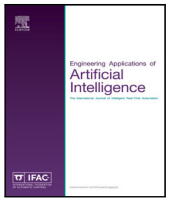
Engineering Applications of Artificial Intelligence 133 (2024) 108065

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

Explained anomaly detection in text reviews: Can subjective scenarios be correctly evaluated?

David Novoa-Paradela *, Oscar Fontenla-Romero, Bertha Guijarro-Berdiñas

Universidade da Coruña, CITIC, Campus de Elviña s/n, 15008, A Coruña, Spain

ARTICLE INFO

Keywords:
Anomaly detection
Text reviews
Transformers
Explainability

ABSTRACT

In the current landscape, user opinions exert an unprecedented influence on the trajectory of companies. In the field of online review platforms, these opinions, transmitted through text reviews and numerical ratings, significantly shape the credibility of products and services. For this reason, detecting inappropriate reviews becomes crucial.

This paper addresses the problem of automatic anomalous review detection using a novel approach based on Anomaly Detection in the field of Natural Language Processing (NLP). Unlike other NLP tasks, anomaly detection in texts is a relatively emerging area. In this paper, we present a pipeline for opinion filtering that poses the problem of discerning between normal opinions containing relevant information about an item and anomalous opinions with unrelated content. Its key functionalities include: Classifying the reviews, assigning normality scores, and generating explanations for each classification, indispensable for the human who normally moderates these platforms.

To evaluate the model, several Amazon datasets were used to demonstrate that the performance obtained is robust, obtaining an average F1 score of 91.4 detecting anomalies in the most complex scenario. In addition, a comparative study of three explainability techniques was conducted with 241 participants to measure the impact on understanding the classifications of the model and to rank their perceived usefulness of explanations.

As a result, we obtained a system with great potential to automate tasks related to online review platforms, offering insights into anomaly detection applications in textual data and showing the difficulties that arise when the task to be explained presents a subjectivity component.

1. Introduction

Nowadays more than ever in history, user opinions about products and services have a great impact on the future of the company that offers them. In such a globalized and highly competitive world, online review platforms, such as electronic commerce (e-commerce), play a crucial role in the credibility of products and services. These reviews usually come in the form of text reviews or numerical ratings made by users, accompanied in some cases by images or videos, and provide other users with information about the product or service they are considering purchasing, which directly influences the number of sales. Most people make purchase decisions based on ratings and reviews from other users (von Helversen et al., 2018).

In the case of many companies, such as Amazon, each product in the store has a list of text reviews published by customers of the platform. Users can access this list of reviews (opinions) to obtain extra information about the product, being able to mark the reviews as useful, which will position those reviews with the most votes at the top of the list. In addition to this, users can report to Amazon if they feel a review is inappropriate, for example, if its content is incorrect. This procedure to rank reviews based on their usefulness and report inappropriate reviews is carried out manually by platform users. As a result, Amazon reported more than 200 million suspected fake reviews in 2020 alone (Amazon). This problem does not only occur on Amazon but affects all platforms that allow their users to post reviews. For example, Tripadvisor uses an automatic system capable of distinguishing between normal, suspicious, and inappropriate reviews (Tripadvisor, 2021). Inappropriate ones are automatically removed (3.1% of review submissions in 2020), while those classified as suspicious are reviewed again by a human moderator (5.1% of review submissions in 2020).

In machine learning (ML), anomaly detection (AD) is the branch that builds models capable of differentiating between normal and abnormal data (Chandola et al., 2009). At first, anomaly detection might seem like a classification problem with only two classes. However, anomalies tend to occur infrequently or are non-existent, so normal

* Corresponding author.
E-mail addresses: david.novoa@udc.es (D. Novoa-Paradela), oscar.fontenla@udc.es (O. Fontenla-Romero), berta.guijarro@udc.es (B. Guijarro-Berdiñas).

https://doi.org/10.1016/j.engappai.2024.108065
Received 3 October 2023; Received in revised form 14 January 2024; Accepted 6 February 2024
Available online 28 February 2024

data prevails in these scenarios. Because of this, it is common for models to be trained only with normal data. The goal of these models is to represent the normal class as well as possible in order to classify new data as normal or anomalous. Under this premise, this type of model could be used to learn which comments are correct and automatically classify all others as anomalous. However, although the technological development of recent years has allowed the construction of very powerful models for Natural Language Processing (NLP) (Chernyavskiy et al., 2021), contrary to other tasks such as Sentiment Analysis (Tabinda Kokab et al., 2022) or Question Answering (Kim et al., 2022), the application of anomaly detection on texts is still at an early stage. As we will see in the next section, there are several proposals in the literature to address the problem of text anomaly detection using machine learning techniques. These works are focused on determining whether reviews have been written genuinely or maliciously, or on finding infrequent comments that can help sellers to improve their product or service. On the contrary, we will focus on the detection of reviews whose content is not sufficiently related to the product to which it is associated and therefore does not provide value to platform users.

For this reason, in this article we present a pipeline that, given text reviews of a product (in this case from Amazon), addresses the problem of opinion filtering as an anomaly detection problem where:

- Reviews containing representative information about the product are considered as the normal class.
- Reviews whose content has little or nothing to do with the product to be represented are considered the anomalous class.

The proposed pipeline allows us to carry out the following tasks:

- Classify reviews as normal or anomalous, allowing us to locate those that do not describe characteristics of the product to which they are associated, and therefore have no value for the users of the platform.
- Issue a normality score associated with each review.
- Generate an explanation that justifies the classification made for each review by the system.

To perform this process, the texts of the reviews are encoded using a pretrained MPNet transformer (Song et al., 2020) and then used to train a DAEF network (Novoa-Paradela et al., 2023), a non-iterative autoencoder model in charge of learning the normality patterns underlying the normal product reviews. In addition, an analysis based on the most frequent product terms is employed to justify the classifications performed by the model. The pipeline's ability to solve the anomaly detection task was evaluated using different datasets created from a large Amazon database (Amazon). Besides, to evaluate the explainability module, a study was carried out to compare three explainability techniques in which a total of 241 people participated. The objective of this study is both to measure the impact of the explanations on the reproducibility of the classification model by the respondents, and the usefulness of these explanations in scenarios in which subjectivity plays such an important role as the one raised in this work.

The contribution of this work can therefore be summarized in two main aspects:

1. Proposal and evaluation of a robust and flexible paradigm, based on machine learning techniques, for the detection of reviews that do not provide value to users of online platforms.
2. Study of the limitations of explainability techniques when operating on scenarios where explainability presents a very strong challenge due to the underlying subjectivity. For this purpose, a human evaluation of the capacity of different explainability techniques has been carried out in a real and infrequent scenario such as the detection of anomalous reviews.

