ТЕХНИЧЕСКИЕ НАУКИ

УДК 004.82

Интеллектуальные методы выявления знаний из баз данных

В.С. Чуб
Донской государственный технический университет (г. Ростов-на-Дону, Российская Федерация)

Аннотация. Выявление знаний в базах данных (БД) представляет собой процесс обнаружения различных закономерностей в больших наборах и хранилищах, для которого был разработан набор моделей интеллектуального анализа, охватывающих классификацию, регрессию, кластеризацию и т. д. Эти модели полезно использовать для решения прикладных задач. В частности, интеллектуальные транспортные системы (ИТС) являются важнейшим компонентом инфраструктуры Умного города. Используя большие данные, а также регрессионные модели, лежащие в основе прогностических модулей, ИТС могут обеспечить анализ дорожной инфраструктуры в режиме реального времени и более эффективное управление дорожным движением, полагаясь на прогнозы трафика как на критический компонент. Применение современных методов выявления знаний в БД позволит повысить их интерпретируемость, решение этой задачи рассматривается на примере анализа транспортного потока. Цель данной статьи — провести сравнительный анализ подходов к моделированию подсистем интеллектуальных транспортных систем, обеспечивающих прогнозирование заторов на дорогах средствами рекуррентных нейронных сетей.

Ключевые слова: база данных, база знаний, рекуррентные нейронные сети, глубокое обучение, транспортный поток

Intelligent methods for discovering knowledge from databases

Vadim S. Chub
Don State Technical University (Rostov-on-Don, Russian Federation)

Abstract. Knowledge discovery in databases is the process of detecting various patterns in large datasets and data warehouses, for which a set of mining models covering classification, regression, clustering, etc. has been developed. These models are useful to use for solving applied problems. In particular, intelligent transport systems (ITS) are an essential component of the smart city infrastructure. Using big data, as well as regression models underlying predictive modules, ITS can provide real-time analysis of road infrastructure and more efficient traffic management, relying on traffic forecasts as a critical component. The use of modern methods for identifying knowledge in the database will increase their interpretability. The solution of this problem is considered by the example of traffic flow analysis. The article objective is to conduct a comparative analysis of approaches to modeling subsystems of intelligent transport systems that provide traffic congestion forecasting by means of recurrent neural networks.

Keywords: database, knowledge base, recurrent neural networks, deep learning, traffic flow

Введение. Процесс выявления знаний, который иногда называют обнаружением знаний в базах данных, представляет собой процедуру извлечения полезной информации из более крупной базы или набора данных. Это популярный метод сбора информации из различных источников и уточнение информации для целевого использования в приложении. Чистые данные имеют мало практической пользы, если их нельзя сортировать, наносить на графики и т. д., вследствие чего невозможно выявить скрытые закономерности и различные взаимосвязи. Это обстоятельство потребовало разработки и внедрения в процесс обнаружения знаний интеллектуальных программных инструментов, таких как машинное обучение и визуализация данных, чтобы позволить находить скрытые закономерности и делать прогнозы на основе предыдущих знаний и данных. В современном мире, основанном на аналитике, данных никогда не бывает слишком много. И исследовательские группы в области искусственного интеллекта (ИИ) добились больших успехов в улучшении методов интеллектуального анализа данных, используемых в процессе обнаружения знаний [1–4].

В рамках данной работы для выявления знаний рассматривается задача предсказания интенсивности транспортного потока. Целью такого прогнозирования является предсказание будущих условий трафика в транспортной сети на основе ретроспективных наблюдений. Эти данные могут быть полезны в приложениях интеллектуальных транспортных систем, таких как контроль заторов на дорогах и управление светофорами.

