникает задача обнаружения разладки, так как резкое изменения состояния больного может указывать на опасность для его жизни.

В обработке сигналов в различных компьютерных системах. Пусть имеются некоторые датчики, мониторящие технологический процесс. Этот процесс находится в некотором нормально функционирующем состоянии. Но в некоторый момент времени может произойти сбой и этой может означать поломку в системе, которую необходимо устранить как можно скорее.

Примеры этих и других исследований можно найти в работах (1) – (5). Как можно заметить все эти задачи требуют в качестве решения нахождения алгоритма, который по стохастическому процессу наблюдений сможет найти момент разладки как можно более точно и с наименьшей задержкой. Из этих

соображений можно выделить две наиболее важных характеристик такой процедуры. Первая связана с как можно большей скоростью обнаружения разладки после того как она произошла – то есть среднее время между моментом разладки и его обнаружением или кратко среднее время запаздывания. Вторая характеристика связана с соображением что алгоритм должен как можно реже ошибаться, выдавая как можно меньше ложных тревог – это среднее время между ложными тревогами.

Чаще всего в качестве модели для обнаружения разладки случайного процесса используют авторегрессию. Основная идея авторегрессии в том, что значения временного ряда в данный момент линейно зависят от предыдущих значений этого же ряда.

Существует множество различных алгоритмов обнаружения момента разладки случайного процесса, но их условно можно разделить на две группы: методы апостериорного обнаружения и последовательные методы.

При апостериорном обнаружении мы предполагаем, что в некоторый момент произошло изменение характеристик процесса. И, чтобы оценить момент этого изменения, мы собираем всю информацию о процессе. И делаем выводы по результатам всей этой информации в целом.

Последовательные методы, напротив, основаны на том, что на каждом шаге наблюдения мы используем только информацию, полученную на предыдущих этапах. И на основе этой информации либо принимается решение о наличии разладки, либо — о продолжении наблюдений. Соответственно, при использовании этого метода возникают ложные тревоги и задержки нахождения момента разладки.

В этой работе процедура обнаружения разладки будет основана на последовательном сравнении оценок параметров на участках наблюдения отстоящих друг от друга на заданную величину. Чтобы построить эти оценки необходимо знать дисперсию шумов, но она часто является неизвестной и необходимо построить ее оценку. Способы нахождения оценки дисперсии были предложены в работах (6) – (9).

Цель и задачи работы. Целью работы является построение процедуры обнаружения разладки авторегрессионного процесса $AR(p)$ с неизвестным параметром дисперсии шума и проверка адекватности ее работы. Для достижения этой цели необходимо выполнить следующий задачи:

1. Построить процедуру для обнаружения разладки случайного процесса
2. Построить процедуру оценки множителя неизвестного шума
3. Провести теоретические исследования свойств построенных процедур
4. Провести исследование работоспособности данного алгоритма на основе сгенерированных данных
5. Применить процедуру к реальным данным и посмотреть на эффективность ее работы.

1 Постановка задачи

Нами будем рассматриваться модифицированный трехэтапный алгоритм обнаружения разладки процесса $AR(p)$ с неизвестной дисперсией гауссовского шума. В статье (11) использовалась процедура последовательной оценки параметров. Мы модифицируем ее, добавив возможность оценки дисперсии неизвестного шума, основанного на алгоритме описанного в статье (10), но обобщенного для $AR(p)$.

На первом этапе мы строим последовательные оценки параметров авторегрессионной модели с использованием метода наименьших квадратов на фиксированных интервалах.

Далее на втором этапе на следующем интервале мы строим оценку множителя для того что бы компенсировать неизвестную дисперсию шума.

На третьем шаге мы получаем последовательность моментов времени $t_1 < t_2 < \dots$ в которые будут приниматься решения для обнаружение момента разладки. И далее определяем правила обнаружения на основе оценок параметров модели, полученных по наблюдениям на интервалах времени $(t[i-1], t[i]]$.

Рассматриваемый случайный процесс рекуррентного типа описывается уравнением

$$x_{k+1} = A_k \cdot \lambda + B\xi_{k+1}, \quad (1)$$

где A_k – матрица размерности $1 * m$ элементы которой зависят от реализации случайного процесса x до момента k включительно;

λ – вектор неизвестных параметров размерности $m * 1$;

B – параметр неизвестной дисперсии шумов; $\{\xi_k\}$, $k \geq 1$ последовательность независимых и одинаково распределенных случайных величин с математическим ожиданием равным нулю и $M \xi_k^2 = 1$

В некоторый неизвестный момент времени v вектор параметров λ меняет свое значение с μ_0 на μ_1 . Предполагается, что величина скачка удовлетворяет условию

$$(\mu_0 - \mu_1)^T (\mu_0 - \mu_1) \geq \Delta, \quad (2)$$

где Δ - фиксированная заданная величина.

Требуется по наблюдениям x_k обнаружить момент разладки v как можно скорее с наименьшим количеством ложных тревог.

2 Построение модели

2.1 Построение решающей процедуры

Опишем из чего будет состоять общая схема процедуры обнаружения разладки случайного процесса. Для начала рассмотрим вариант процедуры, где в (1) известно значение коэффициента дисперсии B . Так как нам неизвестны параметры процесса как до, так и после момента разладки ν , необходимо построить оценки неизвестных параметров на различных участках наблюдаемого процесса. Предлагается использовать метод построение оценок параметров, который был описан в (10).

Суть метода заключается в том, что сначала определяются интервалы наблюдения $[t_{i-1}+1, t_i]$, на каждом из которых вычисляется оценка неизвестных параметров заданного рекуррентного процесса x . На первом шаге положим $t_0 = -1$ и обозначим.

$$C(t_{i-1}, N) = \sum_{k=t_{i-1}}^N \nu_k A_k^T(x) A_k(x), \quad (3)$$

где $\nu_k = \nu_k(x)$ - неотрицательные весовые функции, принадлежащие промежутку $[0,1]$, которые определяются из условия

$$\frac{\nu_{\min}(N_1, k)}{B^2} = \sum_{l=N_1+\sigma}^k \nu_l^2 A_l(x) A_l^T(x), \quad (4)$$

где $\nu_{\min}(N_1, k)$ - минимальное собственное значение матрицы $C(N_1, k)$, где σ - минимальное число наблюдений, при котором матрица $C(N_1, N_1 + \sigma)$ невырождена. Весовые функции на интервале $[N_1, N_1 + \sigma]$ имеют вид

$$\nu_k = \begin{cases} \frac{1}{B\sqrt{A_k(x)A_k^T(x)}}, \\ 0, \end{cases} \quad (5)$$

Момент остановки определим по положительному параметру H из условия:

$$t = t(H) = \inf(N > N_1 : \nu_{\min}(N_1, N) \geq H). \quad (6)$$

Весовую функцию в момент времени t определим из следующих условий:

$$\begin{aligned} \frac{\nu_{\min}(N_1, t)}{B^2} &\geq \sum_{l=N_1+\sigma}^t \nu_l^2 A_l A_l^T, \\ \nu_{\min}(N_1, t) &= H. \end{aligned} \quad (7)$$

Тогда оценка вектора неизвестных параметров на каждом из интервалов имеет вид

$$\tilde{\lambda}(H) = C^{-1}(t, N) \sum_{k=N_1}^t \nu_k A_k^T(x) x_{k+1}, \quad (8)$$

$$C(t, N_1) = \sum_{k=N_1}^t \nu_k A_k^T A_k.$$

В (11) сформулирована и доказана следующая теорема:

Теорема 1. Если веса (5, 7) удовлетворяют условию

$$\sum_{l=0}^{\infty} \nu_l^2 A_l A_l^T = \infty \text{ п.н.}, \quad (9)$$

То момент остановки t (6) конечен почти наверное, и дисперсия оценки (8) ограничена сверху

$$M \|\tilde{\lambda}^*(H) - \lambda\| \leq \frac{H + m - 1}{H^2}. \quad (10)$$

Далее строится процедура, которая сравнивает значения оценок параметров на интервалах отличающихся на заданное натуральное число $l > 1$. Данный алгоритм применим для каждого из интервалов $[t_{i-1} + 1, t_i]$. В данной процедуре будем использовать последовательные оценки параметра λ , зависящего от положительного параметра H , которые состоят из пары $(t, \tilde{\lambda}^*) = (t(H), \tilde{\lambda}(H))$, где t - конечный почти наверное момент остановки, а $\tilde{\lambda}^*$ - оценка вектора параметров λ , построенная по наблюдениям на интервале $[1, t]$. Среднеквадратическое отклонение оценки от истинного значения параметров ограничено сверху величиной:

$$M \|\tilde{\lambda}^*(H) - \lambda\|^2 \leq \frac{P(H)}{H^2}, \quad (11)$$

где $P(H) = \frac{H + m - 1}{H^2}$.

Используя данную процедуру, построим серию последовательных оценок

$$(t_i, \tilde{\lambda}_i^*) = (t_i(H), \tilde{\lambda}_i(H)), \quad (12)$$

где $\{t_i\}$ - возрастающая последовательность моментов остановки, а $\tilde{\lambda}_i$ - гарантированная оценка вектора параметров, построенная на интервале $[t_{i-1} + 1, t_i]$ и обладающая свойством (11). С каждым таким интервалом свяжем решающую статистику вида

$$J_i = (\tilde{\lambda}_i - \tilde{\lambda}_{i-1})^T (\tilde{\lambda}_i - \tilde{\lambda}_{i-1}). \quad (13)$$

Обозначим через Δ минимальную величину сдвига параметра λ в момент разладки. Приведем теорему, доказанную в (11)

Теорема 2. Математическое ожидание решающих статистик J_i удовлетворяет неравенствам

$$M[J_i | t_i < \theta] \leq 4 \frac{P(H)}{H^2}, \quad (14)$$

$$M[J_i | t_{i-1} < \theta \leq t_i] \geq \Delta. \quad (15)$$

Отсюда вытекает алгоритм обнаружения момента разладки. Значение статистики J_i сравнивается с некоторым порогом δ , который удовлетворяет соотношению

$$4 \frac{P(H)}{H^2} < \delta < \Delta - 4 \sqrt{\Delta \frac{P(H)}{H^2}}. \quad (16)$$

Если значение статистики в какой-то момент превышает порог, принимается решение о наличии разладки.

Как ранее было замечено важными метриками качества процедуры являются вероятности ложной тревоги и ложного спокойствия. Пусть P_0 -вероятность ложной тревоги и P_1 -вероятность ложного спокойствия, тогда на каждом отрезке наблюдений $[t_{i-1}+1, t_i]$ их можно грубо оценить сверху по теореме доказанной в (11).

Теорема 3. Вероятности ложной тревоги P_0 и ложного спокойствия P_1 в каждом цикле наблюдений $[t_{i-1}+1, t_i]$ оцениваются сверху величинами

$$P_0 \leq 4 \frac{P(H)}{\delta H^2}, \quad P_1 \leq 4 \frac{P(H)}{(\sqrt{\Delta} - \sqrt{\delta})^2 H^2}. \quad (17)$$

Далее приведем асимптотические свойства решающих статистик. Рассмотрим поведение решающих статистик при больших значениях H . Из теоремы, доказанной в (11), если веса (5,7) удовлетворяю условию

$$\nu_l^k A_l(x) A_l^T(x) \leq \sqrt{H}, \quad (18)$$

то для оценки вектора неизвестных параметров $\tilde{\lambda}$ верно:

$$P\{\|\tilde{\lambda} - \lambda\|^2 > x\} \leq m(1 - 2\Phi(\sqrt{\frac{xH}{m}})), \quad (19)$$

где m – порядок модели авторегрессии,

$\Phi(\cdot)$ – функция стандартного нормального распределения.