We believe that this work can be useful to automate tasks such as those mentioned in online review platforms, in addition to the existing general interest in the application of anomaly detection models on texts, for which it can serve as inspiration to solve similar problems, as well as reflecting on whether it is possible to explain tasks as humanly subjective as this one.

We also consider it interesting to have carried out a human evaluation of the capacity of different explainability techniques in a real and infrequent scenario such as the detection of anomalous reviews, as well as to reflect on whether it is possible to explain tasks as humanly subjective as this one.

This document is structured as follows. Section 2 contains a brief review of the main anomaly detection works in texts and, more specifically, in text reviews. Section 3 describes the ideas taken as the basis for the development of the proposed pipeline and Section 4 describes its operation. Section 5 collects the experimentation carried out and, finally, conclusions are drawn in Section 6.

2. Related work

Anomaly detection has been a consolidated research field for years. Its great utility has allowed its techniques to be applied in numerous areas: medicine (Schneider and Xhafa, 2022), industrial systems (Truong et al., 2022), electronic fraud (Hilal et al., 2022), cybersecurity (Huong et al., 2021), etc. However, when we talk about texts and NLP, there is no massive application of these anomaly detection techniques as in the previous cases. This may be due to the difficulty in defining the idea of an anomaly in texts. Contrary to other scenarios, such as monitoring an industrial system through its sensors, in which the anomalous class will correspond to faults in the system, when the data is text, defining the concept of an anomaly is not trivial.

One of the most important lines of research related to the detection of anomalies in texts is fake reviews detection, also known as spam review detection, fake opinion detection, and spam opinion detection (Mohawesh et al., 2021). The main problem associated with fake review detection is classifying the review as either fake or genuine. There are generally three types of fake reviews (Jindal and Liu, 2008):

- **Type 1 (untruthful opinions)**: Fake reviews describing users who post negative reviews to damage a product's reputation or post positive reviews to promote it. These reviews are called fake or deceptive reviews, and they are difficult to detect simply by reading, as real and fake reviews are similar to each other.
- **Type 2 (reviews on brands only)**: Those that do not comment on the products themselves, but talk about the brands, manufacturers, or sellers of the products. Although they can be useful, they are sometimes considered spam because they are not targeted at specific products.
- **Type 3 (non-reviews)**: Non-reviews that are irrelevant and offer no genuine opinion.

There are a wide variety of ways to distinguish between intentionally written reviews and fake ones in e-commerce scenarios. In the work carried out by Salminen et al. (2022), the authors try to distinguish genuine reviews from fake reviews on Amazon. To have a labeled dataset of fake reviews, they use GPT-2 to artificially generate them. After this, they solve the task of distinguishing between genuine and fake reviews by fine-tuning a pretrained RoBERTa model. Once trained, it is shown that this model is also capable of detecting fake reviews manually written by humans. Birim et al. (2022) proposed, instead of directly handling the encoded text of the reviews, to use relevant information as the review length, purchase verification, sentiment score, or topic distribution as features to represent customer reviews. Based on these features, well-known machine learning classifiers like random forests (RF) are applied for fake detection. In another approach, Vidanagama et al. (2022) incorporate review-related features such as linguistic

features, Part-of-Speech (POS) features, and sentiment analysis features using a domain ontology to detect fake reviews with a rule-based classifier.

If instead of focusing only on reviews we focus on detecting anomalies in texts in a more general way, we could find specific methods to solve this task such as the one developed by Ruff et al. (2019), who presented Context Vector Data Description (CVDD), a text-specific anomaly detection method that allows working with sequences of variable-length embeddings using self-attention mechanisms. To overcome the limitations of CVDD, Mu et al. (2021) proposed tadnet, a textual anomaly detection network that uses an adversarial training strategy to detect anomalous texts in Social Internet of Things. In addition, thanks to the capture of the different semantic contexts of the texts, both models achieve interpretability and flexibility, allowing to detect which parts of the texts have caused the anomaly.

Other authors, instead of developing text-specific AD models, make use of well-known AD and NLP techniques to design architectures that solve the problem. Song and Suh (2019) propose to analyze the accident reports of a chemical processing plant to detect anomalous conditions, defined as unexperienced accidents that occur in unusual conditions. The authors work directly with the original text extracting the meaningful keywords of the reports using the term frequency–inverse document frequency (TF–IDF) index. Based on this, and using the local outlier factor (LOF) algorithm, they identify anomaly accidents in terms of local density clusters, finding four major types of anomaly accidents. Working with the original texts and not with embeddings they achieve a certain interpretability in the results. In another approach presented by Seo et al. (2020), a framework is proposed to identify unusual but noteworthy customer responses and extracting significant words and phrases. The authors use Doc2Vec to vectorize customer responses to which LOF is applied to identify unusual responses and, based on a TF–IDF analysis and the distances in the embedding space, visualize useful information about the results through a network graph.

In some of the works described above, as well as in most of the published works, the detection of fake reviews focuses on detecting type 1 reviews, that is, reviews that positively or negatively describe a product but whose intention is not genuine since they do not come from a real buyer (Salminen et al., 2022; Birim et al., 2022; Vidanagama et al., 2022). Other works seek to detect infrequent reviews to detect interesting aspects of their products and services (Song and Suh, 2019; Seo et al., 2020), while other approaches do not focus exactly on reviews but try to find anomalies in texts in a global way (Ruff et al., 2019; Mu et al., 2021). In most of these works, moreover, explainability is not a major issue. Since there are several works based on these scenarios, in this work we focus on detecting type 2 and 3 fake reviews, which refer to reviews that do not provide information about the product itself, but with a special emphasis on explainability. In this way, the objective of this work is to design a pipeline capable of distinguishing reviews related to a specific product (normal reviews) from reviews that do not and therefore do not provide information to the users that read them (anomalous reviews), for example, because they wrongly describe other products or because they are too generic. In addition, the classifications carried out by the system are explained through an analysis process based on NLP techniques.

3. Background

This section introduces the theoretical foundations as well as the ideas taken as the basis for the proposed pipeline.

3.1. The MPNet model

Although there are machine learning models capable of dealing directly with images or text, the majority of models are usually designed to be trained using numerical vectors that represent the sample data (tabular data). This input format of the data is the usual one for classic anomaly detection methods. In the NLP area, there are multiple techniques to represent texts using vectors of real numbers, which are known as embeddings. These techniques allow the generation of vector spaces that try to represent the relationships and semantic similarities of the language, so that, for example, two synonymous words will be found at a shorter distance in the vector space than two unrelated words.

