

О ПРИМЕНЕНИИ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ОЦЕНКИ ВЕСОВ СВЯЗЕЙ МЕЖДУ КОНЦЕПТАМИ НЕЧЕТКОЙ КОГНИТИВНОЙ КАРТЫ В АДАПТИВНОЙ ОЦЕНКЕ РИСКОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ¹

В статье обсуждаются результаты анализа применения нейронных сетей (НС) в оценке весов связей (ВС) между концептами нечеткой когнитивной карты (НKK) в адаптивной оценке рисков информационной безопасности (РИБ).

Были проанализированы научные публикации в области НKK и НС, определены особенности проектирования архитектуры и обучения НС, был сформирован алгоритм обучения НС проводить оценку весов связи НKK.

Продемонстрирован процесс обучения НС осуществлять оценку ВС, произведенный на обучающем наборе данных. Предложен программный код для решения данной задачи, определена эффективность применения технологий НС и КМ для адаптивной оценки рисков ИБ.

Ключевые слова: *информационная безопасность, адаптивная оценка рисков информационной безопасности, нечеткие когнитивные карты, веса связи, нейронные сети.*

¹ В рамках исполнения гранта ИБ МТУСИ № 40469-23/2022-к 15.06.2022г. протокол №2

ON THE APPLICATION OF NEURAL NETWORKS FOR ESTIMATING THE WEIGHTS OF CONNECTIONS BETWEEN THE CONCEPTS OF A FUZZY COGNITIVE MAP

The article discusses the results of the analysis of the use of neural networks (NN) in estimating the weights of connections (BC) between the concepts of a fuzzy cognitive map (FCM) in adaptive information security risk assessment (ISS).

Scientific publications in the field of NCC and NN were analyzed, the features of designing the architecture and training of NN were determined, and an algorithm for training NN to evaluate the NCC communication weights was formed.

The process of training a NN to evaluate the aircraft produced on a training data set is demonstrated. A program code for solving this problem is proposed, and the effectiveness of using NN and CM technologies for adaptive information security risk assessment is determined.

Keywords: *information security, adaptive information security risk assessment, fuzzy cognitive maps, connection weights, neural networks.*

Введение

В современном информационном обществе вопросы информационной безопасности (ИБ) всегда являются важными и актуальными. Одним из важных компонентов эффективной стратегии ИБ в любой компании или организации является оценка рисков. Традиционные методы оценки рисков ИБ часто основываются на статических моделях и предположениях, которые могут быть недостаточными для учета динамических изменений в информационной системе и угрозах. Адаптивная оценка РИБ стремится преодолеть эту проблему, обеспечивая гибкость и способность модели адаптироваться к новым данным и условиям.

Подход адаптивной оценки РИБ основан на постоянном мониторинге и анализе угроз и уязвимостей, а также на принятии мер по обеспечению безопасности в реальном времени. В этом контексте уместно вспомнить про НКК, которые могут быть применены в адаптивной оценке РИБ.

НКК – это модель, представленная в графическом виде и отражающая связи между различными концептами, факторами и событиями, связанными с ИБ.

НКК позволяют моделировать сложные взаимосвязи и динамические изменения в ИБ-среде, а также имеют способность прогнозировать и адаптироваться к новым угрозам и уязвимостям.

Одним из важных моментов, связанных с работой НКК, является оценка ВС между концептами в НКК. Она позволяет определить степень влияния одного концепта на другой и учесть сложные зависимости и взаимодействия между ними. В этом контексте, НС предлагают эффективный инструмент для оценки и определения ВС на основе обучающих данных.

Целью настоящей работы является рассмотрение возможности применения НС в контексте оценки ВС между концептами в НКК для адаптивной оценки РИБ. Представлена методика оценки ВС с использованием НС, а также приведены результаты экспериментов для оценки эффективности предложенного подхода.

Нечёткие когнитивные карты и нейронные сети в адаптивной оценке рисков ИБ

При использовании НКК [1-3] в адаптив-

ной оценке РИБ, эксперты могут оценивать взаимосвязь между различными факторами: угрозы, уязвимости, защитные меры и последствия. НКК позволяют моделировать и представлять сложные зависимости и взаимодействия между этими факторами [4].