Отсюда следует вероятности ложной тревоги и ложного спокойствия при достаточно больших H можно оценить, как:

$$P_0 \leq m(1 - 2\Phi(\sqrt{\frac{\delta H}{2m}})), \quad (20)$$

$$P_1 \leq m(1 - 2\Phi((\sqrt{\Delta} - \sqrt{\delta})\sqrt{\frac{H}{2m}})). \quad (21)$$

2.2 Построение оценки неизвестной дисперсии

Теперь рассмотрим случай, когда параметр дисперсии шума неизвестен. Построим трехэтапную процедуру оценки параметра λ в модели (1). На первом этапе мы оцениваем параметр авторегрессии с помощью метода наименьших квадратов. На втором этапе мы строим специальный множитель для компенсации неизвестной дисперсии шума. Построение такого множителя для авторегрессии 1 порядка было описано в работе (10). На третьем этапе мы применяем процедуру оценивания параметр авторегрессии описанную ранее.

Чтобы компенсировать неизвестную дисперсию шума, нам нужна базовая оценка неизвестных параметров авторегрессии λ , которую мы получим, используя обычную оценку по методу наименьших квадратов

$$\tilde{\lambda} = (A_k A_k^T)^{-1} A_k x_{k+1}, \quad (22)$$

На заданном интервале $[x_{N_0}, x_{N_0+s})$ значений авторегрессионного процесса, где s – размер промежутка, на котором строится оценка вектора параметров. Тогда компенсирующий коэффициент имеет следующий вид

$$\Gamma_l = C_f(l) \sum_{k=t+1}^{t+l} (x_{k+1} - \tilde{\lambda} A)^2, \quad (23)$$

где коэффициент $C_f(l)$ определяется функцией насыщенности шума

$$C_f(l) = E\left(\sum_{k=1}^l \varepsilon_k^2\right). \quad (24)$$

В случае гауссовского шума, как было доказано в работе (6), можно записать в виде $C_f(l) = 1/(l-2)$.

Следующее свойство, доказанное в работе (10), позволяет нам ограничить сверху математическое ожидание величины:

$$E \frac{1}{\Gamma_l} \leq \frac{1}{B^2} . \quad (25)$$

Тогда мы можем изменить нашу процедуру обнаружения разладки следующим образом

$$\frac{\nu_{\min}(N_1, t)}{\Gamma_{N_1}^2} \geq \sum_{l=N_1+\sigma}^t \nu_l^2 A_l A_l^T, \quad \nu_{\min}(N_1, t) = H , \quad (26)$$

$$t = t(H) = \inf(N > N_1 : \nu_{\min}(N_1, N) \geq H) . \quad (27)$$

3 Моделирование

3.1 Результаты моделирования на сгенерированных данных

Далее будут представлены исследования работоспособности процедуры обнаружения разладки с помощью программы написанной на языке python, реализующей предложенный алгоритм.

Рассмотрим процесс авторегрессии первого, второго и третьего порядка. В таблице 1 приведены параметры данных процессов, где N количество наблюдений случайного авторегрессионного процесса, μ_0 и μ_1 – векторы параметров процесса до и после разладки, ν – момент разладки, B – множитель, определяющий дисперсию шума, s – длина промежутка на котором строится оценка вектора параметров с помощью метода наименьших квадратов, L – длина промежутка на котором строиться оценка B . В качестве ε был взят гауссовский шум с математическим ожиданием 0 и дисперсией 1.

Таблица 1 – Характеристики процедуры и смоделированного процесса

μ_0	μ_1	N	ν	B	H	s	L
-0.5	0.4	20000	10000	1	30	20	20
(-0.2,0.1)	(0.3,-0.2)	20000	10000	1	30	20	20
(0.4, -0.1, 0.2)	(0.2, 0.2, -0.2)	20000	10000	1	30	20	20

Смоделированные данные в диапазонах от 9000-1100 представлены на рисунках 1 – 3.

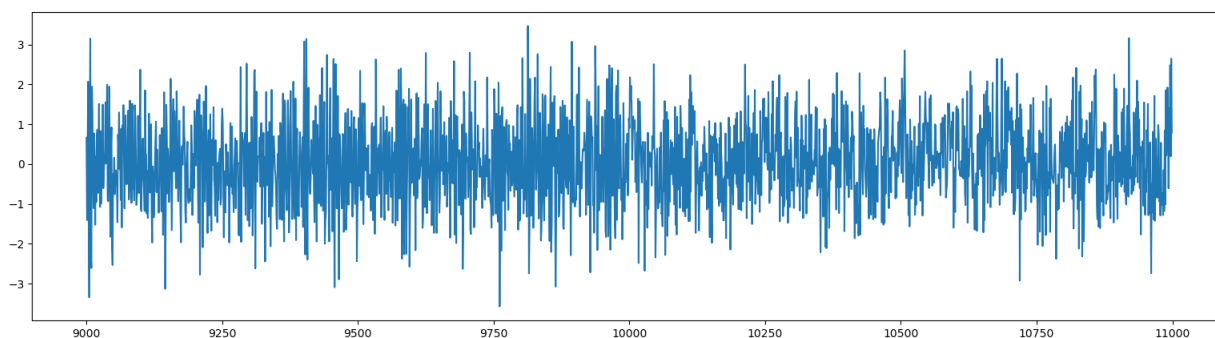


Рисунок 1 – Авторегрессионный процесс 1 порядка

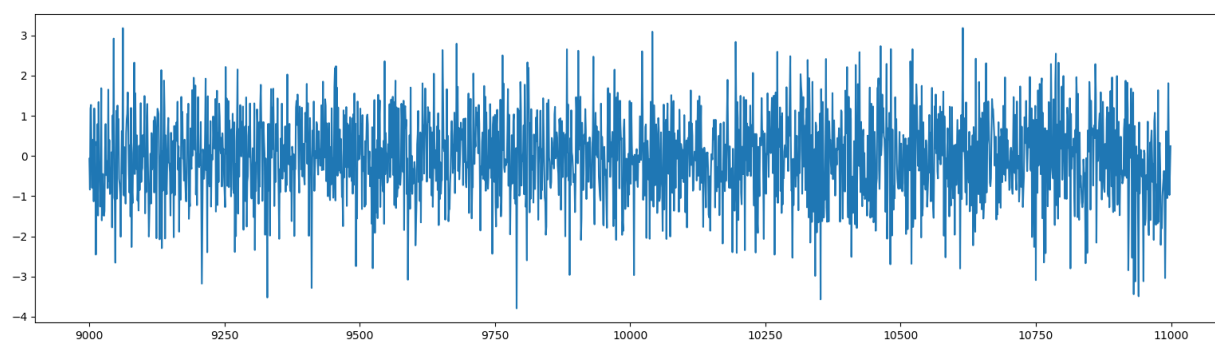


Рисунок 2 – Авторегрессионный процесс 2 порядка

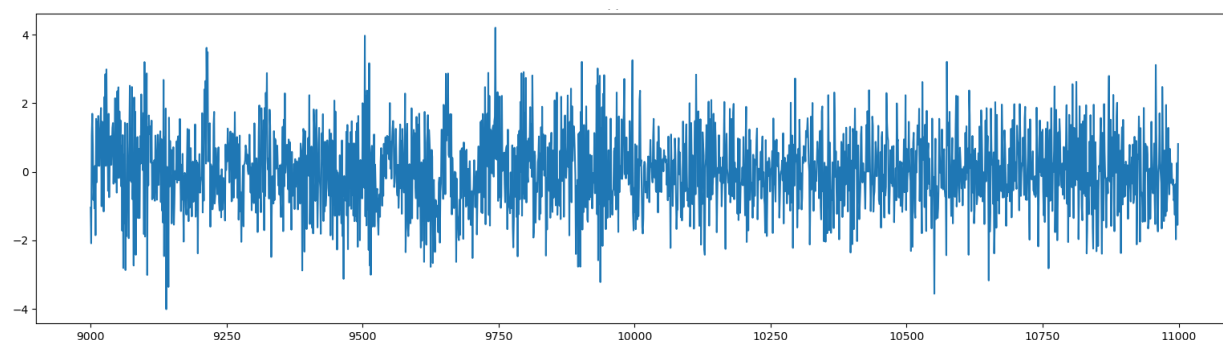


Рисунок 3 – Авторегрессионный процесс 3 порядка

По графикам процесса видно, как он меняет свои характеристики. К смоделированным данным применим модифицированную процедуру нахождения разладки случайного процесса. Построим оценку коэффициента неизвестной дисперсии с помощью описанного алгоритма, перед каждым шагом основной процедуры. На рисунках 4 – 6 изображены графики оценок этого коэффициента.

