

(19)



Евразийское
патентное
ведомство

(11) 046819

(13) B1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2024.04.25

(21) Номер заявки
202393582

(22) Дата подачи заявки
2023.10.23

(51) Int. Cl. G06Q 20/38 (2012.01)
G06N 3/08 (2023.01)

(54) СПОСОБ ВЫЯВЛЕНИЯ МОШЕННИЧЕСКИХ ТРАНЗАКЦИЙ НА ОСНОВЕ АНАЛИЗА ТАБЛИЧНЫХ ДАННЫХ

(43) 2024.04.24

(96) 2023000173 (RU) 2023.10.23

(71)(73) Заявитель и патентовладелец:
ОБЩЕСТВО С ОГРАНИЧЕННОЙ
ОТВЕТСТВЕННОСТЬЮ
"АСТОНЛАБ" (RU)

(72) Изобретатель:
Батыров Ричард Эрикович, Миронов
Валентин Алексеевич (RU)

(74) Представитель:
Котлов Д.В. (RU)

(56) IKRAM UI HAQ et al., "Categorical Features Transformation with Compact One-Hot Encoder for Fraud Detection in Distributed Environment", 16th Australasian Conference, AusDM 2018, Bahrurst, NSW, Australia, November 28-30, 2018 [онлайн] [найдено 14.02.2024]. Найдено в <https://www.researchgate.net/publication/331126668_Categorical_Features_Transformation_with_Compact_One_Hot_Encoder_for_Fraud_Detection_in_Distributed_Environment_16th_Australasian_Conference_AusDM_2018_Bahrurst_NSW_Australia_November_28-30_2018_Revis>
US-A1-20230013152
US-B2-11082438
US-B2-11308497
CN-A-11045857

(57) Изобретение относится к области анализа табличных данных, в частности к способу выявления мошеннических транзакций на основе анализа табличных данных.

046819 B1

046819 B1

046819 B1

Область техники

Настоящее изобретение относится к области анализа табличных данных, в частности к способам выявления мошеннических транзакций на основе анализа табличных данных.

Уровень техники

Из уровня техники известен источник информации, CN 114298712 (A), опубликованный 08.04.2022, раскрывающий способ и применение обнаружения аномальных транзакций криптовалюты. Способ включает в себя этапы выборки данных нормальной транзакции и данных аномальной транзакции криптовалюты на основе цепочки блоков, сериализации данных транзакции одного и того же пользователя в соответствии со временем транзакции, создания образца последовательности транзакций пользователя и маркировки метки нормальной транзакции или аномальной транзакции; формирования метки для образца последовательности транзакций пользователя; образец последовательности пользовательских транзакций кодируется в вектор последовательности пользовательских транзакций, вектор последовательности пользовательских транзакций вводится в модель обнаружения аномальных криптовалютных транзакций для обучения, и модель обнаружения аномальных криптовалютных транзакций реализуется на основе сетевой архитектуры преобразования.

Отличием предлагаемого решения от известных из уровня техники является использование мягкого подхода для представления табличных данных. Дополнительными отличиями является использование подходов softonehot, а также soft one-hot encoding.

Сущность изобретения

Технической задачей, на решение которой направлено заявленное техническое решение, является создание способа выявления мошеннических транзакций на основе анализа табличных данных.

Технический результат заявленного технического решения совпадает с назначением и дополнительно заключается в улучшение качества классификации табличных данных.

Заявленный технический результат достигается за счет осуществления способа выявления мошеннических транзакций на основе анализа табличных данных, включающего этапы, на которых:

входные данные разделяются на категориальные данные и непрерывные данные, причем непрерывные данные представляются действительными числами, а категориальные данные натуральными числами и могут принимать ограниченный набор значений;

категориальные данные кодируются посредством метода быстрого кодирования one-hot encoding, при этом непрерывные данные кодируются методом softonehot;

закодированные данные трансформируются в векторные представления, посредством метода обучения матрицы весов; далее общие векторные представления данных используются как входные данные для модели "трансформер" с глубокой нейронной сетью;

выходные данные из глубокой нейронной сети представляются в виде вектора длины I , после чего вектор скалярно умножается на вектор параметров длины I , причем получаемое значение представляет собой уверенность модели в том, что транзакция мошенническая, причем чем больше полученное значение, тем выше уверенность.