Одним из преимуществ использования НКК в адаптивной оценке РИБ является их способность учитывать неопределенность и нечеткость данных, различные уровни вероятности и степени влияния факторов на РИБ. НКК могут быть адаптированы и обновлены с учетом изменений в ИБ-среде. Это позволяет проводить динамическую оценку РИБ и принимать актуальные решения на основе изменений в концепциях и связях между ними.

Рассмотрим пример НКК для образовательной сферы со следующими концептами:

C1 – Управление доступом;

C2 – Конфиденциальность студентов и персональных данных;

C3 – Физическая безопасность;

C4 – Сетевая безопасность;

C5 – Мошенничество в области исследований и публикаций;

C6 – Управление уязвимостями;

C7 – Социальная инженерия;

C8 – Отказ в обслуживании (DoS);

C9 – Безопасность мобильных устройств;

C10 – Обучение и осведомленность.

Дальнейшим шагом по работе с данной НКК является оценка ВС между ее концептами. Традиционно эту задачу решают экспертным путем, что допускает вероятность рассогласованности экспертов и увеличения сложности оценки ВС с ростом факторов. Для решения этого предложено применить НС, а именно обучить НС самостоятельно оценивать ВС.

Результаты исследований вопросов НС отражены в работах многих исследователей, так в [5] представлена общая схема оценки защищенности объектов информатизации с использованием нейронных сетей. Авторы подчеркивают, что результаты оценки зависят от множества факторов и сложно формально определить итоговый показатель из-за неизвестной взаимосвязи между показателями защищенности. Для решения этой проблемы авторы используют искусственные нейронные сети и анализируют групповые и частные показатели в качестве входных данных, основываясь на отечественных и зарубежных методиках и стандартах. Они также

отмечают важность опыта экспертов для качественной оценки и считают задачу исключения вероятности ошибочного суждения экспертов ключевой.

Одним из важных вопросов в использовании НС является их обучение, в [6] приведен обзор методов обучения глубоких НС, содержащих более одного скрытого слоя. Автор работы указывает, что искусственные НС могут аппроксимировать любую непрерывную функцию с требуемой точностью. Однако, он отмечает, что в настоящее время не существует конструктивного подхода, который гарантированно позволял бы создавать НС с заранее заданными свойствами, что является ограничением для их применения. Автор делает вывод, что на данный момент два типа архитектур НС являются доминирующими: сверточные и рекуррентные сети. Он также отмечает перспективность совместного использования обоих типов архитектур с обучением с подкреплением. Одним из преимуществ рекуррентных НС является возможность обучения без учителя и без предварительного размеченного набора данных. В [7] автор расширяет существующие методы обучения и рассматривает различные способы инициализации весов. Он обсуждает применение принципов обучения с учителем, self-taught learning и transfer learning. В ходе экспериментов автор обнаружил проблему затухания градиента в процессе обучения, при которой первые слои многослойного перцептрона учатся менее эффективно. В связи с этим, автор предлагает использовать итеративный алгоритм инициализации весов, который основан на уточнении весов скрытых слоев НС через решение исходной задачи классификации или регрессии. Он также предлагает использовать этот алгоритм для построения ансамбля НС. Этот аспект работы подчеркивает необходимость исследования особенностей организации ансамблей НС и в целом НС.

Обучение НС проводить оценку весов связи НКК

Ключевыми моментами в создании НС являются выбор ее типа, архитектуры и метода обучения [8]. Наиболее распространенные типы НС, их достоинства, недостатки и применимость в выбранной области приведены в таблице 1 [6, 8, 9]. Основные элементы архитектуры НС приведены в таблице 2 [6]. Распространенные алгоритмы обучения НС – в таблице 3 [6, 7, 9].

Типы нейронных сетей

№	Тип НС	Характеристика
1.	Многослойный персептрон (MLP)	Состоит из нескольких слоев нейронов: входной, скрытые слои и выходной. Каждый нейрон в слоях, кроме входного, получает сигналы от предыдущего слоя и передает свой выход следующему слою. Входные данные распространяются через нейроны слоев вперед, от входного слоя к выходному, без циклических связей. Используется в различных областях, включая распознавание образов, классификацию, регрессию и другие задачи обработки данных.
2.	Рекуррентные НС (RNN)	Используют направленную последовательность связи между узлами. Результат вычислений на каждом этапе используется в качестве исходных данных для следующего. Подходят для анализа последовательных данных: временные ряды, что может быть полезным для анализа данных ИБ. Могут использоваться для анализа последовательностей событий, связанных с ИБ-инцидентом: временная метка, тип инцидента, источник атаки. Могут улавливать долгосрочные зависимости и паттерны, могут обрабатывать входные данные разной длины, учитывать контекст и зависимости между данными в последовательности.
3.	Сверточные НС (CNN)	Используют свертки для извлечения признаков из входных данных. Подходят для анализа изображений и других типов данных, которые можно представить в виде матрицы. Могут использоваться для анализа логов и других текстовых данных. Способны извлекать значимые признаки из входных данных, позволяют моделировать иерархическую и локальную структуру данных, могут быть модульными и легко масштабируемыми.

