Диффузионная модель со скачками с возвращающейся к среднему логнормальной волатильностью

Р. В. Честнов[1]

На данный момент существует множество стохастических моделей, построенных для различных концепций рынка. Однако, практически все подобные модели основаны и протестированы для традиционных рынков, в то время как в настоящее время рынок криптоактивов набирает колоссальные обороты по объемам и капитализации рынка. По данной причине напрашивается идея о создании такой модели, которая была бы построена специально под рынок криптоактивов с учетом всех ее особенностей и моделей поведения. В данной работе мы попробуем посмотреть на некоторые признаки, которые заметны для криптоактивов, и построить стохастическую модель, учитывающую их, после чего сравнить её с другой похожей по структуре моделью. Сразу стоит отметить, что по аналогии с тем, как используются традиционные модели для крипторынков, наша модель для крипторынков так же будет хорошо интерпретировать традиционный рынок. Более того, даже лучше, чем в обратном случае, так как основная идея моей модели заключается в учитывании различных критических событий, происходящих с тем или иным активом, но калибруя модель нужным образом, мы можем их не учитывать.

Ключевые слова: Финансовая математика, стохастическое исчисление, криптоактивы, нейронные сети, уравнение Фейнмана-Каца, модель Бэйтса, функция правдоподобия.

1. Вступление

Текущие реалии финансовых рынков находится в периоде активных преобразований, ведь новые и старые финансовые инструменты сталкиваются на фоне динамичной рыночной обстановки. Традиционные активы, такие как акции и облигации, на протяжении многих лет привлекали внимание финансовых аналитиков и моделей прогнозирования. Однако, в последние десятилетия мы стали свидетелями стремительного развития нового класса активов — криптовалют и криптоактивов. Это привносит важные вызовы в разработке новых финансовых моделей,

[1] *Честнов Роберт Валентинович* — количественный исследователь по DeFi продуктам в Qset, выпускник каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: rvchestnov@gmail.com.

Chestnov Robert Valentinovich — quantitative researcher on DeFi products in Qset, graduate student of Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intellectual Systems.

способных адаптироваться к уникальным особенностям и рискам, связанным с крипторынками. Одной из ключевых особенностей является высокая волатильность и нестабильность цен, что может привести к резким колебаниям. Этот фактор создает повышенные риски как для инвесторов, так и для аналитиков, и подчеркивает важность разработки специализированных моделей, способных учитывать эту динамику цен.

Кроме того, криптовалютные рынки отличаются не только волатильностью, но и своим характером торговли, структурой участников и воздействием различных факторов на динамику цены, таких как новости о регулировании, технические обновления и события в сообществе, что отчасти объясняет такую непредсказуемость в поведении криптоактивов. На данный момент невозможно смоделировать описанные внешние социальные факторы, но в данной работе будет применен классический подход к данной проблеме - отделение подобных деталей от рыночной составляющей цены путем добавления соответствующие параметры в модель.

В предложенной стохастической модели планируется учесть несколько особенностей динамики цен криптоактивов, которые существенно отличаются от традиционных финансовых инструментов. Одной из таких особенностей является ненормальное распределение волатильности, которая характеризуется более высокой вероятностью крупных колебаний цен и резких движений на рынке. В отличие от нормального распределения, которое используется в большинстве стохастических моделей, логнормальное распределение лучше отражает действительность на крипторынках, где наблюдается большое количество экстремальных событий и высокая волатильность. Кроме того, еще одной важной особенностью криптовалютных рынков является наличие скачков цен, то есть внезапных и резких изменений цены, которые могут происходить в результате крупных событий. Учет таких скачков в стохастической модели играет ключевую роль в повышении ее точности и надежности, а также в обеспечении более реалистичного прогнозирования ценовых траекторий на крипторынках. Таким образом, разработка стохастической модели, учитывающая следующие два фактора, представляет собой значимый шаг в совершенствовании аналитических инструментов для криптовалютных рынков. Это позволит не только более точно оценивать риски и возможности инвестиций в криптоактивы, но и развивать более эффективные стратегии управления портфелем в условиях высокой волатильности и нестабильности рыночной среды.

Работа организована по разделам следующим образом: в разделе 2 мы подробнее посмотрим на некоторые характеристики криптоактивов и подробнее распишем процесс создания стохастической модели и идею, которая за ней стоит; в разделе 3 расписана методология вычисления

кумулятивной функции распределения для модели; в разделе 4 мы займемся численным решением уравнения, полученного в предыдущей главе; в разделе 5 нашей задачей будет построение логарифмической функции правдоподобия для калибровки коэффициентов модели и сравнение полученных результатов с моделью Бэйтса. В разделе 6 будут приведены выводы и предложения по доработке работы.

2. Идея создания модели и её описание

Как было упомянуто в введении, первоначальной целью является выделение некоторых особенности поведения криптоактивов, которые нехарактерны для существующих стохастических моделей. Для начала стоит отметить, что на практике и даже в теории в силу сложности стохастического анализа некоторые детали опускаются и не вносятся в модели. К примеру, самая популярная стохастическая модель для описания поведения цены – модель Геометрического Броуновского Движения (GBM) – предполагает, что доходность актива имеет нормальное распределение и что изменения цен независимы и стационарны. GBM широко используется в финансовой математике и анализе рынков для моделирования ценовых процессов. Модель описывается следующим стохастическим уравнением:

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \tag{0}$$

или, переходя к логарифму цены,

$$dX_t = \mu dt + \sigma dW_t; \quad X_t = \ln(S_t) \tag{1}$$

где S_t – цена актива, X_t – лог-цена актива, μ – тренд, σ – волатильность, dW_t – инкремент броуновского движения. Данная модель является наиболее распространённой благодаря удобству использования и простоте реализации, однако это также её главный недостаток. GBM упоминается в данном контексте, чтобы подчеркнуть, что в математической интерпретации жизненных процессов часто приходится жертвовать точностью ради простоты реализации. В последующих исследованиях учёные совершенствовали существующие модели. Среди наиболее распространённых подходов можно выделить преобразование тренда или волатильности из постоянных параметров в параметры, зависящие от времени, выделение различных свойств в динамике активов (например, возврат к среднему, как в модели Орнштайна-Уленбека), адаптацию моделей под соответствующие финансовые инструменты (например, модель Блэка-Шоулза), а также изменение динамики броуновского движения (например, модель дробного геометрического

броуновского движения). Основная цель настоящей работы заключается в создании альтернативной стохастической модели, использующей как известные методы, так и новые подходы.

Для исследования были выбраны 16 криптовалютных токенов, основываясь на рыночной капитализации, торговом объёме и популярности соответствующих блокчейнов и протоколов: BTC, ETH, LINK, MATIC, UNI, MKR, LDO, AAVE, QNT, MANA, CRV, 1INCH, ZRX, FXS, SUSHI и YFI. Очевидно, что при создании модели для крипторынков в первую очередь необходимо учесть их непредсказуемость и значительные колебания в динамике цен. Поэтому волатильность должна быть представлена как величина, зависящая от времени, и выделена в отдельный стохастический процесс, который будет моделироваться соответствующим образом.

Для начала рассмотрим минутные данные за один день по данным токенам и проведём базовый анализ. На практике, без использования сложных инструментов, волатильность часто аппроксимируют через стандартное отклонение по скользящему окну. Применим этот подход для оценки динамики волатильности на концептуальном уровне. Представим графики полученных волатильностей:

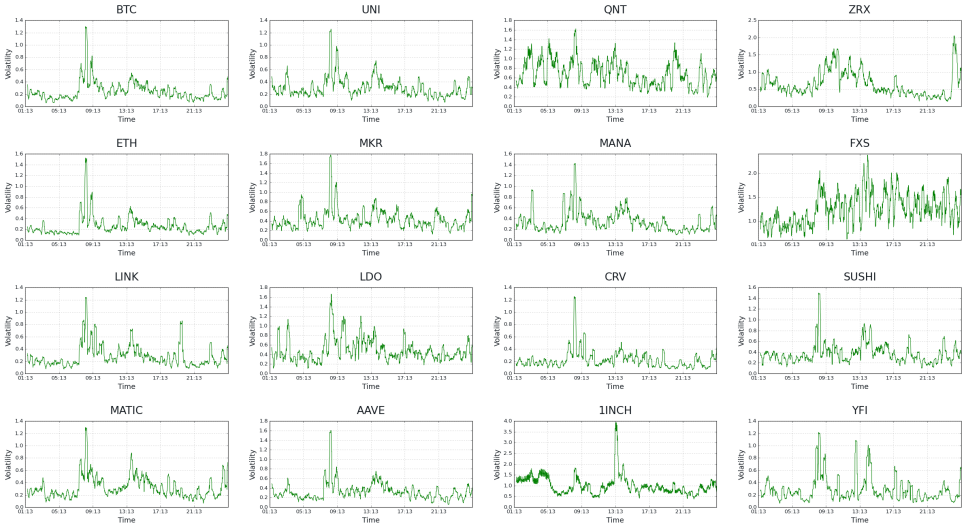

Рисунок 1: Графики аппроксимированных волатильностей для рассматриваемых токенов, построенных по 15-минутному окну

Анализ данных, представленных на Рисунке 1, показывает, что значения волатильности демонстрируют выраженные колебания вверх и вниз на коротких временных интервалах при сохранении относительно постоянного тренда. Это указывает на то, что наиболее корректным решением является описание волатильности в виде процесса, обладающего

