1. Министерство науки и высшего образования Российской Федерации
2. Санкт-Петербургский политехнический университет Петра Великого
3. —
4. Институт прикладной математики и механики
5. Высшая школа кибербезопасности и защиты информации

**Реферат**

«Нейросети на базе Neural Ordinary Differential Equations»

1. по дисциплине «Математический аппарат и средства анализа безопасности программного обеспечения»
2. Выполнил
3. студент гр. 3651003/50801 Корольков А.А

<подпись>

1. Проверил

Калинин М.О

<подпись>

1. Санкт-Петербург
2. 2020

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc34770496)

[ВВЕДЕНИЕ 3](#_Toc34770497)

[1. Архитектура остаточный residual сетей. 4](#_Toc34770498)

[2. Нейронные обыкновенные дифференциальные уравнения 6](#_Toc34770499)

[3. Преимущества ODE 8](#_Toc34770500)

[4. Применение нейронных ОДУ 9](#_Toc34770501)

[4.1. Замена остаточных нейросетей нейронными ОДУ в задаче обучения с учителем 10](#_Toc34770502)

[4.2. Масштабируемые и обратимые нормализующие потоки. 12](#_Toc34770503)

[4.3.Генеративные модели временных рядов 13](#_Toc34770504)

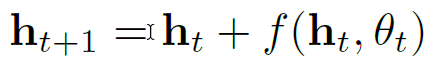
[5. Применение в сфере информационной безопасности 14](#_Toc34770505)

[ЗАКЛЮЧЕНИЕ 15](#_Toc34770506)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 16](#_Toc34770507)

# ВВЕДЕНИЕ

Обыкновенное дифференциальное уравнение (ОДУ) часто используются для описания временных производных физической ситуации, называемых динамикой. Знание динамики позволяет нам моделировать изменение окружающей среды, подобно физическому моделированию, открывая возможность взять любое начальное условие и смоделировать, как оно изменится. Подход с нейронными ОДУ и их изучение через ML устраняет проблему ручного моделирования, затрудняющего интерпретацию данных. Нейронные ОДУ также пригодны для моделирования данных с нерегулярной по времени выборкой данных. Стандартный подход к работе с этими данными заключается в создании временных «корзин» с данными, что приводит к множеству проблем, таких как пустые «корзины» и перекрытия в них. Подход NeuralODE также устраняет эти проблемы, обеспечивая более естественный способ применения ML к нерегулярным временным данным.

Такие модели, как остаточные сети, рекуррентные нейросетевые декодеры и нормализующие потоки, строят сложные преобразования, составляя последовательность преобразований в новое состояние:  где

Эти итерационные обновления можно рассматривать как эйлерову дискретизацию непрерывного преобразования. В пределе делаем меньшие шаги мы параметризуем непрерывную динамику скрытых значений с помощью обыкновенного дифференциального уравнения (ODE), заданного нейронной сетью: 

Начиная с входного слоя h (0), мы можем определить выходной слой h(T) как решение этой задачи начального значения ODE в некоторый момент времени T. Это значение может быть вычислено с помощью решателя дифференциальных уравнений, который оценивает скрытую динамику единицы f везде, где это необходимо для определения решения с требуемой точностью. На рисунке 1 эти два подхода противопоставлены.

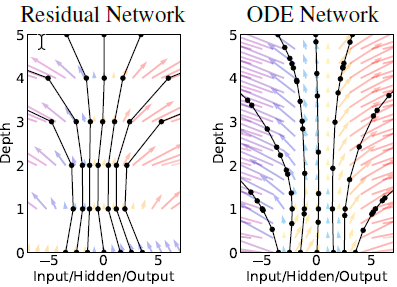


Рисунок 1. Слева: остаточная сеть определяет дискретную последовательность конечных преобразований. Справа: сеть ODE определяет векторное поле, которое непрерывно преобразует состояние. Оба: круги представляют места оценки.

## Архитектура остаточный residual сетей.

Среди нейросетевых архитектур очень популярными оказались архитектуры, использующие механизм residual связей (остаточные сети). Для анализа изображений примером такой архитекту-ры является ResNet [1]. В задаче анализа текстов примером такой архитектуры служит Transformer [2].