В последнее время дорожное движение генерирует большие объемы данных из разнородных источников: датчики на улицах, в транспорте, автоматические системы сбора платы за проезд и т. д. Эти цифровые данные, генерируемые ИТС, используются для более эффективного управления дорожным движением. Для прогнозирования транспортных потоков используются нейронные сети, байесовский подход, статистическое моделирование и гибридные модели [5, 6]. Эти подходы обнаруживают скрытую информацию в данных о трафиках и прогнозируют их предстоящий поток. Динамический характер шаблонов позволяет применять концепцию глубокого обучения. Глубокое обучение — это разновидность машинного обучения, методы которого применялись во многих задачах, таких как классификация, прогнозирование, кластеризация, распознавание образов и т. д. Концепция глубокого обучения заключается в использовании глубоких архитектур, то есть нескольких скрытых слоев, для извлечения неизвестных закономерностей в заданных данных. Алгоритмы глубокого обучения можно использовать для прогнозирования транспортных потоков, которые представляют собой сложные модели трафика, имеют динамический характер. Глубокие нейронные сети повышают точность и уменьшают ошибки предсказания.

Для реализации эксперимента автор использовал платформу Google Colab — это облачная интерактивная среда разработки, созданная для образовательных целей и непосредственных исследований в области машинного обучения. Написание кода осуществляется на языке Python в специальных файлах, так называемых ноутбуках, которые не требуют отдельной настройки и полностью работают в облаке. В базовом режиме Colab предоставляет каждому пользователю до 14 Гб оперативной памяти и около 78 Гб — на физическом носителе. Для работы с данными и применения выбранных методов машинного и глубокого обучения при помощи языка программирования Python используется несколько свободно доступных библиотек, в частности Numpy, Matplotlib, Scikit-Learn, Pandas и Keras.

Основная часть. Для проведения анализа подходов к моделированию подсистем интеллектуальных транспортных систем, обеспечивающих прогнозирование заторов на дорогах, был выбран набор данных из репозитория машинного обучения UCI [7]. Этот набор данных содержит информацию об объеме трафика между станциями метро. Различные атрибуты данных относятся к погоде, выходным дням, праздникам, населению и т. д. Необработанные данные были собраны с датчиков транспортных средств, следующих из Миннеаполиса в сторону Сент-Пола. В этом наборе около 48 000 экземпляров и девять атрибутов данных. Данные содержат записи за каждый день с 2012 по 2018 год. Большинство атрибутов в этом наборе данных являются действительными числами. С их помощью можно спрогнозировать почасовой объем трафика в западном направлении, который является непрерывной числовой характеристикой. Таким образом, решаемая задача относится к регрессии.

Модели глубокого обучения, такие как нейронные сети с долгой кратковременной памятью (Long Short-Term Memory, LSTM), продемонстрировали наилучшие результаты прогнозирования транспортного потока, в несколько раз превзойдя традиционные алгоритмы машинного обучения (такие как линейная регрессия, деревья решений и случайный лес) по величине ошибки — метрикам RMSE и MAE.

В исходном наборе данных 48204 строки и девять атрибутов (признаков). Атрибуты holiday, weather_main, weather_description и date_time являются категориальными, clouds_all и traffic_volume — числовыми, а temp, rain_1h и show_1h — вещественными. В наборе данных отсутствуют пропущенные значения. В представленных сведениях есть данные об 11 праздниках (рис. 1). Максимальный трафик наблюдается в новогодние дни.

