

ISSN 1999-4125 (Print)

ISSN 2949-0642 (Online)

Научная статья

УДК 622.831

DOI: 10.26730/1999-4125-2025-6-145-156

КОНЦЕПТУАЛИЗАЦИЯ АРХИТЕКТУРЫ МОДЕЛИ МАШИННОГО ОБУЧЕНИЯ ДЛЯ КЛАССИФИКАЦИИ ПРОИЗВОДСТВЕННЫХ ТРАВМ НА ГОРНОДОБЫВАЮЩИХ ПРЕДПРИЯТИЯХ ПО ИХ СТЕПЕНИ ТЯЖЕСТИ

Майтак Роман Вячеславович, Пылов Петр Андреевич,
Николаева Евгения Александровна, Дягилева Анна Владимировна

Кузбасский государственный технический университет имени Т. Ф. Горбачева

* для корреспонденции: maytak.roman@mail.ru

Аннотация.

Горнодобывающие предприятия – это производственные комплексы, которые осуществляют разработку месторождений и (или) переработку полезных ископаемых. Как и на любом другом производственном предприятии, работа на горнодобывающем комплексе сопровождается совокупностью рисков и опасностей, которые являются основной причиной возникновения несчастных случаев. Для обеспечения нормальных условий труда в составе каждого предприятия существуют ответственные за техносферную (промышленную) безопасность, которые следят за исполнением регламента техники безопасности производственного процесса. Несмотря на то, что главной задачей этих специалистов является превентивное предупреждение аварийных ситуаций на производстве, они также несут ответственность за безопасность технологических процессов в условиях чрезвычайной ситуации. С наступлением цифровой трансформации многие задачи техносферной и промышленной безопасности были автоматизированы (в том числе интеллектуальными системами) для выполнения более точного превентивного предсказания аварий на производстве. Однако кроме предупреждения аварий и несчастных случаев также существует задача присвоения класса тяжести несчастного случая в тех ситуациях, когда он уже произошел. Меньшая популярность данной тематики (в сравнении с превентивным прогнозом аварий шахтного оборудования) не является основанием для того, чтобы не автоматизировать этот процесс. Поскольку класс тяжести несчастного случая присваивает человек (по определенному набору параметров, описывающих произошедший несчастный случай), то это означает, что сохраняется вероятность возникновения ошибки – человеческого фактора, нивелировать которую может только программная модель. Пробелы в этой области знаний, несомненно, должны восполняться новыми исследованиями и моделями, которые бы позволяли определять класс тяжести несчастного случая без вмешательства человека. Вышеизложенный материал определяет актуальность обозначенной темы исследования, основанной на автоматизации процесса определения класса тяжести несчастного случая на горнодобывающих предприятиях. Объектом исследования является процесс определения класса тяжести несчастного случая по известному набору прецедентов, описывающих несчастный случай. Предметом исследования является разработка модели глубокого обучения для аналитики описательного набора прецедентов.



Информация о статье

Поступила:

22 сентября 2025 г.

Одобрена после

рецензирования:

15 ноября 2025 г.

Принята к публикации:

02 декабря 2025 г.

Опубликована:

22 декабря 2025 г.

Ключевые слова:

шахта, угольная отрасль, интеллектуальный анализ данных, интеллектуальные языковые модели, обработка данных моделями глубокого и машинного обучения

Для цитирования: Майтак Р.В., Пылов П.А., Николаева Е.А., Дягилева А.В. Концептуализация архитектуры модели машинного обучения для классификации производственных травм на горнодобывающих предприятиях по их степени тяжести // Вестник Кузбасского государственного технического университета. 2025. № 6 (172). С. 145-156. DOI: 10.26730/1999-4125-2025-6-145-156, EDN: TRXYOA

Введение (Introduction). Любой тип технологической деятельности человека связан с рисками и опасностями. Горнодобывающие предприятия также не являются исключением в этом трудовом производственном классе.

Условно все возможные варианты опасных ситуаций можно подразделить на две большие категории:

1. Опасности, связанные с профессиональной деятельностью работника;
2. Опасности, не связанные с профессиональной деятельностью работника.

Первая категория опасностей абсорбирует в себе опасности от элементов и условий, возникающих вследствие наличия производственного процесса как такового. Например:

- Инструменты и предметы труда, которые имеют потенциально травмирующую форму (острые края, заостренные кромки и так далее);
- Наличие движущегося оборудования и его частей, конструкция которых способна нанести травму;
- Наличие работ, связанных с монтажом и обслуживанием электрических сетей, находящихся под высоким напряжением;
- Высотные работы.

Вторая категория опасностей включает в себя опасности, которые возникают из-за внешних (естественных или искусственных) условий:

- Физическое размещение производственных объектов в особых условиях пространства: над землей, под землей (шахты и разрезы), на воде, под водой, в безвоздушном пространстве (космосе).

Повреждение производственных объектов в данном случае способно привести к возникновению опасных ситуаций;

- Неблагоприятные природные условия (климатические и физико-географические): полярные, пустынные или необжитые, которые способны привести к опасным событиям;
- Физическое размещение производственных объектов вблизи техногенных источников опасности: атомных электростанций, электростанций, плотин и так далее.

Повреждения и аварии на техногенных объектах могут привести к возникновению цепной аварийной реакции – возникновению аварии на основном производственном предприятии.