Embeddings can be calculated independently for each word of the language (word embeddings), which led to models such as word2vec (Mikolov et al., 2013) or Global Vectors (GloVe) (Pennington et al., 2014). The representation of a sentence (sentence embedding) or a document (document embedding) will therefore be the sum of all the individual representations of the terms that make it up. To obtain representations of fixed length, it is usual to perform operations such as the mean. In certain cases, these operations between embeddings can worsen or even invalidate the final embedding, so specific models have been developed capable of understanding and representing a text as a whole, instead of just encoding it word by word. Among these models are those based on transformers (Vaswani et al., 2017), such as BERT (Devlin et al., 2018), XLNet (Yang et al., 2019), GPT-3 (Brown et al., 2020) or GPT-4 (OpenAI, 2023), which have been trained over large-scale datasets and can solve different tasks, including sentence embedding.

Since it inherits the advantages of the BERT and XLNet models while overcoming their limitations, in this work we used the pre-trained MPNet (Song et al., 2020) model to calculate the embeddings of the reviews. MPNet combines Masked Language Modeling (MLM) and Permuted Language Modeling (PLM) to predict token dependencies, using auxiliary position information as input to enable the model to view a complete sentence and reduce position differences. MPNet maps sentences and paragraphs to a 768 dimensional dense vector space $\mathbf{e} \in \mathbb{R}^{768\times1}$, providing fast and quality encodings. Computing time is a critical aspect in the area in which this work is framed, since in e-commerce platforms (and online reviews in general) we can find a huge number of products and reviews to deal with. Models like GPT-3 or GPT-4 are more advanced, but in addition to not being open source, they demand much higher computational resources, which may be unaffordable or excessive.

3.2. The DAEF network

Anomaly detection is a field with a large number of algorithms that solve the problem of distinguishing between normal and anomalous instances in a wide variety of ways (Chandola et al., 2009; Khan and Madden, 2014). Depending on the assumptions made and the processes they employ, we can distinguish different types of methods. Among the reconstruction-based methods are autoencoder (AE) networks (Vincent et al., 2010), one of the most widely used models. AE is a type of self-associative neural network whose output layer seeks to reproduce the data presented to the input layer after having gone through a dimensional compression phase. In this way, they manage to obtain a representation of the input data in a space with a dimension smaller than the original, learning a compact representation of the data, retaining the important information, and compressing the redundant one.

One of the objectives of our pipeline is to generate a normality score for each review. This score allows us, among other things, to order the different reviews by their level of normality. When AE networks are used in anomaly detection scenarios, the classification is usually carried out based on the reconstruction error that they emit to reproduce in its output the embeddings of the reviews it receives as inputs, which represents the level of normality of the evaluated instance. This reconstruction error can be used as the normality score we are looking for. Due to the speed of its training, we have decided to use DAEF (Deep AutoEncoder for Federated learning) (Novoa-Paradela et al., 2023) as our anomaly detection model. Unlike traditional neural

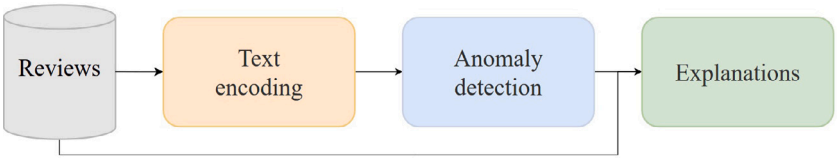


Fig. 1. General modules that form the proposed pipeline.

networks, DAEF trains a deep AE network in a non-iterative way, which drastically reduces its training time without losing the performance of traditional (iterative) AEs.

In DAEF, a first single-layer encoder reduces the dimensionality of the input data and it is adjusted using a Distributed Singular Value Decomposition (DSVD) (Fontenla-Romero et al., 2021) process. This can be accomplished by a low-rank matrix approximation of the input data, which is a minimization problem that tries to approximate a given matrix by another one subject to the constraint that the approximating matrix has reduced rank (Eckart and Young, 1936). The size of this rank is determined by the size of the hidden layer. In the subsequent layers, the decoder part of the network, the goal is to reconstruct the low-dimensional representation of the input at the network's output by training the decoder layer by layer through a non-iterative process. Similar to Extreme Learning Machine Autoencoders (ELM-AE) (Kasun et al., 2013), DAEF employs an auxiliary network to determine the parameters of each layer of the decoder in an unsupervised way, using Regularized One-Layer Neural Networks (ROLANN) as the regularization method (Fontenla-Romero et al., 2021).

The proposed pipeline uses DAEF to classify the embeddings of the reviews, being able to issue a normality score associated with each of the review classifications.

4. The proposed pipeline

The purpose of the proposed pipeline is, given the text reviews of an item, to classify them as normal if they refer to it, and as anomalous if they describe different products or if they are so generic that they do not provide useful information to consumers. These classifications will be accompanied by a normality score and explanations that justify their classification as normal or anomalous. Fig. 1 shows the three modules that are part of the proposed pipeline to solve this task: (1) Text encoding; (2) Anomaly detection; (3) Explainability. In the first phase, the text reviews of the target product are encoded using a pretrained MPNet transformer (Song et al., 2020). In the second phase, a DAEF autoencoder (Novoa-Paradela et al., 2023) is trained using these embeddings to learn the anomaly detection task. Using the reconstruction errors issued by the network and a predefined threshold error, the model can classify reviews as normal (error greater than the threshold) or anomalous (error less than the threshold). For the last phase, we propose a method based on the most frequent normal terms to generate an explanation associated with each classification.

In the following, the architecture modules and operation flow are described. To illustrate this process, we will use a concrete example where the product considered as normal corresponds to "Kind chocolate bars" (Fig. 2).

4.1. Text encoding

The pipeline starts from the list of reviews associated with the product considered as normal, in our example scenario the product "Kind chocolate bars". Given the set of reviews in text format, the objective of the first module is the representation of these reviews using the MPNet transformer model. As discussed in the previous section, MPNet allows mapping sentences and paragraphs into a 768 dimensional dense vector space to be used for different downstream tasks, such as information retrieval, clustering, or sentence similarity. In our case, we use them to perform the anomaly detection task.

In order to carry out this transformation of the textual reviews, we propose the use of the Hugging Face library (Hugging Face). By default, MPNet allows working with texts up to 384 words or tokens. If the input texts contain more than 384 words they will be truncated, although this is not a problem in the context of e-commerce, as most reviews are smaller than this limit. Furthermore, if the environment in which the embeddings are computed has a CUDA-compatible device such as a GPU, this operation will be significantly accelerated.

4.2. Anomaly detection

Once the embeddings of the target product reviews are collected, the anomaly detection model is trained. In this work we propose the use of the DAEF autoencoder network (Novoa-Paradela et al., 2023). As with any other autoencoder, it is based on the assumption that all, or at least most of the training samples belong to the normal class, even though they are not labeled.