В частном варианте реализации предлагаемого способа, входные данные, которые были разделены на непрерывные данные, - это сведения о транзакции, а именно сумма, дата/время, место осуществления транзакции, при этом входные данные, которые были разделены на категориальные, - это данные пользователя, совершающего транзакцию, а именно пол, возраст и история предыдущих транзакций.

Описание чертежей

Реализация изобретения будет описана в дальнейшем в соответствии с прилагаемыми чертежами, которые представлены для пояснения сути изобретения и никоим образом не ограничивают область изобретения.

К заявке прилагаются следующие чертежи:

фиг. 1 иллюстрирует тяжесть хвостов распределения значений в мягком индикаторном векторе при различных значениях t ;

фиг. 2 иллюстрирует распределение мягких индикаторных векторов при обычной инициализации;

фиг. 3 иллюстрирует плотности распределений x и θ при классической инициализации;

фиг. 4 иллюстрирует общую блок-схему заявленного способа;

фиг. 5 иллюстрирует результаты экспериментов;

фиг. 6 иллюстрирует структуру векторного пространства до обучения модели;

фиг. 7 иллюстрирует структуру векторного пространства после обучения модели.

Детальное описание изобретения

В приведенном ниже подробном описании реализации изобретения приведены многочисленные детали реализации, призванные обеспечить отчетливое понимание настоящего изобретения. Однако квалифицированному в предметной области специалисту будет очевидно, каким образом можно использовать настоящее изобретение как с данными деталями реализации, так и без них. В других случаях хорошо известные методы, процедуры и компоненты не были описаны подробно, чтобы не затруднять понимание особенностей настоящего изобретения.

Кроме того, из приведенного изложения будет ясно, что изобретение не ограничивается приведен-

ной реализацией. Многочисленные возможные модификации, изменения, вариации и замены, сохраняющие суть и форму настоящего изобретения, будут очевидными для квалифицированных в предметной области специалистов.

В настоящем техническом решении описывается метод построения векторных представлений непрерывных предикторных переменных, основанный на имитации поведения деревьев решений в парадигме глубокого обучения. Способ позволяет разбить непрерывную переменную на группы во время обучения, при этом, не задавая их до начала процесса обучения, и получает эти группы неявно, то есть не путем трансформации самой переменной.

Изобретение предназначено для решения задач анализа табличных данных, таких как регрессия или классификация. Предпочтительным применением является задача выявления мошеннических транзакций.

Задача выявления мошеннических транзакций - это задача бинарной классификации, в которой по полученным параметрами транзакции, необходимо причислить ее к одному из двух классов (мошенническая или нет).

Однако, модель может обучаться не только на историях транзакций, а на любых табличных данных.

Часть используемой в изобретении архитектуры "Трансформер", была предложена в <https://arxiv.org/abs/1706.03762> для обработки естественного языка, однако применения данной архитектуры ограничено дискретной природой данных естественного языка, которые представляют в виде "токенов". Под дискретными данными понимаются переменные, используемые в анализе, которые могут принимать ограниченное число значений. Например, слово может быть представлено как набор букв и каждая буква принимает одно из 33 значений. Далее эти значения обрабатываются путем построения индикаторных векторных представлений (one-hot encoding), которые представляют из себя вектор, состоящий из нулей, кроме одной позиции соответствующей, например номеру буквы в алфавите. Так буква "б" будет представлена как [0, 1, 0, ..., 0]. Далее используя построенные векторы каждой переменной ставится в соответствие непрерывное векторное представление из R^n , которое уточняется в процессе обучения. Применения данного подхода для построения векторных представлений непрерывных переменных, таких, как, например, объем транзакции, не является эффективным, так как количество различных значений, которое может принимать непрерывная переменная в доступной для обучения выборке может иметь порядок размера выборки, что приведет к слишком большому количеству параметров в модели и проблеме переобучения. Также возможно появления новых значений при применении модели к данным вне тренировочной выборки.