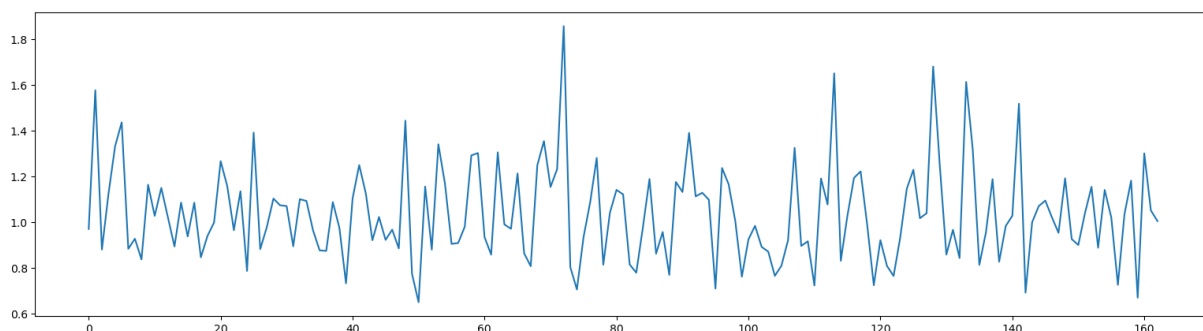


Рисунок 4 – Оценка дисперсии авторегрессии 1 порядка

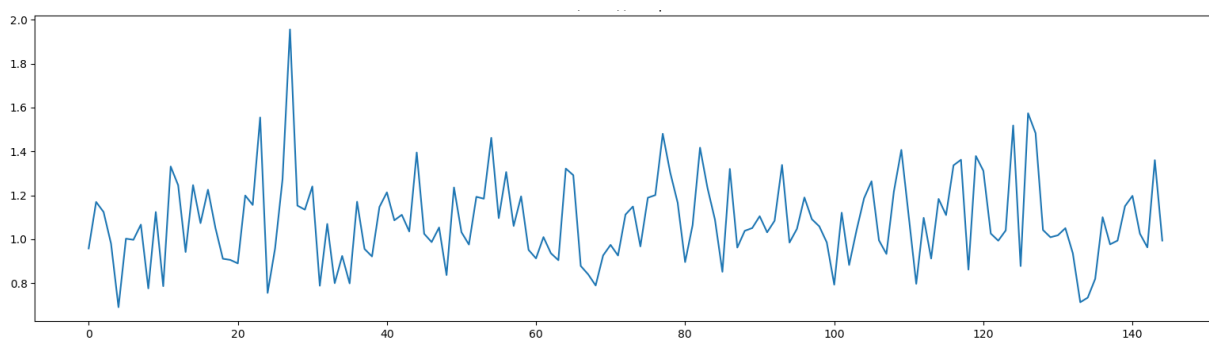


Рисунок 5 – Оценка дисперсии авторегрессии 2 порядка

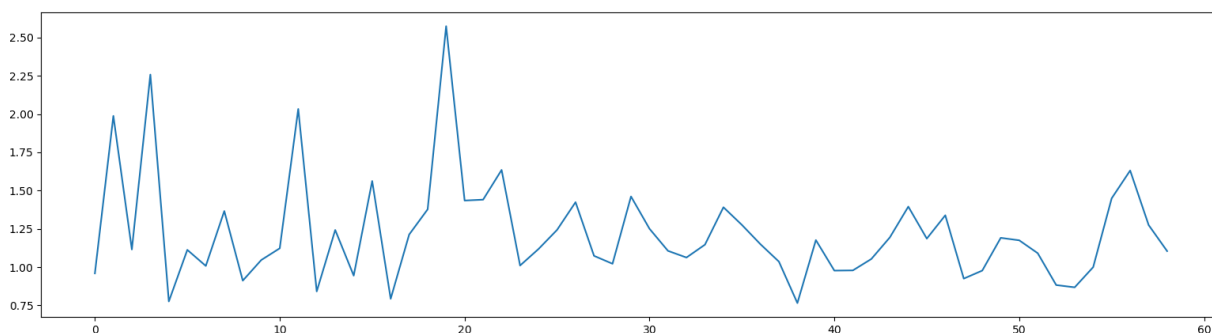


Рисунок 6 – Оценка дисперсии авторегрессии 3 порядка

Далее определяются границы интервалов наблюдения на каждом из которых вычисляется оценка неизвестных параметров. Теперь после нахождения весовых функций на каждом интервале находим оценку вектора неизвестных параметров авторегрессионного процесса. На рисунках 7 – 9 изображены графики изменения оценок каждого из параметров.

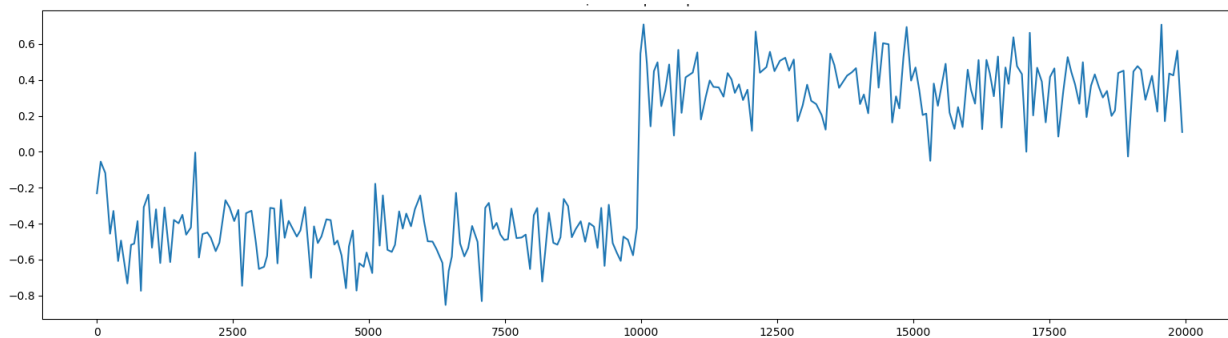


Рисунок 7 – Изменения оценки параметра для авторегрессии 1 порядка

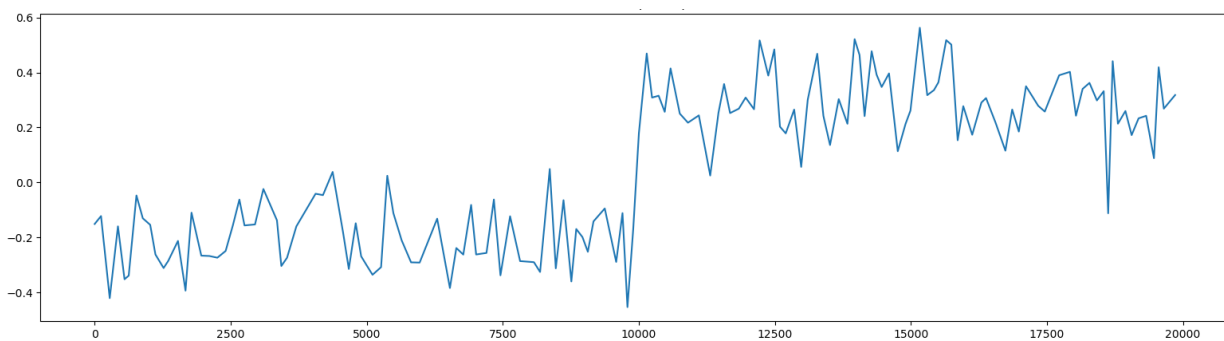
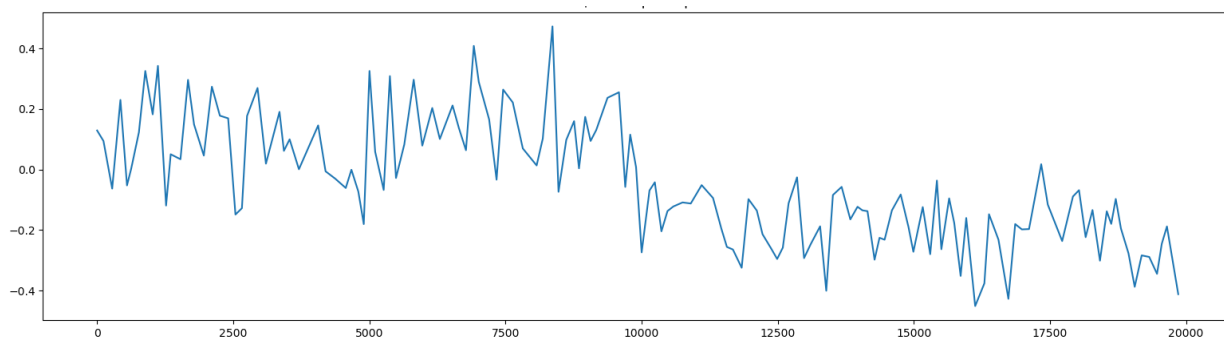


Рисунок 8 – Изменение оценок параметра для авторегрессии 2 порядка



Рисунок 9 – Изменение оценок параметра для авторегрессии 3 порядка

Как можно заметить при большем количестве параметров интервалы оценивания увеличиваются. Момент разладки четко выделяется на всех графиках.

Теперь найдем сам момент разладки используя процедуру при $l = 3$. На рисунках 10 – 12 изображены изменения решающей статистики.

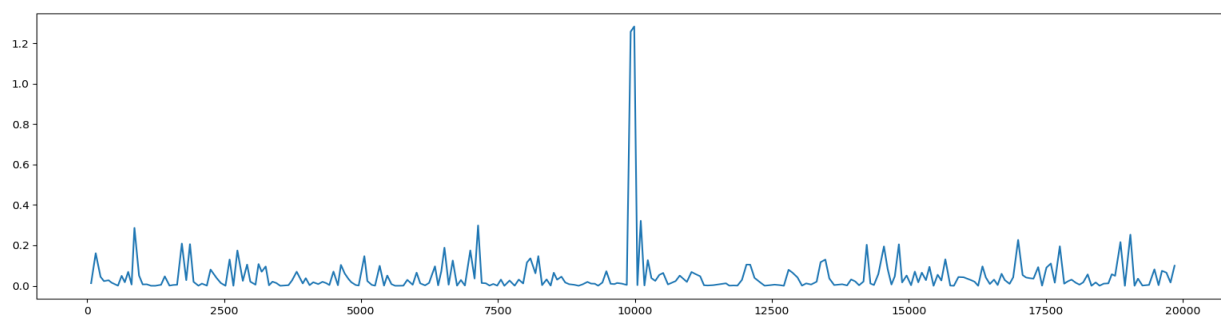


Рисунок 10 – Изменение решающей статистики авторегрессии 1 порядка

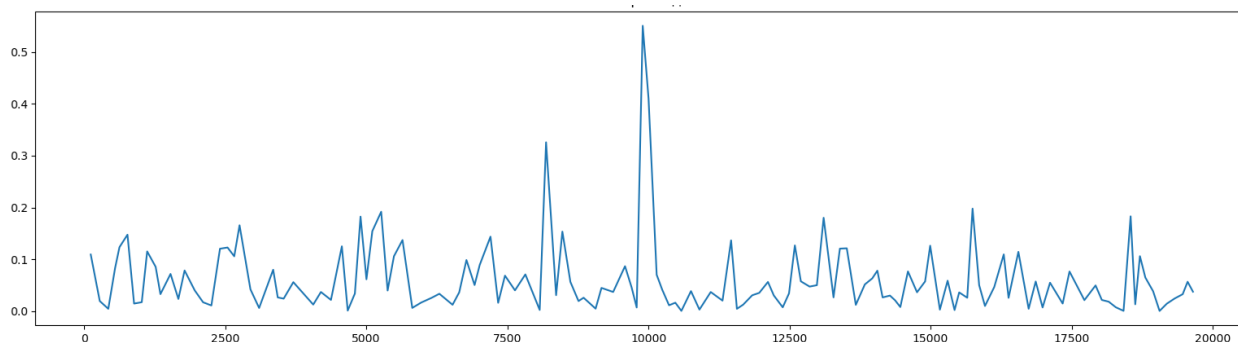


Рисунок 11 – Изменение решающей статистики авторегрессии 2 порядка

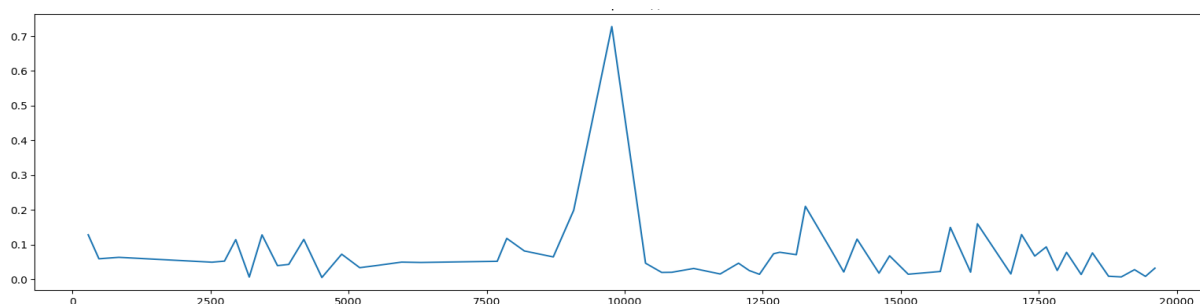


Рисунок 12 – Изменение решающей статистики авторегрессии 3 порядка

В качестве порога δ , по которому будем принимать решение о разладке, можно взять значение 0.4. Тогда можно вычислить выборочную вероятность ложной тревоги P_0 и выборочную вероятность ложного спокойствия P_1 . В таблице 2 приведены полученные результаты.

Таблица 2 – Выборочные вероятности

	1 порядка	2 порядка	3 порядка
P_0	0	0.09	0.003
P_1	0	0.12	0.01

Далее рассмотрим, как влияет значение N на нахождение разладки и построение оценок параметров, на примере авторегрессии 2 порядка с теми же параметрами. На рисунках 13–16 изображены изменения решающей статистики для разных значений N .