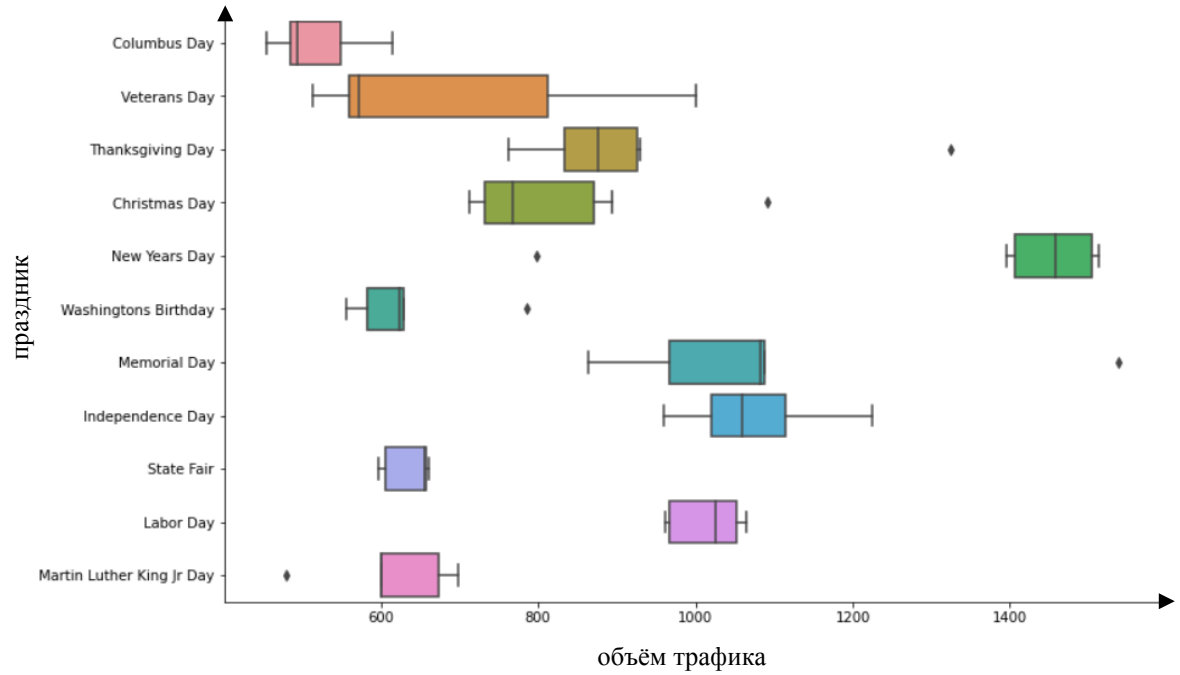

Рис. 1. Распределение данных по праздникам (сформировано на основе открытых данных с использованием Matplotlib)

Описание атрибутов в наборе данных приведено в таблице 1.

Таблица 1

Описание признаков в наборе данных [7]

Атрибут (признак)	Описание
holiday	Национальный праздник США и региональный праздник штата Миннесота; None, если нет
temp	Средняя температура в градусах Кельвина
rain_1h	Количество дождя в мм, выпавшего за час
snow_1h	Количество снега в мм, выпавшего за час
clouds_all	Процент облачного покрытия (от 0 до 100)
weather_main	Краткое текстовое описание текущей погоды
weather_description	Подробное текстовое описание текущей погоды
date_time	Местное время для собранных данных
traffic_volume	Объем трафика на кольцевой автомагистрали, количество транспортных средств в час

В наборе представлены данные об 11 кратких описаниях погоды: облачно, ясно, дождь, моросящий дождь, мгла, дымка, туман, гроза, снег, шквал, смог. Наиболее частыми краткими описаниями погоды являются Clouds (облачно) и Clear (ясно). В приведенных данных есть сведения о 38 подробных описаниях погоды (рис. 2). Наиболее частым подробным описанием погоды является sky is clear (небо ясное).