Для решения проблемы применения архитектуры "Трансформер" к непрерывным данным авторами изобретения предложен подход мягкого индикатора, который позволяет построить векторные представления непрерывных переменных, имитируя подход, используемый деревьями решений в парадигме глубокого обучения.

Подход мягкого индикатора.

Пусть имеется параметр $x \in R$, построим вектор $\theta \in R^k$, $k \in N$, где k - гиперпараметр, определяющий набор подмножеств на которые разбивается переменная x . Обозначим вектором $S \in R^k$, итоговый мягкий вектор индикатор (softonehot), тогда:

$$S_j = \text{softonehot}(x, \theta, j) = \frac{e^{-t|x-\theta_j|}}{\sum_{q=1}^k e^{-t|x-\theta_q|}, t \in R.}$$

Здесь t - гиперпараметр, являющийся коэффициентом температуры, позволяющий контролировать тяжесть хвостов распределения значений в мягком индикаторном векторе (фиг. 1).

Применив функцию softonehot для всех j от 1 до k получим векторную формулировку softonehot:

$$S = \text{softonehot}(x, \theta) = \{\text{softonehot}(x, \theta, 1), \dots, \text{softonehot}(x, \theta, k)\}.$$

Далее для получения финальных векторных представлений $h \in R^v$, $v \in N$, где v - размер векторного представления, вектор S , который можно представить в виде матрицы из $R^{1 \times k}$, умножается на матрицу весов $A \in R^{k \times v}$. То есть

$$h = IA = \text{softonehot}(x, \theta)A.$$

Важно отметить, что операция `softonehot` является дифференцируемой:

$$S_i = \frac{e^{-t|x-\theta_i|}}{\sum_{q=1}^k e^{-t|x-\theta_q|}},$$

$$\frac{\partial}{\partial \theta_j} \log(s_i) = \frac{1}{s_i} \frac{\partial s_i}{\partial \theta_j},$$

$$\frac{\partial s_i}{\partial \theta_j} = s_i \frac{\partial}{\partial \theta_j} \log(s_i),$$

$$\log(s_i) = e^{-t|x-\theta_i|} - \log\left(\sum_{q=1}^k e^{-t|x-\theta_q|}\right),$$

$$\frac{\partial}{\partial \theta_j} \log(s_i) = \frac{\partial}{\partial \theta_j} e^{-t|x-\theta_i|} - \frac{\partial}{\partial \theta_j} \log\left(\sum_{q=1}^k e^{-t|x-\theta_q|}\right),$$

$$\frac{\partial}{\partial \theta_j} e^{-t|x-\theta_i|} = \frac{1_{\{i=j\}} e^{-t|x-\theta_i|} t(x-\theta_i)}{|x-\theta_i|}, \text{ здесь } 1 - \text{ индикаторная функция,}$$

$$\frac{\partial}{\partial \theta_j} \log\left(\sum_{q=1}^k e^{-t|x-\theta_q|}\right) = \frac{s_j t(x-\theta_j)}{|x-\theta_j|},$$

$$\frac{\partial s_i}{\partial \theta_j} = s_i \left(\frac{1_{\{i=j\}} e^{-t|x-\theta_i|} t(x-\theta_i)}{|x-\theta_i|} - \frac{s_j t(x-\theta_j)}{|x-\theta_j|} \right).$$

Что позволяет применять алгоритмы оптимизации к параметру θ и уточнять его в процессе обучения.

Входные данные о пользователях хранятся в отдельной базе данных пользователей. Данные о транзакции принимаются в момент совершения транзакции и представляют собой ее различные параметры (например, время, отправитель, получатель, объем и т.д.).

Таким образом, предлагаемый подход имеет следующий общий вид.