After the network training, for each input example, the network issues a reconstruction error at the output. In this work, reconstruction errors are calculated using Mean Squared Error (MSE). This error quantifies the normality of the review as it passes through the network. The most normal reviews will emit lower reconstruction errors, while the most anomalous reviews will produce the highest values. By setting a threshold error we can classify the reviews as normal or anomalous. The way to define this threshold error is very varied. If we know about the percentage of anomalies in the training dataset, we can use various percentiles to calculate it. If we do not have any information, we can use automatic techniques such as the interquartile range (IQR) of the reconstruction errors of the training examples, defined by:

$$IQR = Q_3 - Q_1 \tag{1}$$

where Q_1 and Q_3 represent the first and the third quartiles. We define two error thresholds, one for outlier errors ($outlier IQR$) and another for extreme outliers ($extreme IQR$), as:

$$outlier IQR = Q_3 + 1.5 \times IQR \tag{2}$$

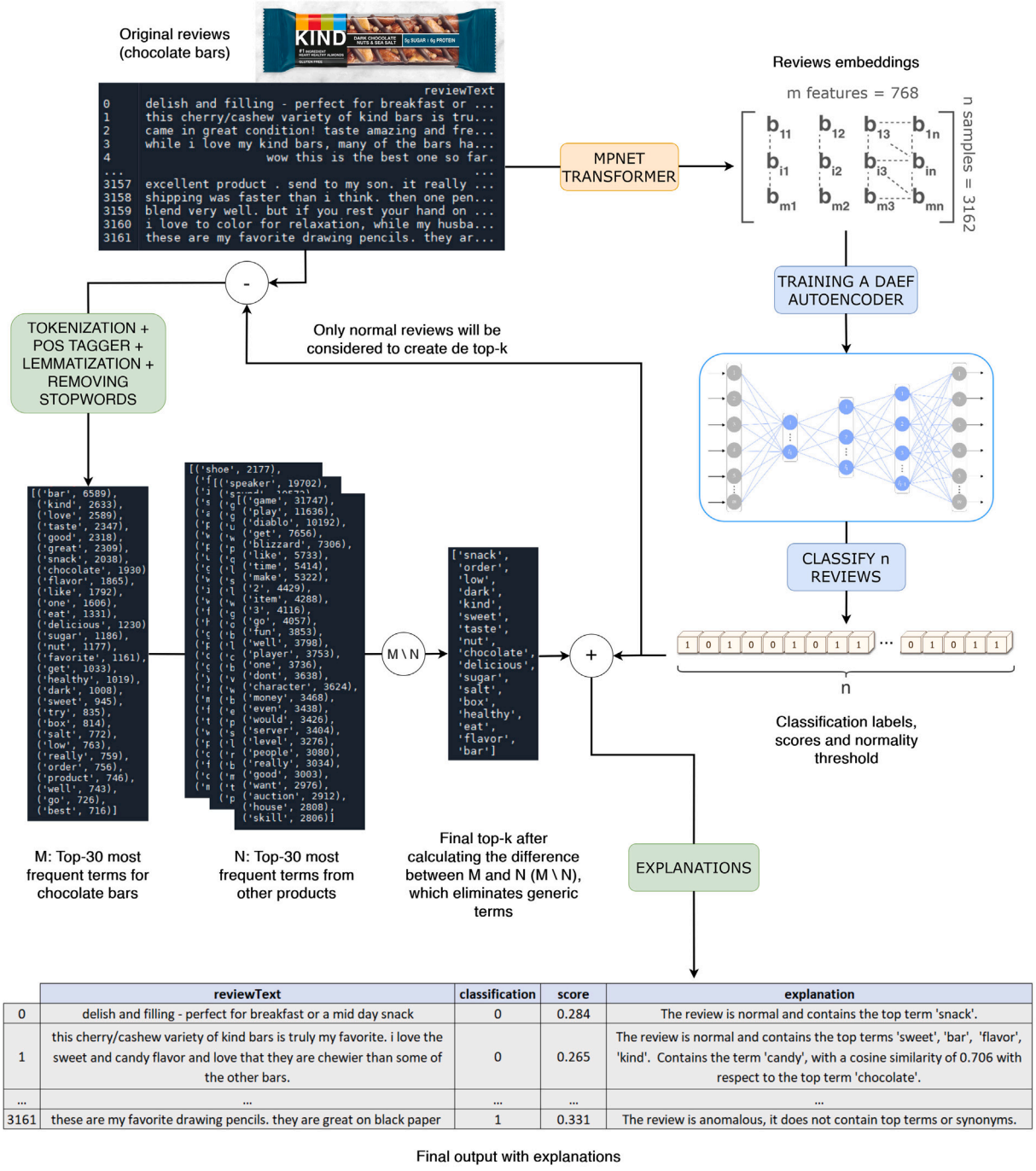
$$extreme IQR = Q_3 + 3 \times IQR \tag{3}$$

In any case, it is recommendable to experiment with different values to adapt the behavior of the system to our needs.

Using this anomaly detection module for every review its classification as normal (0) or anomalous (1) will be available together with the corresponding reconstruction error.

4.3. Explanations

Many explainability techniques base their operation on determining which characteristics of the dataset have most influenced the predictions. For example, in industrial scenarios, it is common for the features of the datasets to come directly from the physical aspects measured by the sensors of the machines, giving rise to variables such as temperatures, pressures, or vibrations. By quantifying the influence of each of these variables on the output of the system we can achieve very useful explanations.



	reviewText	classification	score	explanation
0	delish and filling - perfect for breakfast or a mid day snack	0	0.284	The review is normal and contains the top term 'snack'.
1	this cherry/cashew variety of kind bars is truly my favorite. i love the sweet and candy flavor and love that they are chewier than some of the other bars.	0	0.265	The review is normal and contains the top terms 'sweet', 'bar', 'flavor', 'kind'. Contains the term 'candy', with a cosine similarity of 0.706 with respect to the top term 'chocolate'.
...	...	...	...	...
3161	these are my favorite drawing pencils. they are great on black paper	1	0.331	The review is anomalous, it does not contain top terms or synonyms.

Fig. 2. Proposed pipeline considering the product "Kind chocolate bars" as the normal class.

However, in the scenario proposed in this paper, the data received by the anomaly detection model as input are the reviews' embeddings, these being numerical vectors. The variables that make up the embeddings are not associated with aspects understandable by human beings such as temperatures or pressures, so determining which ones have influenced the most would not provide us with useful information.

To overcome this problem, in this paper we propose an approach based on a statistical analysis of the product reviews. After the classification of the training reviews, the original list is filtered by removing those classified as anomalous by the model. These normal reviews are the ones used to elaborate the explanations of the system. Based on the definitions of normal and anomalous review presented in Section 2, our hypothesis assumes that normal reviews always refer directly or indirectly to the target product so that there will be a list of terms used very frequently among normal texts. The appearance of one or more of these "normal" terms in a review would justify its classification by the system as normal. In the same way, anomalous reviews may be justified with the non-presence of said terms.