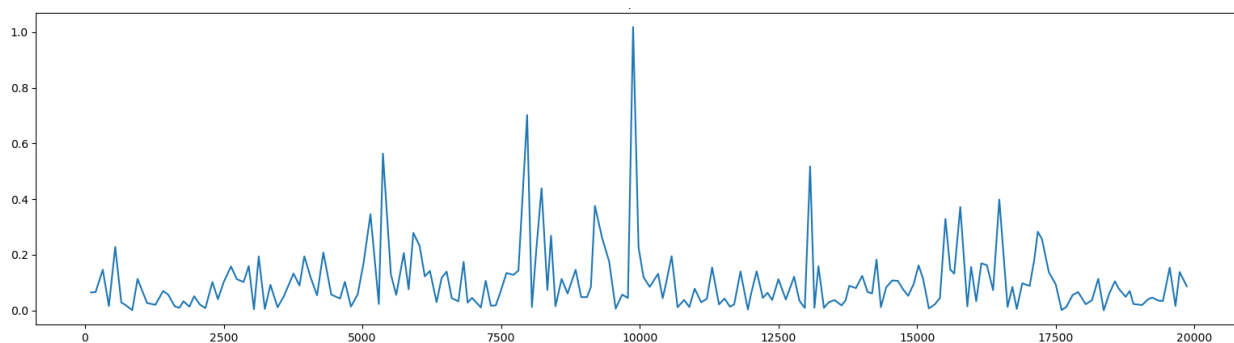


Рисунок 13 – Изменение решающей статистики при значении $N=20$

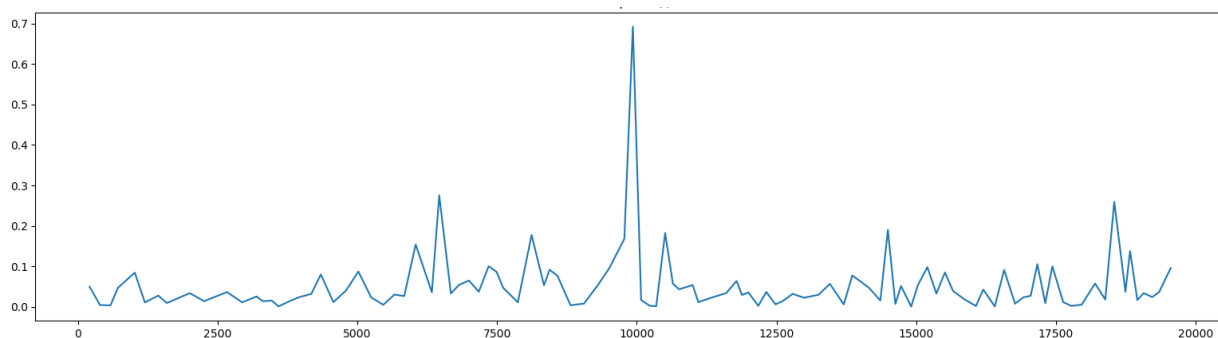


Рисунок 14 – Изменение решающей статистики при значении $N=50$

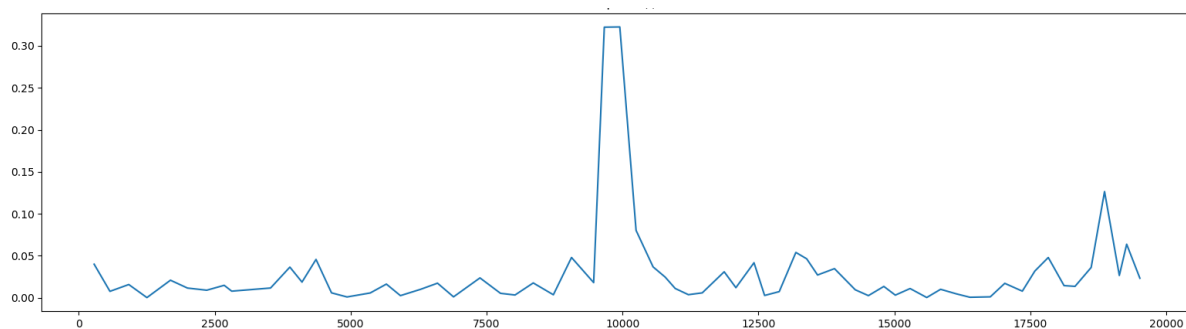


Рисунок 15 – Изменение решающей статистики при значении $N=80$

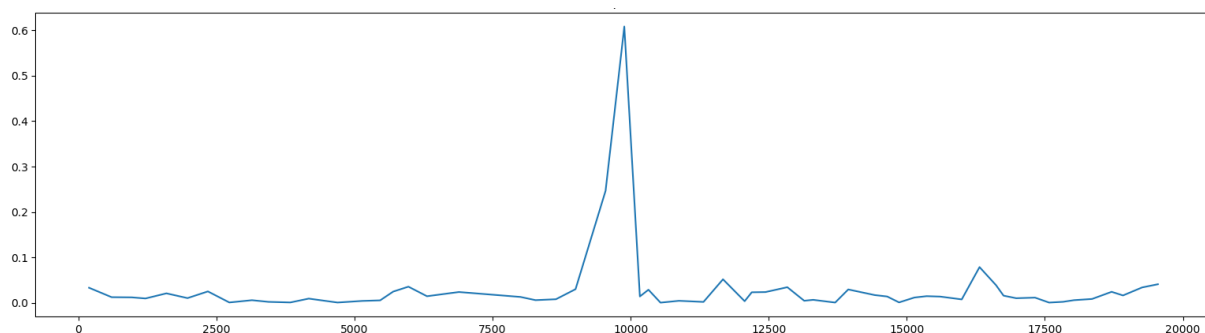


Рисунок 16 – Изменение решающей статистики при значении $N=100$

Из графиков видно, что при увеличении N точность обнаружения разладки становится выше, но интервалы становятся длиннее, что на практике может вызывать проблемы. Далее на рисунках 17 – 20 изображены изменения оценок параметров моделей.