Таблица 2

Архитектура нейронной сети

№	Элемент архитектуры	Характеристика
1.	Входной слой	Для ввода данных: параметры системы безопасности или другие характеристики, которые могут влиять на риски ИБ.
2.	Скрытые слои	Используются для обработки входных данных. Количество слоев и количество нейронов зависит от сложности задачи и размера обучающего набора данных.
3.	Выходной слой	Используются для выдачи результата оценки рисков. Количество нейронов зависит от количества классов рисков, которые необходимо оценивать.
4.	Функция активации	Используются для нелинейной обработки данных в скрытых слоях и выходном слое. Подходящие функции: ReLU, sigmoid или tanh.
5.	Функция потерь	Используются для определения ошибки между предсказанными и фактическими значениями. Подходящие функции: mean squared error, categorical cross-entropy.
6.	Оптимизатор	Для обновления весов НС на каждой итерации обучения. Подходящие оптимизаторы: Adam, SGD или RMSProp.

Таблица 3

Алгоритмы обучения НС

№	Алгоритм	Характеристика
1.	Обратное распространение ошибки	Основывается на минимизации функции потерь, которая измеряет разницу между выходом сети и желаемым выходом. После расчета ошибки, алгоритм обновляет веса в соответствии с градиентом функции потерь.
2.	Метод стохастического градиентного спуска	Обновляет веса в соответствии с градиентом функции потерь, но использует только случайное подмножество обучающих данных на каждой итерации, что позволяет быстрее обучать модель на больших данных.
3.	Метод адаптивного градиентного спуска	Адаптивно изменяет скорость обучения для каждого параметра в соответствии с его градиентом, что позволяет более эффективно обновлять веса в сети.

№	Алгоритм	Характеристика
4.	Метод Адам	Использует комбинацию метода стохастического градиентного спуска с моментом и метода адаптивного градиентного спуска. Это позволяет более эффективно обновлять веса в сети, учитывая как градиент, так и его предыдущие значения.
5.	Метод максимального правдоподобия	Минимизирует разницу между распределением вероятностей, предсказанным моделью, и распределением вероятностей, заданным обучающими данными. Часто используется в задачах классификации.
6.	Метод генетического алгоритма	Эволюционный алгоритм, использует популяцию индивидуальных решений и эмуляцию естественного отбора, чтобы найти оптимальные веса для НС.

На основе вышеперечисленного анализа типов НС, их архитектуры и методов обучения сформируем НС, способную решить поставленную задачу – обучить НС оценивать ВС между концептами НКК.

Для обучения НС проводить оценку ВС в НКК можно использовать тип НС – MLP. Входной слой принимает входные данные, скрытые слои выполняют вычисления, а выходной слой предоставляет оценку ВС НКК. Скрытых слоев может быть несколько, это позволяет моделировать сложные взаимосвязи между концептами в НКК и оценивать ВС более точно. Также MLP можно легко расширить, добавив больше скрытых слоев или изменяя количество нейронов в каждом слое. Размер скрытого слоя является гиперпараметром, который можно настраивать и экспериментировать с разными значениями, чтобы найти наилучшую производительность для решения задач.

В качестве входных данных были использованы данные, полученные при создании НКК: определенные экспертами 10 концептов, результаты оценки ВС, выставленные 10 экспертами. ВС, определенные экспертами, также их можно обозначить как желаемые ВС, представляют собой данные для обучающей выборки. Тестовая выборка задана произвольно.

Подходящими для решения поставленной задачи методами обучения являются: алгоритм обратного распространения ошибки и градиентного спуска. Для его реализации необходимо определить функцию потерь, которая будет минимизироваться в процессе обучения. В данном случае, функция потерь зависит от разницы между оценками ВС НКК и желаемыми значениями. Метод обратного распространения ошибки – для вычисления градиента функции потерь по ВС, и градиентный спуск – для обновления ВС с целью минимизации функции потерь.