To determine which are these normal terms, their frequency is analyzed. To do this, the texts of the reviews are processed using different NLP techniques including tokenization, post tagging, lemmatization, and stop word removal. This task is carried out using the NLTK library (Bird et al., 2009). From these processed text reviews classified as normal, we can calculate the list of most frequent terms. For our specific example (Fig. 2), we can observe the M ranking of the 30 most frequent terms and their number of occurrences. As we can see, among the terms referring to the normal product there are also generic terms that could be common in products of all types, such as *love*, *good*, or *like*.

To eliminate these generic terms, we also built the lists (N) of the most frequent terms for other items belonging to different categories from that of the item being explained (e.g., "fashion", "electronics", "video games", etc.). The terms present in those lists are subtracted from the target ranking (M), giving rise to the final list of frequent terms specific to our normal product.

Employing this final term list and the classifications made by the anomaly detection model the explanations are generated. For each text review, if the review is classified as normal, it is checked if it contains any of the frequent terms from the list or any possible semantically related term, such as a synonym. The cosine similarity calculated by MPNet is used to calculate the similarity between these terms. If the review to be processed was classified as anomalous, it is explained as the non-appearance of normal terms.

Using the classifications and reconstruction errors output by the model, as well as the resulting explanations, the final output of the system is constructed (Fig. 2).

5. Evaluation

In this section, several experiments are presented to show the behavior of the proposed pipeline in real scenarios. The section is divided into two parts: the evaluation of the anomaly detection task (Section 5.1), and the evaluation of explanations (Section 5.2). In the first one, the capability of the pipeline to detect anomalous reviews is evaluated using products from the Amazon platform. The second part discusses the problems derived from evaluating explainability techniques in this same scenario. In addition, a human study is presented to evaluate the benefits of adopting such explanations, both those generated by the explainability technique proposed in this paper and other state-of-the-art techniques.

5.1. Evaluating the anomaly detection task

The anomaly detection task that solves the proposed pipeline can be evaluated following the usual methodology in the field of anomaly detection. In this first part of the evaluation we will describe the test scenario used, the methodology employed and, finally, we will discuss the results obtained.

5.1.1. Experimental setup

The objective of this study is to evaluate the capacity of the pipeline in a real anomaly detection scenario. Although in Section 4.2 we propose the use of DAEF as the anomaly detection method, different alternatives were compared. Specifically, a second non-iterative implementation of autoencoder networks, two boundary-based methods and a density-based method were employed. These methods are Online Sequential Extreme Learning Machine (OS-ELM) (Liang et al., 2006), One-Class Support Vector Machine (OC-SVM) (Wang et al., 2004), Isolation Forest (IF) (Liu et al., 2008), and Local Outlier Factor (LOF) (Breunig et al., 2000) respectively.

For the evaluation, we employed datasets obtained from the large Amazon database (Amazon), which collects reviews of various products from the years 1996 to 2018. The main problem with this dataset is that the reviews that compose it are not labeled as normal or anomalous. Because of this, to simulate a scenario similar to the one described throughout the article, we have decided to select the reviews corresponding to several of the most demanded products. Thus, for a given test we could consider the reviews of one product as normal, and introduce reviews from other products as anomalous. Seven different product categories were selected, for which the two products with the highest number of reviews were used, resulting in a total of 14 products. Table 1 summarizes its characteristics. In all cases, MPNet was used as the model to encode the text reviews.

Besides, we considered two types of tests based on the products used:

Table 1
Characteristics of the products used.

Product	Category	Reviews
Chocolate bars	Grocery and gourmet food	11 526
Anise seeds	Grocery and gourmet food	9083
Colored pencils	Office products	14 340
Ergonomic cushion	Office products	11 942
Gaming mouse	Video games	6462
PS4 membership	Video games	5135
Bluetooth speaker	Electronics	28 539
Wi-Fi range extender	Electronics	20 873
Foot insoles	Amazon fashion	4384
Yoga leggings	Amazon fashion	3889
Hamilton album	Amazon music	3411
Partners album	Amazon music	3243
Hygrometer	Industry and scientific	14 331
Vacuum	Industry and scientific	12 182

- **1 vs. 6 — Far products**: For each of the seven categories, in this type of tests the product with the most reviews from one of the categories has been considered as the normal class, while the product with the highest number of reviews from each of the other six categories was considered the anomalous class. The fact that the product considered normal belongs to a different category should facilitate its distinction.
- **1 vs. 1 — Near products**: For each of the seven categories, in these tests the two products with the most reviews within the same category have been selected. One is considered the normal class and the other the abnormal one. The fact that both products belong to the same category should make it more difficult to distinguish them since they may have common characteristics.

The anomaly detection algorithms were trained using only normal data (the product considered as normal), while the test phase included data from both classes in a balanced manner (50% normal and 50% anomalies).

To evaluate the performance of each algorithm with each combination of hyperparameters a 10-fold was used. The normal data were divided into 10-folds so that, at each training run, 9 folds of normal data were used for training, while the remaining fold and the anomaly set were used together in a balanced manner for testing.

In this work, reconstruction errors at the output of the anomaly detector were calculated using Mean Squared Error (MSE). To establish the threshold above which this error indicates a given instance corresponds to an anomaly, among the various methods available, we employed the interquartile range (IQR) of the reconstruction errors of the training examples. In addition, throughout the tests, we also tested other thresholds using fixed percentiles (Q_{95}, Q_{90}, Q_{80}, Q_{70}, Q_{60} and Q_{50}), since a priori it is not easy to figure out which one can provide the best results, so it is considered as an additional hyperparameter to be taken into account in the case of DAEF and OS-ELM autoencoders. The OC-SVM method constitutes an exception as it automatically assigns a score to each input instance and decides its classification based on an internally calculated value. In the case of IF and LOF, the parameter contamination is used to define the threshold, so it was also considered as an additional hyperparameter.

To measure the performance of the algorithms the F1-score metric was used, considering the anomalous class as the positive one, and based on the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN):

$$F_1 = 2\frac{precision \cdot recall}{precision + recall} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{4}$$

Finally, the combinations of hyperparameters chosen for each algorithm, as well as the error thresholds, were selected using a grid search and are available in Appendix (Tables A.9 and A.10).

All the evaluation tests were performed in a machine equipped with an Intel Core i7-11700k processor and 64 GB of RAM.

Table 2
Average test F1-score ± standard deviation for the 1 vs. 6 datasets.