свойством возврата к среднему значению. Аналогичный метод используется в модели Орнштайна-Уленбека, где динамика процесса также стремится к среднему состоянию. В классической форме процесс Орнштайна-Уленбека применяется для описания динамики цен активов и описывается следующим стохастическим уравнением:

$$dX_t = \theta(\mu - X_t)dt + \sigma dW_t, \tag{2}$$

где μ – среднее значение цены, θ – скорость возврата к среднему, σ – волатильность актива, dW_t – приращение броуновского движения. В нашей модели мы применим данную концепцию для моделирования волатильности. Тем не менее, существует ещё один важный фактор, который необходимо включить в модель волатильности для её усовершенствования. Для того, чтобы выявить этот фактор и лучше понять особенности поведения волатильности, необходимо проанализировать её распределение. Для этого построим гистограммы для полученных ранее данных по волатильностям:

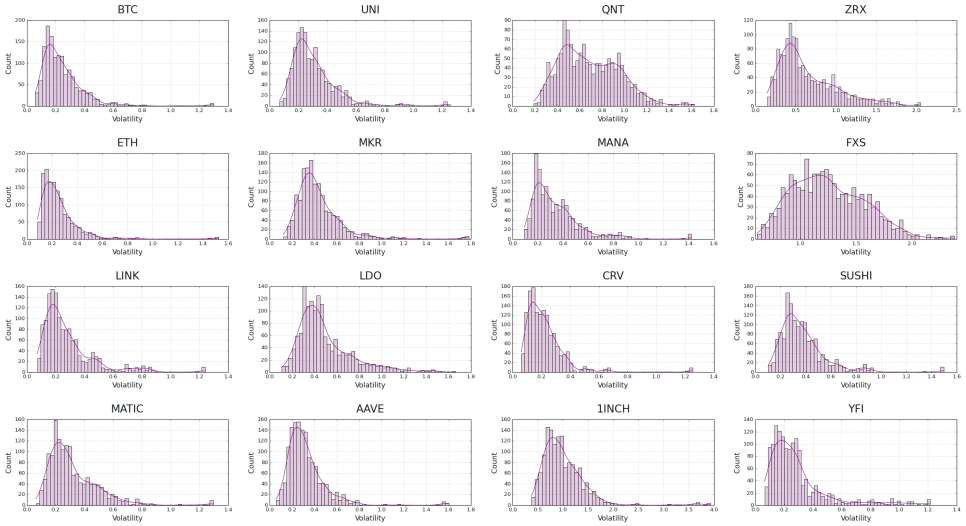

Рисунок 2: Гистограммы аппроксимированных волатильностей для рассматриваемых токенов, построенных по 15-минутному окну

Следует отметить, что в основе большинства стохастических моделей заложено предположение о нормальном распределении волатильности токенов. Даже при использовании модели Орнштейна–Уленбека данное предположение характерно для традиционных финансовых рынков. Однако, исходя из анализа гистограмм, можно заметить, что поведение волатильности криптоактивов ближе к логнормальному распределению, чем к стандартному нормальному. Данная особенность обусловлена спецификой криптовалютных рынков, где значительно чаще наблюдаются

резкие колебания цен и высокие значения волатильности. В связи с этим необходимо учесть данный фактор при построении модели. Одним из решений может стать добавление компоненты волатильности в стохастический процесс, описывающий броуновское движение. Таким образом, в обозначениях для модели волатильности мы получаем следующую стохастическую модель:

$$d\sigma_t = \kappa(\theta - \sigma_t)dt + \gamma\sigma_t dW_t^\sigma, \tag{3}$$

где θ – среднее значение цены, κ – скорость возврата к среднему, γ – коэффициент колебания волатильности, dW_t^σ – приращение броуновского движения, соответствующего процессу σ_t.

Обратимся к анализу гистограмм (Рисунок 2), который демонстрирует, что волатильность характеризуется наличием экстремальных значений, обусловленных значительными колебаниями цен. Эти значения особенно выражены для таких криптоактивов, как UNI, MKR и 1INCH. Следовательно, для более точного описания рынка необходимо учитывать скачки цен на рассматриваемые активы. В целях моделирования таких скачков, предлагается включить компоненту скачков в стохастическое уравнение, описывающее динамику цены актива. Скачки будут моделироваться пуассоновским процессом с интенсивностью λ, а их величина — случайной величиной J, распределённой по нормальному закону с нулевым средним значением и стандартным отклонением ε. Альтернативные подходы к моделированию скачков, такие как метод, описанный в работе [7], также могли бы быть применены. Однако в данной работе нормальное распределение скачков представляется приемлемым и достаточным для описываемой модели. Таким образом, стохастическое уравнение для динамики цен можно записать следующим образом:

$$dX_t = \mu dt + \sigma_t dW_t^X + JdN_t, \tag{4}$$

где X_t – лог-цена актива, μ – тренд, σ_t – волатильность, dW_t^X – инкремент броуновского движения, соответствующего процессу X_t, N_t – пуассоновский процесс с интенсивностью λ, J – размер скачка, который описывается нормальным распределением с нулевым средним и некой дисперсией.

Однако, уравнение в его текущем виде не предоставляет достаточной интерпретации. При работе со случайными процессами мы стремимся наблюдать четкую динамику поведения случайной величины во времени. Тем не менее, в данном виде уравнение не отражает полной картины поведения величины, отвечающей за размер скачков. В связи с этим было предложено новое концептуальное решение, ранее не применявшееся в подобного рода моделях: выделить величину скачка в отдельный процесс

J_t. Такой подход позволяет более точно описать независимые от рынка колебания цен, которые оказывают значительное влияние на динамику актива в конкретные моменты времени, обусловленные срабатыванием пуассоновского процесса. Это нововведение позволяет нам лучше понять вклад скачков в общую динамику цен. Таким образом, можно записать следующее стохастическое уравнение для описания так называемой «нерыночной» компоненты:

$$dJ_t = \varepsilon dW_t^J, \tag{5}$$

где ε – вариация величины скачков, dW_t^J – инкремент броуновского движения, соответствующего процессу J_t. Несмотря на то, что это уравнение отображает динамику изменения скачков, такая запись корректно отображает идею нормального распределения для самой величины скачков в силу свойств винеровского процесса.

Следует подчеркнуть, что в контексте финансовой математики необходимо, чтобы все применяемые случайные процессы являлись мартингалами, поскольку модели, используемые для оценки активов, предполагают отсутствие арбитражных возможностей и справедливую оценку параметров. Однако, сам по себе пуассоновский процесс не является мартингалом. Для того чтобы привести его в соответствие с данными требованиями, необходимо добавить корректирующий инкремент $-\lambda t$, тем самым превращая его в компенсированный пуассоновский процесс

Теорема 1. Пусть $N(t)$ – пуассоновский процесс с интенсивностью λ. Определим *компенсированный пуассоновский процесс* следующим образом:

$$M(t) = N(t) - \lambda t.$$

Тогда процесс $M(t)$ является мартингалом.

Доказательство данной теоремы приведено в учебнике [8]

Таким образом, возникает необходимость записи пуассоновского процесса в виде компенсированного пуассоновского процесса с добавленной компонентой λt. Это позволяет учесть компенсирующий инкремент и сделать процесс мартингалом. В результате, уравнение (4) можно переписать в следующем виде:

$$dX_t = (\mu + \lambda J_t)dt + \sigma_t dW_t^X + J_t d\tilde{N}_t, \tag{6}$$

где $\tilde{N}_t$ – соответствующий компенсированный пуассоновский процесс. Данная запись будет удобной при дальнейших вычислениях.

Прежде чем перейти к окончательной формулировке стохастической системы, важно обратить внимание на возможную корреляцию между винеровскими процессами, описывающими динамику X_t и σ_t. Обозначим

коэффициент корреляции между ними как ρ. Следует также отметить, что винеровский процесс, связанный со скачками J_t, предполагается независимым от процессов X_t и σ_t. Это связано с тем, что J_t представляет нерыночные компоненты, которые не зависят от волатильности и логарифмической цены актива. Данное предположение оправдано, поскольку скачки, описываемые J_t, имеют другую природу и не коррелируют с основными рыночными процессами. Таким образом, объединив выражения (3), (5) и (6) с учётом указанных зависимостей и сделанных предположений, мы получаем следующую стохастическую модель, которая полноценно описывает динамику рассматриваемых процессов:

$$\begin{cases} dX_t = (\mu + \lambda J_t)dt + \sigma_t dW_t^X + J_t d\tilde{N}_t, \\ d\sigma_t = \kappa(\theta - \sigma_t)dt + \gamma\sigma_t dW_t^\sigma, \\ dJ_t = \varepsilon dW_t^J, \\ dW_t^X dW_t^\sigma = \rho, \\ dW_t^X dW_t^J = dW_t^\sigma dW_t^J = 0. \end{cases}$$

Стоит отметить, что по структуре данная модель очень похожа на модель стохастической волатильности Бэйтса, описанную в его работе [1], которая является расширением известной модели Хестона [4], включающим в себя компоненты скачков. Модель Бэйтса описывается следующей стохастической системой:

$$\begin{cases} dS_t = (r - \lambda\mu_J)\, S_t dt + \sqrt{V_t} S_t d\widetilde{W}_t^{(1)} + J S_t d\widetilde{N}_t, \\ dV_t = \kappa\,(\theta - V_t)\, dt + \sigma_v \sqrt{V_t} d\widetilde{W}_t^{(2)}, \\ d\widetilde{W}_t^{(1)} d\widetilde{W}_t^{(2)} = \rho dt, \end{cases}$$

при этом скачки следуют логнормальному распределению:

$$\eta := \ln(1 + J) \sim \mathcal{N}\left(\mu_J, \sigma_J^2\right).$$

Данная модель учитывает как наличие скачков, так и ненормальную структуру распределения волатильности, а также её возврат к среднему значению. Основные отличия заключаются в том, что в модели Бэйтса скачки распределены по логнормальному закону, волатильность имеет гамма-распределение (при предположении, что процесс не вырождается), а величина скачков рассматривается как случайная величина. В нашей же модели величина скачков описывается отдельным случайным процессом. Учитывая структурное сходство предлагаемой модели с моделью Бэйтса, последняя представляется наиболее подходящим кандидатом для сравнения. Поэтому для анализа мы выберем именно её.

