**УДК 621.391.8:004.032.26**

# ОБУЧЕНИЕ МОДЕЛИ 1D-CNN ДЛЯ КЛАССИФИКАЦИИ СОСТОЯНИЙ РАДИОКАНАЛА И ВЫЯВЛЕНИЯ RF-JAMMING

А.О. ПИЛИПЕНКО, С.А. ШВЕД

*Белорусский государственный университет информатики и радиоэлектроники*
*(г. Минск, Беларусь)*

*E-mail: ctepan3310@gmail.com, pilipenkoandrej602@gmail.com*

**Аннотация.** В работе проводится исследование применения одномерной сверточной нейронной сети 1D-CNN, которая может автоматически обнаруживать помехи и попытки какого-либо стороннего воздействия на сигнал (RF-jamming), а также модель способна классифицировать состояния радиоканала. Входными данными для обучения и последующей проверки нейронной сети служит набор active_scan. Данный набор содержит три класса наблюдений: фоновые сигналы (чистый эфир), гауссовский шум (сплошной шум) и однотональные помехи (точечные помехи). Особенность данной нейронной сети в том, что она обучалась с нуля на подготовленных для нее входных данных, в отличие от подхода с дообучением какой-либо сохраненной ранее модели. Для повышения натренированности модели и снижения риска переобучения в работе к входным данным после нормализации был добавлен искусственный шум. Данное действие позволило нейросети даже в самых непростых условиях выделять именно важные признаки, а также предостерегло ее от переобучения (overfitting) и дальнейших грубых ошибок. По итогу точность нашей модели составила 88,89 % на тестовой выборке. Такой процент подтверждает эффективность применения 1D-CNN в качестве отличного учебного прототипа для защиты радиосвязи от различных атак и для обнаружения аномальных состояний радиоканала.

**Abstract**. This paper explores the use of a one-dimensional convolutional neural network (1D-CNN), which can automatically detect interference and attempts at external signal interference (RF jamming), and classify radio channel states. The active_scan dataset serves as the input data for training and subsequent validation of the neural network. This dataset contains three classes of observations: background signals, Gaussian noise, and single-tone interference. A unique feature of this neural network is that it was trained from scratch using pre-prepared input data, as opposed to retraining a previously saved model. To improve the model's fitness and reduce the risk of overfitting, artificial noise was added to the input data after normalization. This step allowed the neural network to identify important features even in the most challenging conditions and also prevented it from simply learning the sample and making further gross errors. As a result, the accuracy of our model on the test set was 88.89%. This percentage confirms the effectiveness of using 1D-CNN as an excellent training prototype for protecting radio communications from various attacks and for detecting abnormal radio channel conditions.

## Введение

В наше время беспроводные системы стали незаменимы и преследуют нас везде. Они используются и в смартфонах, и в компьютерах, и в беспилотниках, и даже в кухонном чайнике. По мере развития технологий программно-определяемого радио (SDR) управление стало гораздо проще и удобнее, но это создало некоторые риски и проблемы в виде, например, создание помех злоумышленниками. Радиофон постоянно усложняется, и это делает ручной мониторинг слишком медленным и неэффективным, в связи с этим все нуждается в быстрых автоматических решениях. Главной задачей здесь является выявление преднамеренных действий злоумышленника, которые способны повредить или полностью оборвать связь. Обычные методы защиты работают по жестким и заранее прописанным правилам, из-за чего они часто ошибаются при малейшем изменении обстановки и не могут адаптироваться в случае каких-либо новых видов атак. Здесь как раз и нужны нейронные сети. Они способны самостоятельно находить скрытые закономерности в самой структуре радиосигнала и быстро подстраиваться под новые условия, основываясь на разных критериях прошлых примеров. В данной работе используется одномерная сверточная сеть (1D-CNN) из-за того, что она умеет анализировать сигнал выделяя его важные признаки в конкретный момент времени. Этот проект посвящен созданию и проверке учебной модели 1D-CNN, которая должна научиться распознавать чистый эфир и атаки, такие как гауссовский шум и однотональные помехи. В исследовании выделим одну особенность, заключающуюся в том, что нейросеть создавалась «с нуля» и обучалась на входных данных active_scan, без использования других заранее готовых моделей или процедур дообучения (Fine-Tuning).