Для входных данных, имеющих вид $Q=(x, y)$, где y - метка, для задачи поиска мошеннических транзакций принимающее значения из $\{0, 1\}$, где 1 означает что транзакция является мошеннической и 0 означает что транзакция не является мошеннической, x является набором параметров транзакции, состоящим из M категориальных параметров и N непрерывных параметров. Представим x в виде:

$$x = \{x_1, \dots, x_M; x_1, \dots, x_N\}.$$

Рассмотрим процесс построения векторных представлений длины d для всех параметров транзакции. Для построения векторного представления категориального параметра x_i построим вектор индикатор E_i длины $|x_i|$, где $|x_i|$ - количество возможных значений, которые может принимать параметр x_i . Тогда для получения финального векторного представления параметра x_i построим матрицу параметров $H_i \in R^{|x_i| \times d}$ и получим финальное представление в виде: $h_i = E_i H_i$.

Для построения векторного представления непрерывного параметра x_j применим функцию `softonehot`, описанную ранее с параметром θ_j , здесь $\theta_j \in R^k$ и умножим результат на матрицу параметров $A_j \in R^{k \times d}$. Таким образом:

$$h_j = \text{softonehot}(x_j, \theta_j) A_j.$$

Конечное векторное представление h всех параметров x примет вид:

$$h = \{h_1, \dots, h_M, h_1, \dots, h_N\} \in R^{(N+M) \times d}.$$

Далее матрица параметров h используется как набор входных данных для глубокой нейронной сети, например для многослойного персептрона матрица параметров распрямляется в вектор из $\mathbb{R}^{(N+M)d}$, или в предпочтительном варианте реализации изобретения матрица параметров h передается без изменений в качестве входных данных для модели "Трансформер".

Важной частью воспроизведения эффективности изобретения является предобработка непрерывных параметров x_j и инициализации параметров θ_j . При инициализации параметров θ_j с использованием стандартного нормального распределения и отсутствии предобработки параметров x_j большинство мягких индикаторных векторов могут иметь распределение как на фиг. 2, что приведет к проблемам в процессе обучения. При стандартизации или мин-макс нормализации параметров x_j и инициализации параметров θ_j с использованием стандартного нормального распределения или равномерного распределения возникает проблема несовпадения плотностей распределений параметров x_j и параметров θ_j , что приводит к большой концентрации потенциальных разбиений в областях с низким числом параметров (фиг. 3). Это приводит к понижению репрезентативности *softonehot* слоя. Для преодоления этих проблем в предпочтительной реализации изобретения рекомендуется использовать квантильную нормализацию параметров x_j (<https://www.tandfonline.com/doi/abs/10.1080/17517575.2020.1856417>).

Полная схема изобретения с использованием архитектуры Трансформер в качестве глубокой нейронной сети представлена на фиг. 4.

Выходной слой сети может меняться в зависимости от задачи. Для задачи нахождения мошеннических транзакций, после представления выхода из глубокой нейронной сети в виде вектора длины l . Далее данный вектор скалярно умножается на вектор параметров длины l , получаемое значение представляет собой уверенность модели в том, что транзакция мошенническая, чем больше значение, тем больше уверенность. Для представления данного значения в виде вероятности используется сигмоидальная функция

$f(x) = \frac{1}{1+e^{-x}}$. Для применения данной модели для задач регрессии сигмоидальная функция не используется и выход модели является предсказанием непрерывного значения. Для задач классификации с количеством классов большим 2 вектор параметров заменяется на матрицу из $\mathbb{R}^{l \times C}$, где C - количество классов, далее для представления результатов умножения в виде вероятностей применяется функция

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$

. Как было отмечено ранее предлагаемая архитектура полностью дифференцируема и может быть обучена с применением современных методов оптимизации (<https://arxiv.org/abs/1711.05101>). В предпочтительной реализации изобретения для задач регрессии рекомендуется предобрабатывать переменные u приводя их к распределению с нулевым средним. И добавлять вектор смещения к финальному слою, инициализируя его при обработке первой группы данных, в качестве значения для инициализации использовать значения отклонения от нуля оригинальных выходов модели. Это приводит к стабилизации обучения и лучшей воспроизводимости модели.