3. Построение уравнения для кумулятивной функции распределения

Для дальнейшего анализа необходимо использовать мощный инструмент для оценки модели и сравнения её характеристик. В качестве такого инструмента предлагается рассчитать переходную функцию плотности вероятности. Поскольку речь идёт о функции плотности вероятности, требуется составить одно из уравнений Колмогорова, которое описывает её изменение в контексте уравнений в частных производных. Однако, прежде чем приступить к его построению, необходимо определить, как будет выглядеть дифференциал некоторой гладкой функции для рассматриваемой модели. Таким образом, мы приходим к следующей теореме:

Теорема 2. Пусть $f : \mathbb{R}^4 \to \mathbb{R}$ – некая дважды дифференцируемая функция. Тогда для нашей модели справедлива следующая формула:

$$\begin{aligned} df(t,X,\sigma,J) = \Bigg[& \frac{\partial f}{\partial t} + \mu\frac{\partial f}{\partial X} + \lambda\big[f(t,X+J,\sigma,J) - f(t,X,\sigma,J)\big] + \\ & + \kappa(\theta-\sigma)\frac{\partial f}{\partial \sigma} + \frac{1}{2}\sigma^2\frac{\partial^2 f}{\partial X^2} + \frac{1}{2}\gamma^2\sigma^2\frac{\partial^2 f}{\partial \sigma^2} + \frac{1}{2}\varepsilon^2\frac{\partial^2 f}{\partial X^J} + \\ & + \rho\gamma\sigma^2\frac{\partial^2 f}{\partial X \partial \sigma}\Bigg] dt + \sigma\frac{\partial f}{\partial X}dW_t^X + \gamma\sigma\frac{\partial f}{\partial \sigma}dW_t^\sigma + \varepsilon\frac{\partial f}{\partial J}dW_t^J + \\ & + \big[f(t,X+J,\sigma,J) - f(t,X,\sigma,J)\big]d\tilde{N}_t \end{aligned}$$

В качестве упрощения можно разложить компонету с компенсированным пуассоновским процессом по формуле Тейлора до второго порядка, после чего мы получим, что

$$f(t,X+J,\sigma,J) - f(t,X,\sigma,J) \approx J\frac{\partial f}{\partial X} + J^2\frac{\partial^2 f}{\partial X^2}$$

Доказательство. Рассмотрим систему, которой описывается наша модель. Свернем компенсированный пуассоновский процесс обратно в стандартный пуассоновский процесс:

$$dX_t = \mu dt + \sigma_t dW_t^X + J_t dN_t.$$

Можно отметить следующее: для всех компонентов системы выполняется многомерная лемма Ито, что позволяет вывести практически всю формулу, приведённую в теореме. Исключение составляет компонент с пуассоновским процессом $J_t dN_t$, поскольку классическая лемма Ито не включает пуассоновские процессы. Таким образом, необходимо доказать лемму Ито для пуассоновского процесса. Доказательство этой

леммы в интегральной форме представлено в работах [5] и [8], однако для наших целей требуется дифференциальная форма. Поэтому мы предлагаем собственное доказательство в дифференциальной форме, которое отличается от упомянутых источников.

Доказывать будем для более простого, но в то же время достаточно общего случая, так как результат будет справедлив для нашего сценария без изменений. Постановка задачи следующая: рассмотрим процесс $dY_t = j(Y_t, t)dN_t$. Задав гладкую функцию $g(t, N_t)$, необходимо вычислить дифференциал $dg(t, Y_t)$.

По определению дифференциала, запишем

$$dg(t, Y_t) = g(t+dt, Y_t+dY_t)-g(t, Y_t) = g(t+dt, Y_t+j(Y_t, t)dN_t)-g(t, Y_t) \quad (7)$$

Для пуассоновского процесса известно следующее свойство: компонента dN_t может принимать только два значения: 1 с вероятностью λdt (процесс активировался) и 0 с вероятностью $1 - \lambda dt$ (процесс не активировался). Соответственно, первое слагаемое мы можем представить в виде суммы двух «ортогональных» компонент:

$$\begin{aligned} g(t + dt, Y_t + j(Y_t, t)dN_t) = \\ = g(t + dt, Y_t + j(Y_t, t))dN_t + g(t + dt, Y_t)(1 - dN_t). \end{aligned} \quad (8)$$

Таким образом, если $dN_t = 1$, то скачок размера $j(Y_t, t)$ активируется, а второе слагвемое обнулится; если $dN_t = 0$, то скачок не активировался (или равен нулю), а первое слагаемое обнулится.

Рассмотрим первое слагаемое. Функцию g мы можем разложить по формуле Тейлора по первому аргументу до первой производной:

$$g(t + dt, Y_t + j(Y_t, t))dN_t = g(t, Y_t + j(Y_t, t))dN_t + \frac{\partial g(t, Y_t + j(Y_t, t))}{\partial t} dtdN_t$$

Заметим, что второе слагаемое обнулится в силу того, что $dtdN_t = 0$ (изометрия Ито для пуассоновского процесса), после чего мы получим, что

$$g(t + dt, Y_t + j(Y_t, t))dN_t = g(t, Y_t + j(Y_t, t))dN_t$$

Аналогично можно разложить по Тейлору второе слагаемое в (8) (компонента с первой производной здесь будет равна нулю по той же причине):

$$g(t + dt, Y_t)(1 - dN_t) = g(t, Y_t)(1 - dN_t)$$

Итого, с учетом полученных разложений формула (8) перепишется в виде

$$g(t + dt, Y_t + j(Y_t, t)dN_t) = g(t, Y_t + j(Y_t, t))dN_t + g(t, Y_t)(1 - dN_t)$$

Подставляя эту формулу в уравнение (7), получим

$$dg(t, Y_t) = g(t, Y_t + j(Y_t, t))dN_t + g(t, Y_t)(1 - dN_t) - g(t, Y_t) = \\ = \big(g(t, Y_t + j(Y_t, t)) - g(t, Y_t)\big)dN_t.$$

Теперь вернемся опять к записи в форме компенсированного пуассоновского процесса, переписав дифференциал в следующем виде:

$$dg(t, Y_t) = \big(g(t, Y_t + j(Y_t, t)) - g(t, Y_t)\big)d(N_t - \lambda t + \lambda t) = \\ = \lambda\big(g(t, Y_t + j(Y_t, t)) - g(t, Y_t)\big)dt + \big(g(t, Y_t + j(Y_t, t)) - g(t, Y_t)\big)d\tilde{N}_t.$$

В итоге, мы получили лемму Ито в дифференциальном виде для компенсированного пуассоновского процесса. Соединяя вместе стандартную многомерную лемму Ито в доказанной только что лемму Ито для компенсированного пуассоновского процесса, мы получим, что

$$df(t, X, \sigma, J) = \Bigg[\frac{\partial f}{\partial t} + \mu\frac{\partial f}{\partial X} + \lambda\big[f(t, X + J, \sigma, J) - f(t, X, \sigma, J)\big] + \\ + \kappa(\theta - \sigma)\frac{\partial f}{\partial \sigma} + \frac{1}{2}\sigma^2\frac{\partial^2 f}{\partial X^2} + \frac{1}{2}\gamma^2\sigma^2\frac{\partial^2 f}{\partial \sigma^2} + \frac{1}{2}\varepsilon^2\frac{\partial^2 f}{\partial X^J} + \\ + \rho\gamma\sigma^2\frac{\partial^2 f}{\partial X \partial \sigma}\Bigg]dt + \sigma\frac{\partial f}{\partial X}dW_t^X + \gamma\sigma\frac{\partial f}{\partial \sigma}dW_t^\sigma + \varepsilon\frac{\partial f}{\partial J}dW_t^J + \\ + \big[f(t, X + J, \sigma, J) - f(t, X, \sigma, J)\big]d\tilde{N}_t.$$

□

Теперь необходимо записать уравнение для переходной функции плотности вероятности $p(t, X, \sigma, J)$. Однако следует отметить один важный аспект: эволюция переходной плотности описывается уравнением Фоккера-Планка (известным также как прямое уравнение Колмогорова). При записи уравнения Фоккера-Планка для интервала времени $\tilde{t} \le t \le T$, необходимо также учесть граничное условие. В нашем случае оно будет задано следующим образом: в момент времени $\tilde{t}$

$$p(t_0, X, \sigma, J) = \delta(X - \tilde{X}) \cdot \delta(\sigma - \tilde{\sigma}) \cdot \delta(J - \tilde{J}),$$

где δ – дельта-функция Дирака, а $\tilde{X}, \tilde{\sigma}, \tilde{J}$ – некие начальные параметры.

