RU EN ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ ISSN 2079-3316
научная статья ПРОГРАММНОЕ И АППАРАТНОЕ ОБЕСПЕЧЕНИЕ РАСПРЕДЕЛЕННЫХ И СУПЕРКОМПЬЮТЕРНЫХ СИСТЕМ

УДК 004.272.45
10.25209/2079-3316-2025-16-3-3-22

Методы квантового машинного обучения для обнаружения атак на программно-конфигурируемые сети

Илья Андреевич **Антонов**[1]✉, Илья Ильич **Курочкин**[2]

[1] Университет науки и технологий МИСИС, Москва, Россия
[2] Институт проблем передачи информации им. А. А. Харкевича РАН, Москва, Россия
[1]✉*m1908142@edu.misis.ru*

Аннотация. Программно-конфигурируемая сетевая архитектура является предпочтительным способом построения больших компьютерных сетей, требующих высокой скорости реагирования на изменения и высокой степени автоматизации. Основной особенностью данной архитектуры является централизованное управление всей сетью с одного контроллера. Тем не менее, такой подход открывает новые возможности для атак на сеть, делая контроллер их главной целью. В данной работе рассматривается возможность применения моделей квантового машинного обучения для обнаружения таких атак.

Ключевые слова и фразы: программно-конфигурируемые сети, информационная безопасность, машинное обучение, нейронные сети, квантовые вычисления, системы обнаружения вторжений, SDN, IDS

Для цитирования: Антонов И. А., Курочкин И. И. *Методы квантового машинного обучения для обнаружения атак на программно-конфигурируемые сети* // Программные системы: теория и приложения. 2025. **Т. 16**. № 3(66). С. 3–22. https://psta.psiras.ru/read/psta2025_3_3-22.pdf

Введение

В последние годы распространение сетевых атак и нарушений безопасности стало растущей проблемой как для организаций, так и для отдельных лиц [1]. Для защиты сетей от них разработаны системы обнаружения вторжений (СОВ, IDS). Однако традиционные системы обнаружения вторжений часто не могут идти в ногу с растущей сложностью и изощренностью современных атак. Следовательно, существует острая необходимость в инновационных подходах, которые могут повысить точность и эффективность обнаружения вторжений. Машинное обучение (ML) является одним из таких подходов, который показал многообещающие результаты в повышении точности и эффективности IDS. Системы обнаружения вторжений на основе ML оказались важным масштабируемым инструментом для защиты сетей от кибератак, они способны к самообучению, а также могут работать при относительно небольших мощностях с достаточной скоростью, в отличие от классических IDS [2].

Квантовые вычисления, являясь относительно молодой областью, предлагают значительное ускорение для многих задач за счет квантового параллелизма и других принципов квантовой физики, поэтому имеет смысл приложить их и к области ML. В данной работе рассматривается задача обнаружения вторжений в программно-конфигурируемых сетях с помощью методов квантового машинного обучения. В частности, рассматривается применение метода VQC, который пользуется большой популярностью у исследователей [3].

Сети традиционной архитектуры широко и успешно используются, но для больших сетей (например, дата-центров) необходима более высокая скорость реагирования на изменения и более высокая степень автоматизации, которые с помощью традиционной архитектуры не могут быть реализованы, даже при наличии высококлассных специалистов. Программно-конфигурируемая сеть (ПКС, SDN) – это концепция сети, которая отделяет плоскость управления от плоскости передачи данных [4]. Основной особенностью данной архитектуры является централизованное управление всей сетью с одного контроллера, что повышает гибкость сети и ее способность расти. Контроллер SDN может выполнять функции, связанные с управлением сетью, такие как конфигурация коммутаторов, упорядоченная доставка пакетов, получение статистических данных коммутатора и другие [5]. Архитектуру SDN можно представить схематично, как показано на рисунке 1 [6].

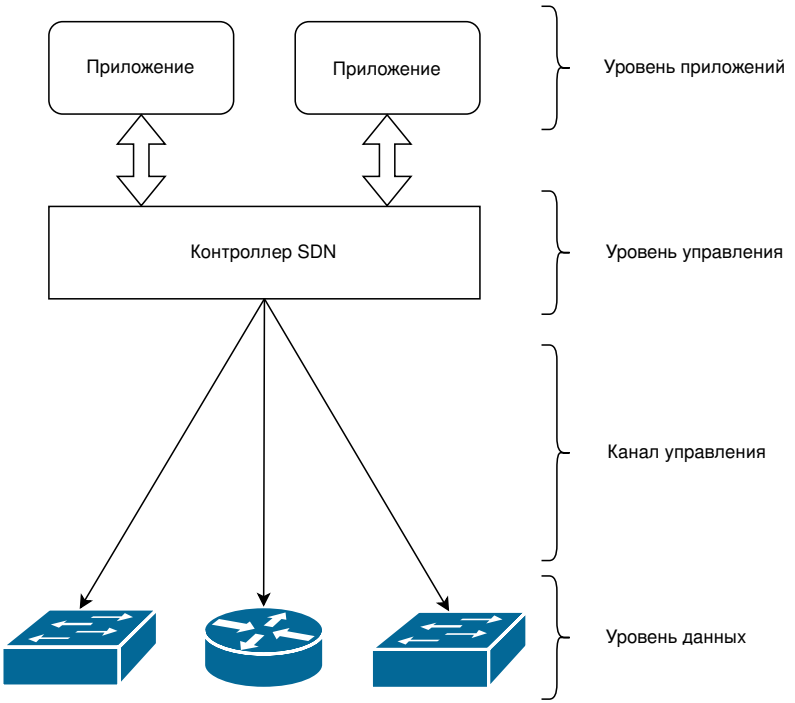

Рисунок 1. Архитектура SDN

Разделение уровня управления сетью и уровня передачи данных в сети SDN приводит к следующим преимуществам [7]:

- Отделение плоскости управления от плоскости данных облегчает управление сетью. Кроме того, сеть становится легче изменять и обновлять, что снижает количество человеческих ошибок.
- ИТ-администраторы могут легко добавлять сетевые устройства или модернизировать сетевую инфраструктуру, не привязываясь к конкретному поставщику оборудования.
- Низкоуровневые инфраструктурные устройства не требуют отдельного программирования, что значительно снижает эксплуатационные расходы по сравнению с обычной сетью.