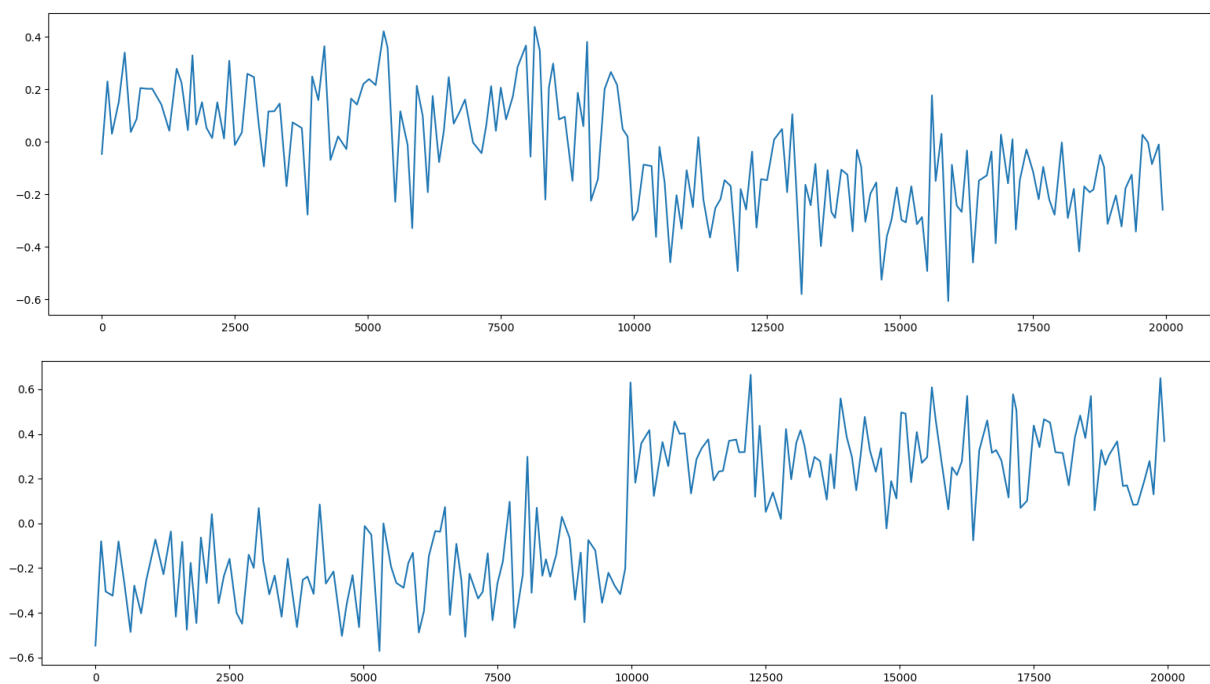


Рисунок 17 – Изменение оценок параметров при значении $N=20$

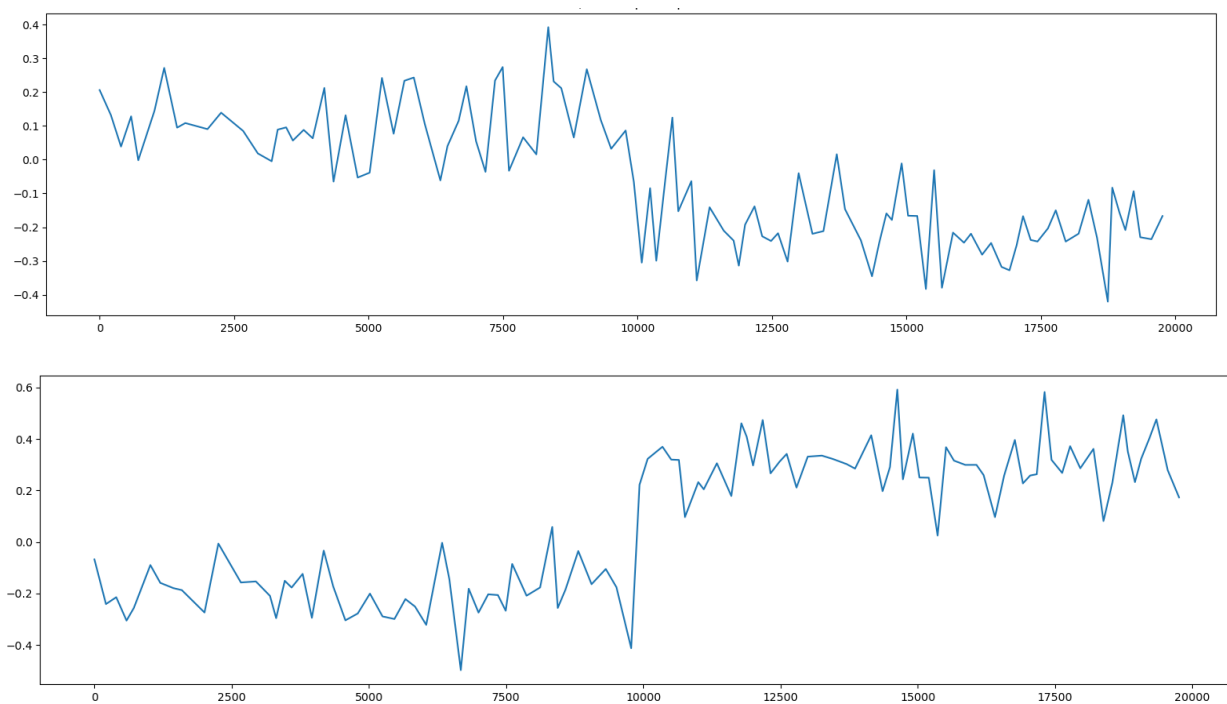


Рисунок 18 – Изменение оценок параметров при значении $H=50$

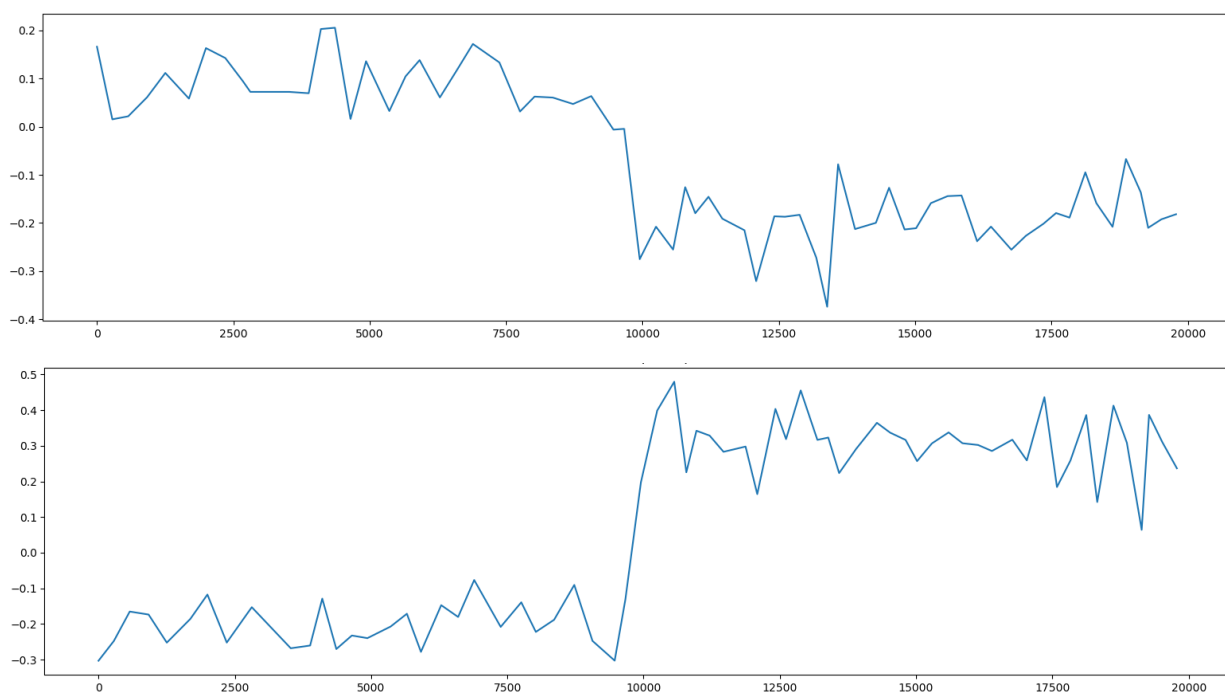


Рисунок 19 – Изменение оценок параметров при значении $H=80$

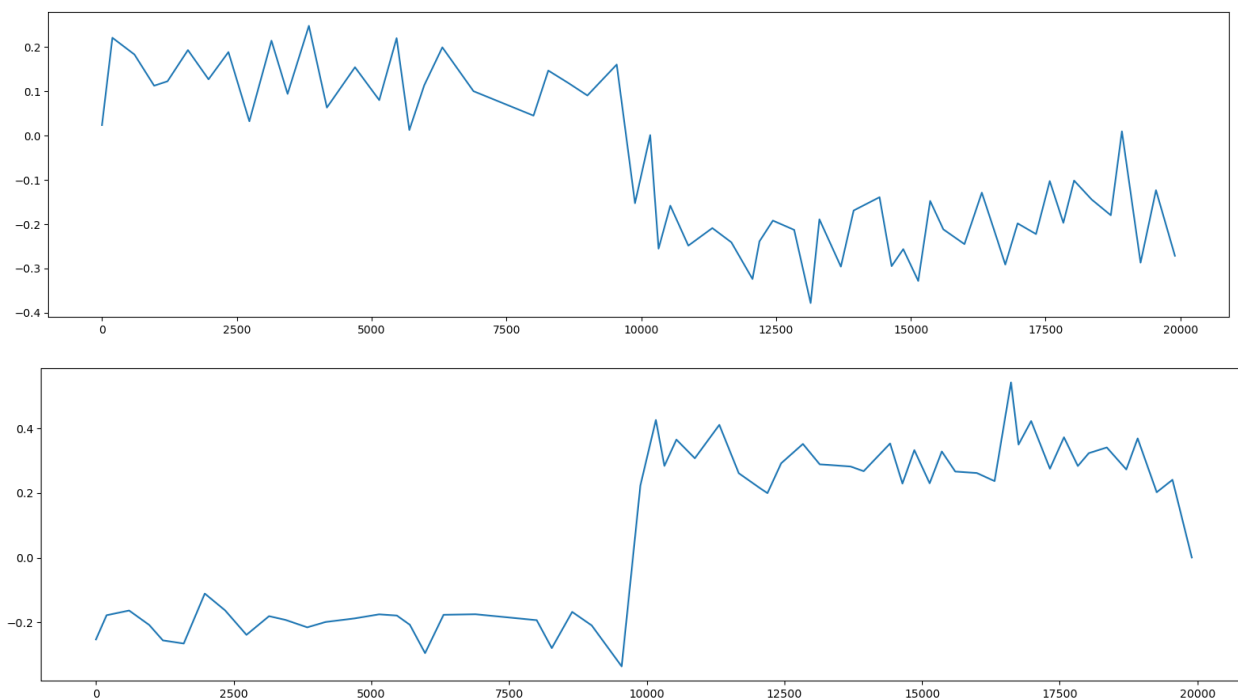


Рисунок 20 – Изменение оценок параметров при значении $H=100$

Далее на рисунках 21 – 24 представлены оценки параметра неизвестной дисперсии при различных шумах.

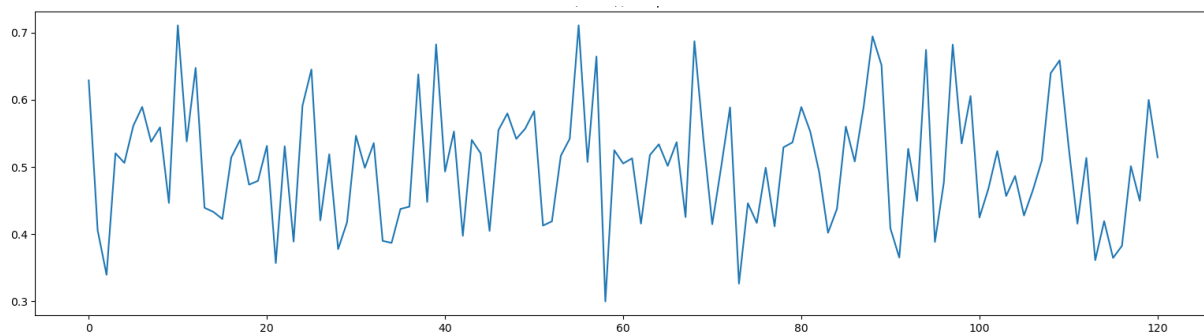


Рисунок 21 – Оценки параметра неизвестной дисперсии при различных шумах при $B=0.5$.

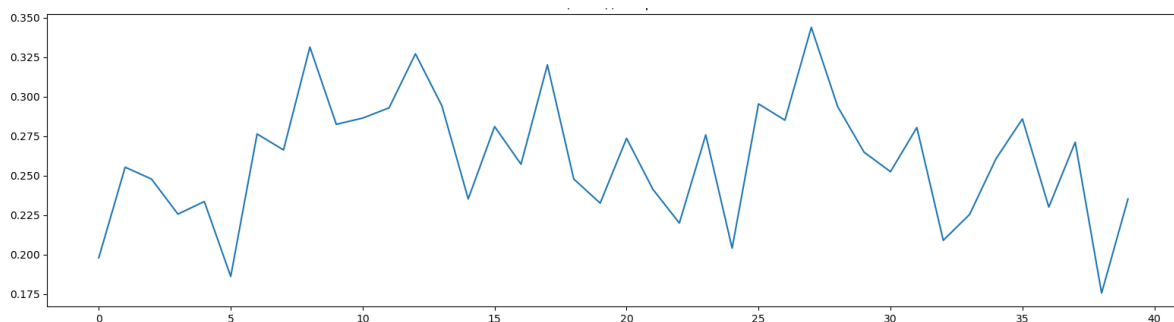


Рисунок 22 – Оценки параметра неизвестной дисперсии при различных шумах при $B=0.25$.

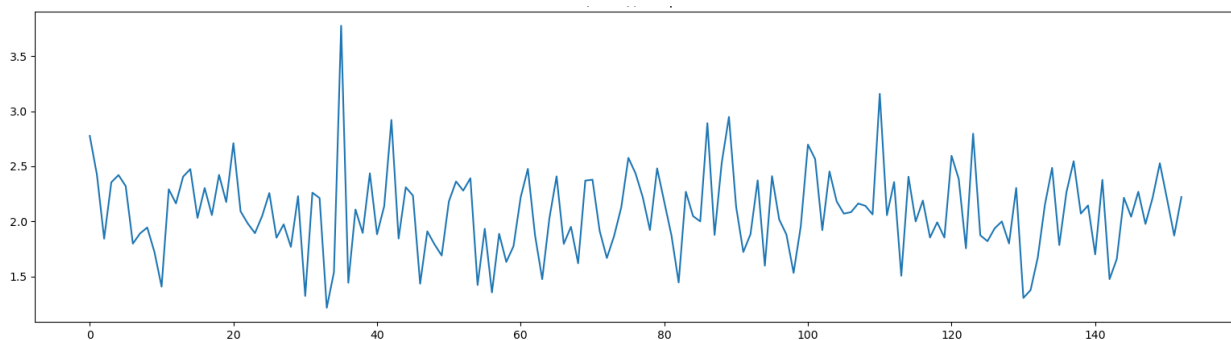


Рисунок 23 – Оценки параметра неизвестной дисперсии при различных шумах при $V=2$.

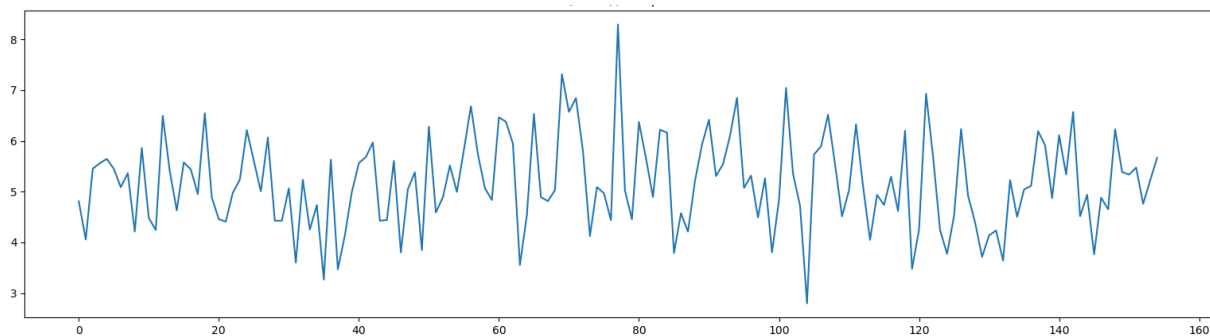


Рисунок 24 – Оценки параметра неизвестной дисперсии при различных шумах при $V=5$.

Из графиков видно, что оценка множителя неизвестной дисперсии хоть и имеет незначительные отклонения, но в среднем близка к теоретическому значению коэффициента.