Методы (Methods). Исследуя плоскость классифицированных опасностей, спроецированную на горнодобывающие

предприятия, можно сделать вывод о том, что промышленные комплексы, осуществляющие добычу полезных ископаемых с помощью системы горных выработок аккумулируют опасности из двух категорий, утвержденных Приказом Минтруда России от 31.01.2022 N 36 «Об утверждении Рекомендаций по классификации, обнаружению, распознаванию и описанию опасностей». Добыча полезных ископаемых задействует сложное техническое оборудование, которое имеет в своем составе большое количество движущихся частей [первая категория]. Нельзя не учитывать и тот факт, что физическое размещение шахт и разрезов организовано под землей, то есть в особых условиях пространства [вторая категория].

Таким образом, возникновению несчастных случаев с сотрудниками горнодобывающих предприятий равносильно могут способствовать факторы обеих категорий (Приказ Минтруда России от 31.01.2022 N 36).

Отметим, что сама по себе опасность является лишь состоянием, которое может привести к одному исходу из двух взаимоисключающих:

- Происходит несчастный случай на производстве;
- Несчастного случая не происходит.

Второй исход является благоприятным и его постоянное присутствие считается атрибутом нормальной рабочей смены. Возникновение первого исхода вызывает последствия, которые требуют оперативного вмешательства и корректировки рабочей смены [1]. Наиболее очевидным последствием является травма человека (сотрудника предприятия или третьего лица). Поскольку травмой считается повреждение тканей организма человека (анатомической целостности) внешним воздействием, то, формально, даже едва заметная царапина также относится к травме. По этой причине одной из важнейших описательных характеристик травмы является ее степень тяжести (класс тяжести).

Ранжирование степени тяжести травмы может корректироваться от предприятия к предприятию, но постоянной остается методика правильного присвоения класса тяжести травмы. Как правило, при фиксации производственной травмы записывается информация о том, что произошло, в формате текстовой описательной характеристики [1]. Присвоение класса возлагается на сотрудника, который должен

будет изучить полученную текстовую информацию и определить степень тяжести.

Такой подход является традиционным и скрывает за собой ряд преимуществ и недостатков: с одной стороны, наиболее быстро и просто зафиксировать информацию о несчастном случае в текстовом виде, а потом передать ее для оценки уполномоченному лицу. Очевидным недостатком подхода является низкая автоматизация процесса и высокая вероятность возникновения ошибки, носящей человеческий фактор – человек (особенно, если

уполномоченное для определения класса опасности лицо является единственным сотрудником филиала предприятия), который постоянно занимается монотонной работой, основанной на продолжительном анализе однотипной (текстовой) информации, гораздо чаще допускает ошибки по естественной причине снижения концентрации внимания [2].

Отметим также, что традиционный подход существует уже очень давно и сложился в те времена, когда не существовало цифрового формата данных [2, 3]. С появлением

Таблица 1. Информация о наборе исследуемых данных

Table 1. Information about the dataset under study

Русскоязычное представление признака	Название признака в проекте решения	Понятийное значение признака
Информация о времени (дате) случившегося несчастного случая в условиях промышленного предприятия	Data	Столбец данных содержит временную метку или информацию о времени/дате произошедшего несчастного случая
Регион (область) географического местоположения произошедшего события несчастного случая	Countries	Фиксирует географические данные о том, в каком регионе был зарегистрирован несчастный случай на горнодобывающем предприятии
Географическое местоположение местности, в которой находится промышленное предприятие горнодобывающей сферы	Local	Город (или его внутренний географический ориентир, например, северо-западная часть), в котором находится горнодобывающее предприятие (или его филиал)
Сектор (отдел) горнодобывающего промышленного предприятия, в котором был зарегистрирован несчастный случай	Industry sector	Определяет, в каком секторе внутреннего административного деления горнодобывающего предприятия был зарегистрирован несчастный случай
Класс тяжести несчастного случая	Accident level	Ранжирован от I до V. Регистрирует, насколько значительным был несчастный случай с точки зрения уровня последствий: I означает «незначительно серьезный», а V характеризует произошедший несчастный случай как «очень серьезный»
Потенциальный уровень аварийной ситуации	Potential Accident Level	В зависимости от уровня тяжести несчастного случая в данном столбце регистрируется, насколько серьезной могла быть авария (из-за других факторов, связанных с несчастным случаем и важностью обязанностей пострадавшего)
Пол пострадавшего	Genre	Определяет пол пострадавшего человека: мужчина или женщина
Тип пострадавшего	Employee or Third Party	Детерминирует пострадавшую сторону: лицо может являться сотрудником горнодобывающего предприятия или быть третьей (независимой) стороной (например, приглашенным деловым партнером или сторонним экспертом)
Критический риск	Critical Risk	Формирует некоторое описание риска, связанного с несчастным случаем
Описание несчастного случая	Description	Содержит в себе информацию с подробным описанием того, как произошел несчастный случай

персональных компьютеров возникла необходимость в цифровой трансформации, так как многие задачи гораздо удобнее и эффективнее решать в цифровом и электронном формате. Однако на сегодняшний день¹ такая методика интегрирована еще не во все сферы технологических предприятий (в частности, горнодобывающих). Задача определения класса тяжести произошедшего несчастного случая по его описательной характеристике может быть автоматизирована методами прикладного искусственного интеллекта, так как для этих целей текстовое описание является необходимым и достаточным условием для детерминирования [4]. Благодаря автоматизации возникает возможность более быстрой аналитики информации и присвоения класса несчастному случаю [1, 2]. Кроме этого, обработка естественного языка моделью искусственного интеллекта позволяет сегментировать формулировки проблем, которые привели к возникновению несчастного случая с человеком, позволяя тем самым выполнять дескрипцию проблемы [4]: модель будет присваивать бинарный признак основы несчастного случая: вычленение отдельных поврежденных частей человеческого организма, которое на данный момент фиксируется только в виде текстового описания, будет доступно в формате дополнительной электронных столбцов данных (например, дополнительные столбец «повреждение верхней левой конечности» будет установлен в соответствие к значению «1», если повреждена левая рука, и к значению «0», если левая рука человека не была повреждена при возникновении несчастного случая) [17, 18].