Normal class	Anomalous class	DAEF	OS-ELM	OC-SVM	IF	LOF
Chocolate bars	[Colored pencils, Gaming mouse, Bluetooth speaker, Foot insoles, Hamilton album, Hygrometer]	93.3 ± 0.9	96.8 ± 0.1	95.4 ± 0.1	97.2 ± 0.1	**97.6 ± 0.1**
Colored pencils	[Chocolate bars, Gaming mouse, Bluetooth speaker, Foot insoles, Hamilton album, Hygrometer]	**96.2 ± 0.5**	**96.1 ± 0.2**	94.5 ± 0.1	94.7 ± 0.2	95.9 ± 0.1
Gaming mouse	[Chocolate bars, Colored pencils, Bluetooth speaker, Foot insoles, Hamilton album, Hygrometer]	93.2 ± 0.7	**95.0 ± 0.1**	93.3 ± 0.2	93.9 ± 0.2	94.4 ± 0.1
Bluetooth speaker	[Chocolate bars, Colored pencils, Gaming mouse, Foot insoles, Hamilton album, Hygrometer]	**94.9 ± 0.6**	**94.8 ± 0.2**	93.9 ± 0.1	93.8 ± 0.2	94.4 ± 0.2
Foot insoles	[Chocolate bars, Colored pencils, Gaming mouse, Bluetooth speaker, Hamilton album, Hygrometer]	96.4 ± 1.1	96.5 ± 0.1	95.2 ± 0.05	93.9 ± 0.4	**96.9 ± 0.1**
Hamilton album	[Chocolate bars, Colored pencils, Gaming mouse, Bluetooth speaker, Foot insoles, Hygrometer]	94.3 ± 0.7	**95.6 ± 0.2**	94.6 ± 0.2	93.8 ± 0.2	93.9 ± 0.1
Hygrometer	[Chocolate bars, Colored pencils, Gaming mouse, Bluetooth speaker, Foot insoles, Hamilton album]	92.7 ± 1.0	**94.4 ± 0.1**	92.5 ± 0.1	92.6 ± 0.5	92.9 ± 0.1

5.1.2. Evaluating the anomaly detection task

Table 2 shows the results of the 1 vs. 6 — Far products experimentation. Statistical tests were carried out to compare the performance of the five approaches. A Kruskal–Wallis and Tukey's HSD test (Ostertagova et al., 2014; Nanda et al., 2021) with a significance level of 5% was used for each dataset to highlight in bold the models that rank first. As can be seen, the performance of the anomaly detection algorithms in general is very close. OS-ELM ranks as the best model for five of the seven datasets, while DAEF and LOF each rank first twice. A Nemenyi statistical test (Demšar, 2006; García and Herrera, 2008) was carried out to compare the global performance of the algorithms. Using a significance level of 5% and the F1-scores of the algorithms for the different datasets, the five methods rank in the same position, represented graphically by Fig. 3. In light of the results, we can affirm that the embeddings produced by the MPNet model provide an encoding with sufficient quality for the anomaly detection models to be able to differentiate between the evaluated products.

Table 3 collects the results of test 1 vs. 1 — Near products. Again, a Kruskal–Wallis and Tukey's HSD test (Ostertagova et al., 2014; Nanda et al., 2021) with a significance level of 5% have been used for each dataset to highlight in bold the models that rank first. In this case, for some products such as the two music albums, the differences between the performance of the algorithms are more remarkable. Once again, OS-ELM continues to rank first for most times (6/14), followed in this case by LOF (5/14), DAEF (3/14), IF (1/14), and OC-SVM (0/14). A Nemenyi statistical test (Demšar, 2006; García and Herrera, 2008) was carried out to compare the global performance of the algorithms. Using a significance level of 5% and the F1-scores of the algorithms for the different datasets, OS-ELM, LOF, DAEF, and OC-SVM are placed in the first position, represented graphically by Fig. 4, while IF is in a lower rank. The overall performance of the anomaly detection methods is still good, although it has been slightly reduced concerning the previous tests, possibly because the task is a little more complicated as the products are semantically closer to each other. Nevertheless, once again, the quality of the embeddings allows a proper differentiation between products. In this type of scenario, new reviews appear over time, even when the model is already in production. This would mean retraining the algorithm from scratch if we want to incorporate it into the model. This problem can be solved by using DAEF or OS-ELM, as they are two of the few anomaly detection models that allow what is known as online or incremental training, so its use can be beneficial. The main difference between both is that DAEF enables the use of deep architectures, while OS-ELM employs autoencoders with a single hidden layer.

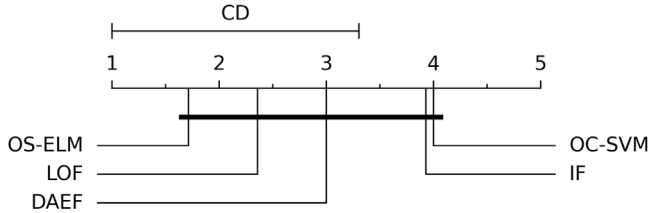


Fig. 3. Graphical representation of Nemenyi test with $\alpha = 0.05$ for the 1 vs. 6 tests. The critical distance (CD) obtained was 2.31.

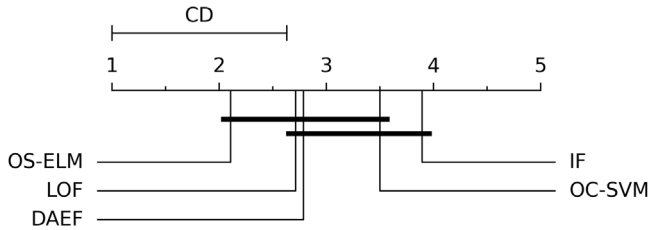


Fig. 4. Graphical representation of Nemenyi test with $\alpha = 0.05$ for the 1 vs. 1 tests. The critical distance (CD) obtained was 1.63.

5.2. Evaluating the explanations for model classifications

In Section 4.3, we proposed an explainability method to support the model decisions based on the occurrence of frequent terms, relying on the hypothesis that normal reviews will tend to use certain terms regularly, while anomalous reviews will not. In this section, this approach is compared, qualitatively through user surveys, with two other popular alternative approaches to achieve such explainability, specifically SHAP (Lundberg and Lee, 2017) and GPT-3 (Brown et al., 2020).

5.2.1. Explanations based on SHAP

Some explainability techniques have been adapted to deal with texts. In the case of NLP models, specifically transformers, SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017) is one of the most widely used techniques. SHAP is a game theory-based approach to explain the output of any machine learning model. SHAP also assigns each feature an importance value for a particular prediction, but it has been

Table 3
Average test F1-score ± standard deviation for the 1 vs. 1 datasets.