**Подготовка данных**

Перед работой с данными необходимо их нормализовать. В данном эксперименте использовался каталог active_scan, в котором данные распределены по трем классам: benign/background/location1 (обычная «тишина» радиоэфира, в которой могут быть какие-то случайные шумы, но нет атаки), malicious/gaussian_noise/10dbm (атака «белым шумом») и malicious/singletone/10dbm (злоумышленник атакует не все частоты, а одну конкретную сильным сигналом). Из каждого CSV-файла были извлечены 3 признака: noise (общий уровень беспорядка в эфире), rssi (сила принимаемого сигнала) и max_magnitude (самый громкий пик в сигнале). Отдельно взятый объект выборки представляет собой временную последовательность длиной 128 отсчетов с тремя признаковыми каналами. Ниже приведена формула нормализации:

$$x_{norm} = ( x - mean(x) ) \ / \ ( std(x) ) , \quad (1)$$

У параметра rssi при извлечении может быть, например, значение -90, а у noise – 0.001. Модель подумает, что раз число больше, то оно важнее. Однако на практике это не так, в связи с этим и необходима нормализация, которая приводит все к единому масштабу (от -3 до 3), чтобы нейросеть обучалась честно.

Для каждого класса использовалось до 150 файлов, поэтому общий объем выборки составлял около 450 последовательностей. Важным моментом является специальный прием регуляризации и стресс-тестирования, который усложняет задачу для модели и позволяет избежать завышенных результатов тестирования на обучающей выборке. Прием заключается в том, что к нормализованным данным добавлялся искусственный гауссовский шум с параметром noise_level = 0.8. Если нейросеть научится распознавать RF-jamming на зашумленных данных, в реальной жизни (где условия хуже, чем в лаборатории) она будет работать стабильнее. Это защищает от переобучения (когда нейросеть заучивает примеры обучающей выборки и в дальнейшем не может правильно классифицировать данные тестовой выборки). Нормализация выполнялась по формуле:

$$x_{norm} = ( x - mean(x) ) \ / \ ( std(x) + 1e\text{-}8 ) , \quad (2)$$

Как уже было сказано, выборка была разделена на обучающую и тестовую части. Обучающая часть составила 80% всей выборки, а тестовая – 20% с фиксированным random_state = 42. Параметр random_state служит для одинакового перемешивания данных при каждом запуске кода. Данное действие стабилизирует результат эксперимента и делает его более проверяемым. Чтобы реализовать многоуровневое распознавание, необходимо было перевести метки в векторный формат one-hot encoding.

Для более наглядного представления того, как проходит обработка информации, а также для понимания общей структуры нейросети приведем общую блок-схему системы на следующем рисунке:

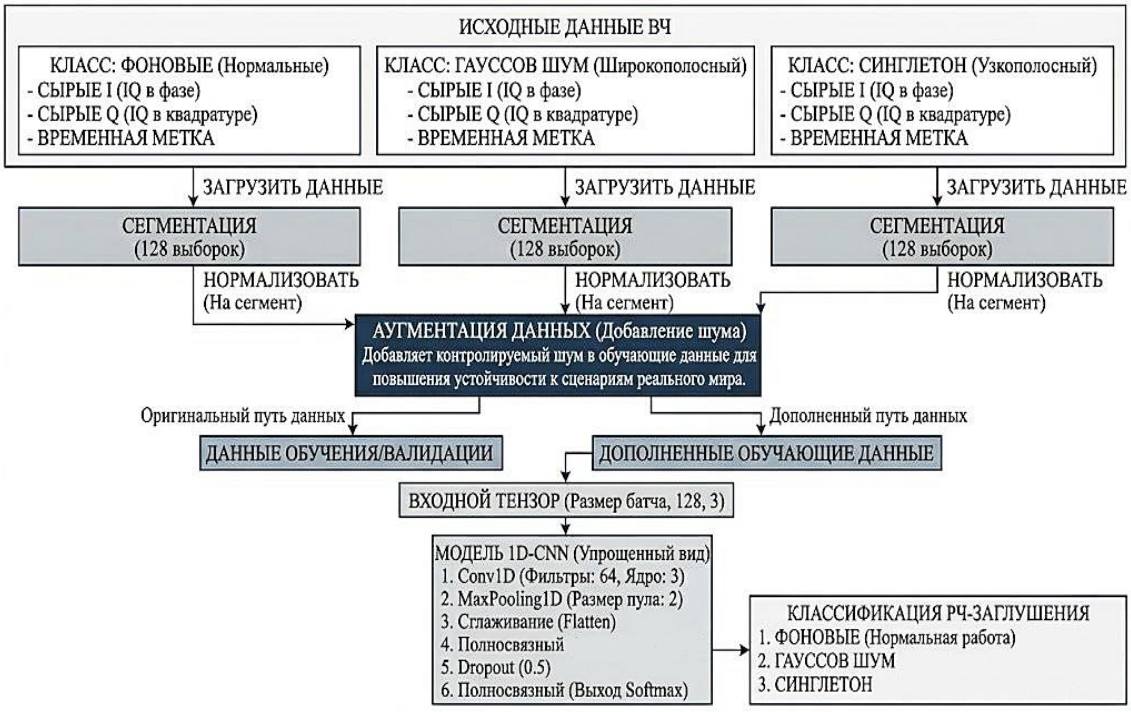



**Рис. 1.** Общая структурная схема подготовки данных и архитектура модели 1D-CNN

На схеме видно, что сначала на вход подаются исходные данные. Важно отметить, что на схеме отражены «общие» данные, а в нашем случае подаются: noise, rssi, max_magnitude. Также в данных есть временные метки, которые являются точным временем (дата, часы, секунды, миллисекунды) замера, когда именно был сделан каждый конкретный замер. Затем все данные сегментируются путем нарезания на блоки по 128 выборок, далее каждый сегмент нормализируется, чтобы выровнять масштаб для всех значений. Следующим очень важным блоком идет аугментация, здесь к обучающим нормализованным данным добавляется шум, чтобы повысить устойчивость модели. Таким образом наши данные начинают формировать так называемый входной тензор, который поступает на вход модели 1D-CNN. Сам тензор представляет собой многоуровневую таблицу с числами (массив) – это главный формат данных, с которыми взаимодействуют нейросети. Он имеет размерность (Batch, 128, 3), где Batch – количество примеров, 128 – длина записи, 3 – количество признаков. Используя сверточный (первый) слой Conv1D, нейросеть извлекает все признаки и уменьшает размерность с помощью слоя MaxPooling (второй). Затем, под действием слоя Flatten (третий), признаки из таблицы выстраиваются в одну длинную линию, для удобства вынесения последующего вердикта. Далее все передается в четвертый слой (полносвязный), имеющий функцию активации ReLU. Чтобы исключить переобучение модели, используется пятый слой Dropout с коэффициентом 0.5, который на этапе обучения случайным образом отключает часть нейронов во время тестирования. Данная структурная составляющая нейросети работает как защитный механизм, не давая 50% нейронов включаться в работу, поэтому сеть начинает работать стабильнее и не просто запоминает конкретные примеры, а обнаруживает и подчеркивает для себя общие черты и закономерности. Финальный шестой слой с функцией Softmax считает и показывает итоговую вероятность принадлежности сигнала к какому-то из трех целевых классов. Вся эта последовательность работы с данными дает гарантии того, что система начнет эффективно распознавать радиоглушения и сам сигнал даже при изменении электромагнитной обстановки и других усложняющих факторах.