В этой архитектуре основным строительным блоком является так называемый «остаточный слой» (см. рисунок 2). В таком блоке активации с некоторого слоя сети «пропускают» несколько свёрток и без изменений добавляются к результату применения этих свёрток. Такой подход решает проблему затухающих градиентов в силу того, что у каждого слоя нейросети есть доступ к непреобразованному градиенту в процессе обратного распространения ошибки.

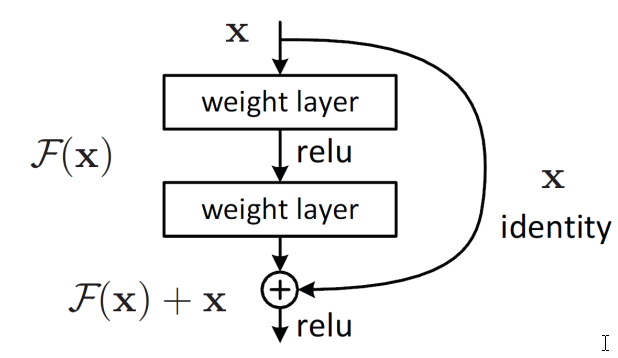


Рисунок 2: Остаточный слой

Остаточные блоки позволили обучать очень глубокие нейросети. Например, у архитектуры ResNet, победившей в соревновании ImageNet в 2015 году, было 152 слоя. Архитектура ResNet очень быстро стала наиболее популярной архитектурой, используемой при переносе обучения, а её модификации до сих пор являются SOTA для многих задач компьютерного зрения.

Опишем данный механизм residual связи. Пусть стандартный слой на вход принимает результат работы предыдущих 𝑡 слоёв сети — представление ℎ𝑡. Тогда выход сети можно задать в следующем виде: ℎ𝑡+1 = 𝑓(ℎ𝑡, 𝜃𝑡),

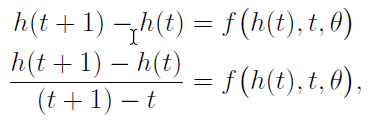
Где 𝑓(ℎ𝑡, 𝜃𝑡),— это какое-то нелинейное преобразование, задающееся на практике с помощью полносвязанных, свёрточных слоёв, их комбинаций и нелинейности.

Пусть теперь 𝑓(ℎ𝑡, 𝜃𝑡) совпадает по размерности с ℎ𝑡. Тогда слоем с residual связью называют следующее преобразование: ℎ𝑡+1 = 𝑓(ℎ𝑡, 𝜃𝑡) + ℎ𝑡, где суммирование происходит поэлементно.

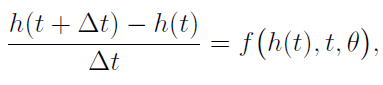
Как было замечено авторами архитектуры ResNet, описанный слой позволяет сети лег-ко «сохранять» информацию с предыдущих слоёв. Благодаря этому удалось существенно увеличить глубину сетей, не теряя при этом качества [1].

## Нейронные обыкновенные дифференциальные уравнения

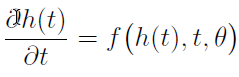
Рассмотрим уравнение ℎ𝑡+1 = 𝑓(ℎ𝑡, 𝜃𝑡) + ℎ𝑡 и возьмём представление ℎ𝑡 как функцию от 𝑡. Тогда, перенеся ℎ𝑡 в левую часть и добавив в функцию 𝑓(·) зависимость от 𝑡, получим:



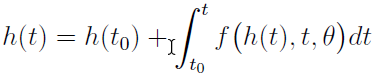
Если рассматривать функцию ℎ(𝑡) как функцию от непрерывного аргумента на некотором отрезке [𝑡0, 𝑡1], то уравнение можно переписать в следующем виде:



и, устремив число слоёв к бесконечности, получим:



Данное уравнение является обыкновенным дифференциальным уравнением (ОДУ). ОДУ с начальным условием ℎ(𝑡0) = ℎ𝑡0 называется задачей Коши и по теореме о существовании и единственности решения задачи Коши ℎ(𝑡) представимо в виде:



При этом значение ℎ(𝑡) ищется приблизительно, с заданной точностью, с помощью численного решения дифференциального уравнения (например методами Adams или Рунге-Кутта):

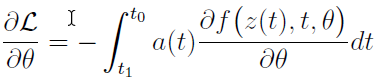


Рассмотрим теперь как оптимизировать выучиваемые параметры 𝜃 функции 𝑓(・) при наличии функции потерь ℒ(︀𝑧(𝑡1))︀. Стандартно, оптимизация ведётся с помощью метода градиентного спуска. Для этого необходимо знать градиент .