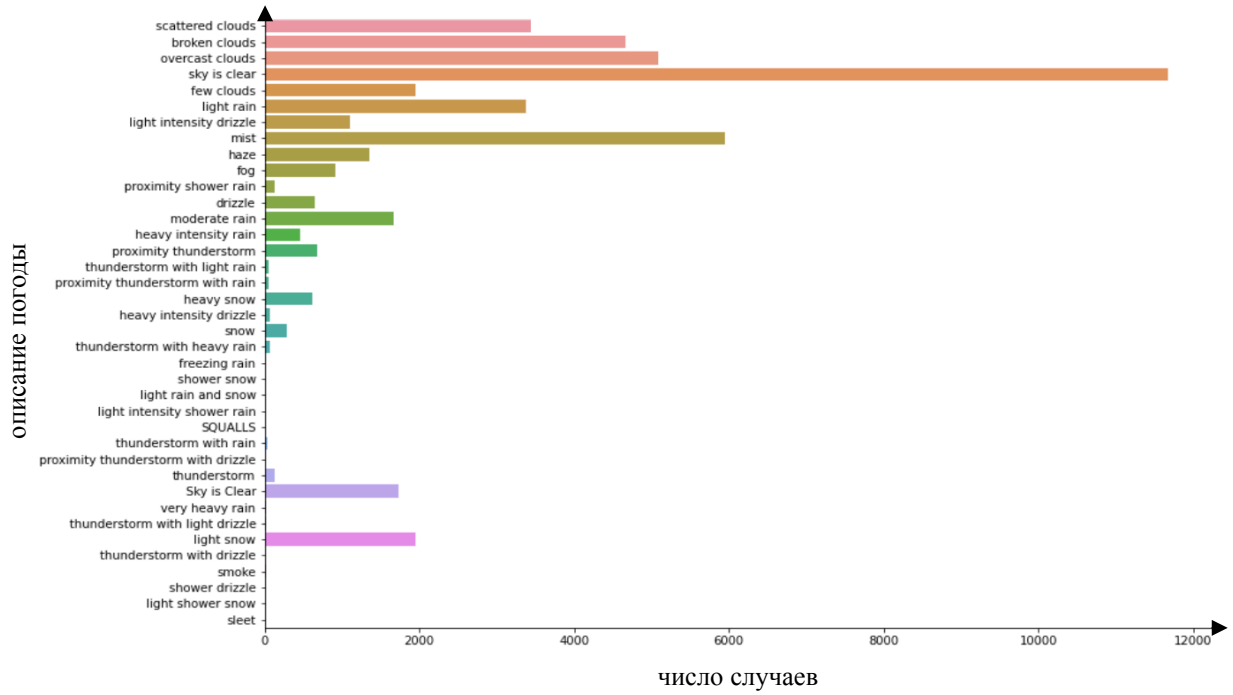

Рис. 2. Распределение данных по подробным описаниям погоды
(сформировано на основе открытых данных с использованием Matplotlib)

Тепловые карты позволят увидеть корреляции между числовыми признаками в наборе данных — как между парами атрибутов, так и по отношению к целевой переменной. Коэффициент корреляции варьируется от –1 до +1. Если значение близко к +1, это означает, что между двумя переменными существует сильная положительная корреляция. Когда оно близко к –1, переменные имеют сильную отрицательную корреляцию. На рис. 3 приведена матрица корреляции для набора данных.

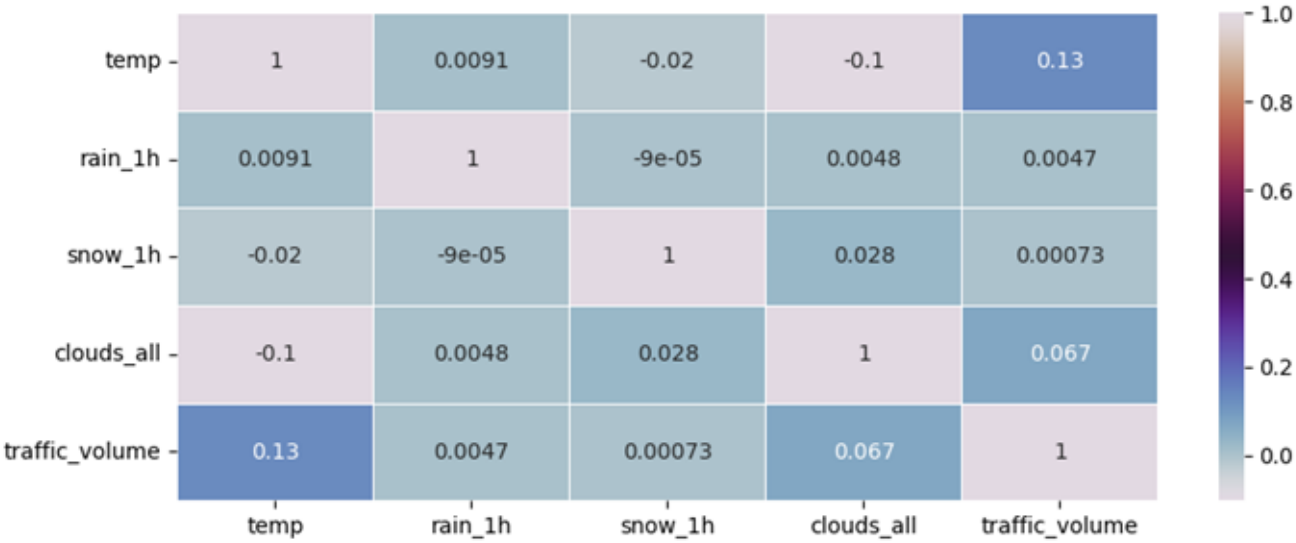

Рис. 3. Матрица коэффициентов корреляции числовых данных в наборе
(сформировано на основе открытых данных с использованием Matplotlib)