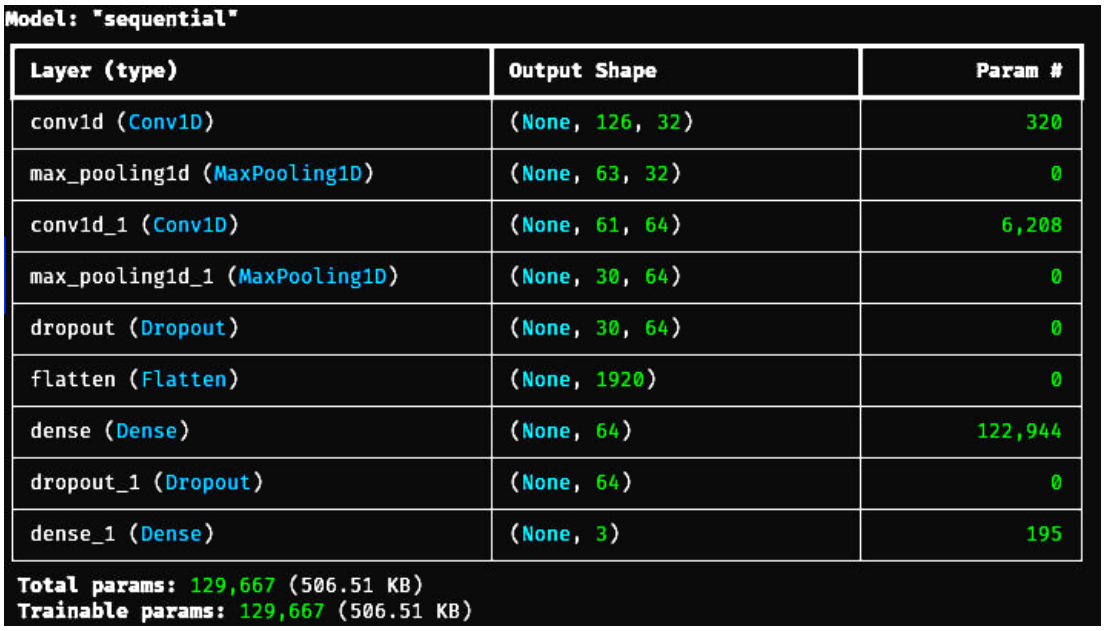
**Архитектура модели**

Для работы с проектом использовалась определенная последовательная архитектура 1D-CNN. Такая модель отлично подойдет для этого случая, так как мы будем анализировать временные ряды. «Скользящие окна» двигаются по данным и находят важные закономерности в соседних местах. На вход модели поступают данные размером 128x3, а это значит, что нейросеть каждый раз анализирует данные последовательности из 128 замеров времени. В каждом таком моменты содержатся три параметра: noise, rssi, max_magnitude. Благодаря такому методу модель воспринимает полную динамику эфира.

**Таблица 1.** Архитектура использованной модели 1D-CNN

| Слой | Тип | Выходная форма | Параметры |
|---|---|---|---|
| conv1d | Conv1D, 32 фильтра, kernel_size=3, ReLU | (None, 126, 32) | 320 |
| max_pooling1d | MaxPooling1D, pool_size=2 | (None, 63, 32) | 0 |
| conv1d_1 | Conv1D, 64 фильтра, kernel_size=3, ReLU | (None, 61, 64) | 6 208 |
| max_pooling1d_1 | MaxPooling1D, pool_size=2 | (None, 30, 64) | 0 |
| dropout | Dropout, 0.5 | (None, 30, 64) | 0 |
| flatten | Flatten | (None, 1920) | 0 |
| dense | Dense, 64, ReLU | (None, 64) | 122 944 |
| dropout_1 | Dropout, 0.5 | (None, 64) | 0 |
| dense_1 | Dense, 3, Softmax | (None, 3) | 195 |

Количество параметров обучения, которые модель выделила для себя, составило 129 667. Число этих параметров показывает количество внутренних связей и коэффициентов, которые модель для себя определила из обучающей выборки, для последующего распознавания сигналов. Выходной слой Softmax формирует вероятности принадлежности объекта к одному из трех классов. Функция Softmax, которая находится на последнем слое нейросети, позволяет из сложных математических расчетов в понятные значения, выраженные в процентах. Поэтому модель после тестов выдает еще и степень своей уверенности в ответе.



Model: "sequential"

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv1d (Conv1D) | (None, 126, 32) | 320 |
| max_pooling1d (MaxPooling1D) | (None, 63, 32) | 0 |
| conv1d_1 (Conv1D) | (None, 61, 64) | 6,208 |
| max_pooling1d_1 (MaxPooling1D) | (None, 30, 64) | 0 |
| dropout (Dropout) | (None, 30, 64) | 0 |
| flatten (Flatten) | (None, 1920) | 0 |
| dense (Dense) | (None, 64) | 122,944 |
| dropout_1 (Dropout) | (None, 64) | 0 |
| dense_1 (Dense) | (None, 3) | 195 |

Total params: 129,667 (506.51 KB)
Trainable params: 129,667 (506.51 KB)

**Рис. 2.** Вывод model summary () для модели 1D-CNN

## Методика обучения

Для реализации обучения модели проекта был выбран оптимизатор Adam, который достаточно плавно корректирует выбранные нейросетью параметры, а также отвечает за настройку функции потерь categorical_crossentropy, которая в свою очередь служит для точного подсчета ошибок модели. Главным критерием является метрика точности accuracy. Она показывает процент состояний эфира, которые были предсказаны (классифицированы) верно. Во время обучения входные данные обрабатываются частями по 32 объекта на протяжении 20 эпох (в каждой эпохе по 32 объекта). Чтобы модель не заучивала выборку и результаты были достоверными, проверка модели проводится на отдельно выделенной тестовой выборке, которую нейросеть до этого не видела.