Для подсчёта градиента вводится вспомогательное понятие сопряжённого (adjoint): 𝑎(𝑡) = , которое показывает, как зависит функция потерь от скрытого состояния.

Для данного уравнения известно начальное состояние в точке 𝑡1. Соответственно, чтобы найти 𝑎(𝑡) в любой точке необходимо решить задачу Коши. Решение вычисляется в обратную сторону, от 𝑡1 к 𝑡0, что соответствует обратному распространению ошибки при обучении нейросетей.

Градиент по параметрам вычисляется следующим образом:



Зная 𝑎(𝑡) в нужных точках, значение этого интеграла считается численными методами.

Заметим, что для решения данной задачи Коши можно использовать любой метод для приближённого решения ОДУ. Одним из преимуществ такого подхода является тот факт, что для решения ОДУ разработаны точные и эффективные методы, которые позволяют явно задавать желаемую точность решения. От точности решения обычно зависит количество оцениваний динамики изменения скрытого состояния с помощью функции f: чем выше заданная точность, тем больше методу требуется обращений к функции f. Таким образом, возникает некоторый компромисс между желаемой точностью и временем обучения, что позволяет адаптировать сложность модели под задачу и имеющиеся ресурсы

Рассмотрим теперь основные достоинства предложенного подхода по сравнению со стандартными нейросетями:

* Использование численных методов при решении задачи Коши позволяет выбирать баланс между точностью решения и временем вычисления;
* Задание нейросетевого ОДУ требует меньшего количество параметров, так как residual сеть должна учить каждый слой по отдельности, а здесь все «слои» связаны одним дифференциальным уравнением, вид которого оптимизируется;
* Решение задачи Коши методом Adams занимают константную память, в отличии от residual сетей, линейно зависящий от числа слоёв. Память является существенным ограничением на размер сети при вычислениях на современных устройствах.

## Преимущества ODE

Модель нейродифференциального уравнения имеет ряд преимуществ по сравнению с обычными нейронными сетями:

* **Эффективное по памяти обучение.** Отсутствие необходимости хранить промежуточные значения, вычисленные в ходе прямого прохода, позволяет обучать модель с независящими от «глубины» затратами по памяти. Под «глубиной» понимается количество вычислений функции в промежуточных точках.
* **Адаптивное время работы.** Современные численные методы решения дифференциальных уравнений позволяют явно указывать желаемую точность численного решения, подбирая требуемую дискретность сетки самостоятельно (Рис. 3 a). Естественно, с измельчением сетки возрастает и время работы метода (Рис. 3 b). Таким образом, существует явный способ контролировать соотношение между численной точностью решения и вычислительными затратами. Данное свойство метода позволяет обучать модель с высокой точностью, а во время тестирования намеренно понижать её, тем самым сокращая объём требуемых вычислительных ресурсов. Это может быть полезно, например, для запуска модели на устройстве с низкой вычислительной мощностью (например, телефоне).

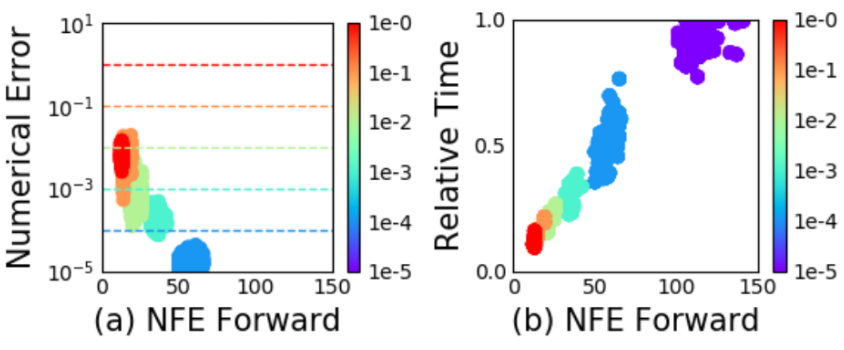


Рис. 3: (а) Численная ошибка уменьшается с увеличением количества промежуточных вычислений; (b) Время работы растёт с увеличением количества промежуточных вычислений. (NFE - number of function evaluations)

* **Непрерывные нормализационные потоки**. Идея нейродифференциальных уравнений, примененная к модели нормализационных потоков [3], породила новый класс моделей, который авторы статьи назвали «Непрерывные нормализационные потоки».