По матрице коэффициентов корреляции можно сказать, что все столбцы положительно коррелируют с целевой переменной traffic_volume. Но взаимосвязи как между парами атрибутов, так и между любым из атрибутов и целевой переменной являются незначительными.

Гистограммы распределения числовых данных по каждому признаку позволяют увидеть распределение данных в наборе (рис. 4).

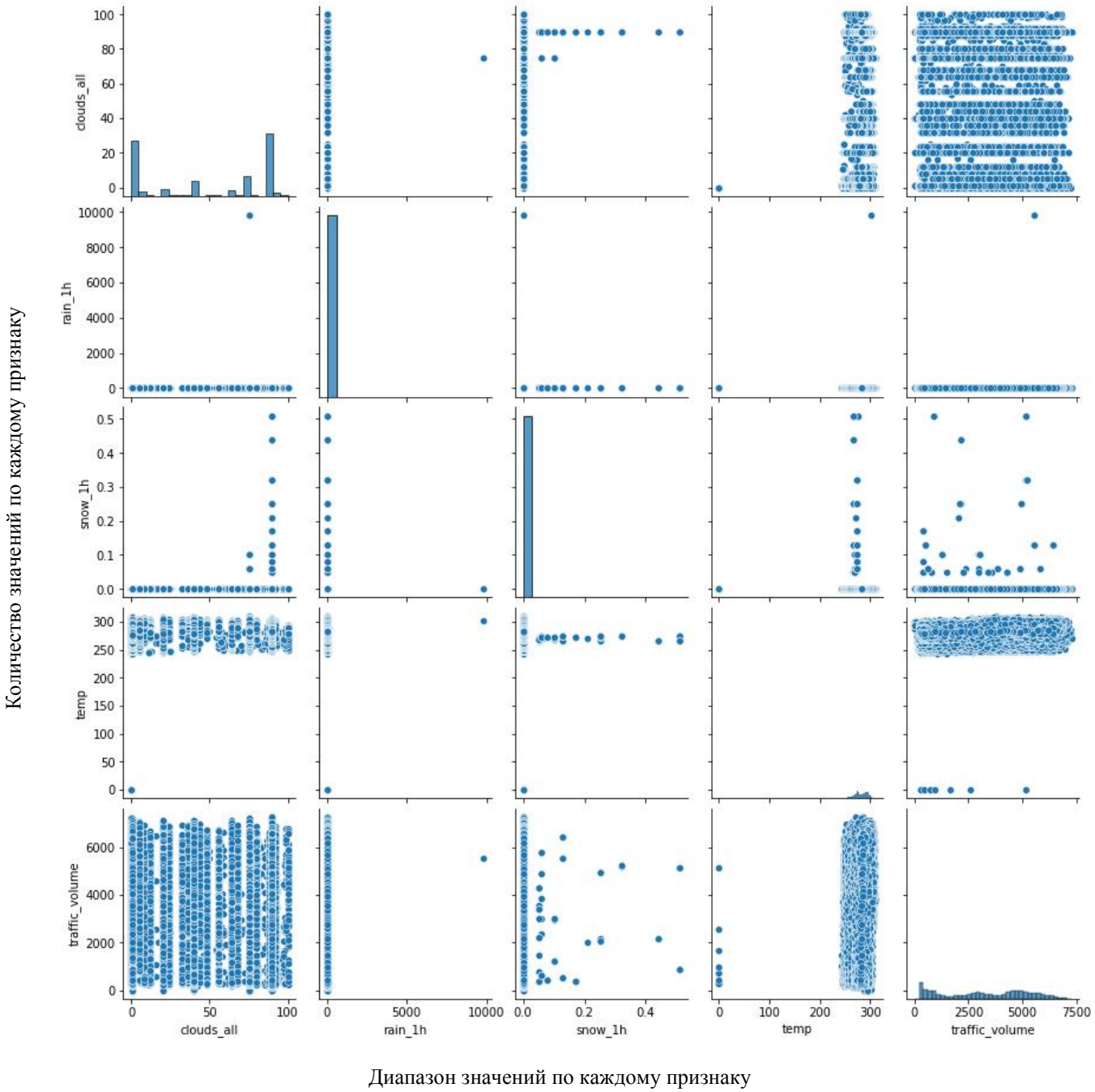

Рис. 3. Гистограммы распределения числовых значений признаков в наборе данных (сформировано на основе открытых данных с использованием Matplotlib)

Гистограммы позволяют выявить следующие данные:

– в cloud_all больше значений в диапазоне 90–100 и 0–10;

– атрибуты rain_1h и snow_1h имеют низкие десятичные значения и находятся в диапазоне 0–5 и 0,0–0,05 соответственно;

– в столбце temp минимальное значение составляет около 220, а максимальное — 310, и этот атрибут имеет хорошее распределение в диапазоне 250–310;

– в столбце traffic_volume данные нормально распределены по всему диапазону.

Эксперимент состоит в том, что данные из используемого набора загружаются из csv-файла для их дальнейшей обработки и анализа. На этапе подготовки используются методы предварительной обработки (в том числе удаление выбросов), а затем выполняется обучение и тестирование различных моделей глубокого обучения. Глубокие LSTM-сети обучаются на временных рядах, извлекаемых из набора данных. После оценки

полученных результатов лучшая модель (с наименьшей величиной ошибки) может быть использована в веб-приложении для предсказания объема трафика на ранее неизученном экземпляре данных.