Модель трансформер, как и любая сложная нейронная сеть, - это инструмент для создания более информативных векторных представлений и построения векторного пространства имеющего структуру, связанную с задачей. Если взять входные данные в трансформер, которые также являются векторами и рассмотреть структуру этого векторного пространства, мы обнаружим отсутствие какой-либо структуры, векторы будут нормально распределены в этом пространстве (фиг. 6). Пространство выходных векторов, в свою очередь, будет иметь структуру, которая позволит построить решающую плоскость (геометрический смысл "после чего вектор скалярно умножается на вектор параметров длины l , причем получаемое значение представляет собой уверенность модели в том, что транзакция мошенническая"), как предложено в изобретении (фиг. 7).

При конфигурации изобретения существует набор параметров, которые не уточняются в процессе обучения. Такие параметры называются гиперпараметрами. Лучший набор гиперпараметров находится с помощью валидационного набора данных и специальных алгоритмов оптимизации (<https://arxiv.org/abs/1907.10902>).

Для изобретения существует следующий набор гиперпараметров:

процент зануляемых нейронов в процессе регуляризации "дропаут" (<https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf>);

параметр температуры t ;

размеры линейных слоев внутри архитектуры трансформер;

размер векторного представления параметров;

размер параметра δ при построении мягких векторов-индикаторов;

процент зануляемых нейронов в остаточном соединении;

число слоев архитектуры трансформер;

коэффициент скорости обучения;

коэффициент затухания весов.

Результаты экспериментов.

Для экспериментов используются различные открытые наборы данных с разными задачами, такими как бинарная классификация, многоклассовая классификация и регрессия. Используются набор данных California Housing (https://www.dcc.fc.up.pt/~Itorgo/Regression/cal_housing.html) для задачи регрессии, содержащий 8 непрерывных переменных и 0 категориальных. Всего в данном наборе 20640 объектов, для оценки качества модели используется метрика корень из средней квадратической ошибки. Также используется набор данных Adult (<https://www.kaggle.com/datasets/uciml/adult-census-income>), содержащий 6 непрерывных параметров и 8 категориальных параметров. Всего в данном наборе 48842 объекта, для оценки качества модели используется метрика точность. Также используется набор данных Janis (https://link.springer.com/chapter/10.1007/978-3-030-05318-5_10), содержащий 54 непрерывных параметра и 0 категориальных. Всего в данном наборе 83733 объекта, для оценки качества модели используется метрика точность. Так же был использован набор данных IEEE-CIS Fraud detection, состоящий из 590540 объектов, содержащий 206 непрерывных переменных и 46 категориальных.

При проведении экспериментов исходный набор данных разбивается на 3 поднабора тренировочный, тестовый и валидационный. Тренировочный набор используется для применения алгоритмов оптимизации. Валидационный набор используется для оптимизации набора гиперпараметров.

Алгоритм оптимизации набора гиперпараметров осуществляет 100 итерация, гиперпараметры оптимизируются в следующих промежутках:

- процент зануляемых нейронов в процессе регуляризации "дропаут" от 0 до 0.3;
- параметр температуры t от 2 до 10;
- размеры линейных слоев внутри архитектуры трансформер от одного до трех размеров векторного представления параметров;
- размер векторного представления параметров от 32 до 128;
- размер параметра θ при построении мягких векторов-индикаторов от 272 до 592;
- процент зануляемых нейронов в остаточном соединении от 0.3 до 0.5;
- число слоев архитектуры трансформер от 6 до 9;
- коэффициент скорости обучения от 0.00001 до 0.001;
- коэффициент затухания весов от 10^{-6} до 0.001.

Для предотвращения переобучения используется подход ранней остановки, с параметром терпения равным 60.

Для получения оценки финального качества модели используется лучший набор параметров согласно качеству на валидационном наборе данных. Далее модель обучается 15 раз и используется среднее качество среди 15 экспериментов. Обучения проводилось на одной видеокарте Tesla T4. Результаты экспериментов представлены на фиг. 5. Моделями, которые были приведены для сравнения являются FT-Transformer (<https://arxiv.org/abs/2106.11959>), в основе которой используется архитектура Трансформер, но используется другой способ построения векторных представлений непрерывных параметров. TabNet (<https://arxiv.org/abs/1908.07442>) - модель которая является одной из первых попыток адаптации архитектуры Трансформер к табличным данным, но Трансформер здесь применяется не в чистом виде, а как часть специализированной архитектуры. MLP (multilayer perceptron) - многослойный перцептрон, классическая архитектура нейронной сети. ResNet (<https://arxiv.org/abs/1512.03385>) - модель с применением остаточных соединений.