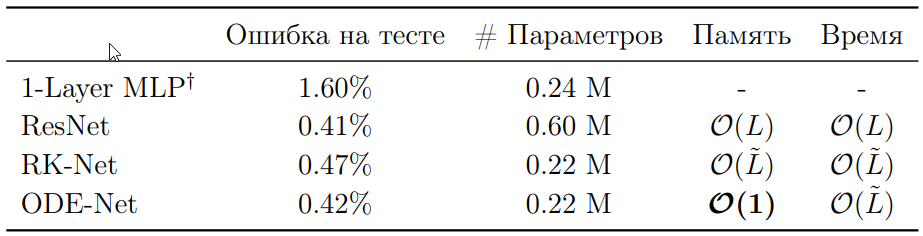
## Применение нейронных ОДУ

Имея алгоритм для обучения параметров нейросети, задающей неройнное ОДУ, можно применять описанную модель для решения привычных задач машинного обучения. Рассмотрим несколько примеров, приведенных авторами статьи [4].

## Замена остаточных нейросетей нейронными ОДУ в задаче обучения с учителем

Нейронные ОДУ могут быть встроены в архитектуру нейронных сетей подобно свёрточным или остаточным слоями. Становится возможным сравнение качества нейронной сети, состоящей из нескольких остаточных слоёв, и нейронной сети, в которой все остаточные слои заменены на одно нейронное ОДУ. Результаты экспериментов авторы статьи [4] приводят в таблице 1 ниже.

Таблица 1: Результаты экспериментов на MNIST. L обозначает количество слоёв в сети ResNet. L˜ обозначает количество оцениваний скрытого состояния при приближённом решении ОДУ.



RK-Net в таблице обозначает нейронное ОДУ, обучение которого происходило напрямую через подсчёт градиентов всех вычислительных операций в использующемся методе для решения ОДУ. Как видно из таблицы, качество такой нейросети хуже.

Нейронное ОДУ, обучаемое с помощью вычисления сопряжённого состояния, показывает лучшее качество, имея константные затраты по памяти . Более того, в ходе проведения экспериментов авторы статьи выяснили, что количество обращений к функции динамики при обратном проходе во время обучения примерно в два раза меньше, чем во время прямого прохода, что свидетельствует о том, что предложенный метод оптимизации не только более эффективен по памяти чем прямой подсчёт градиентов, но ещё и требует гораздо меньше операций.

Как было указано ранее, увеличение глубины традиционных нейросетей позволило существенно улучшить качество их работы. Возникает закономерный вопрос: будет ли прирост в качестве при объединении в одной нейросети нескольких блоков нейронных ОДУ? Для этого авторы статьи провели эксперименты. Рассмотрим нейросетевую архитектуру с одним блоком нейронного ОДУ, использованную в статье, и добавим ещё один блок после первого. Сравнение качества двух моделей можно видеть на рисунке 4.

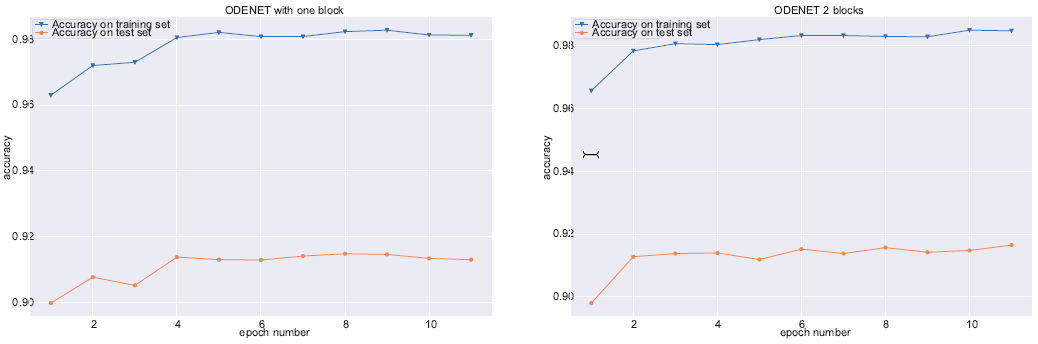


Рисунок 4: Качество двух моделей. Добавление второго ODE блока не даёт прироста в качестве, при этом увеличивая работы модели приблизительно в два раза. (a) Качество модели с одним ODE блоком. Количество параметров: 233k. (b) Качество модели с двумя ODE блоками. Количество параметров: 308k