Обучение НС проводить оценку ВС НКК будет выполняться итеративно. На каждой итерации, входные данные НКК будут подаваться на вход НС, оценки ВС будут вычисляться, а затем будут сравниваться с желаемыми значениями. После этого, будет происходить обновление ВС с использованием метода обратного распространения ошибки и градиентного спуска. Процесс будет повторяться до достижения требуемой точности или сходимости. Алгоритм обучения НС проводить оценку ВС НКК представлен на рис. 1.

Рассмотрим более подробно данный алгоритм:

1. Ввод 10 концептов НКК, каждый представлен в виде вектора, где каждый элемент соответствует определенному признаку или характеристике концепта.

2. Ввод желаемых ВС. На данном этапе вводится матрица ВС, которая была определена экспертами:

3. Инициализация ВС НС и параметров обучения.

4. Прямое распространение. Вычисление выходного значения НС на основе текущих ВС. Входные данные подаются на вход НС, и происходит прямое распространение сигнала. Каждый нейрон вычисляет свой выход на основе входных данных и текущих ВС.

5. Вычисление ошибки. Результаты прямого распространения сравниваются с желаемыми значениями ВС. Разница между предсказанными значениями и желаемыми значениями представляет ошибку.

6. Обратное распространение ошибки. Ошибка распространяется обратно через НС. Каждый нейрон вычисляет свою часть ошибки на основе ошибки предыдущего слоя и своей функции активации.

7. Обновление ВС. ВС обновляются с использованием алгоритма обратного распространения ошибки и градиентного спуска. Градиент ошибки по ВС вычисляется, и ВС об-

$$W = \begin{bmatrix} 1 & 0.7 & 0.5 & 0.6 & 0 & 0.4 & 0.3 & 0.2 & 0.3 & 0.6 \\ 0.7 & 1 & 0 & 0.6 & 0.4 & 0.3 & 0.2 & 0 & 0 & 0.5 \\ 0.5 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 \\ 0.6 & 0.6 & 0 & 1 & 0.3 & 0.5 & 0.4 & 0.2 & 0.3 & 0.5 \\ 0 & 0.4 & 0 & 0.3 & 1 & 0.2 & 0.6 & 0 & 0 & 0.4 \\ 0.4 & 0.3 & 0 & 0.5 & 0.2 & 1 & 0 & 0 & 0 & 0.3 \\ 0.3 & 0.2 & 0 & 0.4 & 0.6 & 0 & 1 & 0 & 0 & 0.4 \\ 0 & 0 & 0 & 0.4 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.3 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0.6 & 0.5 & 0.4 & 0.5 & 0.4 & 0.3 & 0.4 & 0 & 0 & 1 \end{bmatrix}$$