Normal class	Anomalous class	DAEF	OS-ELM	OC-SVM	IF	LOF
Chocolate bars	Anise seeds	91.6 ± 2.0	92.2 ± 0.1	90.3 ± 0.1	91.6 ± 0.6	**92.8 ± 0.1**
Anise seeds	Chocolate bars	**92.3 ± 0.6**	90.1 ± 0.1	91.4 ± 0.1	89.9 ± 0.3	90.4 ± 0.1
Colored pencils	Ergonomic cushion	96.3 ± 1.1	96.9 ± 0.1	94.9 ± 0.1	97.0 ± 0.2	**97.0 ± 0.1**
Ergonomic cushion	Colored pencils	96.4 ± 0.8	96.4 ± 0.1	94.4 ± 0.1	96.3 ± 0.2	**96.6 ± 0.1**
Gaming mouse	PS4 membership	93.4 ± 1.1	**94.1 ± 0.1**	92.0 ± 0.1	93.7 ± 0.2	93.9 ± 0.1
PS4 membership	Gaming mouse	92.6 ± 1.2	**94.0 ± 0.2**	92.0 ± 0.2	90.0 ± 0.3	90.3 ± 0.1
Bluetooth speaker	Wi-Fi range extender	90.5 ± 2.5	90.8 ± 0.1	91.3 ± 0.1	**93.1 ± 0.2**	92.7 ± 0.1
Wi-Fi range extender	Bluetooth speaker	92.8 ± 0.6	**93.7 ± 0.1**	92.5 ± 0.1	92.1 ± 0.3	91.9 ± 0.1
Foot insoles	Yoga leggings	**91.2 ± 0.5**	90.4 ± 0.1	90.8 ± 0.1	90.6 ± 0.2	**91.2 ± 0.1**
Yoga leggings	Foot insoles	90.2 ± 1.8	**91.2 ± 0.2**	90.7 ± 0.3	87.9 ± 0.6	88.4 ± 0.1
Hamilton album	Partners album	82.5 ± 1.3	**83.4 ± 0.3**	80.3 ± 0.3	62.4 ± 3.4	72.0 ± 0.5
Partners album	Hamilton album	84.1 ± 1.6	**86.9 ± 0.1**	83.8 ± 0.1	75.9 ± 1.3	80.7 ± 0.2
Hygrometer	Vacuum	**91.8 ± 0.4**	89.8 ± 0.1	90.5 ± 0.1	88.5 ± 0.6	89.3 ± 0.1
Vacuum	Hygrometer	93.6 ± 1.1	93.8 ± 0.1	92.6 ± 0.1	93.6 ± 0.2	**93.9 ± 0.1**

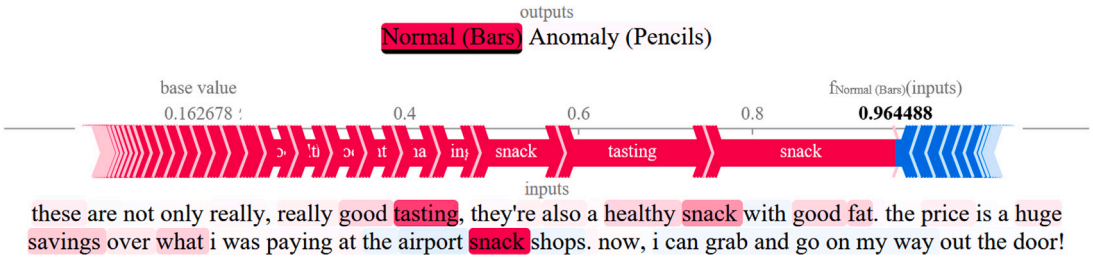


Fig. 5. Explanation generated by SHAP for the anomalous reviews detection problem raised in this work. In this scenario, the reviews to be analyzed correspond to the product "chocolate bars" (Bars). The review of this example was correctly classified as normal. The most influential terms in its classification as normal are marked in red, while the terms that promote the opposite class are highlighted in blue, in this case practically none since we are dealing with an obvious case. Greater intensity implies greater influence. In this case, the review terms that have most influenced its classification as normal are "snack" and "tasting".

extended to provide interpretability to models that use embeddings as input. In these cases, it can quantify the importance of each word of the original text in the prediction, which generates a quite understandable interpretation for a human being.

In this work, we considered that SHAP could be a good alternative to our proposal to generate the explanations associated with the classification of the reviews. Fig. 5 shows the explanation generated by SHAP for a review classified as normal, where it can be seen that each term of the review has an associated score representing its influence on the classification. Despite this, for the evaluation of explanation techniques and to facilitate the understanding of the explanation by the final users (many of them unfamiliar with these techniques), during this work the explanations generated by SHAP were simplified, showing only the five most influential terms, instead of all.

5.2.2. Explanations based on GPT-3

There is no doubt that GPT-3 has started a technological revolution at all levels. Its availability to the general public has led to the discovery of a large number of unimaginable features before its release. The original idea was to create a high-level conversational bot, trained with a large amount of text available on the web, such as books, online encyclopedias, or forums. However, its deep understanding of language has far exceeded the preset idea of a chatbot. GPT-3 is capable of successfully carrying out tasks that go beyond writing a joke or summarizing a novel, GPT-3 is capable of analyzing and developing code in multiple programming languages, generating SEO positioning strategies, or carrying out NLP tasks traditionally solved by ad hoc models, such as sentiment analysis.

Due to its enormous potential, in this work, we have decided to study GPT-3 as an explainability model. To do this, first of all, we have introduced a prompt in which we describe the task that our anomaly detection model is carrying out, as well as the format with which we would like to work in future prompts (see Fig. 6). After this, and following the predefined format, GPT-3 is ready to generate the explanations (see Fig. 7). Although GPT-3 does not have direct knowledge of the anomaly detection model, its ability to generate consistent and intuitive responses can be of great help to humans reading the explanations.

We should note that since GPT-3 trains on a wide range of data from the internet, including data that may contain biases, there is a risk that the generated explanations may reflect or amplify those biases. It is essential to exercise caution and perform a critical analysis of the explanations generated, considering their context and possible inherent biases.

Table 4
Area of knowledge of the survey participants.

Knowledge area	Participants
Engineering and architecture	87 (36.1%)
Social and legal sciences	77 (32.0%)
Natural sciences	33 (13.7%)
Arts and humanities	23 (9.5%)
Health sciences	17 (7.1%)
Others	4 (1.7%)

5.2.3. How to evaluate explainability

Unlike other tasks such as anomaly detection, the quality of the explanations generated to provide certain interpretability to a model is not easily measurable. In a scenario like the one described in this paper, the subjectivity of the users plays a strong role when determining whether an explanation is appropriate or not. Due to this, we have decided to carry out a comparative study of the three explainability techniques using a survey. This survey was disseminated through the students, professors, and research and administrative staff of our university, giving rise to a total of 241 participants. Table 4 shows the number of participants by area of knowledge. To build the survey, reviews from the "1 vs. 6" scenario described in Table 2 were used, considering "chocolate bars" as the normal class, and using DAEF as an anomaly detection method.