При добавлении дополнительного блока количество параметров в сети возрастает до 308 тысяч. При этом среднее время работы на одном мини-батче увеличивается вдвое, так как теперь происходит в два раза больше обращений к методу для приближённого решения ОДУ. Однако прироста в качестве не наблюдается. Если рассматривать ResNet как схему Эйлера, то увеличение количества слоёв даёт более точную оценку на динамику, чем можно объяснить улучшение качества. При последовательном использовании блоков нейронных ОДУ, которые описывают непрерывное преобразование скрытого представление, отсутствие прироста можно объяснить тем, что выразительной способности функции, параметризующей динамику, хватает, чтобы описать преобразование скрытого представления в рамках одного блока. Второй блок в таком случае бесполезен.

## **Масштабируемые и обратимые нормализующие потоки.**

В современных генеративных моделях остро стоит проблема mode collapsing. Генератор выдает ограниченное количество разных образцов, не покрывая даже обучающую выборку. Это связано с тем, что GAN реализует неявную вероятностную модель, то есть доступа к плотности нет. Это сделано для возможности приближения сложных распределений, при этом распределение моделируется нейронной сетью. Но такая нейронная сеть может лишь сэмплировать точки, не выдавая их плотности. Чтобы строить сложные распределения и иметь доступ к плотности, можно брать случайную величину с простым распределением и преобразовывать её с помощью непрерывных и обратимых преобразований. Тогда для вычисления в точке плотности нового распределения необходимо знать только плотность в этой точке старого распределения и якобиан преобразования. Вычисление якобиана имеет сложность O(D3 ) , где D — размерность пространства. Имеются преобразования с сильными ограничениями, вычисление якобиана которых имеет меньшую сложность. Но из-за ограничений этих преобразований получение сложных распределений из простых становится трудоёмким. Модель, представленная в статье, позволяет находить плотность в точке нового распределения за линейное время. Более того, нет ограничений на преобразования. Единственное ограничение: шаги по t должны быть достаточно маленькими.

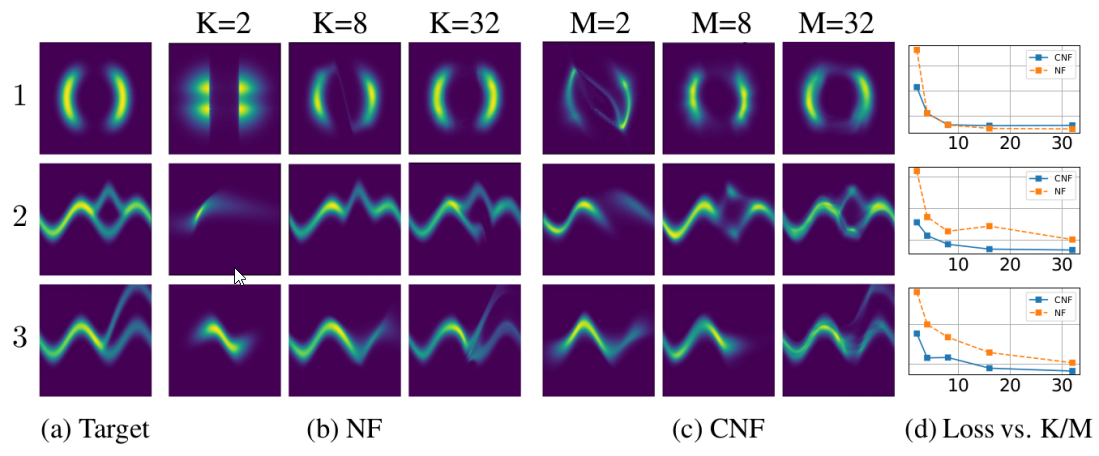


Рисунок 5. Сравнение дискретных и непрерывных нормализующих потоков. K - глубина дискретного потока, M - «ширина» непрерывного потока