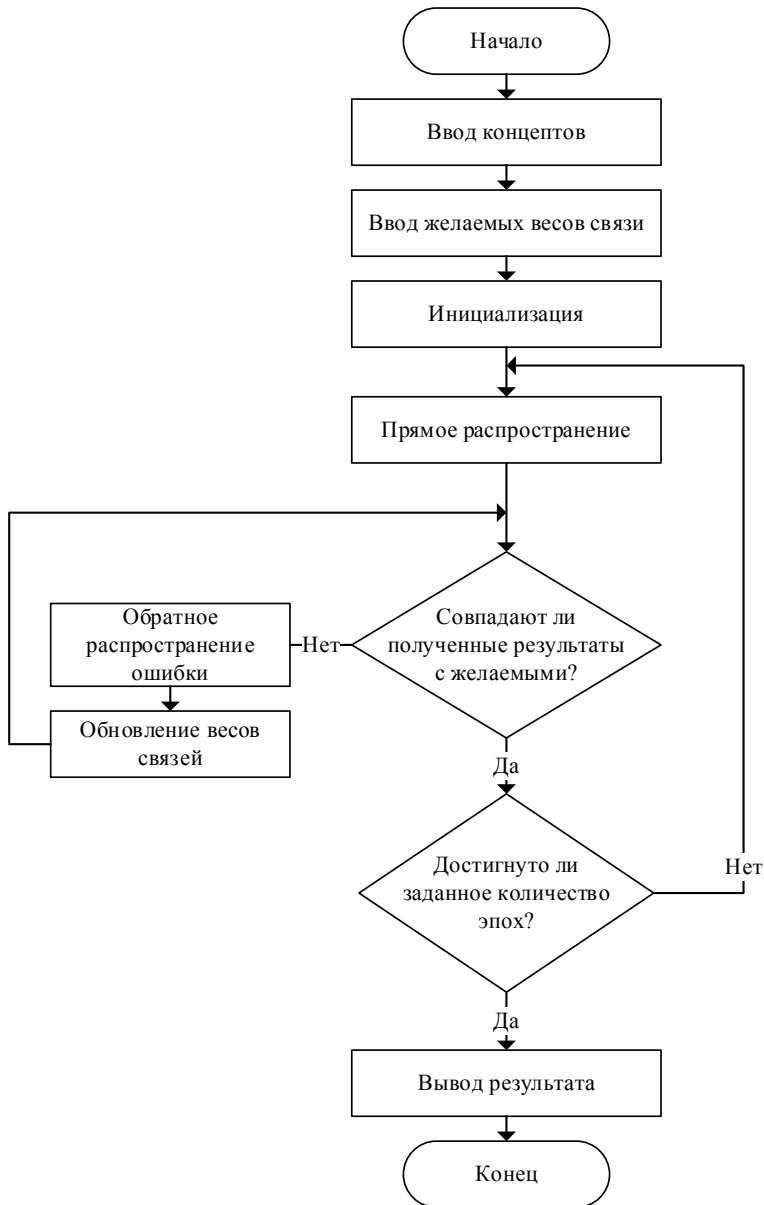


Рис. 1. Алгоритм обучения НС проводить оценку весов связи НКК

новляются в направлении, противоположном градиенту, чтобы минимизировать ошибку.

8. Итерационный процесс. Процессы прямого и обратного распространения ошибки повторяются для каждого обучающего при-

мера в обучающем наборе данных. Обновление ВС происходит на каждой итерации, пока не достигнута заданная точность или не пройдено определенное количество эпох.

9. Завершение обучения. Обучение НС завершается, когда достигнуты требуемые кри-

терии остановки, например, заданное количество эпох или достижение определенной точности на обучающем наборе данных.

Ниже приведен программный код на Python для обучения MLP с использованием алгоритма обратного распространения ошибки и градиентного спуска.

```
# Импорт необходимых библиотек
import numpy as np

class MLP:
    def __init__(self, input_size, hidden_size, output_size):
        # Инициализация весовых коэффициентов
        self.W1 = np.random.randn(input_size, hidden_size)
        self.b1 = np.zeros((1, hidden_size))
        self.W2 = np.random.randn(hidden_size, output_size)
        self.b2 = np.zeros((1, output_size))
    def forward(self, X):
        # Прямое распространение сигнала
        self.hidden_output = np.dot(X, self.W1) + self.b1
        self.hidden_activation = self.sigmoid(self.hidden_output)
        self.output = np.dot(self.hidden_activation, self.W2) + self.b2
        return self.output
    def backward(self, X, y, learning_rate):
        # Обратное распространение ошибки
        delta_output = self.output - y
        delta_hidden = np.dot(delta_output, self.W2.T) * self.sigmoid_derivative(self.hidden_output)
        # Обновление весовых коэффициентов
        self.W2 -= learning_rate * np.dot(self.hidden_activation.T, delta_output)
        self.b2 -= learning_rate * np.sum(delta_output, axis=0, keepdims=True)
        self.W1 -= learning_rate * np.dot(X.T, delta_hidden)
        self.b1 -= learning_rate * np.sum(delta_hidden, axis=0)
    def train(self, X, y, epochs, learning_rate):
        # Обучение MLP
        for epoch in range(epochs):
            output = self.forward(X)
            self.backward(X, y, learning_rate)
    def sigmoid(self, x):
        # Сигмоидная функция активации
        return 1 / (1 + np.exp(-x))
    def sigmoid_derivative(self, x):
        # Производная сигмоидной функции активации
```

```
return self.sigmoid(x) * (1 - self.sigmoid(x))
# Подготовка обучающих данных
X = np.array([[0, 0, 0, 0, 0, 0, 0, 0, 0],
              [0, 1, 1, 0, 1, 0, 0, 0, 1, 0],
              [1, 0, 0, 1, 0, 1, 0, 0, 0, 1],
              [0, 1, 1, 0, 1, 0, 0, 0, 1, 0],
              [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
              [0, 1, 1, 0, 1, 0, 0, 0, 1, 0],
              [1, 0, 0, 1, 0, 1, 0, 0, 0, 1],
              [0, 1, 1, 0, 1, 0, 0, 0, 1, 0],
              [1, 0, 0, 1, 0, 1, 0, 0, 0, 1],
              [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]])
# Входные данные для K1
# Входные данные для K2
# Входные данные для K3
# Входные данные для K4
# Входные данные для K5
# Входные данные для K6
# Входные данные для K7
# Входные данные для K8
# Входные данные для K9
# Входные данные для K10
y = np.array([[1, 0.7, 0.6, 0, 0.4, 0.3, 0.2, 0.3, 0.6],
              [0.7, 1, 0, 0.6, 0.4, 0.3, 0.2, 0, 0, 0.5],
              [0.5, 0, 1, 0, 0, 0, 0, 0, 0, 0.4],
              [0.6, 0.6, 0, 1, 0.3, 0.5, 0.4, 0.2, 0.3, 0.5]],)
# Желаемые значения ВС для K1
# Желаемые значения ВС для K2
# Желаемые значения ВС для K3
# Желаемые значения ВС для K4
# Желаемые значения ВС для K5
# Желаемые значения ВС для K6
# Желаемые значения ВС для K7
# Желаемые значения ВС для K8
# Желаемые значения ВС для K9
# Желаемые значения ВС для K10
# Создание и обучение MLP
mlp = MLP(input_size=10, hidden_size=4, output_size=4)
mlp.train(X, y, epochs=1000, learning_rate=0.1)
# Тестирование обученной MLP
test_input = np.array([[0, 1, 0, 1, 0, 1, 0, 1, 0, 1]])
# Входные данные для тестирования (концепт 5)
predicted_output = mlp.forward(test_input)
print(predicted_output)
Приведенный код создает класс MLP, ко-
```