3.2 Моделирование на реальных данных

Приведем пару примеров работы алгоритма на реальных данных. В качестве реальных данных были взяты котировки акций Яндекс в интервале с 4 июня 2014 года по 30 декабря 2021 года взятых с периодичностью в час, которые приведены на рисунке 25.

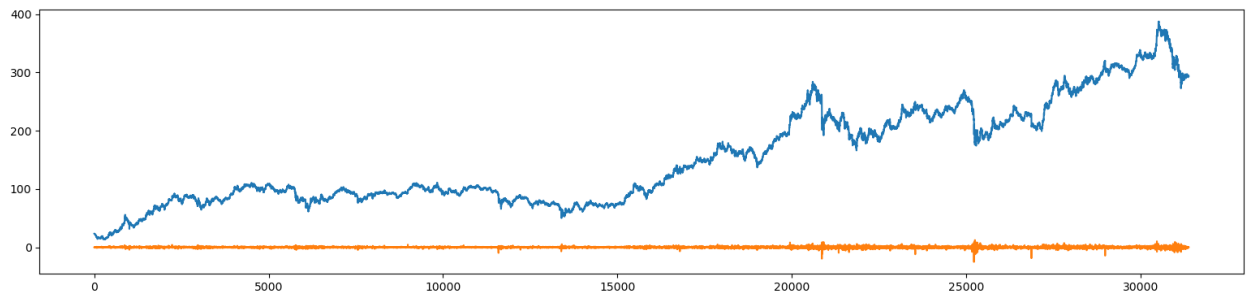


Рисунок 25 — Изменение котировок акций

Поскольку в модели предполагается, что среднее значение процесса равно нулю, преобразуем исходные данные следующим образом: $x_i = x_i - x_{i-1}$. Преобразованный процесс приведен на рисунке 26.

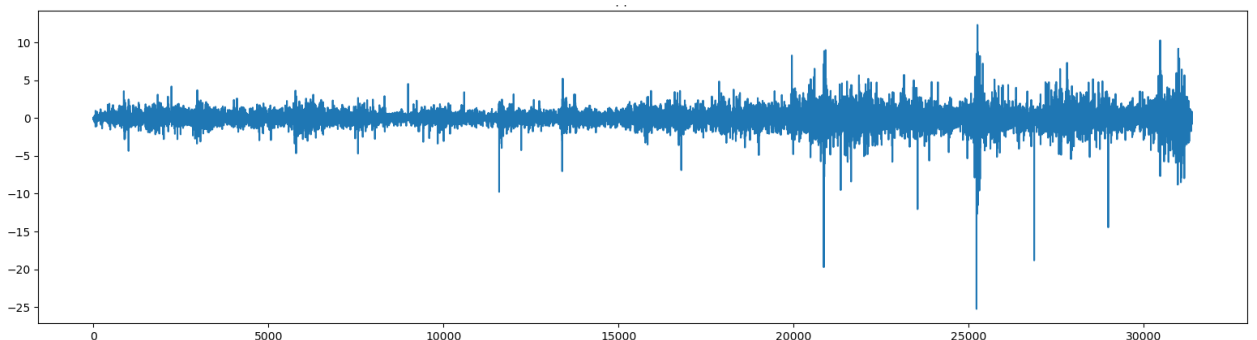


Рисунок 26 — Первые разности котировок акций Яндекс

Далее в ходе нескольких опытов были найдены характеристики, при которых процедура отработывает наиболее корректно. Параметры модели $AR(2)$, $N = 50$, $s = 20$, $L = 20$, $l = 1$. На рисунках 27 – 29 приведены найденные характеристики процесса.

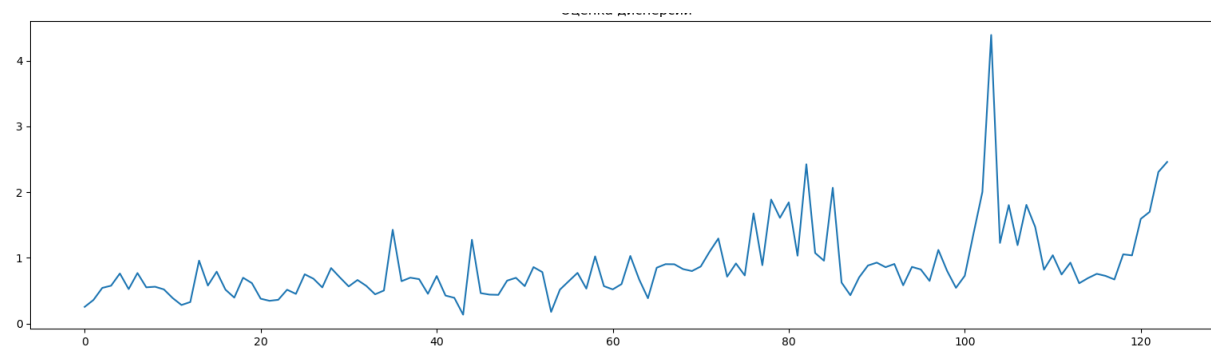


Рисунок 27 — Изменение оценок неизвестной дисперсии

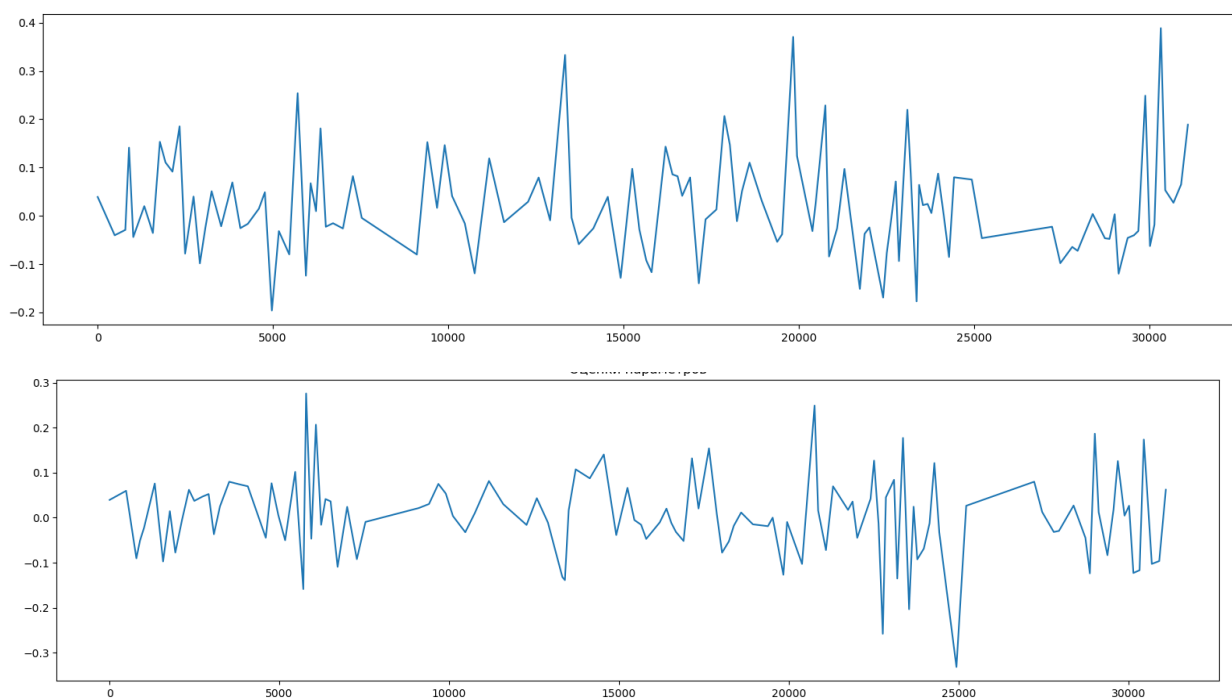


Рисунок 28 — Изменение оценок параметров

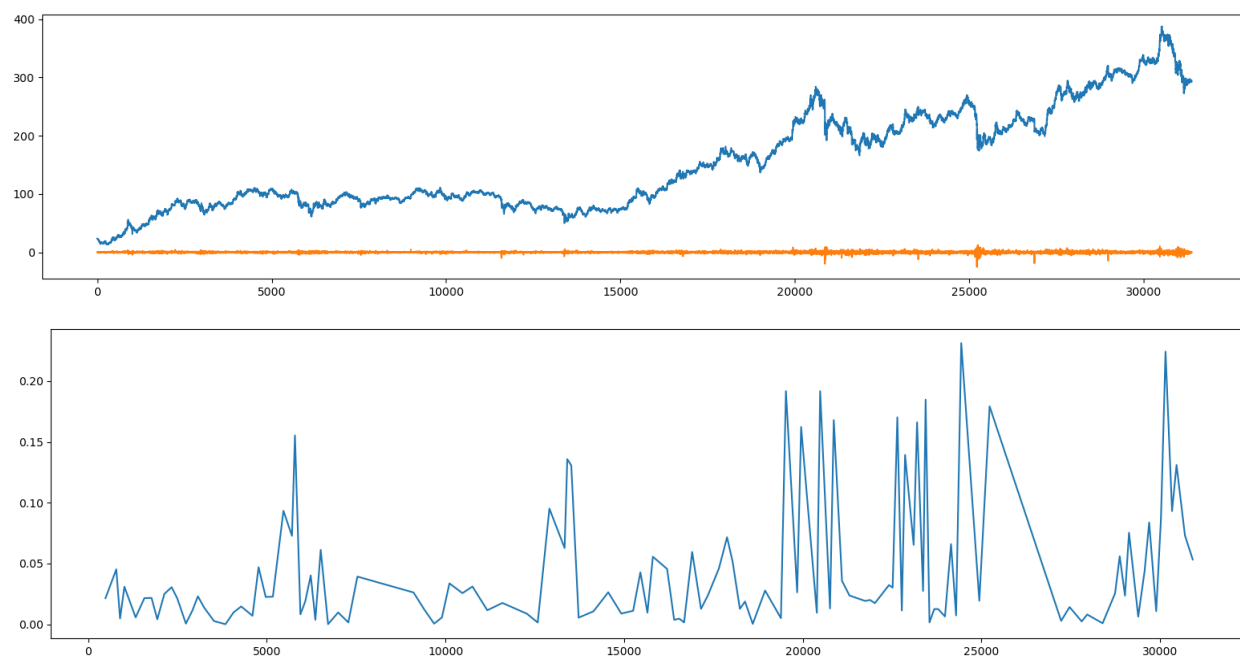


Рисунок 29 — Изменение решающей статистики при значении $N=50$

При значении порога δ , равного 0.12 будет обнаружено 13 точек разладки случайного процесса, большинство из которых соответствуют изменению трендов котировок

Далее возьмем другую компанию. Теперь в качестве реальных данных возьмем котировки акций Сбербанка в интервале с 5 января 2000 года по 30 декабря 2021 года взятых с периодичностью в час.

На рисунке 30 приведен график изменения стоимости котировок.