Таким образом, кроме автоматизированного присвоения класса тяжести несчастному случаю модель искусственного интеллекта сформирует дополнительный набор данных, который позволяет подробнее рассмотреть проблемы методами компьютерной аналитики [3, 4].

Результаты исследования (Result). Представленный для исследования набор данных содержит информацию об атрибутах данных (прецедентах) и целевом прогнозируемом столбце (класс тяжести несчастного случая) [16, 19]. Каждая строчка (кортеж) датасета – это зарегистрированный произошедший несчастный случай. Удобнее всего данные можно представить в формате сводной результирующей таблицы. В Таблице 1 зафиксировано наименование признака (такое, каким оно будет представлено в дальнейшем проекте решения), его русскоязычное представление и поясняющая семантику термина описательная характеристика.

Таблица 1 в описательной форме позволяет предоставить информацию о прецедентах набора исследуемых данных. По этой информации уже можно выполнить детальную математическую аналитику признаков датасета для того, чтобы уже на начальном шаге определить параметры для разработки оптимальной модели глубокого обучения.

Большинство данных Таблицы 1 могут быть закодированы в виде номинативных признаков, что позволит использовать их в моделях глубокого обучения, основанных на вероятностных эвристиках [2, 3]. Однако для реализации модели распознавания и аналитики естественного языка требуется разработать модель глубокого обучения, основанную на векторизации текстовой информации [4]. Внутренняя математическая основа таких моделей основана не на вероятностных эвристиках, поэтому для объединенного анализа как текстовых, так и номинативных прецедентов необходимо будет разработать две модели глубокого обучения, а затем провести их взаимную коалесценцию [4, 5]. Коалесценция моделей глубокого обучения резко отличается от понятия и операции ансамблирования [3, 5]. Главной особенностью ансамблевых моделей является независимая работа нескольких алгоритмов прикладного искусственного интеллекта, после завершения цикла которых производится взвешенное голосование и избирается лучшая точность решения какого-то одного алгоритма для каждого нового случая данных (кортежа). Коренным отличием операции коалесценции является конкатенация обобщающей (аппроксимирующей) функции нескольких моделей в одну общую структуру модели. Таким образом, атомарные модели глубокого обучения уже не существуют по отдельности, а рассчитывают итоговое значение точности при решении каждой новой строки данных (кортежа) сообща [4, 5].

Программная реализация будет выполнена на высокоуровневом языке программирования Python, так как это обусловлено следующими факторами:

- Большинство подключаемых программных библиотек для реализации алгоритмов машинного обучения написаны на низкоуровневых языках программирования (C, C++). Такой подход позволяет получать максимальную временную эффективность при относительно низких аппаратных затратах;
- Язык программирования Python написан таким образом, что обеспечивается и сохраняется возможность переноса написанных на нем программ на другие языки программирования (в частности, возможен перенос разработанных моделей машинного

¹ По состоянию на конец февраля 2024 года.

обучения с уже рассчитанными весами и параметрами);

- Язык относится к интерпретируемым типам языков программирования, что составляет основу для написания скриптов выполнения программного кода. Скрипты имеют большую популярность, особенно тогда, когда задачу требуется решить при минимальном задействовании аппаратных и программных ресурсов.

На основе имеющегося набора данных необходимо разработать модель глубокого обучения, которая будет определять класс тяжести несчастного случая на основе прецедентов набора данных (таблица 1). Поскольку номинативные прецеденты и признаки, содержащие информацию на естественном языке, должны обрабатываться разными моделями прикладного искусственного интеллекта [6, 7], то предварительная обработка датасета должна быть выполнена таким образом, чтобы на основе очищенных данных можно было одинаково хорошо построить любую модель глубокого обучения [7, 8].

Наиболее оптимальным программным вариантом построения нейронной сети будет реализация архитектуры из трех полносвязных слоев [9]. При этом, чтобы избежать

потенциально возможного переобучения, необходимо добавить после двух первых полносвязных слоев слой прореживания [10], после которых будут следовать слои нормализации. Такая архитектура позволит повысить стабильность процесса обучения нейронной сети и сформировать высокую обобщающую способность [2]. Графическое представление описанной архитектуры представлено на Рис. 1.

Обсуждение (Discussion). Перейдем к программной реализации модели нейронной сети, представленной на Рис. 1. Первоначально определим новые независимые переменные (Рис. 2) для критериев:

- набор номинативных прецедентов обучающей выборки;
- целевой столбец ответов для обучающей выборки;
- набор номинативных прецедентов для тестовой выборки;
- целевой столбец ответов для тестовой выборки.