торый представляет MLP с одним скрытым слоем. В методе forward выполняется прямое распространение сигнала, а в методе backward производится обратное распространение ошибки и обновление весовых коэффициентов. Метод train используется для обучения MLP на обучающих данных.

Чтобы оценить работоспособность предложенного подхода, важно учитывать то, насколько быстро и хорошо НС обучается оценивать ВС НКК. Для этого был использован показатель средней абсолютной ошибки (MAE, Mean Absolute Error) [10]. MAE представляет собой разность между спрогнозированным и фактическим значениями. Чем ближе MAE к нулю, тем точнее модель. MAE рассчитывается по формуле:

$$MAE = (1/n) * \sum |y - \hat{y}| \quad (2)$$

где y – действительные значения, \hat{y} – предсказанные значения, n – количество примеров данных.

Во время тестирования НС на предложенных входных данных было произведено несколько итераций, и на 200 итерации MAE максимально приблизилось к нулю. Таким образом, предложенный подход с сочетанием НКК и НС является применимым к решению вопроса совершенствования процесса оценки рисков ИБ.

Заключение

В работе предложен подход, сочетающий в себе технологии НКК и НС для этапов оценки рисков ИБ. Подход представляет собой применение НКК, содержащей в себе концепты по возможным угрозам ИБ в образовательной сфере. Для оценки ВС было предло-

жено применение НС, с целью устранения разрозненности мнений экспертов во время оценки и устранения необходимости работать с большими данными. Что ускоряет процесс проведения оценки, и делает ее адаптивной.

Для обучения НС требуются входные данные, которые формируются на основе работы экспертов. Были приглашены 10 экспертов ВУЗа из области ИБ: преподаватели, специалисты по ЗИ. Опираясь на открытые источники данных: отчеты компаний, занимающихся вопросами ИБ, статистика утечек данных за крайние годы, БДУ ФСТЭК, а также на свой опыт, эксперты сформировали 10 концептов для НКК в сфере образования, а также расставили ВС между ними. Эти данные далее были использованы для обучения НС расставлять ВС. Был сформирован алгоритм обучения НС, выбрана ее архитектура, представлен программный код на Python для решения данной задачи. Критерием применимости этих технологий является качество и скорость обучения НС оценивать ВС. Для оценки этого критерия был использован показатель средней абсолютной ошибки – MAE.