### 4.3.Генеративные модели временных рядов

Пусть имеется временной ряд xt1 , . . . , xtN , . . . , xtM . Причем значения xt1 , . . . , xtN мы наблюдаем — это обучающая выборка, а значения xtN+1 , . . . , xtM предсказываем — тестовая выборка. Рассматривается следующая модель. Обучающая выборка подается на вход реккурентной нейросети (кодировщик). Кодировщик выдает параметры, как правило, нормального распределения: матожидание µ и стандартное отклонение σ. Из полученного распределения сэмплируется случайная величина zt0 ∼ q(zt0 |µ, σ). Значение zt0 подается на вход другой реккурентной нейросети (декодировщик), которая обучается восстанавливать значения xt1 , . . . , xtN , поданные на вход кодировщику. Затем генерируются оставшиеся «тестовые» значения. Реккурентная сеть использует дискретные преобразования вида z(t + 1) = z(t) + f(z(t), t), которые авторы статьи заменили на непрерывный аналог ОДУ-сетью. Как видно на рисунке 6, новый подход лучше восстанавливает рассматриваемые временные ряды. Кроме того, в силу непрерывности, имеется возможность достаточно просто получать значения временного ряда в любые моменты времени.

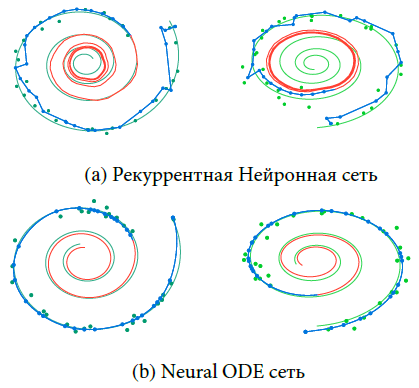


Рисунок 6. Сравнение реконструкции и интерполяции траекторий между Neural ODE и простой Рекуррентной сетью. Исходная траектория отмечена зеленой линией. Обучающая выборка отмечена красной линией.

## Применение в сфере информационной безопасности

Как было показано в разделе 4.1 изученный тип сетей хорошо подходит для анализа безопасности в больших сенсорных системах, SIEM (Security information and event management) систем и систем обнаружения сетевых атак, так как нейросетевые ОДУ требует меньшего количество параметров, в сравнении с остаточными сетями. Это позволяет более эффективно применять ODE нейросеть экономя память и время во время обучения и тестирования модели. По этой же причине данный вид сетей может эффективно решать задачи анализа ВПО.

Также рассмотренный тип нейронных сетей позволяет обучать модель с высокой точностью, а во время тестирования намеренно понижать её, тем самым сокращая объём требуемых вычислительных ресурсов. Это может быть полезно, например, для запуска модели на устройстве с низкой вычислительной мощностью. Что делает выгодным применение именно этого типа нейросетей в сфере IoT и мобильных устройств для решения как задач описанных выше , так и для более экономного решения задач нейросетей других типов, например: нейронные ODE смогут эффективно решить задачу распознавания объектов на фото, что можно использовать для реализации биометрической аутентификации по лицу пользователя.

Другим применением нейронных ODE могут являться системы безопасности в которых необходимо учитывать и рассчитывать физические величины, которые часто являются непрерывными величинами. Пример такой системы : система безопасности беспроводной сети, в которой необходимо учитывать физическое распространения сигнала, которое может быть эффективно рассчитано с применением нейронных ODE.

# **ЗАКЛЮЧЕНИЕ**

Нейродифференциальные(ODE) уравнения являются очень интересным классом моделей, который имеет свои несомненные достоинства. Первые эксперименты подтверждают применимость данных моделей на практике и демонстрируют сопоставимые с традиционными моделями результаты. Однако, поскольку данная область является достаточно новой и малоизученной, на текущий момент не было проведено полномасштабных экспериментов. Несмотря на то, что данное направление для исследований является очень перспективным, станет ли модель нейродифференциальных уравнений такой же популярной и широко используемой, как модель ResNet — пока что не известно. Но уже сейчас существуют возможности применения данного вида нейросетей в сфере информационной безопасности. Они описаны в разделе 5 и используют преимущества нейронных ОДУ описанные в разделе 3.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Deep Residual Learning for Image Recognition —Url: <https://arxiv.org/abs/1512.03385> — (Дата обращения: 05.03.2020).
2. Attention is All you Need—Url: <https://papers.nips.cc/paper/7181-attention-is-all-you-need> — (Дата обращения: 05.03.2020).
3. Variational inference with normalizing flows.—Url: <https://arxiv.org/abs/1505.05770> — (Дата обращения: 05.03.2020).
4. Neural Ordinary Differential Equations—Url: <https://arxiv.org/abs/1806.07366> — (Дата обращения: 05.03.2020).