Обнаружение вторжений – это задача классификации, распределяющая входящий сетевой трафик на классы: обычный и вредоносный [8]. Сами атаки на программно-конфигурируемые сети можно классифицировать на 4 категории [9]: удаленные атаки на плоскость управления сетью, локальные

атаки на плоскость управления сетью, атаки на канал управления сетью (особенно, протокол OpenFlow, используемый в SDN) и атаки на уровень передачи данных.

1. Особенности квантовых вычислений

Квантовые вычисления – это относительно новая область, которая объединяет компьютерные науки, математику и физику. Эта область исследует возможности использования принципов квантовой физики для создания квантовых компьютеров, использующих квантовые биты (кубиты), которые могут содержать комбинации значений 0 и 1 в суперпозиции одновременно.

Так как кубиты могут находиться одновременно в нескольких состояниях, временные затраты на расчет каждого состояния не требуются, а общее время расчётов уменьшается. Именно такая возможность позволяет решать некоторые сложные задачи быстрее классических компьютеров [10, 11].

Классический компьютер с памятью из n бит может одномоментно выполнить некоторую операцию только над одним из $2n$ возможных наборов. Чтобы вычислить значение некоторой булевой функции от n аргументов для всех значений аргументов с помощью классического компьютера, придётся по очереди перебирать все $2n$ наборов и вычислять значение функции на каждом наборе по отдельности. Благодаря квантовой суперпозиции, в квантовом компьютере с n кубитами можно одновременно представить все $2n$ набора и выполнять операции над всеми наборами сразу. В результате возможно вычислить значение функции сразу для всех $2n$ комбинаций значений аргументов [12].

Кубит можно определить, как вектор единичной длины в двумерном гильбертовом пространстве над полем комплексных чисел. Состояния 0 и 1 вместе представляют собой базисные вектора. Каждое из состояний кубита может быть представлено суперпозицией двух базисных состояний, которые обычно обозначаются как $|0\rangle$ и $|1\rangle$. Формула, представляющая кубит в суперпозиции, выглядит следующим образом:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

α и β - комплексные числа, известные как амплитуды. Сумма их квадратов должна быть равна 1, поэтому можно сказать, что они определяют вероятности обнаружения кубита в каждом из базисных состояний.

Однако создание и манипулирование кубитами само по себе не является достаточным для выполнения сложных операций. Для реализации квантовых алгоритмов необходимо уметь манипулировать состояниями кубитов с высокой степенью контроля. Это достигается с помощью квантовых вентилей [10]. Квантовые вентили представляют собой аналоги классических логических вентилей, которые выполняют различные операции над кубитами, такие как логические, унитарные и нелинейные. Квантовые вентили используются для создания квантовых цепей и выполнения последовательности операций, составляющих квантовый алгоритм. Благодаря квантовым вентилям получается достичь эффективности и гибкости квантовых вычислений.

Квантовые вентили можно представить в виде унитарных матриц, которые преобразуют входной кубит в выходной кубит. Для любого входного состояния кубита, квантовый вентиль выполняет преобразования в соответствии с определенными правилами. Например, вентиль NOT инвертирует состояние кубита. Тем не менее, квантовые вычисления на данный момент используются только в исследовательских целях и не распространены для массового использования. Рассмотрим основные препятствующие этому проблемы:

(*1*) Декогеренция представляет собой явление, при котором квантовая система взаимодействует с окружающей средой, что приводит к потере квантовых свойств и, следовательно, к ухудшению точности и стабильности вычислений [13]. Максимальное время жизни квантовой системы, когда она пригодна для квантовых вычислений, крайне мало, а по окончании этого времени система начнет выдавать белый шум вместо вероятностных распределений.

(*2*) Одним из наиболее актуальных вызовов является масштабируемость квантовых систем. В настоящее время существуют квантовые компьютеры с небольшим числом кубитов, однако для решения реальных задач требуется значительное увеличение их мощности [11]. Разработчикам необходимо искать пути увеличения числа кубитов и улучшения их производительности, чтобы обеспечить масштабируемость и применение квантовых систем для решения широкого спектра задач.

(*3*) Другим важным вызовом является сложность реализации квантовых систем. В отличие от классических вычислительных систем, которые имеют долгую историю и разработанную методологию, квантовые системы находятся на стадии активных исследований и разработок. Например, нет четких методик устранения “багов”, специфичных для

квантовых компьютеров, из-за чего во время их работы возникают ошибки [14].

Рассмотрим также основные приложения квантовых вычислений:

(1) Алгоритм Шора [15, 16] представляет собой квантовый алгоритм, способный эффективно факторизировать большие целые числа. Факторизация является процессом разложения числа на простые множители. На классических компьютерах, работающих на классических алгоритмах, факторизация достаточно больших чисел требует экспоненциального времени, что делает ее практически невозможной для чисел с достаточной длиной для применения в криптографических системах. Алгоритм Шора, используя возможности квантовых компьютеров, способен произвести факторизацию числа не просто за полиномиальное время, а за время, ненамного превосходящее время умножения целых чисел. Таким образом, с помощью этого алгоритма (при использовании квантового компьютера с несколькими тысячами логических кубитов) становится возможным взлом криптографических систем с открытым ключом.