Процедура создания дубликатов набора данных необходима по двум причинам:

1. В процессе построения модели нейронной сети могут быть повреждены данные,

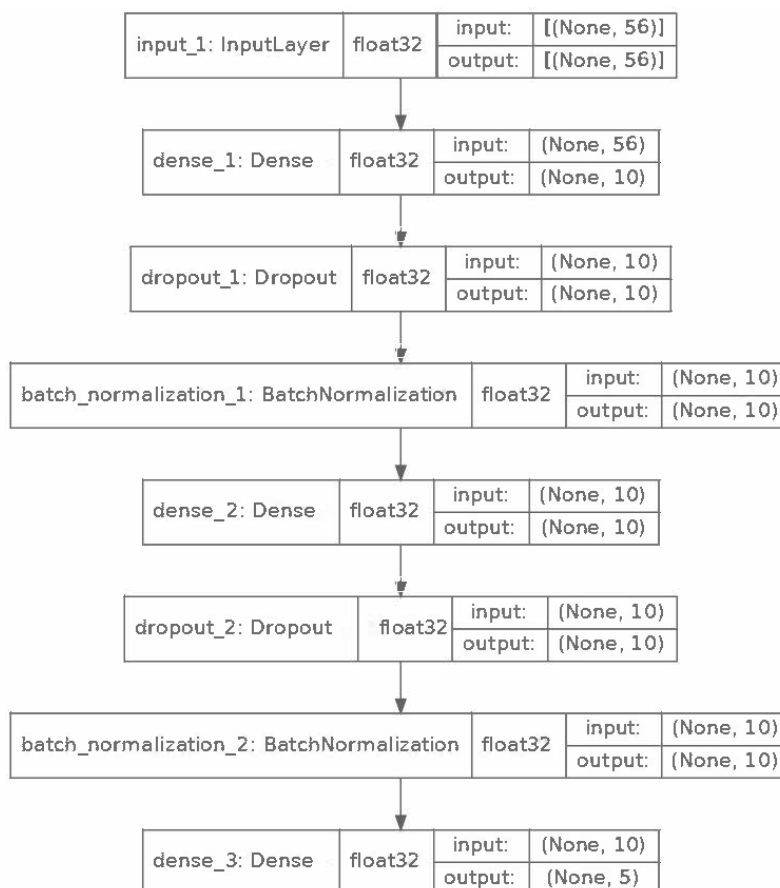


Рис. 1. Архитектура слоев последовательной модели нейронной сети для анализа номинативных признаков

Fig. 1. Architecture of layers of a sequential neural network model for analysing nominative features

поэтому обязательно должна присутствовать копия первоначальных элементов;

2. Очищенные данные требуются не только для одной модели глубокого обучения, их также будет задействовать нейронная сеть для обработки естественного языка.

При анализе номинативных признаков целевая переменная должна быть также закодирована в формате вектора [2, 11]. Этот прием позволяет построить более точные аппроксимирующие связи при формировании обобщающей способности. Реализация кодировки представлена на Рис. 3. Поскольку целевая переменная теперь представлена в «новом», модифицированном формате, то соответствующему масштабированию должны быть подвергнуты и признаки набора данных.

Теперь, когда все подготовлено для программной реализации архитектуры модели нейронной сети, можно воспроизвести ее

алгоритмическое описание (рисунок 4).

В качестве метрики контроля качества обучения модели был выбран критерий accuracy [2, 12]. Он является численным отношением правильно классифицированных моделью объектов выборки ко всем исследуемым объектам набора данных.

Результаты исследования (Results). Перед инициацией процесса обучения нейронной сети необходимо убедиться в верном переносе схемы (представленной на Рис. 1) архитектуры модели в ее программный вид (представленный на Рис. 4). Для этого достаточно вызвать метод «summary», который выводит строковую сводку необходимой информации о разработанной нейронной сети (Рис. 5).

Отметим, что индексы слоев (Рис. 5) не обязательно должны совпадать с обозначениями архитектуры (Рис. 1) [15]. Главным критерием, регулирующим правильность переноса, является

```
X_nominative_train = X_train
X_nominative_test = X_test
y_nominative_train = y_train
y_nominative_test = y_test

print('Длина обучающей выборки номинативных прецедентов : {}'.format(X_nominative_train.shape[0]))
print('Длина обучающей выборки (серии) целевого класса : ({}))\n'.format(y_nominative_train.shape[0]))
print('Длина тестовой выборки номинативных прецедентов : {}'.format(X_nominative_test.shape[0]))
print('Длина тестовой выборки (серии) целевого класса : ({}))\n'.format(y_nominative_test.shape[0]))

Длина обучающей выборки номинативных прецедентов : 334
Длина обучающей выборки (серии) целевого класса : (334)

Длина тестовой выборки номинативных прецедентов : 84
Длина тестовой выборки (серии) целевого класса : (84)
```

Рис. 2. Определение новых независимых фреймов данных для построения модели нейронной сети
Fig. 2. Definition of new independent data frames for building the neural network model

```
# Преобразуем векторы номинативных признаков в векторы с односторонним кодированием
y_nominative_train = np_utils.to_categorical(y_nominative_train)
y_nominative_test = np_utils.to_categorical(y_nominative_test)
scaler_X = StandardScaler()
X_nominative_train.iloc[:, :6] = scaler_X.fit_transform(X_nominative_train.iloc[:, :6])

X_nominative_test.iloc[:, :6] = scaler_X.fit_transform(X_nominative_test.iloc[:, :6])
```