Дальнейшее развитие данного исследования заключается в проведении экспериментов по совершенствованию НС, корректировке ее архитектуры, размерности слоев с целью подбора наиболее оптимальной версии, а также поиск других блоков из предложенной функциональной модели, в которых можно использовать такую комбинацию технологий как НС и НКК.

Литература

1. Фомин Г. П., Сухорукова И. В., Грибов А. Ф. Адаптивная многокритериальная система управления рисками/ Вестник ростовского государственного экономического университета (РИНХ), 2022. № 1 (77). С. 98-103. URL: <https://www.elibrary.ru/item.asp?id=49424923> (дата обращения 30.07.2023 г.)
2. Глушенко С. А. Адаптивная нейро-нечеткая система оценки рисков информационной безопасности организации/ Бизнес-информатика, 2017. № 1 (39). С. 68-77. URL: <https://bijournal.hse.ru/data/2017/08/30/1173956328/%D0%93%D0%BB%D1%83%D1%88%D0%B5%D0%BD%D0%BA%D0%BE%20%D0%A0%D0%A3%D0%A1%D0%9F%D0%95%D0%A0%D0%95%D0%92%D0%9E%D0%94%20-%20v2F.pdf> (дата обращения 30.07.2023 г.)
3. Джамай Е. В., Зинченко А. С. Разработка адаптивной многокомпонентной системы управления риском высокотехнологичных предприятий/ Вестник Московского государственного областного университета. Серия: Экономика, 2022. № 2. URL: <https://cyberleninka.ru/article/n/razrabotka-adaptivnoy-mnogokomponentnoy-sistemy-upravleniya-riskom-vysokotekhnologichnyh-predpriyatiy> (дата обращения 30.07.2023 г.) DOI: 10.18384/2310-6646-2022-2-49-55
4. Заграновская А. В. Системный анализ на основе нечетких когнитивных карт/Вестник Российского экономического университета им. Г. В. Плеханова, 2018. №4. URL: <https://cyberleninka.ru/article/n/sistemnyy-analiz-na-osnove-nechetkih-kognitivnyh-kart/viewer> (дата обращения 25.08.2023 г.)
5. Пивкин Е.Н., Белов В.М., Белкин С.А. К вопросу об анализе защищенности объектов информа-

тизации с использованием нейронных сетей/ Доклады Томского государственного университета систем управления и радиоэлектроники, 2014. № 2 URL: <https://journal.tusur.ru/storage/44773/30.pdf?1465976630> (дата обращения 25.08.2023 г.)

6. Созыкин А. В. Обзор методов обучения глубоких нейронных сетей/ Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика, 2017. № 3. Том 6. URL: <https://cyberleninka.ru/article/n/obzor-metodov-obucheniya-glubokih-neyronnyh-setey> (дата обращения 29.08.2023 г.)

7. Дрокин И. С. Об одном алгоритме последовательной инициализации весов глубоких нейронных сетей и обучении ансамбля нейронных сетей/ Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления, 2016. № 4. URL: <https://cyberleninka.ru/article/n/ob-odnom-algoritme-posledovatel'noy-initsializatsii-vesov-glubokih-neyronnyh-setey-i-obucheni-ansamblya-neyronnyh-setey> (дата обращения 10.09.2023 г.)

8. Макаренко А. В. Глубокие нейронные сети: зарождение, становление, современное состояние/ Проблемы управления, 2020. № 2. URL: <https://cyberleninka.ru/article/n/glubokie-neyronnye-seti-zarozhdenie-stanovlenie-sovremennoe-sostoyanie> (дата обращения 12.09.2023 г.)

9. Гудфеллоу Ян, Бенджио Иосуа, Курвилль Аарон. Глубокое обучение. 2018

10. How to interpret MAE. URL: <https://stephenallwright.com/interpret-mae/?ref=helenkapatsa.ru> (дата обращения 15.09.2023)

References

1. Fomin G. P., Sukhorukova I. V., Gribov A. F. Adaptivnaya mnogokriterial'naya sistema upravleniya riskami/ Vestnik rostovskogo gosudarstvennogo ekonomicheskogo universiteta (RINKH), 2022. № 1 (77). S. 98-103. URL: <https://www.elibrary.ru/item.asp?id=49424923> (дата обращения 30.07.2023 г.)