Вычислительная система, способная обеспечивать обработку данных, необходимую для реализации заявленного решения, в общем случае, содержат такие компоненты как: один или более процессоров, по меньшей мере одну память, средство хранения данных, интерфейсы ввода/вывода, средство ввода, средства сетевого взаимодействия.

При исполнении машиночитаемых команд, содержащихся в оперативной памяти, конфигурируют процессор устройства для выполнения основных вычислительных операций, необходимых для функционирования устройства или функциональности одного, или более его компонентов.

Память, как правило, выполнена в виде ОЗУ, куда загружается необходимая программная логика, обеспечивающая требуемый функционал. При осуществлении работы предлагаемого решения выделяют объем памяти, необходимый для осуществления предлагаемого решения.

Средство хранения данных может выполняться в виде HDD, SSD дисков, рейд массива, сетевого хранилища, флэш-памяти и т.п.

Средство позволяет выполнять долгосрочное хранение различного вида информации, например, вышеупомянутых файлов с наборами данных пользователей/пассажиры, базы данных, содержащих записи измеренных для каждого пользователя временных интервалов, идентификаторов пользователей и т.п.

Интерфейсы представляют собой стандартные средства для подключения и работы периферийных и прочих устройств, например, USB, RS232, RJ45, COM, HDMI, PS/2, Lightning и т.п.

Выбор интерфейсов зависит от конкретного исполнения устройства, которое может представлять собой персональный компьютер, мейнфрейм, серверный кластер, тонкий клиент, смартфон, ноутбук и т.п.

В качестве средств ввода данных в любом воплощении системы, реализующей описываемый спо-

соб, может использоваться клавиатура. Аппаратное исполнение клавиатуры может быть любым известным: это может быть, как встроенная клавиатура, используемая на ноутбуке или нетбуке, так и обособленное устройство, подключенное к настольному компьютеру, серверу или иному компьютерному устройству. Подключение при этом может быть, как проводным, при котором соединительный кабель клавиатуры подключен к порту PS/2 или USB, расположенному на системном блоке настольного компьютера, так и беспроводным, при котором клавиатура осуществляет обмен данными по каналу беспроводной связи, например, радиоканалу, с базовой станцией, которая, в свою очередь, непосредственно подключена к системному блоку, например, к одному из USB-портов. Помимо клавиатуры, в составе средств ввода данных также может использоваться: джойстик, дисплей (сенсорный дисплей), проектор, тачпад, манипулятор мышь, трекбол, световое перо, динамики, микрофон и т.п.

Средства сетевого взаимодействия выбираются из устройства, обеспечивающий сетевой прием и передачу данных, например, Ethernet карту, WLAN/Wi-Fi модуль, Bluetooth модуль, BLE модуль, NFC модуль, IrDa, RFID модуль, GSM модем и т.п. С помощью средств обеспечивается организация обмена данными по проводному или беспроводному каналу передачи данных, например, WAN, PAN, ЛВС (LAN), Интранет, Интернет, WLAN, WMAN или GSM.

Компоненты устройства сопряжены посредством общей шины передачи данных.

В настоящих материалах заявка было представлено предпочтительное раскрытие осуществление заявленного технического решения, которое не должно использоваться как ограничивающее иные частные воплощения его реализации, которое не выходит за рамки испрашиваемого объема правовой охраны и является очевидными для специалистов в соответствующей области техники.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ выявления мошеннических транзакций на основе анализа табличных данных, включающий этапы, на которых:

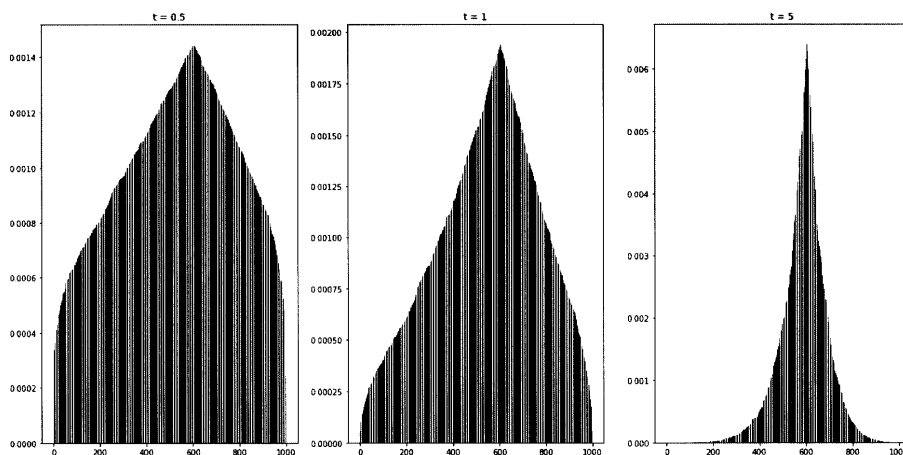
входные данные разделяются на категориальные данные и непрерывные данные, причем непрерывные данные представляются действительными числами, а категориальные данные натуральными числами и могут принимать ограниченный набор значений;

категориальные данные кодируются посредством метода быстрого кодирования one-hot encoding, при этом непрерывные данные кодируются методом softonehot;

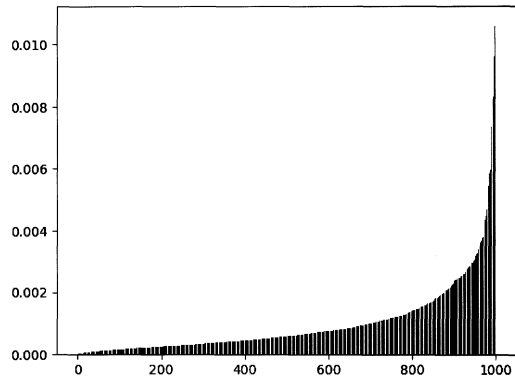
закодированные данные трансформируются в векторные представления, посредством метода обучения матрицы весов; далее общие векторные представления данных используются как входные данные для модели "трансформер" с глубокой нейронной сетью;

выходные данные из глубокой нейронной сети представляются в виде вектора длины I , после чего вектор скалярно умножается на вектор параметров длины I , причем получаемое значение представляет собой уверенность модели в том, что транзакция мошенническая, причем чем больше полученное значение, тем выше уверенность.

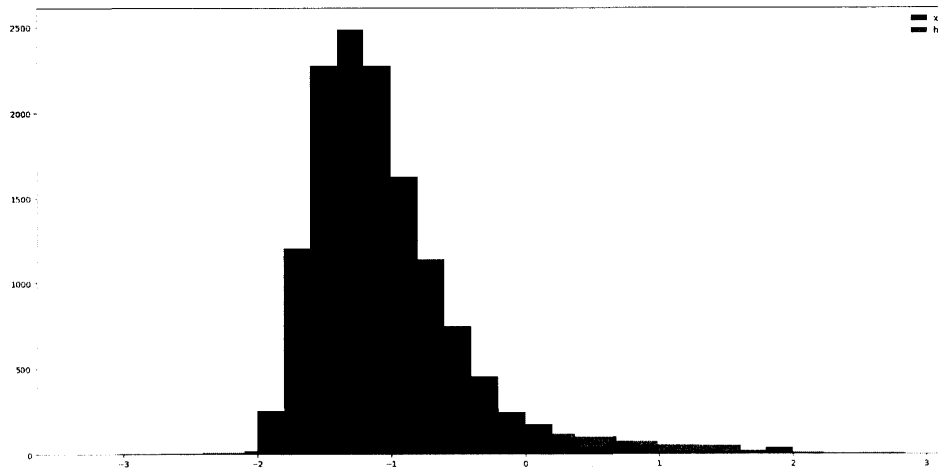
2. Способ по п.1, в котором входные данные, которые были разделены на непрерывные данные, - это сведения о транзакции, а именно сумма, дата/время, место осуществления транзакции, при этом входные данные, которые были разделены на категориальные, - это данные пользователя, совершающего транзакцию, а именно пол, возраст и история предыдущих транзакций.