Рис. 3. Кодировка переменной в формате вектора и соответствующее ей масштабирование данных прецедентов
Fig. 3. Variable encoding in vector format and its corresponding scaling of precedent data

```
input1 = Input(shape=(X_nominative_train.shape[1],))
dense_layer_1 = Dense(10, input_dim=X_nominative_train.shape[1], \
    activation='relu', kernel_initializer='he_uniform', kernel_regularizer=l2(1e-9), \
    kernel_constraint=unit_norm()(input1))
drop_out_layer_1 = Dropout(0.2)(dense_layer_1)
batch_norm_layer_1 = BatchNormalization()(drop_out_layer_1)
dense_layer_2 = Dense(10, activation='relu', kernel_initializer='he_uniform', kernel_regularizer=l2(1e-9), \
    kernel_constraint=unit_norm()(batch_norm_layer_1))
drop_out_layer_2 = Dropout(0.5)(dense_layer_2)
batch_norm_layer_2 = BatchNormalization()(drop_out_layer_2)
dense_layer_3 = Dense(5, activation='softmax', kernel_regularizer=l2(1e-9), kernel_constraint=unit_norm()(batch_norm_layer_2))

model = Model(inputs=input1, outputs=dense_layer_3)

opt = optimizers.Adamax(lr=0.01)
model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['acc'])
```

Рис. 4. Программная реализация архитектуры модели нейронной сети
Fig. 4. Software implementation of the neural network model architecture

последовательность слоев в модели нейронной сети. Идентичность слоев на Рис. 1 и 5 подтверждает правильность алгоритмизации модели. После завершения обучения необходимо детерминировать точность разработанной модели на основе метрики ассигасу (Рис. 6).

Отметим, что индексы слоев (Рис. 5) не обязательно должны совпадать с обозначениями архитектуры (Рис. 1) [15]. Главным критерием, регулирующим правильность переноса, является последовательность слоев в модели нейронной сети. Идентичность слоев на Рис. 1 и 5

подтверждает правильность алгоритмизации модели. После завершения обучения необходимо детерминировать точность разработанной модели на основе метрики ассигасу (Рис. 6).

Анализ снижения ошибки с увеличением числа эпох позволит отследить на графике, был ли достигнут минимум ошибки [11, 13]. На основании этого можно будет однозначно определить, достаточное ли количество эпох было выбрано для обучения модели.

Первоначально алгоритмизуем функционал сбора статистических данных (Рис. 7).

```
# Проверка соответствия реализованной модели своей схематичной структуре
print(model.summary())
```

Model: "model_12"

Layer (type)	Output Shape	Param #
input_19 (InputLayer)	[(None, 57)]	0
dense_65 (Dense)	(None, 10)	580
dropout_54 (Dropout)	(None, 10)	0
batch_normalization_24 (Batch Normalization)	(None, 10)	40
dense_66 (Dense)	(None, 10)	110
dropout_55 (Dropout)	(None, 10)	0
batch_normalization_25 (Batch Normalization)	(None, 10)	40
dense_67 (Dense)	(None, 5)	55

=====
Total params: 825
Trainable params: 785
Non-trainable params: 40

None

Рис. 5. Строковая сводка информации о модели нейронной сети

Fig. 5. String summary of information about the neural network model

```
# Оцениваем точность разработанной модели
_, train_accuracy = model.evaluate(X_nominative_train, y_nominative_train, batch_size=8, verbose=0)
_, test_accuracy = model.evaluate(X_nominative_test, y_nominative_test, batch_size=8, verbose=0)

print('Точность на обучающей выборке: %.2f' % (train_accuracy*100))
print('Точность на тестовой выборке: %.2f' % (test_accuracy*100))
```

Точность на обучающей выборке: 73.95
Точность на тестовой выборке: 73.81

Рис. 6. Оценка точности обученной нейронной сети на тестовых данных

Fig. 6. Evaluation of the accuracy of the trained neural network on test data

```
epochs = range(len(training_history.history['loss'])) # Получаем данные о порядковом номере эпохи обучения

plt.plot(epochs, training_history.history['loss'], label = 'Обучающая выборка', color = 'red')
plt.plot(epochs, training_history.history['val_loss'], label = 'Тестовая выборка', color = 'green')
plt.legend(loc = 'upper right')
plt.title('Функция потерь на обучающей и тестовой выборке');
```

Рис. 7. Определение необходимой статистики для построения графика зависимости числа эпох от величины ошибки

Fig. 7. Determination of the necessary statistics for plotting the dependence of the number of epochs on the magnitude of the error

Программный функционал (Рис. 7) позволяет получить графическую взаимосвязь итераций эпох от снижения ошибки (Рис. 8).

Из Рис. 8 следует, что приблизительно после 40 эпохи обучения на тестовых данных перестала заметно снижаться величина ошибки –

это означает, что количество эпох обучения было достаточным для формирования обобщающей способности модели [14].

Скачкообразное поведение функции зависимости на обучающей выборке подтверждает необходимость наличия



Рис. 8. График зависимости эпох от величины ошибки
Fig. 8. Graph of dependence of epochs on the magnitude of error

```
plt.plot(epochs, training_history.history['acc'], label = 'Обучающая выборка', color = 'red')
plt.plot(epochs, training_history.history['val_acc'], label = 'Тестовая выборка', color = 'green')
plt.legend(loc = 'lower right')
plt.title ('Точность на обучающей и тестовой выборке данных');
```

Рис. 9. Программный код для отрисовки графика зависимости точности модели глубокого обучения от числа эпох

Fig. 9. Programme code for drawing the graph of dependence of deep learning model accuracy on the number of epochs



Рис. 10. График зависимости точности на обучающей и тестовой выборке от числа эпох обучения
Fig. 10. Graph of the dependence of accuracy on the training and test samples on the number of training epochs


```
# Сохраняем веса номинативной модели
model.save_weights("model.h5")
print("Веса номинативной модели нейронной сети были сохранены в корневой папке")

# Сохраняем саму номинативную модель
model.save("nominative_model.h5")
print("Модель сохранена в корневой папке")
```

Веса номинативной модели нейронной сети были сохранены в корневой папке
Модель сохранена в корневой папке

Рис. 11. Сохранение номинативной модели нейронной сети и ее весовых коэффициентов
Fig. 11. Preserving the nominal model of the neural network and its weight coefficients

прореживающих и нормализующих слоев, так как без них нейронная сеть имела бы оптимистически заниженную кривую (обозначенную красным цветом) снижения ошибки на обучающей выборке и совершенно отличающиеся значения кривой (обозначенной зеленым цветом) на тестовой выборке.