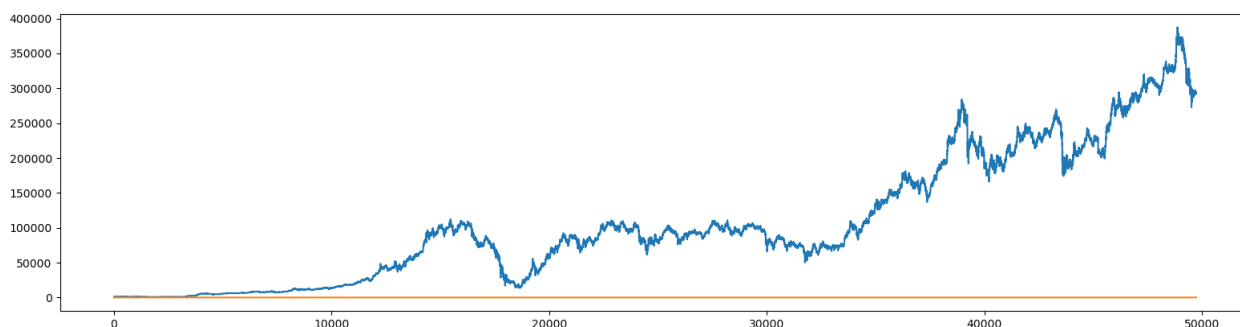


Рисунок 30 — Изменение котировок акций Сбербанка

Из-за аналогичных соображений, преобразуем исходные данные следующим образом: $x_i = \frac{(x_i - x_{i-1})}{x_{i-1}} * 100$. Преобразованный процесс приведен на рисунке 31.

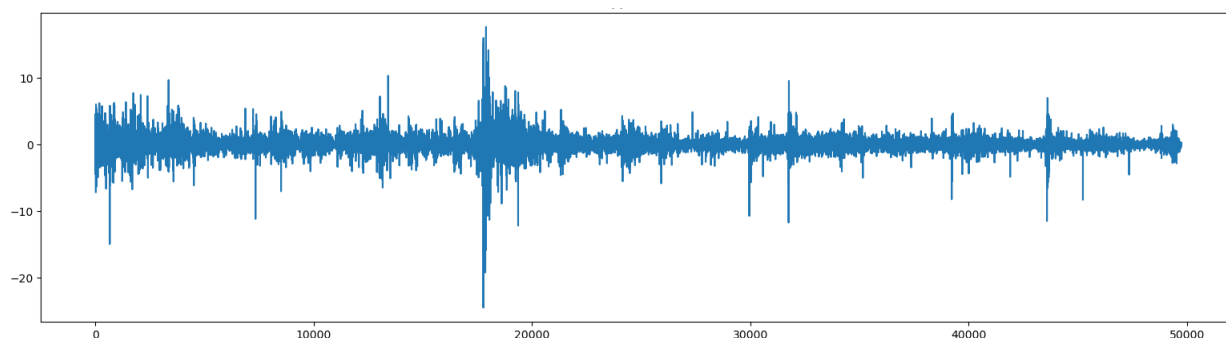


Рисунок 31 — Преобразованные исходные данные

В качестве параметров модели были выбраны следующие величины: AR(2), $H = 50$, $s = 20$, $L = 20$, $l = 1$.

Ниже на рисунке 32–33 приведен график изменения оценок коэффициента дисперсии и параметров модели.

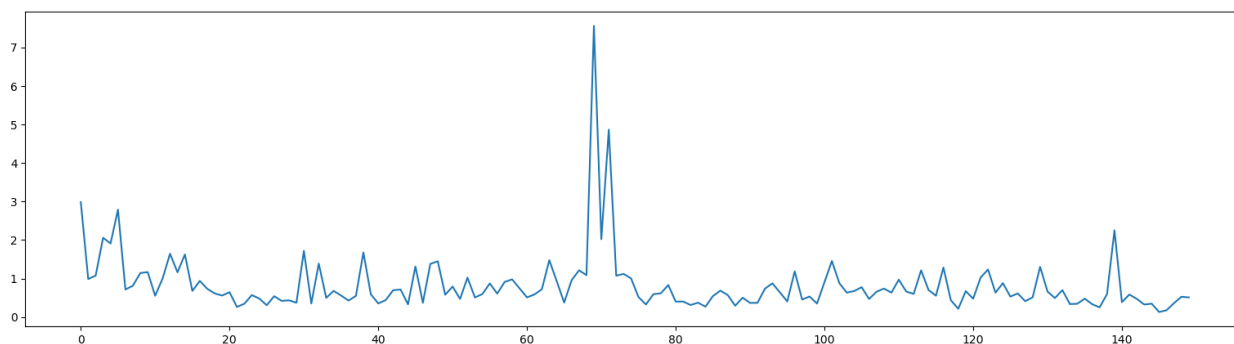


Рисунок 32 — Изменение оценок неизвестной дисперсии

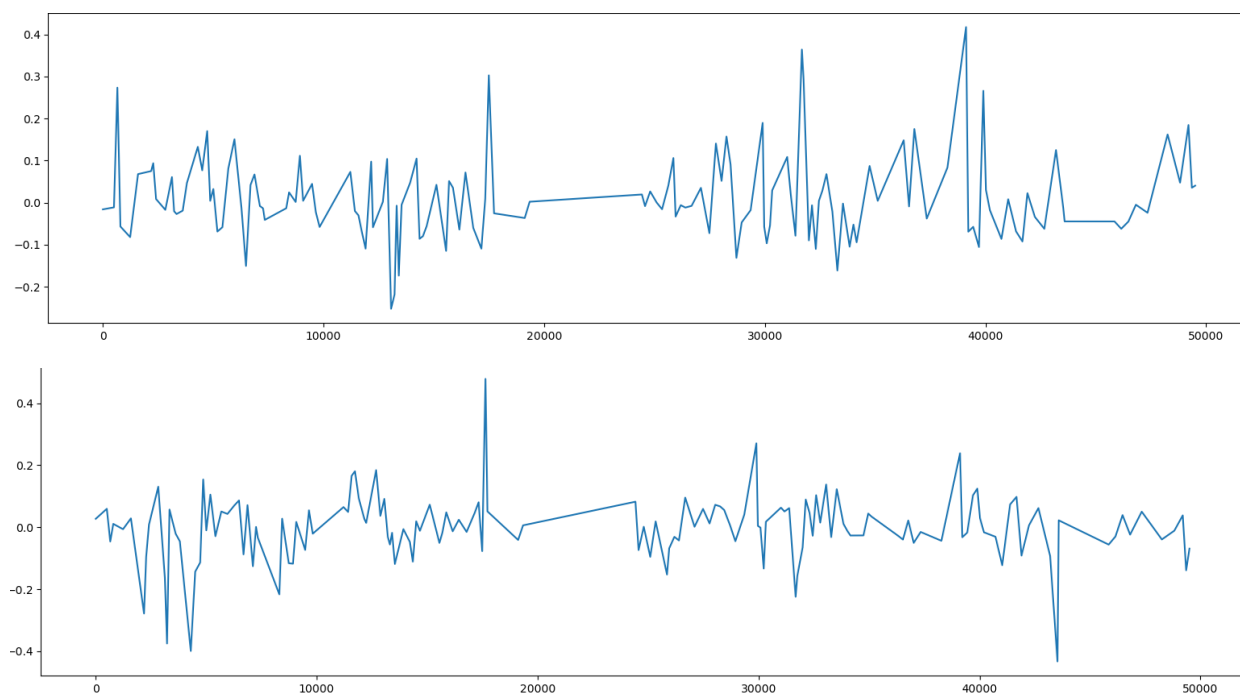


Рисунок 33 – Изменение оценок параметров модели

И наконец получим изменения решающей статистики, которые приведены на рисунке 34. Возьмем порог δ равного 0.12. При данном пороге обнаружено 19 точек разладки случайного процесса, которые соответствуют датам, приведенным в таблице 3, где J – значение решающей статистики, (t, T) – границы интервала на котором произошла разладка.

Можно увидеть, что алгоритм обнаружил все значительные изменения в поведении акций компании. Например, такие, как начало мирового финансового кризиса 2008 года, кризис 2014 года и начало пандемии 2020 года.

Как можно заметить многие интервалы достаточно широкие (до нескольких месяцев). При уменьшении N можно добиться более узких интервалов, но будет увеличено количество возможных ложных тревог.

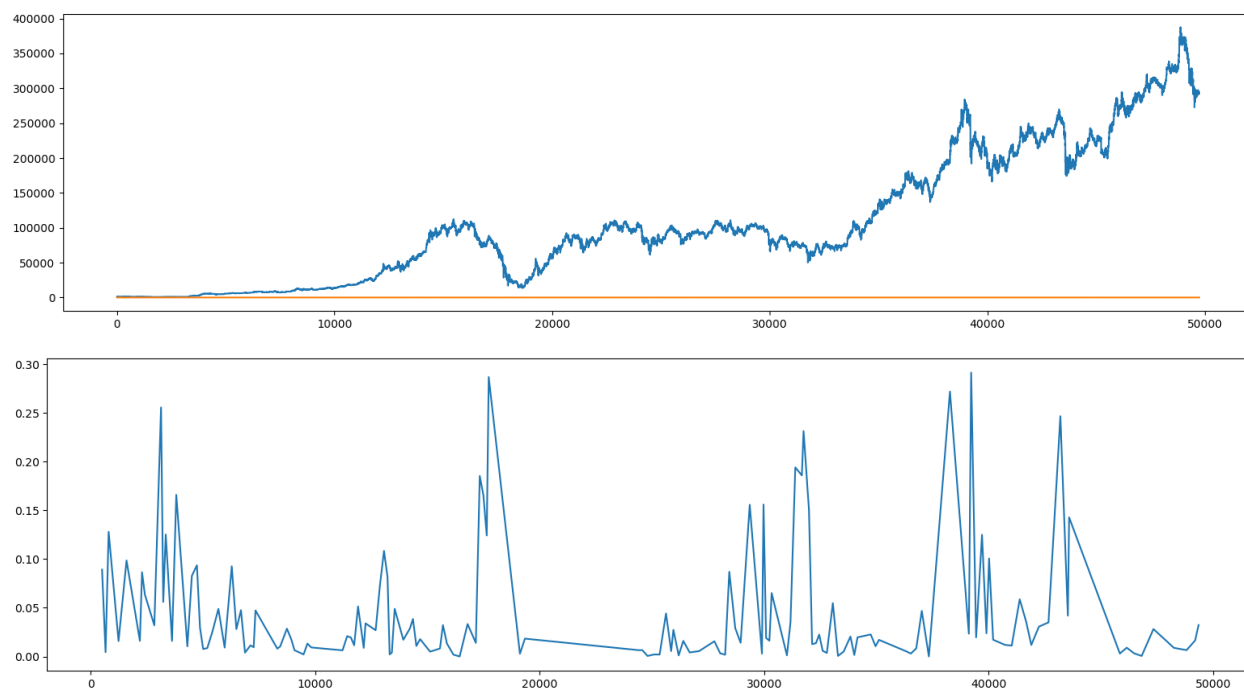


Рисунок 34 – Изменение решающей статистики

Таблица – 3 Даты, найденные с помощью решающей процедуры на исторических данных акций сбербанка

J	t	T
0.128026	2000.06.21	2000.09.18
0.25573932	2001.10.23	2001.11.13
0.12531761	2001.12.05	2002.02.06
0.16601948	2002.03.19	2002.06.24
0.18541894	2008.07.02	2008.07.30
0.16554485	2008.07.30	2008.08.25
0.12426203	2008.08.25	2008.09.09
0.28699575	2008.09.09	2009.05.13
0.15577099	2013.11.22	2014.02.21
0.15602754	2014.03.05	2014.03.25
0.19418692	2014.10.17	2014.12.04
0.18598462	2014.12.04	2014.12.16
0.23152735	2014.12.16	2015.01.28
0.15036394	2015.01.28	2015.02.19
0.2720719	2017.11.03	2018.03.22
0.29154403	2018.04.09	2018.05.15
0.12497863	2018.06.25	2018.07.25
0.24684305	2020.01.06	2020.02.28
0.14289304	2020.03.10	2020.12.04

ЗАКЛЮЧЕНИЕ

В данной работе рассмотрена процедура обнаружения момента разладки стохастического процесса авторегрессии $AR(p)$ с неизвестным параметром множителя дисперсии шума.