(2) Алгоритм Гровера [17] позволяет найти нужный элемент базы данных за полиномиальное время, что делает его значительно более эффективным по сравнению с классическими алгоритмами. Это имеет потенциальное применение в таких областях, как поиск в больших базах данных, оптимизация и распознавание образов.

(3) Квантовые компьютеры могут обрабатывать большие матрицы, а также ускорять различные операции линейной алгебры, значительно улучшая традиционные приложения машинного обучения [3]. В частности, было показано, что при использовании методов квантового машинного обучения можно достичь улучшения производительности (увеличение значений метрик или скорости работы) по сравнению с классическими методами, в том числе и для решения задачи обнаружения вторжений в сетях SDN [18, 19].

2. Методы квантового машинного обучения

Рассмотрим основные методы квантового машинного обучения в сравнении с их классическими аналогами.

К-ближайших соседей (kNN) – это алгоритм классификации, состоящий из трех шагов: вычисление расстояния относительно обучающих элементов; нахождение k элементов, ближайших к тестовому экземпляру; предсказание метки класса посредством голосования большинства.

В квантовом kNN вначале все элементы переводятся в пространство квантовых векторов состояний, затем вычисляются расстояния между классифицируемым объектом и всеми объектами тренировочной выборки с помощью, например, евклидовой метрики [20], SWAP-теста [21] или других. Далее класс присваивается объекту по большинству, находящемуся в числе его ближайших соседей, как и в классическом kNN. Благодаря свойству суперпозиции кубитов все расстояния от обучающих элементов вычисляются одновременно (квантовый параллелизм), что дает большой прирост к скорости работы алгоритма [20].

Метод опорных векторов (SVM) с помощью функции-ядра переводит выборку в пространство более высокой размерности, в котором данные становятся линейно разделимыми по классам, и проводит между ними разделяющую гиперплоскость. Квантовый SVM работает схожим образом: сначала происходит кодирование данных в квантовое пространство, затем к ним применяется квантовое ядро для перевода в пространство большей размерности, после чего проводится разделяющая гиперплоскость и выполняется классификация. В качестве методов оптимизации может выступать алгоритм Гровера [22] или алгоритм HHL [23], которые используют квантовый параллелизм и дают существенный прирост в скорости работы [24].

В квантовых сверточных нейронных сетях (QCNN) [25] изображение сначала кодируется в квантовую схему с использованием карты признаков. Затем к закодированному изображению применяются чередующиеся сверточные и объединяющие слои, уменьшая размерность схемы до тех пор, пока не останется только один кубит. Выход этого оставшегося кубита измеряется для классификации входного изображения. Квантовый сверточный слой состоит из серии двухкубитных унитарных операторов, которые распознают и определяют связи между кубитами в схеме; квантовый pooling-слой уменьшает количество кубитов, выполняя операции над каждым кубитом до определенной точки, а затем отбрасывая определенные кубиты в определенном слое. В QCNN каждый слой содержит параметризованные схемы, что означает, что выход может быть изменен путем настройки параметров каждого слоя. Во время обучения эти параметры корректируются для минимизации функции потерь.

Метод VQC (variational quantum circuit), использованный в данной работе, наиболее популярен у исследователей. Концептуально его показывает в виде схемы рисунок 2, навеянный рисунком 13 из [3]. Метод VQC представляет собой квантовый аналог нейронной сети. На входе квантовой

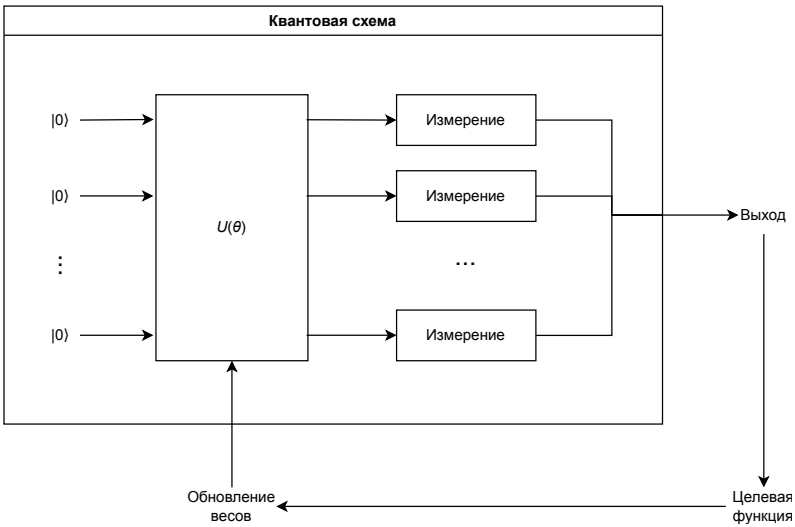

Рисунок 2. Представление метода VQC

схемы (Quantum circuit) расположено некоторое количество кубитов, соответствующее количеству признаков в наборе данных, а на выходе формируется решение задачи (например, вероятностное распределение для задачи классификации). Предварительно переведенные в пространство квантовых векторов (с помощью карты признаков) данные поступают на вход этой схеме, после чего она «обучается». Процесс обучения происходит с задействованием классического компьютера: на нем рассчитывается функция потерь и обновляются веса для квантовой схемы, после чего квантовая схема снова начинает работу.

3. Набор данных

В данной работе используется набор данных InSDN [7], опубликованный в 2020 году и содержащий различные виды трафика, актуального на сегодняшний день (YouTube, чаты, электронная почта и др.) и характерного для ПКС. Набор данных содержит типы сетевого трафика, представленные в таблице 1. Набор данных несбалансирован, поэтому для балансировки набора данных в некоторых экспериментах использовался алгоритм SMOTE (Synthetic Minority Over-sampling Technique), основная идея которого – генерация искусственных объектов в наборе данных, «похожих» на существующие, но не дублирующих их [26, 27].