Аналогичным образом проанализируем поведение точности модели нейронной сети в зависимости от порядковых номеров эпох обучения. Первоначально реализуем программный код графического представления информации (Рис. 9).

На основе программного кода (Рис. 9) получаем график (представленный на Рис. 10) зависимости точности модели от итераций эпох обучения.

Обсуждение (Discussion). Согласно полученным данным (Рис. 10), точность плавно возрастала с самых начальных эпох обучения. С ростом числа итераций эпох точность увеличивалась лишь незначительно, что характеризует окончательно сформированную обобщающую способность. Отметим, что искусственное увеличение точности (переобучение) было исключено прореживающими слоями в реализованной архитектуре нейронной сети.

Финальный шаг разработки текущей версии номинативной модели нейронной сети – сохранение весовых коэффициентов модели и самой сети в локальном хранилище для дальнейшего использования (модифицирования). Для этого вызовем соответствующие методы сохранения (Рис. 11).

Выводы (Reference). В результате программной реализации, основанной на автоматизации процесса определения класса тяжести несчастного случая в горнодобывающих промышленности, были решены следующие задачи:

1. Для решения задачи определения класса тяжести несчастного случая по набору номинативных признаков целесообразно использовать модель нейронной сети с тремя плотно связанными скрытыми слоями и двумя слоями пакетной нормализации;

2. Для решения задачи определения класса тяжести несчастного случая по описанию текстового признака целесообразно использовать модель рекуррентной нейронной сети с одним слоем встраивания, одним двунаправленным слоем и одним слоем агрегирующего типа;
3. Доказана возможность применения разработанной модели на основе объективно полученного значения точности, которое превышает значение 0.7 по прикладной метрике ассюрасу.

Цифровая трансформация помогает автоматизировать задачи техносферной и промышленной безопасности. Но на производстве также существует задача присвоения класса тяжести несчастного случая в тех ситуациях, когда он уже произошел. Данную оценку присваивает человек, это означает, что сохраняется вероятность возникновения ошибки, нивелировать которую поможет созданная программная модель.

СПИСОК ЛИТЕРАТУРЫ

1. Киселев А. Промышленная безопасность опасных производственных объектов М. : Альфа-Пресс, 2017. 240 с.
2. Serdyuk D., Braga O., Siohan O. Audio-visual speech recognition is worth 32x32x8 voxels // 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). 2021.
3. Sung Y.-L., Cho J., Bansal. M. V1-adapt: Parameter-efficient transfer learning for vision and language tasks // In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022. Pp. 5227-5237.
4. Srinivasan T., Sanabria R., Metze F. Looking enhances listening: Recovering missing speech using im- ages. ICASSP. 2020.
5. Synnaeve G., Xu Q., Kahn J., Likhomanenko T., Grave E., Pratap V., Sriram A., Liptchinsky V., Collobert R. End-to-end asr: from supervised to semi-supervised learning with modern architectures. 2019. arXiv preprint arXiv:1911.08460.
6. Bishop C.M. Pattern Recognition and Machine Learning // Springer. 2006. 738 с.
7. Солтис М. Введение в анализ алгоритмов СПб. : ДМК. 2019. 269 с.
8. Домингос П. Верховный алгоритм М. :

«МАНН, ИВАНОВ И ФЕРБЕР». 2016. 315 с.

9. Гифт Н. Программный ИИ. СПб. : Питер. 2019. 304 с.

10. Кормен Т., Лейзерсон Ч. Алгоритмы: построение и анализ, 3-е издание М. : ООО И. Д. Вильямс, 2013. 1328 с.

11. Xu Q., Baevski A., Likhomanenko T., Tomassello P., Conneau A., Collobert R., Synnaeve G., Auli M. Self-training and pre-training are complementary for speech recognition // In ICASSP. 2021.

12. Sanabria R., Caglayan O., Palaskar Sh., Elliott D., Barrault L., Specia L., Metze F. How2: a large-scale dataset for multimodal language understanding // In Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL). 2018.

13. Пылов П. А., Майтак Р. В., Зайцева Е. Г. Применение мультимодального трансформера для прогнозирования выходных параметров насыщенных углеводородных соединений из состава тяжелой нефти в присутствии катализаторов // Труды Института системного программирования РАН. 2023. Т. 35. № 5. С. 229-244. DOI: 10.15514/ISPRAS-2023-35(5)-15. EDN NLWIYD.

14. Роджерс Д. Алгоритмические основы машинной графики. М. : МИР. 2022. 687 с.

15. Босов А. Е. Машинное обучение и дискретная математика: как использование комбинаторики и теории графов улучшает

производительность алгоритмов машинного обучения // В сборнике: Наука и молодежь: новые идеи и решения. Материалы XVII Международной научно-практической конференции молодых исследователей. Волгоград, 2023. С. 193-194.