Простейшая LSTM называется базовой, или ванильной (Vanilla LSTM). Эта сеть состоит также из выходного слоя для получения отклика модели и так называемого слоя дропаута. Слои дропаута используются для снижения переобучения. Коэффициент соответствует долям связей между слоями, которые выродятся в ноль и не будут меняться в процессе обучения. В данной работе входной слой ванильной LSTM состоит из 64 нейронов, а коэффициент дропаута равен 0,2. Схема модели приведена на рис. 5. На входном слое 24 нейрона (24 признака в наборе данных).

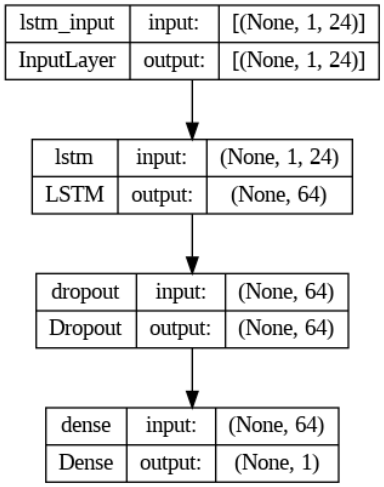

Рис. 4. Схема модели ванильной LSTM-сети (сформировано на основе открытых данных с использованием Tensorflow)

Для всех моделей LSTM были созданы признаки, включающие временные ряды из входного набора данных. Временной ряд — это последовательность чисел, упорядоченных по временному индексу, которую можно представить как список или столбец упорядоченных значений. Для создания нового набора данных по временным рядам была написана специальная функция, которая принимает четыре аргумента: последовательность наблюдений в виде списка или двумерного массива, количество «запаздывающих» (более ранних) наблюдений в качестве входных данных (по умолчанию 1), количество наблюдений в качестве выходных данных (по умолчанию 1), а также логическое значение, следует ли удалять строки с отсутствующими значениями (по умолчанию Истина) [8]. Функция возвращает одно значение — набор признаков, который может быть использован для решения задачи регрессии при помощи LSTM-моделей.