Таблица 1. Виды трафика в наборе данных InSDN

Тип трафика	Количество записей
Нормальный	68424
Атака BFA	1405
Атака DDoS	121942
Атака DoS	53616
Атака Probe	98129
Атака Web-Attack	192
Атака BOTNET	164
Атака U2R	17

Учитывая, что для симуляции квантовых методов машинного обучения на классическом компьютере требуются значительные ресурсы, количество признаков в наборе данных было сокращено с 48 исходных до 10 с помощью функционала SelectKBest из библиотеки Scikit-Learn.

Оставшиеся признаки для экспериментов с задачей бинарной классификации: Protocol, Fwd Pkt Len Min, Bwd Pkt Len Min, Bwd Pkt Len Mean, Bwd Pkt Len Std, Fwd IAT Tot, Pkt Len Min, Pkt Len Mean, Pkt Len Std, Pkt Size Avg.

Оставшиеся признаки для экспериментов с задачей многоклассовой классификации: Protocol, Bwd Pkt Len Min, Bwd Pkt Len Mean, Bwd Pkt Len Std, Flow Pkts/s, Bwd Pkts/s, Pkt Len Min, Pkt Len Mean, Pkt Len Std, Pkt Size Avg.

Помимо сокращения количества признаков набор данных также был разделен на 3000 порций, и на вход методу VQC подавалось по одной порции за раз для минимизации использования памяти. Это тоже повлияло на значение метрики, поскольку качество модели, обучаемой на полном наборе данных, как правило, выше, чем для модели, обучаемой на том же наборе, но порционно.

4. Эксперименты и результаты

В процессе работы были проведены эксперименты с моделью квантового машинного обучения VQC и моделью классического машинного обучения XGBoost для сравнения. Эксперименты проводились на машине со следующими характеристиками: процессор Intel Xeon E5-2695 v2, 32 Гб оперативной памяти, ОС Windows 10.

Результаты работы моделей представлены в таблице 2. Модель XGBoost была взята как эталон, поскольку метод градиентного бустинга показывает лучшие результаты классификации табличных данных в большинстве случаев.

Для каждой модели было проведено 6 экспериментов:

(1) Бинарная классификация (нормальный или атакующий трафик).

(2) Бинарная классификация (нормальный или атакующий трафик) с предварительной нормализацией набора данных с помощью метода минимакс.

(3) Бинарная классификация (нормальный или атакующий трафик) с предварительной нормализацией набора данных с помощью метода минимакс и балансировкой с помощью метода SMOTE.

(4) Многоклассовая классификация (нормальный трафик или один из нескольких типов атак).

(5) Многоклассовая классификация (нормальный трафик или один из нескольких типов атак) с предварительной нормализацией набора данных с помощью метода минимакс.

(6) Многоклассовая классификация (нормальный трафик или один из нескольких типов атак) с предварительной нормализацией набора данных с помощью метода минимакс и балансировкой с помощью метода SMOTE.

Реализация модели VQC взята из библиотеки *Qiskit Machine Learning* для языка Python, также в работе были использованы инструменты из библиотеки *Qiskit*[1]. Модель *VQC* была взята с параметрами по умолчанию, в качестве метода оптимизации используется *COBYLA* (Constrained Optimization By Linear Approximation optimizer), в качестве карты признаков использована *ZZFeatureMap*, в качестве квантовой схемы использована *RealAmplitudes*. Модель XGBoost была взята из одноименной библиотеки *XGBoost* с параметрами по умолчанию.

Для оценки качества моделей были использованы меры accuracy и F1. Значение accuracy есть отношение количества правильно классифицированных объектов к общему количеству элементов выборки, а значение F1-меры сочетает в себе две другие функции оценки качества: precision (доля объектов, действительно принадлежащих данному классу

[1]Прискорбно, что фирма IBM закрыла доступ к своим сайтам (*прим. ред.*)

относительно всех объектов, которые модель отнесла к этому классу) и recall (доля найденных классификатором объектов, принадлежащих классу, относительно всех объектов этого класса), что облегчает понимание, насколько качественно работает модель.

Для запуска экспериментов для модели VQC не был использован квантовый компьютер, все вычисления выполнялись с помощью симуляции квантовой архитектуры на классическом компьютере. Из-за этого появились ограничения по скорости работы моделей и используемой памяти: было принято решение сократить количество признаков (как уже говорилось в разделе «набор данных») и сам набор данных разделить на небольшие части и дообучать модель итеративно. Перечисленные ограничения сказались на времени обучения модели (как мы видим, оно на несколько порядков выше по сравнению с классическим методом) и значениях метрик.

Модель XGBoost обучалась также на наборе данных с сокращенным количеством признаков, но на всем наборе данных целиком, а не порциями, поскольку градиентный бустинг не поддерживает механизм дообучения и требует на вход полный набор данных за раз.

Из результатов экспериментов мы видим, что нормализация набора данных приводит к увеличению значений метрик VQC, а также балансировка набора данных методом SMOTE в эксперименте с многоклассовой классификацией, поскольку в этом случае присутствует несколько классов, представленных крайне малым количеством объектов. Нормализация не оказывает сильного влияния на классический метод XGBoost в силу особенностей его архитектуры, а балансировка набора данных не увеличивает значения метрик для обеих моделей в эксперименте с бинарной классификацией, поскольку в этом случае оба класса представлены достаточным количеством объектов.