Как было уже отмечено ранее, отличительной чертой модели является то, что она полностью обучается самостоятельно. Для этого специально была проигнорирована технология Fine-Tuning, которая позволяет взять уже готовую модель и дообучить ее в процессе использования под определенные задачи. В коде программы отсутствуют загрузки весов из сторонних файлов, а также отсутствует заморозка отдельных слоев сети.

Чтобы повысить достоверность полученных данных после тестирования модели, был специально добавлен искусственный шум. Данный прием после нормализации необходимо рассматривать как регуляризующий. Таким образом имитируются более сложные условия для модели, что заставляет ее работать в усиленном режиме и не искать каких-то лазеек, задействовав все свои ресурсы. Нейросеть ищет все более незаметные признаки сигналов и помех, а не просто запоминает примеры. Так добавление шума делает систему более надежной и подготовленной.

## Результаты эксперимента

После завершения всех циклов анализа данных тестирующей выборки модель выдала значение точности равное 88.89%. Ценность результата в первую очередь в том, что он получен на предварительно усложненной выборке, в которую были добавлены шумы. Также стоит отметить, что обучающая выборка была небольшая, и модель не в полной мере успела наработать навык классификации. Хорошим знаком является и то, что на графике ниже видно: линия точности, установленной на тестирующей выборке, идет рядом с линией точности, установленной на обучающей выборке. Это происходит как раз из-за того, что нейросеть смогла правильно обучиться определять целевые факторы, а не просто заучила примеры из обучающей выборки. Теперь она может применять свои знания к другим случаям, к которым изначально модель не была готова. Так как линии достаточно синхронны, можно сделать вывод, что в работе удалось избежать переобучения нейронной сети, а это является одной из главных задач машинного обучения.