Новый набор данных строится как таблица, где каждый столбец соответствующим образом назван как по номеру переменной, так и по временному шагу. Это позволяет создавать множество различных задач прогнозирования типа последовательности временных шагов из заданного одномерного или многомерного временного ряда. Стандартной практикой прогнозирования временных рядов является использование «запаздывающих» наблюдений (например t–1) в качестве входных переменных для прогнозирования текущего временного шага (t). Этот тип называется одношаговым прогнозированием. Другой тип использует прошлые наблюдения для прогнозирования последовательности будущих наблюдений. Это можно назвать прогнозированием последовательности, или многоэтапным прогнозированием. Еще один важный тип временных рядов называется многомерными временными рядами. Здесь могут подаваться на вход наблюдения за несколькими различными показателями, и интерес состоит в прогнозировании одного или нескольких из них. К созданным признакам применен метод предварительной обработки, а именно нормализация значений в диапазоне от 0 до 1.

В обучающей выборке для нейронных сетей 4438 строк и 25 атрибутов (включая целевую переменную), в тестовой — 968 строк и 25 атрибутов (включая целевую переменную). Все созданные в этом подразделе модели обучались в течение 100 эпох с возможностью ранней остановки (если величина ошибки не уменьшается в течение 10 последовательных эпох) и размером мини-выборки в 64 экземпляра. В качестве метрики на этапе обучения используется среднеквадратическая ошибка (Mean Squared Error, MSE), которая описана формулой (1). Здесь N — число наблюдений. Для вычисления функции используются истинное значение целевой переменной y_i и отклик модели $\tilde{y}_i$ для некоторого входного вектора x_i.

$$L = \frac{1}{N}\sum_i^N (y_i - \tilde{y}_i)^2. \quad (1)$$

В данной работе в качестве основы для ансамблевой (сложенной) модели использовалась сеть LSTM. Голосование не применялось. Ансамбль состоит из двух LSTM-сетей с 64 нейронами каждая и с коэффициентами дропаута 0,2 и 0,3 соответственно [9]. На рис. 6 проиллюстрирована схема этой модели.

Нейронная сеть с двунаправленной долгосрочной памятью (Bi-LSTM) — это модель, которая получает информацию о последовательности в обоих направлениях: назад, из конца в начало или вперед, из начала в конец. Двунаправленный режим отличает Bi-LSTM от обычной LSTM-сети. Такой подход позволяет сети лучше обучаться на каждом временном экземпляре из набора данных, что приводит к лучшему изучению признаков. Модель Bi-LSTM состоит из 64 нейронов и имеет коэффициент дропаута 0,2 [1]. Схема этой модели приведена на рис. 7.

Сравнение всех моделей глубокого обучения выполнялось по величине ошибок RMSE — квадратный корень из MSE, описываемой формулой (1) — и MAE (Mean Absolute Error), которая описана формулой (2). Здесь N — число наблюдений. Для вычисления функции используются истинное значение целевой переменной y_i и отклик модели $\tilde{y}_i$ для некоторого входного вектора x_i.

$$L = \frac{1}{N}\sum_i^N |y_i - \tilde{y}_i| \quad (2)$$

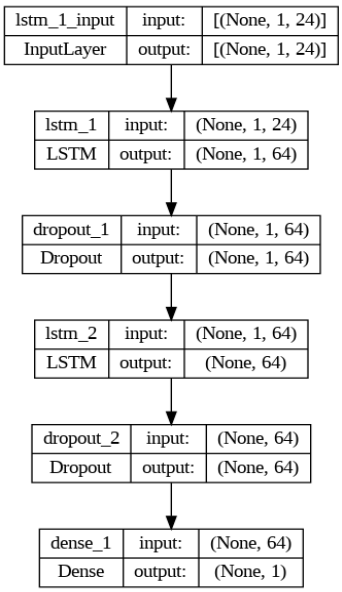

Рис. 5. Схема модели сложенной LSTM-сети
(сформировано на основе открытых данных с использованием Tensorflow)

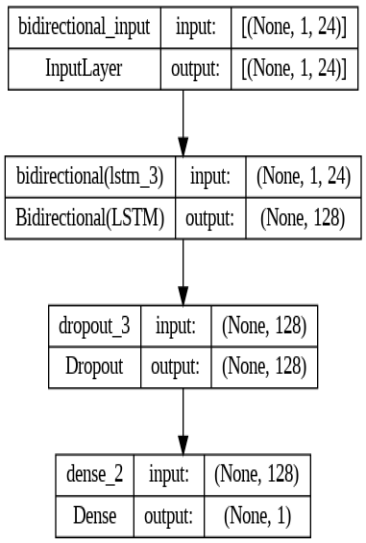

Рис. 6. Схема модели двунаправленной LSTM-сети
(сформировано на основе открытых данных с использованием Tensorflow)

Результаты сравнения моделей приведены в таблице 2.