Для экспериментов с нормализацией и балансировкой набора данных приведем графики, отражающие динамику изменения значений метрик во время обучения модели VQC по порциям (рисунки 3 и 4). Среди результатов экспериментов особое внимание хочется обратить на значения accuracy в экспериментах с бинарной классификацией (выделены жирным шрифтом в таблице 2).

Таблица 2. Результаты экспериментов

Эксперимент	VQC			XGBoost		
	Accuracy	F1-мера	время работы	Accuracy	F1-мера	время работы
Бинарная классификация	0,771	0,656	721985 с	0,982	0,970	1,622 с
- она же после минимакс	**0,916**	0,841	722345 с	**0,982**	0,970	1,639 с
- она же после минимакс и SMOTE	**0,915**	0,841	954277 с	**0,980**	0,968	7,703 с
Многоклассовая классификация	0,423	0,170	835228 с	0,976	0,824	26,068 с
- она же после минимакс	0,480	0,201	836823 с	0,976	0,824	22,133 с
- она же после минимакс и SMOTE	0,624	0,210	984248 с	0,958	0,702	81,959 с

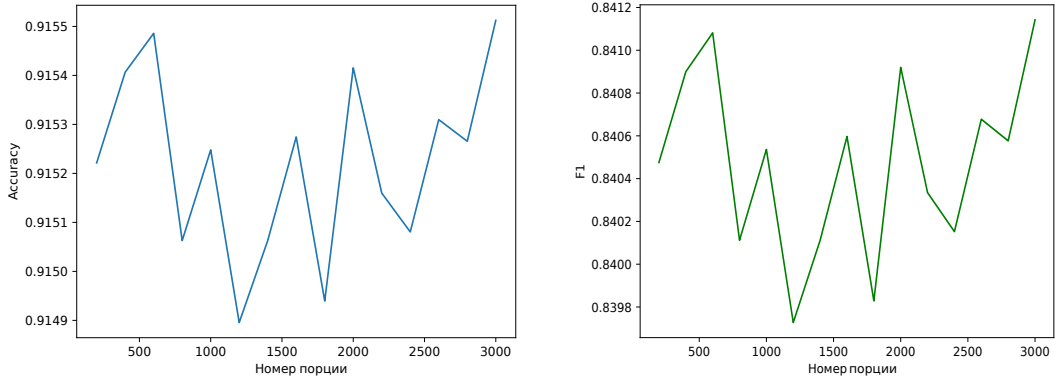

Рисунок 3. Динамика изменения метрик accuracy и F1 во время обучения модели VQC в экспериментах с бинарной классификацией на нормализованных данных

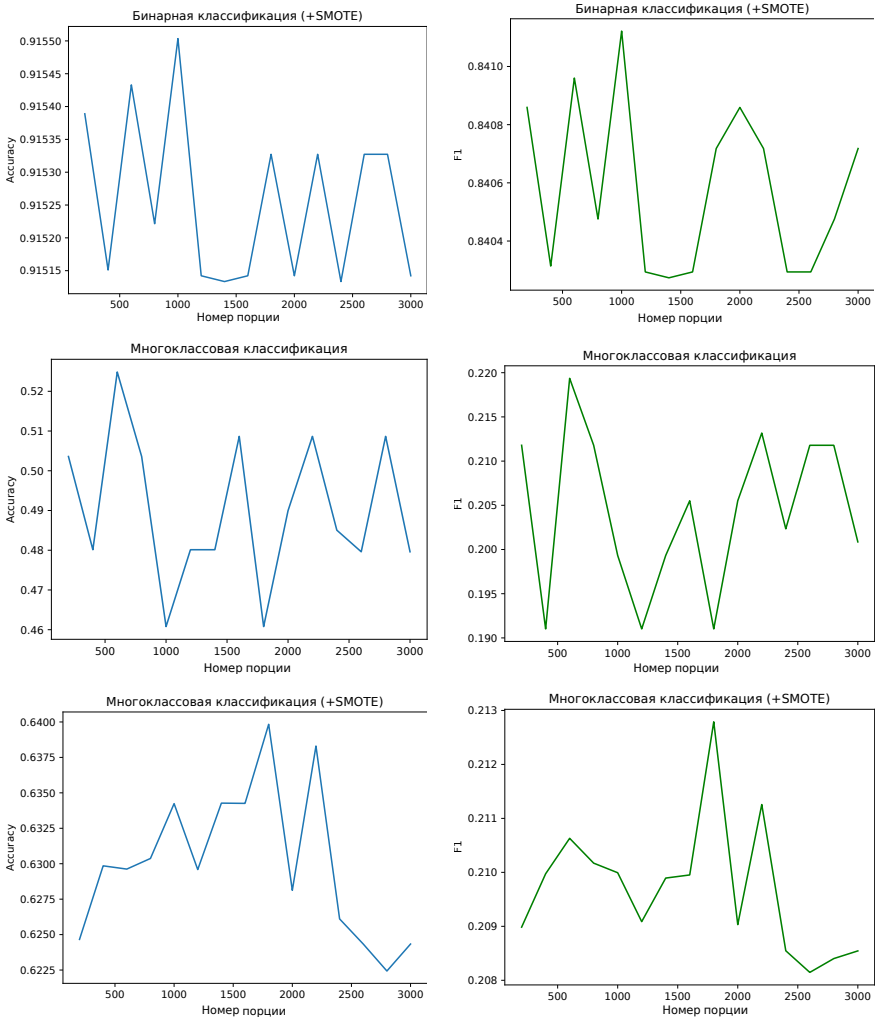

Рисунок 4. Динамика изменения метрик accuracy и F1 во время обучения модели VQC в экспериментах на нормализованных данных

Пока результат работы квантового метода VQC хуже, чем эталонного для задачи XGBoost, но мы видим, что уже получается приблизиться к эталону, результаты работы моделей сопоставимы. При будущих доработках квантовых компьютеров вполне возможно, что квантовые

методы смогут превзойти классические, как по значениям метрик, так и по скорости работы.