Здесь возникает основная проблема: уравнение Фоккера-Планка для предложенной модели является слишком сложным для аналитического решения. Тем не менее, ограничение с дельта-функцией можно эффективно обработать аналитически. Однако, при численном решении уравнения Фоккера-Планка возникает необходимость в численной аппроксимации дельта-функции, так как ее невозможно воспроизвести

в чистом виде для использования в вычислительных алгоритмах. Среди существующих методов численной аппроксимации дельта-функции наиболее известен подход, основанный на сеточных методах. В этом случае дельта-функция представляется в виде треугольника, где боковые вершины располагаются в соседних узлах сетки, а третья вершина — в граничной точке. Высота треугольника подбирается таким образом, чтобы его площадь была равна единице. Основная идея метода заключается в том, что по мере уменьшения шага разбиения треугольник сжимается, сохраняя площадь равной единице, постепенно приближаясь к истинной дельта-функции. Однако данный метод не представляется возможным в нашем случае, так как для более точного анализа предполагается численное решение параметрического уравнения в частных производных. Это означает, что постоянные параметры модели также будут рассматриваться как меняющиеся величины в определённом диапазоне. Таким образом, функция плотности вероятности будет зависеть от 14 переменных, включая исходные переменные и изменяющиеся параметры. Применение сеточной аппроксимации в данном случае неприемлемо ввиду ограничений вычислительных ресурсов. Для адекватного разбиения интервалов, которое достаточно точно аппроксимировало бы дельта-функцию (например, минимум 100 узлов), сеточный метод потребовал бы порядка 10^{28} итераций, включающих решение систем линейных уравнений. Это делает задачу вычислительно невыполнимой в разумные сроки.

Учитывая вышеописанные соображения, необходимо избежать непосредственного использования дельта-функции. Но стоит отметить, что преимуществом дельта-функции является её простая и удобная запись при интегрировании. Именно поэтому возникает желание перейти к интегральной форме, которая позволит сохранить эти свойства. Однако, если в уравнении Фоккера-Планка мы будем брать интеграл от дельта-функции, то аналогично потребуется интегрирование функции плотности вероятности. Такая логика приводит нас к следующему решению, позволяющему избежать прямого взаимодействия с дельта-функцией — это переход от функции плотности вероятности к кумулятивной функции распределения. Этот подход позволит сохранить аналитическую простоту, избегая при этом сложностей, связанных с численной аппроксимацией дельта-функции.

Кумулятивная функция распределения для нашей модели определяется следующим образом:

$$F(t, X, \sigma, J; t', \tilde{X}, \tilde{\sigma}, \tilde{J}) := Prob\left[\, X(t') \leq \tilde{X}; \sigma(t') \leq \tilde{\sigma}; J(t') \leq \tilde{J} \,\right] =$$

$$= \int\limits_{-\infty}^{\tilde{X}} \int\limits_{-\infty}^{\tilde{\sigma}} \int\limits_{-\infty}^{\tilde{J}} p(t, X, \sigma, J; t', X', \sigma', J') dX' d\sigma' dJ',$$

где соответственно $Prob$ – вероятность.

Записать уравнение Фоккера-Планка для кумулятивной функции распределения не представляется возможным, так как оно справедливо исключительно для функции плотности вероятности. В связи с этим целесообразно перейти к уравнению Фейнмана-Каца (также известному как обратное уравнение Колмогорова), применимому к нашей модели:

Теорема 3. Формула Фейнмана-Каца для кумулятивной функции распределения в нашей модели записывается следующим образом:

$$\frac{\partial F}{\partial t} + \mu \frac{\partial F}{\partial X} + \lambda J \frac{\partial F}{\partial X} + \lambda J^2 \frac{\partial^2 F}{\partial X^2} + \kappa(\theta - \sigma) \frac{\partial F}{\partial \sigma} + \\ + \frac{1}{2}\sigma^2 \frac{\partial^2 F}{\partial X^2} + \frac{1}{2}\gamma^2 \sigma^2 \frac{\partial^2 F}{\partial \sigma^2} + \frac{1}{2}\varepsilon^2 \frac{\partial^2 F}{\partial X^J} + \rho\gamma\sigma^2 \frac{\partial^2 F}{\partial X \partial \sigma} = 0$$

Доказательство. Выпишем еще раз формулу Ито для нашей модели из Теоремы 1, подставив в нее кумулятивную функцию распределения:

$$dF(t, X, \sigma, J) = \left[\frac{\partial F}{\partial t} + \mu \frac{\partial F}{\partial X} + \lambda J \frac{\partial F}{\partial X} + \lambda J^2 \frac{\partial^2 F}{\partial X^2} + \kappa(\theta - \sigma) \frac{\partial F}{\partial \sigma} + \right. \\ \left. + \frac{1}{2}\sigma^2 \frac{\partial^2 F}{\partial X^2} + \frac{1}{2}\gamma^2 \sigma^2 \frac{\partial^2 F}{\partial \sigma^2} + \frac{1}{2}\varepsilon^2 \frac{\partial^2 F}{\partial X^J} + \rho\gamma\sigma^2 \frac{\partial^2 F}{\partial X \partial \sigma} \right] dt + \\ + \sigma \frac{\partial F}{\partial X} dW_t^X + \gamma\sigma \frac{\partial F}{\partial \sigma} dW_t^\sigma + \varepsilon \frac{\partial F}{\partial J} dW_t^J + \\ + \left[\lambda J \frac{\partial F}{\partial X} + \lambda J^2 \frac{\partial^2 F}{\partial X^2} \right] d\tilde{N}_t$$

В силу того, что функция $F(t, X, \sigma, J)$ является мартингалом (доказательство есть в учебнике Шрива [8]) и учитывая, что процессы W_t^X, W_t^σ, W_t^J, $\tilde{N}_t$ являются мартингалами, множитель при dt должен обнуляться. Это условие непосредственно приводит к уравнению Фейнмана-Каца, представленному в формулировке теоремы. □

При этом необходимо учитывать граничное условие. В отличие от уравнения Фоккера-Планка, уравнение Фейнмана-Каца имеет обратный ход, то есть граничное условие в уравнении Фейнана-Каца имеет противоположную границу, то есть для точки T при $\tilde{t} \leq t \leq T$. Поэтому

граничное условие запишется в следующей форме:

$$F(T,X,\sigma,J;T,\tilde{X},\tilde{\sigma},\tilde{J}) = \int\limits_{-\infty}^{\tilde{X}}\int\limits_{-\infty}^{\tilde{\sigma}}\int\limits_{-\infty}^{\tilde{J}} \delta(X-X')\cdot\delta(\sigma-\sigma')\cdot\delta(J-J')dX'd\sigma'dJ'.$$

А значение этого интеграла мы уже можем выписать в явном виде, воспользовавшись определением дельта-функции. Получим

$$F(T,X,\sigma,J;T,\tilde{X},\tilde{\sigma},\tilde{J}) = \mathbb{I}(X \le \tilde{X})\cdot\mathbb{I}(\sigma \le \tilde{\sigma})\cdot\mathbb{I}(J \le \tilde{J}),$$

где $\mathbb{I}$ – индикаторная функция: функция равна 1, если индикаторное условие выполнено, иначе равна нулю. Так, мы перешли от дельта-функции к индикаторной функции, которую можно задать явным способом без каких-либо аппроксимаций

Таким образом, мы пришли к следующей задаче: необходимо найти решение для следующего уравнения в частных производных на интервале $\tilde{t} \le t \le T$ со следующими граничными условием:

$$\frac{\partial F}{\partial t} + \mu\frac{\partial F}{\partial X} + \lambda J\frac{\partial F}{\partial X} + \lambda J^2\frac{\partial^2 F}{\partial X^2} + \kappa(\theta-\sigma)\frac{\partial F}{\partial \sigma} + \\ \frac{1}{2}\sigma^2\frac{\partial^2 F}{\partial X^2} + \frac{1}{2}\gamma^2\sigma^2\frac{\partial^2 F}{\partial \sigma^2} + \frac{1}{2}\varepsilon^2\frac{\partial^2 F}{\partial X^J} + \rho\gamma\sigma^2\frac{\partial^2 F}{\partial X\partial\sigma} = 0$$

$$F(T,X,\sigma,J;T,\tilde{X},\tilde{\sigma},\tilde{J}) = \mathbb{I}(X \le \tilde{X})\cdot\mathbb{I}(\sigma \le \tilde{\sigma})\cdot\mathbb{I}(J \le \tilde{J}).$$

После того как будет найдено численное решение уравнения для кумулятивной функции распределения, значение искомой функции плотности вероятности можно будет вычислить методом численного дифференцирования. Это достигается с помощью разностной схемы для третьей производной по параметрам $\tilde{X},\tilde{\sigma},\tilde{J}$. Данный подход позволяет получить значение функции плотности вероятности, что и является нашей конечной целью.