2. Glushenko S. A. Adaptivnaya neyro-nechetkaya sistema otsenki riskov informatsionnoy bezopasnosti organizatsii/ Biznes-informatika, 2017. № 1 (39). S. 68-77. URL: <https://bijournal.hse.ru/data/2017/08/30/1173956328/%D0%93%D0%BB%D1%83%D1%88%D0%B5%D0%BD%D0%BA%D0%BE%20%D0%A0%D0%A3%D0%A1%D0%9F%D0%95%D0%A0%D0%95%D0%92%D0%9E%D0%94%20-%20v2F.pdf> (дата обращения 30.07.2023 г.)

3. Dzhamay Ye. V., Zinchenko A. S. Razrabotka adaptivnoy mnogokomponentnoy sistemy upravleniya riskom vysokotekhnologichnykh predpriyatii/ Vestnik Moskovskogo gosudarstvennogo oblastnogo universiteta. Seriya: Ekonomika, 2022. № 2. URL: <https://cyberleninka.ru/article/n/razrabotka-adaptivnoy-mnogokomponentnoy-sistemy-upravleniya-riskom-vysokotekhnologichnykh-predpriyatii> (дата обращения 30.07.2023 г.) DOI: 10.18384/2310-6646-2022-2-49-55

4. Zagranovskaya A. V. Sistemnyy analiz na osnove nechetkikh kognitivnykh kart/ Vestnik Rossiyskogo ekonomicheskogo universiteta im. G. V. Plekhanova, 2018. №4. URL: <https://cyberleninka.ru/article/n/sistemnyy-analiz-na-osnove-nechetkikh-kognitivnykh-kart/viewer> (дата обращения 25.08.2023 г.)

5. Pivkin Ye.N., Belov V.M., Belkin S.A. K voprosu ob analize zashchishchennosti ob'yektov informatizatsii s ispol'zovaniyem neyronnykh setey/ Doklady Tomskogo gosudarstvennogo universiteta sistem upravleniya i radioelektroniki, 2014. № 2 URL: <https://journal.tusur.ru/storage/44773/30.pdf?1465976630> (дата обращения 25.08.2023 г.) 6. Sozykin A. V. Obzor metodov obucheniya glubokikh neyronnykh setey/ Vestnik Yuzhno-Ural'skogo gosudarstvennogo universiteta. Seriya: Vychislitel'naya matematika i informatika, 2017. № 3. Tom

6. URL: <https://cyberleninka.ru/article/n/obzor-metodov-obucheniya-glubokih-neyronnyh-setey> (дата обращения 29.08.2023 г.)

7. Drokin I. S. Ob odnom algoritme posledovatel'noy initsializatsii vesov glubokikh neyronnykh setey i obucheni ansamblya neyronnykh setey/ Vestnik Sankt-Peterburgskogo universiteta. Prikladnaya matematika. Informatika. Protsessy upravleniya, 2016. № 4. URL: <https://cyberleninka.ru/article/n/ob-odnom-algoritme-posledovatel'noy-initsializatsii-vesov-glubokih-neyronnyh-setey-i-obucheni-ansamblya-neyronnyh-setey> (дата обращения 10.09.2023 г.)

8. Makarenko A. V. Glubokiye neyronnyye seti: zarozhdeniye, stanovleniye, sovremennoye sostoyaniye/ Problemy upravleniya, 2020. № 2. URL: <https://cyberleninka.ru/article/n/glubokie-neyronnye-seti-zarozhdenie-stanovlenie-sovremennoe-sostoyanie> (дата обращения 12.09.2023 г.)

9. Gudfellow Yan, Bendzhio Ioshua, Kurvill' Aaron. Glubokoye obucheniye. 2018

10. How to interpret MAE. URL: <https://stephenallwright.com/interpret-mae/?ref=helenkapatsa.ru> (дата обращения 15.09.2023)

ПАЛЮТИНА Галия Наилевна, аспирант, младший научный сотрудник Федерального государственного бюджетного образовательного учреждения высшего образования «Ростовский

государственный экономический университет (РИНХ)». 344002, г. Ростов-на-Дону, ул. Большая Садовая, д. 69. E-mail: hello616@yandex.ru

PALYUTINA Galiya Nailevna, postgraduate student, junior researcher at the Federal State Budgetary Educational Institution of Higher Education "Rostov State Economic University (RINH)". 344002, Rostov-on-Don, st. Bolshaya Sadovaya, 69. E-mail: hello616@yandex.ru