Заключение

Проведенная работа показала, что квантовые методы машинного обучения уже сейчас можно ограниченно применять и исследовать. На данный момент они требуют гораздо больше ресурсов для симуляции на классическом компьютере, чем привычные нам классические методы, но при большем распространении квантовых компьютеров и увеличении числа кубитов в них QML-методы смогут составить конкуренцию традиционным ML-моделям (а возможно и превзойти их) благодаря квантовому параллелизму и существенному увеличению скорости работы. По этой причине уже сейчас необходимо изучать их, в частности, применительно к задаче обнаружения вторжений в сетях SDN, поскольку в будущем СОВ на основе методов квантового машинного обучения вероятно смогут анализировать гораздо большие объемы информации с большей скоростью, что сделает их значительно эффективнее, чем традиционные IDS.

В данной работе проведен ряд экспериментов для метода квантового машинного обучения VQC и показано, что он может достигать результатов, сопоставимых с классическим методом градиентного бустинга, который является эталоном для задачи классификации табличных данных. При запуске модели VQC на квантовом компьютере, а не в симуляции, можно получить и прирост в скорости работы, помимо приемлемых результатов метрик.

Полученный в работе результат не является конечным, можно выделить два направления исследований для его улучшения. Во-первых, можно улучшать производительность текущей модели VQC путем подбора гиперпараметров и дополнительной предобработки данных. Во-вторых, можно рассмотреть и другие методы квантового машинного обучения, такие как QSVM, квантовые нейронные сети, гибридные квантово-классические архитектуры. Движение в обоих направлениях может привести к более эффективному решению задачи обнаружения вторжений в программно-конфигурируемых сетях.

Список использованных источников

[1] Kukliansky A., Orescanin M., Bollmann C., Huffmire T. *Network anomaly detection using quantum neual networks on noisy quantum computers* // IEEE Transactions on Quantum Engineering.– 2024.– Vol. **5**.– Pp. 1–11. ↑4

[2] Кажемский М. А., Шелухин О. И. *Многоклассовая классификация сетевых атак на информационные ресурсы методами машинного обучения* // Труды учебных заведений связи.– 2019.– Т. **5**.– № 1.– С. 107–115. ↑4

[3] Zeguendry A., Jarir Z., Quafafou M. *Quantum machine learning: A review and case studies* // Entropy.– 2023.– Vol. **25**.– No. 2.– id. 287.– 41 pp. ↑4, 8, 9

[4] Volkov S. S., Kurochkin I. I. *Extraction of traffic features in software-defined networks using an SDN-controller*, Conference: 9th International Conference "Distributed Computing and Grid Technologies in Science and Education" (Dubna, Russia, July 5–9, 2021), CEUR Workshop Proceedings.– vol. **3041**.– 2021.– Pp. 553–557. ↑4

[5] Фёдоров Н. К. *Программно-конфигурируемые сети. Проблемы при переходе к сетям ПКС* // StudNet.– 2022.– Т. **5**.– № 4.– С. 3137–3148. ↑4

[6] Forbacha S. C., Kinteh M. K., Hamza E. M. *Enhanced attacks detection and mitigation in software defined networks* // American Journal of Computing and Engineering.– 2024.– Vol. **7**.– No. 3.– Pp. 40–80. ↑4

[7] Elsayed M. S., Le-Khac N. A., Jurcut A. D. *InSDN: A novel SDN intrusion dataset* // IEEE Access.– 2020.– Vol. **8**.– Pp. 165263–165284. ↑5, 10

[8] Wu Z., Wang J., Hu L., Zhang Zh., Wu H. *A network intrusion detection method based on semantic Re-encoding and deep learning* // Journal of Network and Computer Applications.– 2020.– Vol. **164**.– id. 102688. ↑5

[9] Yoon C., Lee S., Kang H., Park T., Shin S., Yegneswaran V., Porras Ph., Gu G. *Flow wars: Systemizing the attack surface and defenses in software-defined networks* // IEEE/ACM Transactions on Networking.– 2017.– Vol. **25**.– No. 6.– Pp. 3514–3530. ↑5

[10] Vedral V., Plenio M. B. *Basics of quantum computation* // Progress in Quantum Electronics.– 1998.– Vol. **22**.– No. 1.– Pp. 1–39. ↑6, 7

[11] Yang Z., Zolanvari M., Jain R. *A survey of important issues in quantum computing and communications* // IEEE Communications Surveys & Tutorials.– 2023.– Vol. **25**.– No. 2.– Pp. 1059–1094. ↑6, 7

[12] Торгаев С. Н., Шульга И. Д., Юрченко Е. А., Громов М. Л. *Основы квантовых вычислений*, учебное пособие.– Томск: STT.– 2020.– ISBN 978-5-93629-656-7.– 88 с. ↑6

[13] Bacciagaluppi G. *The role of decoherence in quantum mechanics* // *The Stanford Encyclopedia of Philosophy*, substantive revision Thu Jan 23, 2025, eds. Edward N. Zalta, Uri Nodelman.– Metaphysics Research Lab, Stanford University.– 2025. ↑7

[14] Luo J., Zhao P., Miao Zh., Zhao J. *A comprehensive study of bug fixes in quantum programs* // *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering*, SANER (Honolulu, HI, USA, 15–18 March 2022).– IEEE.– 2022.– ISBN 978-1-6654-3786-8.– Pp. 1239–1246. ↑8

[15] Shor P. W. *Algorithms for quantum computation: discrete logarithms and factoring* // *Proceedings 35th Annual Symposium on Foundations of Computer Science* (Santa Fe, NM, USA, 20–22 November 1994).– IEEE.– 1994.– ISBN 0-8186-6580-7.– Pp. 124–134. ↑8