4. Численное решение уравнения Фейнмана-Каца

Как упоминалось ранее, стандартные численные методы для решения нашего уравнения не подходят из-за проблемы, известной как «проклятие размерности». По этой причине для решения уравнения мы будем использовать нейронные сети. Данный подход является оптимальным, поскольку позволяет работать с высокоразмерными задачами и не требует фиксированного разбиения пространства. В частности, мы будем применять методологию Deep Galerkin Method (DGM), предложенную Sirignano и Spiliopoulos [6], которая также использована в работе [3]. Мы

будем следовать оригинальной модели, описанной в [6], с небольшими модификациями.

Скорость и качество обучения нейронных сетей зависят от специфики задачи, поскольку каждая структура сети оптимизирована для определённого класса задач. Например, сверточные нейронные сети (CNN) наиболее эффективны для задач распознавания изображений, в то время как сети типа LSTM (Long Short-Term Memory) лучше подходят для работы с временными рядами, где требуется моделирование прогнозов или решение задач оптимизации. В нашем случае мы воспользуемся модифицированной структурой нейронной сети, предложенной Sirignano и Spiliopoulos [6]. В своей работе они отмечают, что данная структура, являющаяся адаптацией LSTM, не только хорошо работает с последовательными данными, но и эффективно справляется с функциями, демонстрирующими «резкие повороты» в результате наложения граничных условий. Это особенно важно для нашей модели, где граничное условие представлено индикаторной функцией, имеющей ступенчатую форму.

Выбранная структура сетей описывается в виде следующей рекуррентной формы:

$$S^1 = \tanh\left(W^1\vec{x} + b^1\right),$$

$$Z^\ell = \tanh\left(U^{z,\ell}\vec{x} + W^{z,\ell}S^\ell + b^{z,\ell}\right), \quad \ell = 1,\dots,L$$

$$G^\ell = \tanh\left(U^{g,\ell}\vec{x} + W^{g,\ell}S^1 + b^{g,\ell}\right), \quad \ell = 1,\dots,L$$

$$R^\ell = \tanh\left(U^{r,\ell}\vec{x} + W^{r,\ell}S^\ell + b^{r,\ell}\right), \quad \ell = 1,\dots,L,$$

$$H^\ell = \tanh\left(U^{h,\ell}\vec{x} + W^{h,\ell}\left(S^\ell \odot R^\ell\right) + b^{h,\ell}\right), \quad \ell = 1,\dots,L,$$

$$S^{\ell+1} = \left(1 - G^\ell\right) \odot H^\ell + Z^\ell \odot S^\ell, \quad \ell = 1,\dots,L,$$

$$f(t,x;\Theta) = WS^{L+1} + b,$$

где $\vec{x}$ – вектор входных параметров для нашего уравнения, L – количество скрытых слоев, M – количество узлов в слое, D – количество входных параметров, $\odot$ – операция покомпнонетного умножения векторов. Параметры модели обозначим за Θ, где

$$\Theta = \Big\{W^1, b^1, \left(U^{z,\ell}, W^{z,\ell}, b^{z,\ell}\right)_{\ell=1}^{L}, \left(U^{g,\ell}, W^{g,\ell}, b^{g,\ell}\right)_{\ell=1}^{L},$$
$$\left(U^{r,\ell}, W^{r,\ell}, b^{r,\ell}\right)_{\ell=1}^{L}, \left(U^{h,\ell}, W^{h,\ell}, b^{h,\ell}\right)_{\ell=1}^{L}, W, b\Big\}.$$

Параметры в Θ обладают следующими размерностями:

$$W^1 \in \mathbb{R}^{M\times(D+1)},\ b^1 \in \mathbb{R}^M,$$
$$U^{z,\ell} \in \mathbb{R}^{M\times(D+1)},\ W^{z,\ell} \in \mathbb{R}^{M\times M},\ b^{z,\ell} \in \mathbb{R}^M,$$
$$U^{g,\ell} \in \mathbb{R}^{M\times(D+1)},\ W^{g,\ell} \in \mathbb{R}^{M\times M},\ b^{g,\ell} \in \mathbb{R}^M,$$
$$U^{r,\ell} \in \mathbb{R}^{M\times(D+1)},\ W^{r,\ell} \in \mathbb{R}^{M\times M},\ b^{r,\ell} \in \mathbb{R}^M,$$
$$U^{h,\ell} \in \mathbb{R}^{M\times(D+1)},\ W^{h,\ell} \in \mathbb{R}^{M\times M},\ b^{h,\ell} \in \mathbb{R}^M,$$
$$W \in \mathbb{R}^{1\times M},\ b \in \mathbb{R}.$$

Следует отметить, что в качестве функции активации выбран гиперболический тангенс. Это обусловлено тем, что данная функция является гладкой, и для нашей задачи важно, чтобы функция распределения также обладала свойством гладкости. Перечисленные в Θ параметры будут инициализироваться однородным распределением по Ксавье. Для дальнейшей работы были выбраны следующие параметры: $L = 4, M = 64$.

Также закрепим, что в процессе обучения будут оптимизироваться не только переменные $t, X, \sigma, J, \tilde{X}, \tilde{\sigma}, \tilde{J}$, но и все остальные параметры модели: $\mu, \kappa, \theta, \gamma, \varepsilon, \rho, \lambda$. Таким образом, входной вектор будет содержать 14 элементов, то есть $D = 14$.

Теперь необходимо сформулировать функцию потерь. Следует использовать такую функцию, которая учитывает как соответствие самому уравнению, так и выполнение граничных условий. Определим её в общем виде для некоторой функции $u = u(t, \vec{x})$, предполагая, что имеем уравнение в частных производных следующего вида:

$$\frac{\partial u}{\partial t}(t, \vec{x}) + \mathcal{L}u(t, \vec{x}) = 0, \quad (t, \vec{x}) \in [t_0, T] \times \Omega,$$

$$u(t = T, \vec{x}) = u_T(\vec{x}).$$

где $\mathcal{L}$ – некий дифференциальный оператор, действующий на переменную $\vec{x}$, Ω – область определения для переменной $\vec{x}$, а граничное условие в момент времени T задается некоторой функцией u_T. Тогда определим функцию потерь следующим образом:

$$Loss_1(f) = \left\| \frac{\partial f}{\partial t}(t, \vec{x}; \Theta) + \mathcal{L}f(t, \vec{x}; \Theta) \right\|^2_{[0,T]\times\Omega,\nu},$$

$$Loss_2(f) = \|f(u, \vec{x}; \Theta) - g(t, \vec{x})\|^2_{[0,T]\times\partial\Omega,\nu},$$
$$Loss(f) = \alpha Loss_1(f) + Loss_2(f).$$

В данной формулировке суть заключается в том, что мы стремимся приблизить функцию f, чтобы минимизировать отклонение от истинного значения функции. Аналогичная логика применяется для функции g и граничного условия. В конструкции функции потерь для обеих компонент используется норма в пространстве L^2 по распределению ν, то есть

$$\|f(y)\|^2_{\Lambda,\nu} = \int_\Lambda |f(y)|^2 \nu(y) dy,$$

где $\nu(y)$ – функция плотности распределения, которое соответствует распределению, по которому инициализируются параметры, а Λ – область, в которой задаются переменные. Это делается для того, чтобы учесть концентрацию сгенерированных значений в нужных областях. Однако в нашем случае параметры инициализируются с равномерным распределением, поэтому плотность рассматривается как некоторая константа. Кроме того, в модель был добавлен параметр α, который отвечает за вес ошибки дифференциального оператора относительно ошибки, связанной с граничным условием. Согласно работе [3], оптимальное значение для параметра α равно 100, однако мы будем использовать значение $\alpha = 128$, чтобы сохранить пропорцию между количеством узлов в слоях, что также влияет на размер выборки для алгоритма.

Теперь, собрав все элементы воедино, представим следующий алгоритм для численного решения:

1) На каждом n-ом шаге генерируется батч из M многомерных точек $(t_n, \vec{x}_n)$, которые инициализируются в области $[0, T] \times \Omega$.

2) Для каждой такой выборки вычисляется средня квадратичная ошибка на основе построенной ранее функции потерь:

$$MSE\left(\Theta_n, \vec{x}_n, t_n\right) = \frac{\alpha}{M} \sum_{i=1}^{M} \left(\frac{\partial}{\partial t} f\left(t_{ni}, \vec{x}_{ni}; \Theta_n\right) + \mathcal{L} f\left(t_{ni}, \vec{x}_{ni}; \Theta_n\right) \right)^2 + \\ + \frac{1}{M} \sum_{i=1}^{M} \left(f\left(T, \vec{x}_{ni}; \Theta_n\right) - u_T\left(\vec{x}_{ni}\right) \right)^2$$

3) На основе функции среднеквадратичной ошибки, по точкам $(t_n, \vec{x}_n)$ обновляются параметры, заложенные в Θ. Для обновления параметров используется алгоритм адаптивной оценки для моментов (ADAM) с темпом обучения $lr = 0.0001$

4) Все описанные выше шаги выполняются итеративно до тех пор, пока значение ошибки не достигнет установленного минимального порога, который в нашем случае равен $e = 0.0001$.