16. Mikhailov A. M., Rabeeakh S. K. Trading bots that use machine learning // В сборнике: Наука преобразует реальность. Материалы Международной междисциплинарной научно-практической студенческой конференции. Отв. редактор Е. А. Маклакова. Воронеж, 2023. С. 13-19.

17. Шаповал И. В., Рузаков В. О., Каримова Л. К., Мулдашева Н. А., Бейгул Н. А., Ильина Л. А., Волгарева А. Д. Производственный травматизм на горнодобывающих предприятиях республики Башкортостан в 2017-2022 годах // Безопасность и охрана труда. 2024. № 1 (98). С. 8-12.

18. Крупенченкова С. А. Производственный травматизм на предприятиях горнодобывающей промышленности // В сборнике: Наука и практика: актуальные вопросы, достижения и инновации. сборник статей VI Международной научно-практической конференции. Пенза, 2025. С. 25-27.

19. Карначев И. П., Челтыбашев А. А., Судак С. Н. Практическая реализация концепции культуры промышленной безопасности на примере крупнейших горных предприятий // Безопасность техногенных и природных систем. 2022. № 3. С. 24-31.

© 2025 Авторы. Эта статья доступна по лицензии Creative Commons «Attribution» («Атрибуция») 4.0 Всемирная (<https://creativecommons.org/licenses/by/4.0/>)

Авторы заявляют об отсутствии конфликта интересов.

Об авторах:

Майтак Роман Вячеславович, магистр, Кузбасский государственный технический университет имени Т. Ф. Горбачева, 650000, Россия, г. Кемерово, ул. Весенняя, 28, e-mail: maytak.roman@mail.ru

Пылов Петр Андреевич, аспирант, Кузбасский государственный технический университет имени Т. Ф. Горбачева, 650000, Россия, г. Кемерово, ул. Весенняя, 28, e-mail: gedrosten@mail.ru

Николаева Евгения Александровна, кандидат физико-математических наук, заведующая кафедрой математики, Кузбасский государственный технический университет имени Т. Ф. Горбачева, 650000, Россия, г. Кемерово, ул. Весенняя, 28, e-mail: nikolaevaea@yandex.ru

Дягилева Анна Владимировна, кандидат технических наук, доцент кафедры математики, Кузбасский государственный технический университет имени Т. Ф. Горбачева, 650000, Россия, г. Кемерово, ул. Весенняя, 28, e-mail: dyagileva1952@mail.ru

Заявленный вклад авторов:

Майтак Роман Вячеславович – сбор и анализ данных, нейронная сеть.

Пылов Петр Андреевич – концептуализация исследования, программная реализация, научный менеджмент, выводы.

Николаева Евгения Александровна – выводы, оформление текста.

Дягилева Анна Владимировна – постановка задачи, обзор литературы.

Все авторы прочитали и одобрили окончательный вариант рукописи.

Original article

CONCEPTUALIZING THE ARCHITECTURE OF A MACHINE LEARNING MODEL FOR CLASSIFYING OCCUPATIONAL INJURIES IN MINING ENTERPRISES ACCORDING TO THEIR SEVERITY

Roman V. Maitak, Pyotr A. Pylov,
Evgenia A. Nikolaeva, Anna V. Diaghileva

T.F. Gorbachev Kuzbass State Technical University

* for correspondence: maytak.roman@mail.ru



Article info

Received:

22 September 2025

Accepted for publication:

15 November 2025

Accepted:

02 December 2025

Published:

22 December 2025

Keywords: mine, coal industry, data mining, intelligent data analysis, intelligent language models, data processing by deep and machine learning models.

Abstract.

Mining companies are production complexes that develop deposits and/or process minerals. As in any other production enterprise, work in a mining complex is associated with a number of risks and hazards that are the main cause of accidents. In order to ensure normal working conditions, each company has an industrial safety officer who ensures compliance with the safety regulations of the production process. Although the main task of these specialists is to prevent industrial accidents, they are also responsible for process safety in an emergency situation. With the advent of digital transformation, many process and industrial safety tasks have been automated (including intelligent systems) to enable more accurate preventive prediction of industrial accidents. However, in addition to preventing accidents and incidents, there is also the task of assigning an accident severity class in situations where an accident has already occurred. The lesser popularity of this topic (compared to preventive prediction of mining equipment accidents) is no reason not to automate this process. Since the severity class of an accident is assigned by a human (based on a certain set of parameters describing the accident that has occurred), this means that there is still a possibility of human error, which can only be compensated by a software model. The gaps in this area of knowledge should undoubtedly be filled by new studies and models that would allow the severity class of an accident to be determined without human intervention. The above material determines the relevance of the proposed research topic based on the automation of the process of determining the severity class of an accident in mining enterprises. The subject of the study is the process of determining the severity class of an accident using a known set of precedents describing the accident. The subject of the study is the development of a deep learning model for the analysis of a descriptive set of precedents.