[16] Shor P. W. *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer* // SIAM Journal on Computing.– Т. **26**.– № 5.– С. 1484–1509. arXiv quant-ph/9508027 ↑8

[17] Grover L. K. *A fast quantum mechanical algorithm for database search* // *STOC'96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (Philadelphia, Pennsylvania, USA, 22–24 May 1996).– 1996.– ISBN 978-0-89791-785-8.– Pp. 212–219. ↑8

[18] Tychola K. A., Kalampokas T., Papakostas G. A. *Quantum machine learning — an overview* // Electronics.– 2023.– Vol. **12**.– No. 11.– id. 2379.– 21 pp. ↑8

[19] Kalinin M., Krundyshev V. *Security intrusion detection using quantum machine learning techniques* // Journal of Computer Virology and Hacking Techniques.– 2023.– Vol. **19**.– No. 1.– Pp. 125–136. ↑8

[20] Zardini E., Blanzieri E., Pastorello D. *A quantum k-nearest neighbors algorithm based on the Euclidean distance estimation* // Quantum Machine Intelligence.– 2024.– Vol. **6**.– No. 1.– id. 23.– 22 pp. ↑9

[21] Basheer A., Afham A., Goyal S. K. *Quantum k-nearest neighbors algorithm.*– 2021.– 21 pp. arXiv 2003.09187 ↑9

[22] Anguita D., Ridella S., Rivieccio F., Zunino R. *Quantum optimization for training support vector machines* // Neural Networks.– 2003.– Vol. **16**.– No. 5–6.– Pp. 763–770. ↑9

[23] Harrow A. W., Hassidim A., Lloyd S. *Quantum algorithm for linear systems of equations* // Physical review letters.– 2009.– Vol. **103**.– No. 15.– id. 150502. ↑9

[24] Yang J., Awan A. J., Vall-Llosera G. *Support vector machines on noisy intermediate scale quantum computers.*– 2019.– 12 pp. arXiv 1909.11988 ↑9

[25] Meedinti G. N., Srirekha K. S., Delhibabu R. *A quantum convolutional neural network approach for object detection and classification.*– 2023.– 16 pp. arXiv 2307.08204 ↑9

[26] Chawla N. V., Bowyer K. W., Hall L. O., Kegelmeyer W. P. *SMOTE: Synthetic minority over-sampling technique* // Journal of Artificial Intelligence Research.– 2002.– Vol. **16**.– No. 1.– Pp. 321–357. ↑10

[27] Volkov S. S., Kurochkin I. I. *Network attacks classification using Long Short-term memory based neural networks in Software-Defined Networks* // Procedia Computer Science.– 2020.– Vol. **178**.– Pp. 394–403. ↑10

Поступила в редакцию 10.03.2025;
одобрена после рецензирования 26.04.2025;
принята к публикации 22.05.2025;
опубликована онлайн 18.07.2025.

Рекомендовал к публикации *к.т.н. В. П. Фраленко*

Информация об авторах:

Илья Андреевич Антонов

Выпускник магистратуры Национального исследовательского технологического университета МИСИС по направлению 09.04.03 Прикладная информатика. Области научных интересов: методы машинного обучения, нейронные сети, квантовые вычисления.

0009-0002-9820-1558

e-mail: *m1908142@edu.misis.ru*

Илья Ильич Курочкин

Кандидат технических наук, заведующий лабораторией Ц-1 Института проблем передачи информации им. А.А. Харкевича Российской академии наук. Области научных интересов: моделирование телекоммуникационных сетей, облачные и распределенные вычисления, распределенное глубокое обучение.

0000-0002-0399-6208

e-mail: *kurochkin@iitp.ru*

Авторы внесли равный вклад в подготовку публикации.

Декларация об отсутствии личной заинтересованности: *благополучие авторов не зависит от результатов исследования.*

PROGRAM SYSTEMS: THEORY AND APPLICATIONS ISSN 2079-3316
Research Article HARDWARE AND SOFTWARE FOR DISTRIBUTED AND SUPERCOMPUTER SYSTEMS

UDC 004.272.45
10.25209/2079-3316-2025-16-3-3-22

Quantum machine learning methods for intrusion detection in software-defined networks

Il'ya Andreevich **Antonov**[1], Il'ya Il'ich **Kurochkin**[2]

[1] National University of Science and Technology MISIS, Moscow, Russia
[2] Institute for Information Transmission Problems of RAS, Moscow, Russia
[1]✉*m1908142@edu.misis.ru*

Abstract. Software-defined network architecture is the preferred way to build large computer networks that require high responsiveness to change and a high degree of automation. The main feature of this architecture is the centralized management of the entire network from a single controller. However, this approach opens new opportunities for attacks on the network, making the controller their main target. This paper explores the possibility of applying quantum machine learning models to detect such attacks. (*In Russian*).

Key words and phrases: software-defined networks, information security, machine learning, neural networks, quantum computing, intrusion detection systems, SDN, IDS

2020 *Mathematics Subject Classification*: 65Y05; 68Q10

For citation: Il'ya A. Antonov, Il'ya I. Kurochkin. *Quantum machine learning methods for intrusion detection in software-defined networks*. Program Systems: Theory and Applications, 2025, **16**:3(66), pp. 3–22. (*In Russ.*). `https://psta.psiras.ru/read/psta2025_3_3-22.pdf`

References

[1] A. Kukliansky, M. Orescanin, C. Bollmann, T. Huffmire. “Network anomaly detection using quantum neual networks on noisy quantum computers”, *IEEE Transactions on Quantum Engineering*, **5** (2024), pp. 1–11.