Таблица 2

Сравнительный анализ LSTM-сетей

Модель глубокого обучения	MAE	RMSE
Ванильная LSTM	365,05	515,92
Сложенная LSTM	357,1	507,51
Двунаправленная LSTM	373,08	515,71

Таким образом, модель сложенной LSTM-сети продемонстрировала лучшую производительность.

Заключение. Глубокое обучение позволяет динамически извлекать признаки из данных, в отличие от традиционных алгоритмов машинного обучения, где отбор признаков по-прежнему остается проблемой. Глубокие нейронные сети имеют большое количество приложений в области взаимодействия с базами данных, которые используются для хранения и извлечения, что является важным компонентом любого программного приложения. Глубокие нейронные сети требуют больших объемов данных для обучения, поэтому современные архитектуры глубоких (многослойных) искусственных сетей все чаще используются в задачах обработки больших данных, в том числе при извлечении признаков из баз данных больших объемов.

В работе был описан набор данных, проведена предварительная обработка, а также осуществлен сравнительный анализ некоторых методов глубокого обучения для выявления знаний (LSTM-сетей). Результаты показали, что более современные алгоритмы хорошо работают в области предсказания транспортного трафика на выбранном наборе данных, и по этой причине могут быть использованы в качестве полноценных подсистем ИТС как средство прогнозирования заторов на дорогах, необходимых для моделирования потока. Модель сложенной LSTM-сети продемонстрировала лучшую производительность по метрикам MAE и RMSE. Тем не менее, полученные результаты могут быть улучшены, например, за счет использования более сложных нейросетевых моделей, основанных на современных алгоритмах глубокого обучения.

Список литературы

1. Bin Y. Application of AI technology in big data network security defense. *Computer products and circulation*. 2019;03:142.

2. Ning W. Application of big data and artificial intelligence technology in computer network. *Electronic technology and software engineering*. 2019;08:12.

3. Ting G. Application of artificial intelligence in computer network technology in the age of big data. *Electronic technology and software engineering*. 2019;01:6.

4. Xin Z. Application of AI in computer network technology under the background of big data era. *Information and computer (theoretical version)*. 2019;08:113–114.

5. Yaguang Li, Cyrus Shahabi. A brief overview of machine learning methods for short-term traffic forecasting and future directions. *SIGSPATIAL Special.* 2018;10(1):3–9. https://doi.org/10.1145/3231541.3231544

6. Haitao Yuan, Guoliang Li. A Survey of Traffic Prediction: From Spatio-Temporal Data to Intelligent Transportation. *Data Science and Engineering.* 2021;6:63–85. https://doi.org/10.1007/s41019-020-00151-z

7. *Metro Interstate Traffic Volume Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume (дата обращения: 03.07.2023).

8. Rabby F., Yazhou Tu, Hossen I., Lee Insup, Maida AS, Xiali Hei. Stacked LSTM based deep recurrent neural network with kalman smoothing for blood glucose prediction. *BMC Medical Informatics and Decision Making*. 2021;21:101. https://doi.org/10.1186/s12911-021-01462-5

9. Власов А.А. *Теория транспортных потоков*. Пенза: ПГУАС, 2014. 124 с.

Об авторе:

Чуб Вадим Сергеевич, аспирант кафедры «Вычислительные системы и информационная безопасность» Донского государственного технического университета (РФ, 344003, г. Ростов-на-Дону, пл. Гагарина, 1), vadim-chub13@mail.ru

About the Author:

Vadim S. Chub, postgraduate student of the Computing Systems and Information Security Department, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), vadim-chub13@mail.ru