Теперь необходимо определить, какие параметры будут подаваться на вход алгоритма. Поскольку в дальнейшем планируется калибровка коэффициентов модели, нас не устраивает обучение на фиксированных коэффициентах, как это предполагается в уравнениях. Таким образом, на вход, помимо дифференцируемых параметров $X, \sigma, J, \tilde{X}, \tilde{\sigma}, \tilde{J}$, мы также подадим постоянные коэффициенты модели: $\mu, \kappa, \theta, \gamma, \varepsilon, \rho, \lambda$. Эти коэффициенты будут инициализированы в пределах выбранных нами интервалов. Определим следующие диапазоны для данных параметров:

$$X, \tilde{X} \in [-7.0, 7.0]; \quad \sigma, \tilde{\sigma} \in [0.0, 2.0]; \quad J, \tilde{J} \in [-2.0, 2.0];$$
$$\mu \in [-1.0, 1.0]; \quad \kappa \in [0.5, 1.5]; \quad \theta \in [0.2, 1.1]; \quad \gamma \in [0.0, 0.8];$$
$$\varepsilon \in [0.0, 1.0]; \quad \rho \in [-0.5, 0.5]; \quad \lambda \in [0.0, 3.0]; \quad t \in [0.0, 1.0].$$

Модель запускается на 100,000 эпох. Конечно, ввиду большого количества изменяющихся параметров, следовало бы выбрать большее количество итераций, что улучшило бы качество прогнозов. Однако, исходя из ограничений моих вычислительных ресурсов, которые позволяют выполнить 100,000 итераций за приблизительно 27 часов, было принято решение остановиться на данном количестве эпох. Этого оказалось достаточно для получения адекватных результатов.

Для многомерной задачи подобного рода сложно представить визуализацию функции плотности распределения. Тем не менее, мы можем, как минимум, построить тепловую карту (heatmap) и визуализировать изменения плотности «сверху».

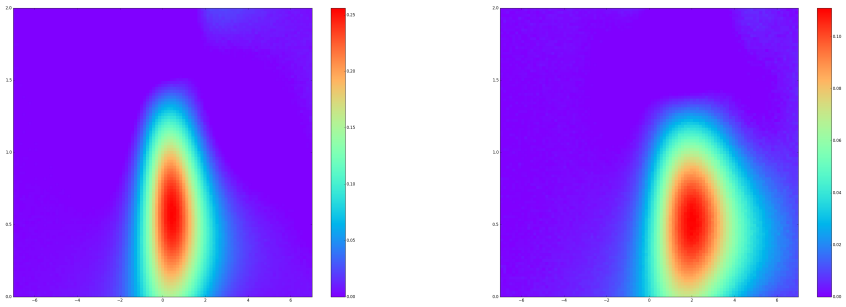

Рисунок 3: Heatmap-ы для вычисленной функции плотности вероятности. Оси x соответствует величина $\tilde{X}$, оси y – величина $\tilde{\sigma}$. Параметры рисунка слева: $X = 0.25$, $\sigma = 0.4$, $J = 0.5$, $\tilde{J} = 0.9$, $\mu = 0.2$, $\kappa = 0.3$, $\theta = 0.4$, $\gamma = 0.3$, $\epsilon = 1.0$, $\rho = -0.25$, $\lambda = 0.25$; параметры рисунка слева: $t = 0.0$, $X = 1.0$, $\sigma = 0.4$, $J = 1.0$, $\tilde{J} = 0.9$, $\mu = 0.2$, $\kappa = 0.5$, $\theta = 0.3$, $\gamma = 0.3$, $\epsilon = 0.6$, $\rho = 0.3$, $\lambda = 0.1$;

На Рисунке 3 видно, что если «разрезать» графики, то есть зафиксировать одну из осевых переменных, полученная функция демонстрирует

поведение, характерное для функции плотности распределения. Также следует отметить, что перед запуском оптимизатора для нашей модели была проведена проверка на простом примере стандартного геометрического броуновского движения. В этом тестовом случае оптимизатор успешно справился с решением уравнения Фейнмана-Каца. Таким образом, исходя как из визуальной проверки, так и из проверки на более простой модели, можно считать результаты достоверными.

Таким образом, этап решения уравнения Фейнмана-Каца был успешно завершён, что позволяет нам переходить к калибровке коэффициентов модели.

5. Калибровка параметров и сравнение с моделью Бэйтса

Пройдя сложный путь поиска решения уравнения Фейнмана-Каца, логично воспользоваться полученной плотностью распределения для калибровки параметров модели. Для этого мы применим один из наиболее надёжных методов математической статистики, специально предназначенный для поиска оптимальных параметров модели — метод максимального правдоподобия. Суть данного метода заключается в том, что при наличии выборки мы вычисляем функцию правдоподобия, представляющую собой кумулятивное произведение условных вероятностей, после чего находим её максимальное значение. Аргументы, при которых достигается максимальное значение функции, будут являться оптимальными оценками для модели. Однако к функции правдоподобия необходимо относиться с особой осторожностью.

Калибровка параметров будет проводиться для каждого токена по отдельности. Напомним, что у нас имеются ежеминутные данные за один день, что даёт временные ряды для цен и волатильностей на $1440 - 15 = 1425$ наблюдений (вычитаем 15, так как волатильность вычисляется по 15-минутному окну). В данной методологии мы будем исходить из того, что волатильность, оценённая как стандартное отклонение по скользящему окну, хотя и не является абсолютно точной оценкой, но даёт достаточно адекватные результаты. Мы разделим данные на обучающую выборку (1000 наблюдений) и тестовую выборку (оставшиеся 425 наблюдений). Обучающая выборка будет использована для калибровки постоянных коэффициентов модели $\mathbf{z} = (\mu, \kappa, \theta, \gamma, \varepsilon, \rho, \lambda)$, а тестовая — для фиксации оптимальных коэффициентов и сравнения модели с моделью Бэйтса.

Перед построением функции правдоподобия необходимо отметить следующее: изначально уравнение Фейнмана-Каца решалось для кумулятивной функции распределения с введением дополнительных

параметров $\tilde{t}, \tilde{X}, \tilde{\sigma}, \tilde{J}$ для существующих переменных t, X, σ, J. Эта функция по определению описывает вероятность того, что, находясь в момент времени t с определёнными значениями X, σ, J, к моменту времени $\tilde{t}$ значения переменных не превысят $\tilde{X}, \tilde{\sigma}, \tilde{J}$. Однако при переходе от функции распределения к функции плотности вероятности интерпретация переменных с тильдами меняется: мы переходим от вероятности не превысить определённые значения к вероятности достижения этих значений. Таким образом, функция $p(t, X, \sigma, J; \tilde{t}, \tilde{X}, \tilde{\sigma}, \tilde{J})$ выражает вероятность того, что переменные X, σ, J в момент времени t примут значения $\tilde{X}, \tilde{\sigma}, \tilde{J}$ к моменту времени $\tilde{t}$. Это важно учитывать при дальнейшем анализе.

Таким образом, функция правдоподобия может быть записана в следующем виде:

$$LH(\mathbf{z}) = \prod_{i=1}^{n} p\big[(X_i, \sigma_i, J_i) \mid \big\{(X_k, \sigma_k, J_k)\big\}_{k=0}^{i-1};\ \mathbf{z}\big],$$

где соответственно $p\big[(X_i, \sigma_i, J_i) \mid \big\{(X_k, \sigma_k, J_k)\big\}_{k=0}^{i-1};\ \mathbf{z}\big]$ – условная вероятность: вероятность того, что лог-цена, волатильность и размер скачка в момент времени i будут равны (X_i, σ_i, J_i) при условии, что те же параметры в предыдущие моменты времени равнялись соответственно $\big\{(X_k, \sigma_k, J_k)\big\}_{k=0}^{i-1}$. Всё это предполагается при фиксированных значениях постоянных параметров модели, $\mathbf{z}$. В силу того, что функция плотности вероятности была построена только для единичного временного интервала и что лучшей оценкой для переменной в момент времени i является та же переменная в момент времени $i-1$ (в силу мартингального свойства), будет корректно записать функцию правдоподобия для нашего случая таким образом:

$$LH(\mathbf{z}) = \prod_{i=1}^{n} p\big[(X_i, \sigma_i, J_i) \mid (X_{i-1}, \sigma_{i-1}, J_{i-1});\ \mathbf{z}\big],$$

Однако следует напомнить, что граничное условие для уравнения Фейнмана-Каца задается на противоположной границе временного интервала, что приводит к обратному ходу времени, начиная с момента T. В связи с этим возникает необходимость «перевернуть» функцию правдоподобия, учитывая данный обратный ход:

$$LH(\mathbf{z}) = \prod_{i=1}^{n} p\big[(X_{i-1}, \sigma_{i-1}, J_{i-1}) \mid (X_i, \sigma_i, J_i);\ \mathbf{z}\big],$$

то есть в соответствии с граничным условием для обратного уравнения Колмогорова, мы должны рассматривать вероятность параметров в

прошлом при условии известных значений в будущем, при этом подразумевается, что постоянные параметры модели заданы вектором $\mathbf{z}$).

Теперь необходимо разобраться с определением оставшихся переменных. Очевидно, что в качестве временного параметра мы выбираем $t = 0$. Процесс X_t является наблюдаемым, поэтому значения X_{i-1} и X_i мы получаем напрямую из тренировочных данных по токенам. Что касается процесса волатильности, на данном этапе мы не производили новых вычислений, поэтому значения σ_{i-1} и σ_i берем из ранее вычисленных волатильностей, полученных с помощью стандартного отклонения цен по скользящему окну. Однако процесс J_t вызывает определённые трудности, так как не вполне очевидно, как корректно выделить скачки исходя из динамики цены.