[2] M. A. Kazhemskij, O. I. Sheluxin. “Multiclass classification of attacks to information resources with machine learning techniques”, *Trudy uchebnyx zavedenij svyazi*, **5**:1 (2019), pp. 107–115 (in Russian).

[3] A. Zeguendry, Z. Jarir, M. Quafafou. “Quantum machine learning: A review and case studies”, *Entropy*, **25**:2 (2023), id. 287, 41 pp.

[4] S. S. Volkov, I. I. Kurochkin. “Extraction of traffic features in software-defined networks using an SDN-controller”, Conference: 9th International Conference "Distributed Computing and Grid Technologies in Science and Education" (Dubna, Russia, July 5–9, 2021), CEUR Workshop Proceedings, vol. **3041**, 2021, pp. 553–557.

[5] N. K. Fyodorov. “Software-defined networks. Problems in the transition to software-defined networks”, *StudNet*, **5**:4 (2022), pp. 3137–3148 (in Russian).

[6] S. C. Forbacha, M. K. Kinteh, E. M. Hamza. “Enhanced attacks detection and mitigation in software defined networks”, *American Journal of Computing and Engineering*, **7**:3 (2024), pp. 40–80.

[7] M. S. Elsayed, N. A. Le-Khac, A. D. Jurcut. “InSDN: A novel SDN intrusion dataset”, *IEEE Access*, **8** (2020), pp. 165263–165284.

[8] Z. Wu, J. Wang, L. Hu, Zh. Zhang, H. Wu. “A network intrusion detection method based on semantic Re-encoding and deep learning”, *Journal of Network and Computer Applications*, **164** (2020), id. 102688.

[9] C. Yoon, S. Lee, H. Kang, T. Park, S. Shin, V. Yegneswaran, Ph. Porras, G. Gu. “Flow wars: Systemizing the attack surface and defenses in software-defined networks”, *IEEE/ACM Transactions on Networking*, **25**:6 (2017), pp. 3514–3530.

[10] V. Vedral, M. B. Plenio. “Basics of quantum computation”, *Progress in Quantum Electronics*, **22**:1 (1998), pp. 1–39.

[11] Z. Yang, M. Zolanvari, R. Jain. “A survey of important issues in quantum computing and communications”, *IEEE Communications Surveys & Tutorials*, **25**:2 (2023), pp. 1059–1094.

[12] S. N. Torgaev, I. D. Shul’ga, E. A. Yurchenko, M. L. Gromov. *Fundamentals of quantum computing*, uchebnoe posobie, STT, Tomsk, 2020, ISBN 978-5-93629-656-7, 88 pp. (in Russian).

[13] G. Bacciagaluppi. “The role of decoherence in quantum mechanics”, *The Stanford Encyclopedia of Philosophy*, substantive revision Thu Jan 23, 2025, eds. Edward N. Zalta, Uri Nodelman, Metaphysics Research Lab, Stanford University, 2025.

[14] J. Luo, P. Zhao, Zh. Miao, J. Zhao. "A comprehensive study of bug fixes in quantum programs", *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering*, SANER (Honolulu, HI, USA, 15–18 March 2022), IEEE, 2022, ISBN 978-1-6654-3786-8, pp. 1239–1246.

[15] P. W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring", *Proceedings 35th Annual Symposium on Foundations of Computer Science* (Santa Fe, NM, USA, 20–22 November 1994), IEEE, 1994, ISBN 0-8186-6580-7, pp. 124–134.

[16] P. W. Shor. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", *SIAM Journal on Computing*, **26**:5, pp. 1484–1509. arXiv quant-ph/9508027

[17] L. K. Grover. "A fast quantum mechanical algorithm for database search", *STOC'96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (Philadelphia, Pennsylvania, USA, 22–24 May 1996), 1996, ISBN 978-0-89791-785-8, pp. 212–219.

[18] K. A. Tychola, T. Kalampokas, G. A. Papakostas. "Quantum machine learning — an overview", *Electronics*, **12**:11 (2023), id. 2379, 21 pp.

[19] M. Kalinin, V. Krundyshev. "Security intrusion detection using quantum machine learning techniques", *Journal of Computer Virology and Hacking Techniques*, **19**:1 (2023), pp. 125–136.

[20] E. Zardini, E. Blanzieri, D. Pastorello. "A quantum k-nearest neighbors algorithm based on the Euclidean distance estimation", *Quantum Machine Intelligence*, **6**:1 (2024), id. 23, 22 pp.

[21] A. Basheer, A. Afham, S. K. Goyal. *Quantum k-nearest neighbors algorithm*, 2021, 21 pp. arXiv 2003.09187

[22] D. Anguita, S. Ridella, F. Rivieccio, R. Zunino. "Quantum optimization for training support vector machines", *Neural Networks*, **16**:5–6 (2003), pp. 763–770.

[23] A. W. Harrow, A. Hassidim, S. Lloyd. "Quantum algorithm for linear systems of equations", *Physical review letters*, **103**:15 (2009), id. 150502.

[24] J. Yang, A. J. Awan, G. Vall-Llosera. *Support vector machines on noisy intermediate scale quantum computers*, 2019, 12 pp. arXiv 1909.11988

[25] G. N. Meedinti, K. S. Srirekha, R. Delhibabu. *A quantum convolutional neural network approach for object detection and classification*, 2023, 16 pp. arXiv 2307.08204

[26] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer. "SMOTE: Synthetic minority over-sampling technique", *Journal of Artificial Intelligence Research*, **16**:1 (2002), pp. 321–357.

[27] S. S. Volkov, I. I. Kurochkin. "Network attacks classification using Long Short-term memory based neural networks in Software-Defined Networks", *Procedia Computer Science*, **178** (2020), pp. 394–403.