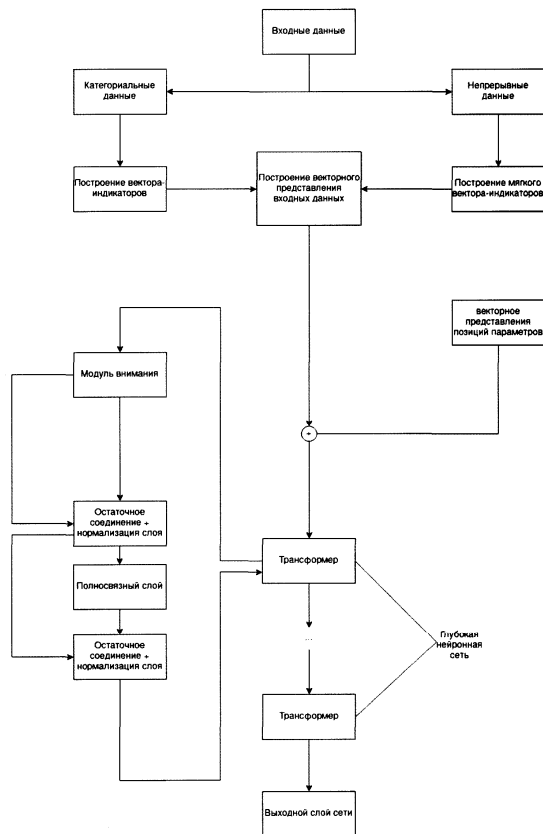
Фиг. 1



Фиг. 2



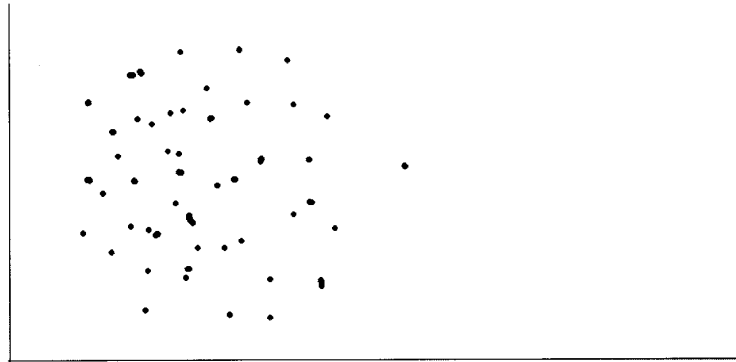
Фиг. 3



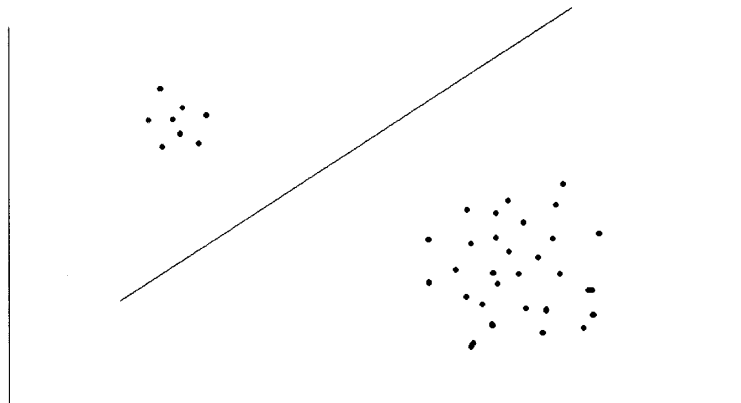
Фиг. 4

	California housing - RMSE	Adult - Accuracy	Jannis - Accuracy
Softenshot	0.4505	0.8643	0.7320
FT-Transformer	0.4591	0.8587	0.7323
TabNet	0.5100	0.8502	0.7230
MLP	0.4985	0.8522	0.7195
ResNet	0.4856	0.8544	0.7280

Фиг. 5



Фиг. 6



Фиг. 7