For citation: Maitak R.V., Pylov P.A., Nikolaeva E.A., Diaghileva A.V. Conceptualizing the architecture of a machine learning model for classifying occupational injuries in mining enterprises according to their severity. *Vestnik Kuzbasskogo gosudarstvennogo tekhnicheskogo universiteta*=Bulletin of the Kuzbass State Technical University. 2025; 6(172):145-156. (In Russ., abstract in Eng.). DOI: 10.26730/1999-4125-2025-6-145-156, EDN: TRXYOA

REFERENCES

1. Kiselev A. Promishlennaya bezopasnost opasnikh proizvodstvennikh obektov M.: Alfa-Press; 2017. 240 P.
2. Serdyuk D., Braga O., Siohan O. Audio-visual speech recognition is worth 32x32x8 voxels. *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. 2021.
3. Sung Y.-L., Cho J., Bansal. M. Vi-adapter: Parameter-efficient transfer learning for vision and language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022. Pp. 5227-5237.
4. Srinivasan T., Sanabria R., Metze F. Looking enhances listening: Recovering missing speech using images. *ICASSP*. 2020.
5. Synnaeve G., Xu Q., Kahn J., Likhomanenko T., Grave E., Pratap V., Sriram A., Liptchinsky V., Collobert R. End-to-end asr: from supervised to semi-supervised learning with modern architectures. arXiv preprint arXiv:1911.08460, 2019.
6. Bishop C.M. Pattern Recognition and Machine Learning. *Springer*. 2006. 738 s.
7. Soltis M. Vvedenie v analiz algoritmov SPb.: DMK; 2019. 269 s.

8. Domingos P. Verkhovniy algoritm. M.: «MANN, IVANOV I FERBER»; 2016. 315 s.
9. Gift N. Programmnii II. SPb.: Piter; 2019. 304 s.
10. Kormen T., Leizerson Ch. Algoritmi: postroenie i analiz, 3-e izdanie M.: OOO I.D. Vilyams; 2013. 1328 s.
11. Xu Q., Baevski A., Likhomanenko T., Tomasello P., Conneau A., Collobert R., Synnaeve G., Auli M. Self-training and pre-training are complementary for speech recognition. In *ICASSP*. 2021.
12. Sanabria R., Caglayan O., Palaskar Sh., Elliott D., Barrault L., Specia L., Metze F. How2: a large-scale dataset for multimodal language understanding. In *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*. 2018.
13. Pilov P.A., Maitak R.V., Zaitseva Ye.G. Primenenie multimodalnogo transformera dlya prognozirovaniya vikhodnykh parametrov nasishchennykh uglevodorodnykh soedinenii iz sostava tyazheloi nefi v prisutstvii katalizatorov. *Trudi Instituta sistemnogo programmirovaniya RAN*. 2023; 35(5):229-244. DOI: 10.15514/ISPRAS-2023-35(5)-15. EDN NLWIYD.
14. Rodzhers D. Algoritmicheskie osnovi mashinnoi grafiki. M.: MIR; 2022. 687 c.
15. Bosov A.Ye. Mashinnoe obuchenie i diskretnaya matematika: kak ispolzovanie kombinatoriki

i teorii grafov uluchshaet proizvoditelnost algoritmov mashinnogo obucheniya. V *sbornike: Nauka i molodyozh: novie idei i resheniya. Materiali XVII Mezhdunarodnoi nauchno-prakticheskoi konferentsii molodikh issledovatelei*. Volgograd, 2023. S. 193-194.

16. Mikhailov A.M., Rabeeakh S.K. Trading bots that use machine learning. V *sbornike: Nauka preobrazhaet realnost. Materiali Mezhdunarodnoi mezhdistsiplinarnoi nauchno-prakticheskoi studencheskoi konferentsii*. Otv. redaktor Ye.A. Maklakova. Voronezh, 2023. S. 13-19.

17. Shapoval I.V., Ruzakov V.O., Karimova L.K., Muldasheva N.A., Beigul N.A., Ilina L.A., Volgareva A.D. Proizvodstvennii travmatizm na gornodobivayushchikh predpriyatiyakh respubliky Bashkortostan v 2017-2022 godakh. *Bezopasnost i okhrana truda*. 2024; 1(98):8-12.

18. Krupnenchenkova S.A. Proizvodstvennii travmatizm na predpriyatiyakh gornodobivayushchei pro-mishlennosti. V *sbornike: Nauka i praktika: aktualnie voprosi, dostizheniya i innovatsii. sbornik statei VI Mezhdunarodnoi nauchno-prakticheskoi konferentsii*. Penza, 2025. S. 25-27.

19. Karnachev I.P., Cheltibashev A.A., Sudak S.N. Prakticheskaya realizatsiya kontseptsii kulturni promishlennoi bezopasnosti na primere krupneishikh gornikh predpriyatii. *Bezopasnost tekhnogennikh i prirodnikh sistem*. 2022; 3:24-31.

© 2025 The Authors. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

The authors declare no conflict of interest.

About the authors:

Roman V. Maitak, Master's degree, T.F. Gorbachev Kuzbass State Technical University, 28 street Vesennyyaya, Kemerovo, 650000, Russian Federation, e-mail: maytak.roman@mail.ru

Pyotr A. Pylov, PhD student, T.F. Gorbachev Kuzbass State Technical University, 28 street Vesennyyaya, Kemerovo, 650000, Russian Federation, e-mail: gedrosten@mail.ru

Evgenia A. Nikolaeva, Associate Professor, T.F. Gorbachev Kuzbass State Technical University, 28 street Vesennyyaya, Kemerovo, 650000, Russian Federation, e-mail: nikolaevaea@yandex.ru

Anna V. Diaghileva, Candidate of Technical Sciences, Associate Professor of the Department of Mathematics, T.F. Gorbachev Kuzbass State Technical University, 28 street Vesennyyaya, Kemerovo, 650000, Russian Federation, e-mail: dyagileva1952@mail.ru

Contribution of the authors:

Roman V. Maitak – data collection and analysis, neural network.

Pyotr A. Pylov – conceptualization of research, software implementation, scientific management, conclusions.

Evgeniya A. Nikolaeva – conclusions, text design.

Anna V. Diaghileva – problem statement, literature review.

All authors have read and approved the final manuscript.