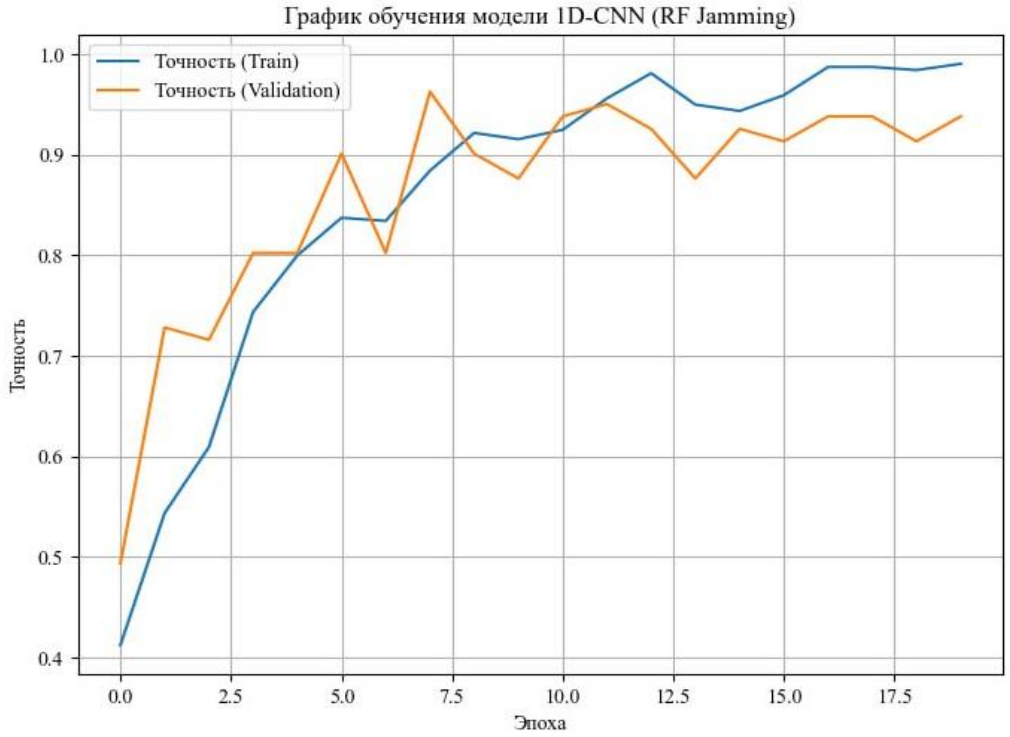



**Рис. 3.** График обучения модели 1D-CNN: точность на обучающей и тестирующей выборках

На графике в самом начале мы также видим резкий рост точности обучающей выборки, что говорит о том, что в отрезке этих эпох модель обучается достаточно быстро, находя очевидные отличия между разными помехами. Но с некоторого момента график начинает расти не так резко, потому что нейросеть начинает все более глубоко вникать в данные выборки и искать все более тонкие и сложные детали сигналов. Нужно сказать и о том, что точность на графике сразу не подскочила до 100%. Данная ситуация нормальна, так как именно это и делает наш эксперимент более реалистичным. В настоящее время достичь идеальных результатов – это неправдоподобно, так как в любой модели и любой выборке есть доля неопределенности. Большее доверие вызывают стандартные результаты точности, ведь на практике часто бывает так, что около-максимальные результаты просто являются ошибкой в программном обеспечении, или недостаточно реалистичных условий существований сигнала.

Несмотря на то, что данная модель достаточно хороша, стоит помнить: проделанная работа является учебно-исследовательской. Если необходимо будет превратить этот проект во что-то более масштабное (какой-либо сильный промышленный инструмент), его придется испытывать на совершенно других данных (более сложные и вариативных).

## Соответствие статьи программной реализации

Содержание статьи приведено в полном соответствие с фактической программной реализацией для обеспечения максимально высокой точности описания. В проекте используются реальные входные данные из файлов набора active_scan, в этих файлах есть три ключевых параметра (noise, rssi, max_magnitude), на основе которых и анализируется сигнал. Архитектура 1D-CNN, описанная в работе, содержит в себе два сверточных блока и слои Dropout, завершается она выходным слоем, который отвечает за классификацию на три класса. Именно такой набор структурных элементов модели и показывает логику написанного программного кода, а также структуру используемых библиотек. В данной работе не предполагалось использование дополнительных источников в виде весов заранее сохраненных моделей, поэтому в проекте нигде не упоминалось о применении Fine-Tuning. Данный факт достаточно важен для понимания того, как именно обучалась модель, так как она на протяжении всех эпох развивалась и обучалась полностью самостоятельно. В связи с этим в целях сохранения академической строгости много моментов пришлось менять, опираясь на большинство рядовых случаев обучения нейросети. Например, было написано «график обучения», а не использовалось словосочетание «график дообучения».

## Заключение

В ходе работы была весьма успешно, ссылаясь на результаты тестирования, построена модель 1D-CNN, которая предназначена для классификации состояний радиоканала и выявления признаков RF-jamming, используя для этого постоянно меняющиеся параметры noise, rssi и max_magnitude. Одновременное использование всех трех признаков дало возможность модели гораздо глубже анализировать сигнал и его помехи, а также находить отличия там, где более простые методы анализа будут бессильны. Как уже было отмечено, модель обучалась полностью самостоятельно на наборе входных данных active_scan. Это позволило не дать повлиять на нейросеть каким-либо сторонним настройкам, созданных для немного других целей, а также позволило удостовериться в правильности работы программного обеспечения в чистом виде. Для того, чтобы модель была более устойчива к реальным помехам, в работе был добавлен искусственный шум, благодаря этому действию оценка качества работы нейросети была проведена достоверно. Только методом добавления шума можно подготовить модель к более «суровым» условиям существования сигнала в реальном эфире. По итогам тестовой выборки была получена точность 88.89%. Такие высокие показатели подтверждают, что одномерные сверточные сети достаточно эффективны для создания фундамента различных учебных прототипов систем обнаружения аномалий в радиоэфире. В последующем для развития проекта необходимо будет расширять набор входных данных, а также более детально проводить подбор шума и сравнивать получившиеся результаты с классическими случаями машинного обучения. Кроме того, отличной перспективой обладает внедрение готовой модели в реальные программно-определяемые радиосистемы (SDR) или составляющие сетевой защиты (IDS/IPS). Такая тенденция развития предоставит возможность для перехода от простых учебных тестов к созданию полноценного инструмента, который будет служить для обеспечения безопасности беспроводных коммуникаций.

## Список использованных источников

1. O'Shea T. J., West N. Radio Machine Learning Dataset Generation with GNU Radio // Proceedings of the 6th GNU Radio Conference. - 2016.
2. Goodfellow I., Bengio Y., Courville A. Deep Learning. - MIT Press, 2016.
3. Chollet F. Deep Learning with Python. - Manning Publications, 2021.
4. Сергиенко А. Б. Цифровая обработка сигналов. - СПб.: Питер, 2002.
5. Документация TensorFlow/Keras: Sequential model, Conv1D, Dropout, categorical crossentropy.