Мы можем сделать упрощённое предположение, что значения J_{i-1} и J_i равны нулю (предполагая, что в среднем скачки не наблюдаются). Однако в этом случае теряется смысл введения скачков в модель, если мы избегаем их учета. Поэтому предлагается следующая аппроксимация: предположим, что значение скачка в момент времени $i-1$ вычисляется как разница между логарифмическими ценами X_{i-1} и X_i. Идея заключается в том, чтобы задать допустимые пороговые значения изменения лог-цены (симметричные относительно нуля, так как скачки в нашей модели распределены нормально с нулевым средним). Если изменение лог-цены превышает порог, это будет интерпретироваться как активация скачка в соответствующий момент времени. В таком случае значение скачка будет равно разнице между ближайшей границей и фактическим изменением лог-цены. Если же порог не был превышен, то скачка не было, и значение J_{i-1} равно нулю.

Почему мы можем поступить таким образом? В данной модели и в других временных рядах изменения лог-цен величины скачков могут быть настолько малы, что их невозможно выделить из динамики лог-цены. Для нашей текущей задачи идентифицировать малые скачки практически невозможно, но это не является нашей основной целью. Мы стремимся моделировать экстремальные изменения цены, а малые скачки являются побочным эффектом моделирования. Поэтому целесообразно разделить все скачки на наблюдаемые и ненаблюдаемые, сосредоточив внимание на крупных скачках, а относительно малые значения можно игнорировать без существенной потери точности.

Для моделирования наблюдаемых скачков, как уже было отмечено, необходимо определить границы допустимых изменений лог-цены. Это можно сделать либо вручную, анализируя графики, либо задать границы в виде формулы. Если задавать границы в виде формулы, нам необходимо учитывать зависимость от волатильности. Однако, поскольку волатильность аппроксимирована на основе самой лог-

цены, скачок лог-цены вызовет скачок волатильности, что приведет к некорректному заданию границ. Поэтому более разумно задать фиксированный граничный интервал $[-b,\ b]$, где $b > 0$ — параметр, который может варьироваться для разных токенов.

Таким образом, модель значений скачков будет записана следующим уравнением:

$$J_{i-1} = \operatorname{sign}(\Delta X_{i-1}) \cdot \max\big\{|\Delta X_{i-1}| - b, 0\big\},$$

где $\Delta X_{i-1} = X_{i-1} - X_i$ – приращение логарифмической цены в момент времени i. Таким образом, если абсолютное значение изменения лог-цены не превышает заданную границу, то максимум второго множителя достигается при нулевом значении скачка, в результате чего данный скачок классифицируется как несущественный, и его значение обнуляется (в упомянутом ранее предположении, что несущественный скачок является частью процесса X_t). В противном случае, при пересечении границы, с помощью функции знака определяется направление скачка.

Прежде чем перейти к обсуждению реализации оптимизатора, необходимо учесть следующее: поскольку мы работаем с относительно большой тренировочной выборкой и вероятностями, перемножение множества вероятностей порядка тысячи раз неизбежно приведет к обнулению функции правдоподобия. Это явление называется underflow, и его необходимо избежать. Для этого мы перейдём от обычной функции правдоподобия к её логарифму: благодаря монотонности логарифма, оптимальные решения для исходной и логарифмической функций будут достигаться при одном и том же наборе параметров. Более того, мы сразу перейдём к отрицательной функции правдоподобия, поскольку в рамках кода задача заключается в минимизации, а не максимизации. Таким образом, мы будем рассматривать следующую отрицательную логарифмическую функцию правдоподобия:

$$LLH^{-}(\mathbf{z}) = \sum_{i=1}^{n} -\ln p\big[(X_{i-1}, \sigma_{i-1}, J_{i-1}) \mid (X_i, \sigma_i, J_i);\ \mathbf{z}\big].$$

Наконец, можно перейти к описанию реализации оптимизатора. Здесь возникает серьёзная проблема: использование нейронных сетей для поиска решения переходной функции плотности вероятности (в нашем случае, с использованием библиотеки PyTorch для языка Python) накладывает ограничения на использование известных библиотек для оптимизации, таких как SciPy или OR-Tools. Это связано с тем, что PyTorch использует свои собственные тензоры для представления данных, которые не воспринимаются сторонними библиотеками. Более того, такие библиотеки не поддерживают концепции PyTorch, такие как

вычисление градиентов, что делает невозможным запуск оптимизатора на основе этих инструментов. Можно было бы рассмотреть метод Монте-Карло, его использование для задач оптимизации не является корректным. Данный метод лучше подходит для задач, где приближение к решению достигается за счёт большого количества симуляций, таких как нахождение интегралов или оценка средних значений. Однако в задаче поиска оптимальной точки на гиперплоскости размерности 7 метод Монте-Карло неэффективен, поскольку его использование можно сравнить с поиском иголки в стоге сена. В результате остаётся единственный вариант — использовать библиотеку PyTorch для решения задачи оптимизации.

Однако и здесь возникает проблема: модели оптимизации в PyTorch не позволяют задать явные границы для оптимизируемых параметров. Поскольку других подходящих инструментов нет, необходимо нивелировать эту проблему путём введения дополнительных условий. Мы решаем это путём включения штрафов за выход за границы в нашу отрицательную логарифмическую функцию правдоподобия, которая будет одновременно выступать в роли функции потерь для оптимизатора.

Важно отметить, что параметры будут калиброваться в пределах тех же интервалов, которые прежде использовались при решении параметрического уравнения Фейнмана-Каца. Если параметры выйдут за эти границы, поведение переходной функции плотности вероятности станет непредсказуемым, что негативно скажется на процессе оптимизации. Выпишем ещё раз эти интервалы:

$$\mu \in [-1.0, 1.0]; \quad \kappa \in [0.5, 1.5]; \quad \theta \in [0.2, 1.1]; \quad \gamma \in [0.0, 0.8];$$

$$\varepsilon \in [0.0, 1.0]; \quad \rho \in [-0.5, 0.5]; \quad \lambda \in [0.0, 3.0].$$

Также введем следующие обозначения: $\mathbf{c_{min}}, \mathbf{c_{max}}$ – вектора минимальных и максимальных границ для вектора параметров $\mathbf{z}$.

Теперь вернёмся к введению штрафов. Идея заключается в том, чтобы добавлять штрафы к функции потерь при выходе параметров за установленные границы. Построим штрафную функцию следующим образом:

$$Penalty(\mathbf{z_n}) = \pi \left|\left| \max\{\mathbf{z_n} - \mathbf{c_{max}}; \mathbf{0}\} + \max\{\mathbf{c_{min}} - \mathbf{z_n}; \mathbf{0}\} \right|\right|_1,$$

где $\mathbf{z_n}$ – вектор параметров на n-ой итерации, $\mathbf{0}$ – нулевой вектор, операция максимума берется покомпонентно, $\|\cdot\|_1$ – норма в пространстве l_1, которая вычисляется как сумма абсолютных значений компонент вектора, π – штрафной коэффициент, который регулирует силу штрафа. Так, предложенная штрафная функция показывает, насколько сильно алгоритм отдалился от границы; если же вектор параметров находится

внутри установленных границ, то размер штрафа равен нулю. Далее в качестве штрафного коэффициента будем использовать $\pi = 10$.

В итоге осталось объединить нашу отрицательную логарифмическую функцию правдоподобия со штрафной функцией. Сделав это, получим окончательную форму функции потерь:

$$Loss(\mathbf{z_n}) = \sum_{i=1}^{n} -\ln p\big[(X_{i-1}, \sigma_{i-1}, J_{i-1}) \mid (X_i, \sigma_i, J_i);\ \mathbf{z_n}\big] + Penalty(\mathbf{z_n}).$$

Для решения данной задачи мы применили алгоритм стохастического градиентного спуска (SGD) с темпом обучения $lr = 0.001$. В качестве начальных параметров $\mathbf{z_0}$ были взяты середины интервалов, задающих границы параметров: $\mathbf{z_0} = \frac{1}{2}(\mathbf{c_{max}} + \mathbf{c_{min}})$.

Теперь у нас есть все необходимое для сравнения результатов с моделью Бэйтса. Для этой модели были воспроизведены все шаги, выполненные для нашей модели. Единственное существенное отличие заключается в природе скачков: в то время как в процессе решения уравнения Фейнмана-Каца в нашей модели скачки выделены в отдельный процесс, в модели Бэйтса параметры логнормального распределения скачков являются одними из постоянных коэффициентов модели. Соответственно, величина скачка моделируется как случайная величина, распределённая по логнормальному закону с соответствующими параметрами.