Разработана процедура обнаружения разладки случайного процесса и приведены ее свойства. Так же был приведен алгоритм построения множителя неизвестного шума, что позволила протестировать процедуру на реальных данных. Исследуемая процедура реализована в виде программного кода на языке программирования Python.

Эффективность разработанной процедуры, протестирована как на сгенерированных данных, где теоретические параметры были известны, так и на реальных данных котировок акций крупных российских компаний.

В качестве итогового вывода можно утверждать о работоспособности процедуры в ситуациях, когда дисперсия шума неизвестна.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Никифоров И. В. Последовательное обнаружение изменения свойств временных рядов // М.: Наука, –1983. – С. 200.
2. Tartakovsky A. Sequential analysis: hypothesis testing and changepoint detection / A. Tartakovsky, I. Nikiforov, M. Basseville. // Boca Raton: A CHAPMAN & HALL BOOK, – 2015. – С. 584.
3. Shiryaev A. N. Quickest detection problems: fifty years later // Sequential analysis. – 2010. – Vol. 29. – С. 345 – 385.
4. Москвичева М.Г. Анализ методов исследования процессов с разладками // Ученые записки УлГУ. Сер. Математика и информационные технологии. УлГУ. Электрон. журн. – 2019. – №1. – С. 70-77.
5. Вероятностно-статистические методы обнаружения спонтанно возникающих эффектов / А. Н. Колмогоров, Ю. В. Прохоров, А. Н. Ширяев // Тр. МИАН СССР. – 1988 – том 182. – С. 4–23.
6. On a Guaranteed Estimation of Autoregressive Parameters for an Unknown Variance of Noise / A. Dmitrienko, V. Konev // Automat. Remote Control. – 1994 – №55. – С. 218–228.
7. Novikov A. Sequential Estimation of Parameters of Processes of Diffusion Type. // Math. Notes – 1972 – №12. – С. 627–638.
8. Non-asymptotic Confidence Estimation of the Parameters in Stochastic Regression Model with Gaussian Noises / V. Konev, S. Vorobeychikov // Sequential Analysis – Design Methods and Applications– 1972 – №36 – С. 55–75.
9. Sriram T. Fixed Size Condence Region for Parameters of Threshold AR(1) Models. // J. Stat. Plan. and Inference – 2001 – С. 293–304.
10. Non-asymptotic Confidence Estimation of the Autoregressive Parameter in AR(1) Process with an Unknown Noise Variance/ S. Vorobeychikov, Y. Burkatovskaya // Austrian Journal of Statistics – 2000 – С. 19–26.
11. Обнаружение разладки стохастического процесса с неизвестными параметрами / С. Э. Воробейчиков, Ю. Б. Буркатовская // Automation and Remote Control – 2000 – С. 425–437.

ПРИЛОЖЕНИЕ А

Код имитационной модели

```
import numpy as np
```

```
import copy
```

```
class Model:
```

```
    def __init__(self, h, data, order=2):
```

```
        self.h = h
```

```
        self.order = order
```

```
        self.N = len(data)
```

```
        self.t = [0]
```

```
        self.r = []
```

```
        self.list_v = []
```

```
        self.L = []
```

```
        self.list_J = []
```

```
        self.X = data
```

```
        self.sigma = 1
```

```
        self.C_border = []
```

```
        self.indent = 0
```

```
        self.sigma_estimate_list = []
```

```
    def _MNK(self, N0, N1):
```

```
        A = np.array([self.X[N0 - i:N1 - i] for i in range(self.order, 2 * self.order)])
```

```
        A = A[::-1]
```

```

b = np.array(self.X[N0:N1]).reshape(-1, 1)

return np.dot(
    np.dot(
        np.linalg.inv(
            np.dot(A, A.transpose()))), A), b).transpose()[0][::-1]

def Estimate_var(self, N0, N1, N2):
    sum = 0
    mnk = self._MNK(N0, N1)

    for i in range(N1, N2 - 1):
        A = self.X[i - self.order: i]
        A = A[::-1]
        sum += (np.array(self.X[i + 1]) -
            np.dot(mnk, np.array(A).reshape(-1, 1))) ** 2

    var = (sum / (N2 - N1 - 1)) ** (1 / 2)
    return var[0]

def _C(self, i, v, C_sum):

    A = np.array(self.X[i - self.order:i]).reshape(-1, self.order)
    A = A[::-1]
    C_sum = C_sum + v * np.dot(A.transpose(), A)

    return min(np.linalg.eig(C_sum)[0]), C_sum

def _Sum_right(self, i, v, sum):

```

```

A = np.array(self.X[i - self.order:i]).reshape(-1, self.order)
A = A[::-1]
sum = sum + (v ** 2) * np.dot(A, A.transpose())[0]
#self._ = (A, v)
return sum

def _start_param(self, x0):
    list_v = []
    v_min = np.zeros((self.order, self.order))
    C_sum = 0
    if x0 < self.order:
        return [0] * self.order, 0, 0

    for i in range(x0 - self.order, x0):
        v = 0

        list_v.append(v)

        A = np.array(self.X[i - self.order:i]).reshape(-1, self.order)
        A = A[::-1]
        v_min += list_v[-1] * np.dot(A.transpose(), A)
        C_sum += (list_v[-1] ** 2) * np.dot(A, A.transpose())[0][0]
        #print(A)
    #print(list_v, v_min, C_sum)
    return list_v, v_min, C_sum

def V_slope(self, k=0, v_interval=0, v_min=None, C_sum=0):
    if v_min is None:
        v_interval, v_min, C_sum = self._start_param(k)

```

```

v_interval.append(0.5)

C_, mat = self._C(k, v_interval[-1], v_min)
Sum_ = self._Sum_right(k, v_interval[-1], C_sum)
p = 2
diff = Sum_ - C_ / (self.sigma ** 2)
while (abs(diff) > 1e-6):

    p = p * 2

    if diff < 0:
        v_interval[-1] = v_interval[-1] + 1 / p
    else:
        v_interval[-1] = v_interval[-1] - 1 / p
    if 1 / p < 1e-10:
        break

    C_, mat = self._C(k, v_interval[-1], v_min)
    Sum_ = self._Sum_right(k, v_interval[-1], C_sum)

    diff = Sum_ - C_ / (self.sigma ** 2)

v_min = mat
C_sum = Sum_
#print(v_interval[-1])
return v_interval, v_min, C_sum

def r_t(self, MNK_size, D_size, var=None):

    t_last_index = self.order

```



```

self.indent = MNK_size + D_size

while (self.N - self.order > t_last_index + MNK_size + D_size + 2):

    k0 = t_last_index
    if var is None:
        self.sigma = self.Estimate_var(k0 + 2, k0 + MNK_size + 2, k0 +
MNK_size + D_size + 2)
        self.sigma_estimate_list.append(self.sigma)
    else:
        self.sigma = var

    k0 = k0 + MNK_size + D_size
    k_last = k0 + 1
    v, S, C_sum = self.V_slove(k=k0)

    while min(np.linalg.eig(S)[0]) < self.h and k_last < self.N - self.order:
        v, S, C_sum = self.V_slove(k_last, v, S, C_sum)
        k_last = k_last + 1
        #print(self.list_v, '\n', S, C_sum)
        self.r.append(min(np.linalg.eig(S)[0]))

    self.list_v.append(v)
    self.C_border.append(S)
    #print(v[-1])

    t_last_index = k_last
    self.t.append(t_last_index - 1)

```

```

def Lambd(self):

    for i in range(len(self.t) - 1):
        c = np.linalg.inv(self.C_border[i])
        S = np.zeros((self.order, 1))

        for j in range(len(self.list_v[i])):

            T = self.t[i] + j + self.indent - 1
            A = np.array(self.X[T - self.order : T]).reshape(-1, self.order)
            A = A[::-1]
            S += A.transpose() * self.list_v[i][j] * self.X[T+1]

        self.L.append(np.dot(c, S))

def J(self,offset):
    self.offset = offset
    for l in range(len(self.L) - offset):
        k = (self.L[l+offset] - self.L[l])

        self.list_J.append(np.dot(k.transpose(), k))

import matplotlib.pyplot as plt
import numpy as np
class Plot():
    def __init__(self, model):
        self.model = model

    def time_step(self):

```

```

plt.figure(figsize=(20,5))
plt.scatter(self.model.t, np.zeros(len(self.model.t)))
plt.title("Отрезки времени")
def sum_step(self):
    plt.figure(figsize=(20,10))
    plt.step(np.array(range(len(self.model.r))),self.model.r)
    plt.title("лесенка сумм")

def estimate_params(self):
    for i in range(self.model.order):
        plt.figure(figsize=(20, 5))
        plt.plot(self.model.t[:-1],np.array(self.model.L).reshape(-
1,self.model.order)[:i,i])
        plt.title("Оценки параметров")

def data(self, diff = False):
    plt.figure(figsize=(20, 5))
    if diff:
        plt.plot(list(range(9000,11000)),self.model.X[9000:11000])
    else:
        plt.plot(self.model.X)
    plt.title("Данные")
def change_detection(self):
    plt.figure(figsize=(20, 5))
    plt.plot(self.model.t[1:-
self.model.offset],np.array(self.model.list_J).reshape(-1,1))
    plt.title("Точки разладки")
def estimate_var(self):
    plt.figure(figsize=(20, 5))
    plt.plot(np.array(self.model.sigma_estimate_list).reshape(-1,1))

```

```
plt.title("Оценка дисперсии")  
def plot(self):  
    self.data()  
    self.time_step()  
    self.sum_step()  
    self.estimate_var()  
    self.estimate_params()  
    self.change_detection()
```

Уважаемые пользователи! С 4 по 25 июня 2022 года проверки на тарифе Free могут отключаться в период высокой нагрузки. Это временная мера для работы сервиса. К сожалению, мы не можем сказать точных сроков восстановления, проверки на тарифе Free будут включены при снижении нагрузки.

ТАРИФ

Free

ИЗМЕНИТЬ

БАЛЛЫ

0

ПОЛЬЗОВАТЕЛЬ

konstantinlihota1@gmail.com

ПРОВЕРИТЬ ДОКУМЕНТ

ГЛАВНАЯ / КАБИНЕТ / РЕЗУЛЬТАТЫ ПРОВЕРКИ / КРАТКИЙ ОТЧЕТ

Краткий отчет

Получить полный отчет

- ПАРАМЕТРЫ ПРОВЕРКИ
- ЭКСПОРТ
- ИСТОРИЯ ОТЧЕТОВ
- ВЫЙТИ В КАБИНЕТ
- ЕЩЕ...

Диплом_Лихота_Константин_финал

ПРОВЕРЕНО: 04.06.2022 11:16:46

№	Доля в отчете	Источник	Актуальна на	Модуль поиска
[01]	1,49%	http://vital.lib.tsu.ru/vital/access/services/Download/vital:...	07 Сен 2020	Интернет Free
[02]	0,63%	Обнаружение локальных свойств анализируемых сигн...	20 Сен 2019	Интернет Free
[03]	0%	Модифицированный алгоритм обнаружения разладки ...	01 Мая 2020	Интернет Free

- ЗАИМСТ
- 2,62%
- САМОЦ
- 0%
- ЦИТИР
- 0%
- ОРИГИ
- 97,38%
- ИСТОЧ
- ЕЩЕ НА
- ИСТОЧ
- ЗАИМСТ