В результате была составлена Таблица 1, содержащая итоговые результаты оптимизации логарифмической функции правдоподобия, включая максимальные значения функции для модели Бэйтса. Стоит отметить, что функция правдоподобия, помимо использования для калибровки коэффициентов модели, также служит показателем качества интерпретации рыночных процессов. Как было упомянуто ранее, функция правдоподобия строится как произведение условных вероятностей, что по сути отражает вероятность точного предсказания временного ряда. Таким образом, функция правдоподобия может интерпретироваться как ответ на вопрос: «насколько точно мы предсказываем следующее значение временного ряда?», а лучшая предсказательная способность, в свою очередь, указывает на лучшую интерпретацию актива или рынка, на котором он представлен. По результатам, представленным в таблице, можно заметить, что максимальные значения функции правдоподобия для нашей модели в среднем несколько выше. Однако, эта информация не даёт окончательной сравнительной оценки: необходимо проанализировать поведение функций правдоподобия на тестовой выборке с уже зафиксированными оптимальными коэффициентами.

	μ	κ	θ	γ	ε	ρ	λ	LLH_Our	LLH_Bates
LINK	-0.4792	1.0616	0.2172	0.1813	0.8323	0.4379	1.6761	-1439.5780	-1650.1839
BTC	-0.7391	1.2236	0.2046	0.1587	0.8148	0.1976	1.2809	-1537.4155	-1419.7142
CRV	-0.2301	1.1125	0.2354	0.4477	0.8766	0.4780	0.8428	-1567.0087	-1507.2024
ETH	-0.7755	1.1199	0.2104	0.0198	0.8490	0.1639	0.8450	-1591.1027	-1560.5366
UNI	-0.4072	1.1098	0.2451	0.7735	0.6434	-0.2716	1.5069	-1653.7539	-1744.5925
AAVE	-0.3200	1.0446	0.2675	0.3351	0.9205	0.1190	0.8333	-1666.4852	-1737.6021
MKR	-0.2887	0.7957	0.2032	0.5857	0.9971	0.0233	1.9902	-1482.6937	-1776.7665

Таблица 1. Итоговые значениоткалиброванных параметров на примере семи токенов в результате оптимизации логарифмической функции правдоподобия (первые 7 столбцов) и максимальные значения функции для нашей модели и для модели Бэйтса (последние 2 столбца)

Значения логарифмической функции правдоподобия являются отрицательными, поскольку в исходной версии функции правдоподобия (до применения логарифмов) мы перемножали вероятности, значения которых лежат в диапазоне от 0 до 1, следовательно, логарифмы этих вероятностей всегда будут отрицательными. Также стоит отметить, что наш оптимизатор выявил несколько интересных особенностей: например, известно, что токены BTC и ETH обладают относительно высокой корреляцией между собой, что проявилось как в схожести значений откалиброванных параметров, так и в максимальных значениях функции правдоподобия для этих активов. В целом, результаты выглядят удовлетворительными, но окончательные выводы можно будет сделать только после анализа на тестовой выборке при оптимальных параметрах.

После аналогичного построения Таблицы 2 для результатов на тестовой выборке с использованием откалиброванных параметров мы получили гораздо более обнадёживающие результаты.

	LLH_our	LLH_Bates
LINK	-894.723091	-1119.705687
BTC	-931.342194	-960.669007
CRV	-1139.196365	-1324.843547
ETH	-923.042812	-931.730988
UNI	-793.611184	-1181.945911
AAVE	-803.573028	-1099.465584
MKR	-734.850167	-727.403466
mean	-888.619834	-1049.394884

Таблица 2. Сравнение значений логарифмической функции правдоподобия на тестовой выборке для нашей модели и модели Бэйтса. Последняя строка в таблице отвечает за среднее значение функций для семи активов

В Таблице 2 наблюдаются более высокие значения функций правдоподобия, что объясняется меньшим размером тестовой выборки по сравнению с тренировочной. Тем не менее, по результатам этой таблицы видно, что в среднем наша модель гораздо лучше предсказывает поведение рынка по сравнению с моделью Бэйтса. Особого внимания заслуживает случай токена CRV, для которого были получены относительно более низкие значения функции правдоподобия. Это связано с тем, что на тестовой выборке было зарегистрировано множество скачков, которые, хотя и значимы, но не превышают ранее установленные границы для наблюдаемых скачков. В результате максимальное значение функции правдоподобия для CRV оказалось значительно ниже. Однако, несмотря на аналогичное моделирование скачков в модели Бэйтса, наша модель всё же демонстрирует более точную интерпретацию подобных случаев.

6. Заключение

В итоге нам удалось разработать новую модель ценообразования и выписать для неё систему стохастических уравнений, которая точно описывает поведение рынка криптоактивов в сравнении с моделью Бэйтса, имеющей схожую структуру. В данной работе мы не только учли такие ключевые особенности, как логнормальное распределение волатильности, её возврат к среднему значению и наличие пуассоновских скачков, характерных для криптоактивов, но также предложили принципиально новую концепцию стохастической модели, в которой величина скачков описывается отдельным стохастическим процессом. Это позволяет более точно выделить данную компоненту в динамике изменения цен, что даёт как и гибкость в её моделировании, так и лучшую интерпретацию скачков. Кроме того, мы успешно составили и решили параметрическое уравнение Фейнмана-Каца с помощью нейронных сетей, что позволило нам точно определить значение функции плотности вероятности без привязки к разбиению. После получения функции плотности вероятности мы использовали метод максимального правдоподобия для калибровки модели на исторических данных, а затем провели сравнение результатов на тестовой выборке. Сравнение с такой сложной моделью, как модель Бэйтса, показало, что наша модель в среднем лучше интерпретирует динамику цены, что является важным достижением.

В дальнейшем данную модель можно усовершенствовать и применять не только к крипторынкам, но и к традиционным финансовым рынкам, таким как опционы или процентные ставки. В частности, модель можно обобщить, например, перейдя от обычного пуассоновского процесса к составному пуассоновскому процессу, что позволит более точно моделировать скачки. Также можно рассмотреть возможность добавления

корреляции между винеровским процессом, описывающим скачки, и винеровскими процессами, описывающими цену и волатильность. Кроме того, при составлении уравнения Фейнмана-Каца можно не прибегать к разложению разности функций со скачком и без него в ряд Тейлора до второй производной, что также может повысить точность модели.

Список литературы

[1] Bates, D., *Jumps and Stochastic Volatility: Exchange Rates Processes Implicit in Deutsche Mark Options*, The Review of Financial Studies 9, 1996.

[2] Cornelis W Oosterlee, Lech A Grzelak, *Mathematical Modeling and Computation in Finance*, World Scientific Publishing Europe Ltd, 2020.

[3] Haozhe Su, M.V. Tretyakov, David P. Newton, *Deep Learning of Transition Probability Densities for Stochastic Asset Models with Applications in Option Pricing*, arXiv:2105.10467, 2023.

[4] Heston, S., *A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options*, The Review of Financial Studies 6, 1993.

[5] *Notes on Stochastic Finance, Chapter 20: Stochastic Calculus for Jump Processes*, Nanyang Technological University.

[6] Justin Sirignano, Konstantinos Spitiopoulos, *A deep learning algorithm for solving partial differential equations*, arXiv:1708.07469, 2018.

[7] S. G. Kou, *A Jump-Diffusion Model for Option Pricing*, Department of Industrial Engineering and Operations Research, Columbia University, New York, 2002.

[8] Steven E. Shreve, *Stochastic Calculus for Finance II, Continuous-Time Models*, Springer Science + Business Media, Inc., 2004.

Jump diffusion model with mean-reverting lognormal volatility

Chestnov R.V.

Currently, there are numerous stochastic models built for different market concepts. However, almost all of these models are based and tested for traditional markets, while at present the cryptoasset market is gaining tremendous momentum in terms of volume and market capitalisation. For this reason, the idea arises to create such a model,

which would be built specifically for the cryptoasset market, taking into account all its peculiarities and behavioural patterns. In this paper, we will try to look at some of the features that are noticeable for cryptoassets and build a stochastic model that takes them into account, and after that, we will compare it with another model that is similar in structure. It is notable right away that, similar to how traditional models are used for cryptomarkets, our model for cryptoassets will interpret the traditional market just as well. Moreover, even better than in the opposite case, because the main idea of our model is to take into account various critical events that occur with this or that asset, but by calibrating the model in the right way, we can ignore them.

Key words: Financial mathematics, stochastic calculus, cryptoassets, neural networks, Feynman-Kac equation, Bates model, likelihood function.

References

[1] Bates, D., *Jumps and Stochastic Volatility: Exchange Rates Processes Implicit in Deutsche Mark Options*, The Review of Financial Studies 9, 1996.

[2] Cornelis W Oosterlee, Lech A Grzelak, *Mathematical Modeling and Computation in Finance*, World Scientific Publishing Europe Ltd, 2020.

[3] Haozhe Su, M.V. Tretyakov, David P. Newton, *Deep Learning of Transition Probability Densities for Stochastic Asset Models with Applications in Option Pricing*, arXiv:2105.10467, 2023.

[4] Heston, S., *A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options*, The Review of Financial Studies 6, 1993.

[5] *Notes on Stochastic Finance, Chapter 20: Stochastic Calculus for Jump Processes*, Nanyang Technological University.

[6] Justin Sirignano, Konstantinos Spitiopoulos, *A deep learning algorithm for solving partial differential equations*, arXiv:1708.07469, 2018.

[7] S. G. Kou, *A Jump-Diffusion Model for Option Pricing*, Department of Industrial Engineering and Operations Research, Columbia University, New York, 2002.

[8] Steven E. Shreve, *Stochastic Calculus for Finance II, Continuous-Time Models*, Springer Science + Business Media, Inc., 2004.