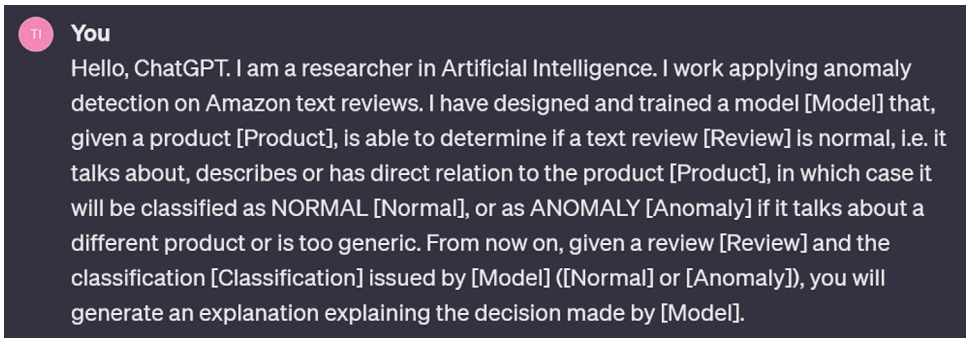


Fig. 6. Initial prompt to present the anomaly detection problem to be solved to GPT-3.

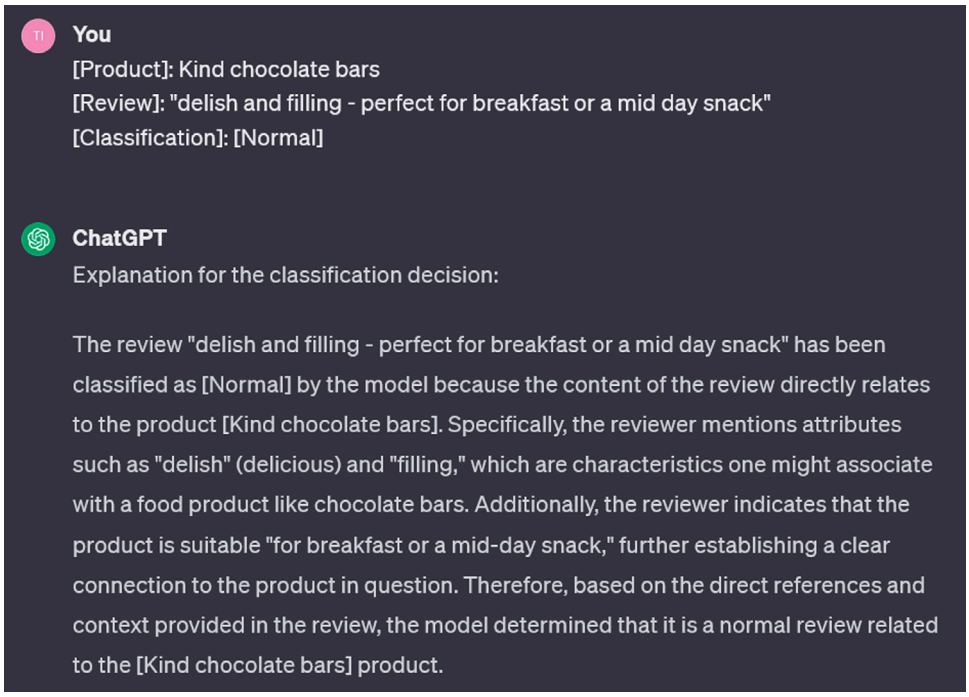


Fig. 7. Prompt in which the classification of a review is explained by GPT-3. The product considered as normal is chocolate bars and the review has been classified as normal by the anomaly detection model.

The survey consists of two different tests: (1) Forward simulation (Hase and Bansal, 2020), which allows measuring the effect of explanations on users; (2) Personal utility, which allows measuring the utility of the explanations based on the personal preferences of the users.

5.2.4. Forward simulation

This test is inspired by the work carried out by Hase and Bansal (2020) and is divided into four phases: a Learning phase, a Pre-prediction phase, a Learning phase with explanations, and a Post-prediction phase. To begin, users are given 20 examples from the model's validation set with reviews and model predictions with no explanations. Then they must predict the model's output for 10 new reviews. Users cannot access the example reviews from the learning phase while they are in this prediction phase. Next, they return to the same learning examples, now with explanations included. Finally, they predict the behavior of the model again on the same instances as in the first round of prediction. The classes of reviews chosen for each of the phases described are balanced between normal and abnormal. By design, any improvement in user performance in the Post prediction phase is attributable only to the addition of explanations that helped the user to better understand the behavior of the model.

Fig. 8 represents this procedure, where x represents a review, $\hat{y}$ is the class predicted by the model, and $\tilde{y}$ the class predicted by the human simulation. Taking this into account, the Explanation Effect can be calculated as follows:

$$Explanation\ Effect = Post\ Accuracy - Pre\ Accuracy \tag{5}$$

where the pre and post accuracies area calculated comparing the user's prediction against the model's prediction. In order not to bias the results, we have decided that each person surveyed will only participate in the forward simulation of a single explainability technique, so that the total number of participants was divided randomly into three groups, each one associated with one of the three techniques: Terms frequency (78), SHAP (89), and GPT-3 (77). The three groups of participants faced the same reviews throughout the four phases of the test, only the explanations presented in the third phase of the test (learning with explanations) varied depending on the assigned group/technique. Throughout the survey, we have warned users several times that they should try to simulate the behavior of the anomaly detection model, instead of ranking the reviews using their personal criteria.

Tables 5 and 6 show the results of this test. As can be seen in the first table, the average explanation effect of the three explainability

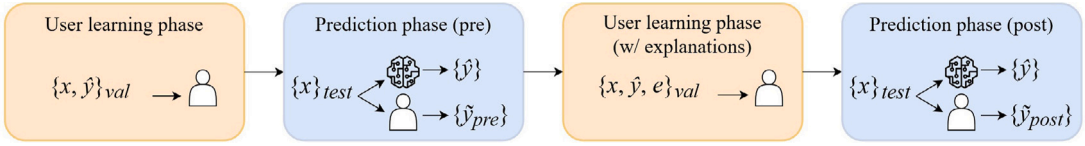


Fig. 8. Forward simulation test procedure to measure human users' ability to understand and predict model behavior. To isolate the impact of explanations, baseline accuracy is measured first, followed by accuracy measurement after users have access to explanations of the model's behavior. The explained examples are different from the test instances.

Table 5
Forward simulation tests. Average accuracy and Explanation effect (± standard deviation) for each explainability technique.

Technique	Pre accuracy	Post accuracy	Explanation effect
Terms frequency	76.2 ± 13.0	72.8 ± 13.8	−3.4 ± 12.7
SHAP	72.4 ± 14.9	71.9 ± 14.9	−0.5 ± 13.8
GPT-3	69.7 ± 15.2	70.1 ± 17.1	0.4 ± 14.3

Table 6
Forward simulation tests. Average accuracy and explanation effect (± standard deviation) broken down by participants' area of knowledge.

Area of knowledge	Pre accuracy	Post accuracy	Explanation effect
Engineering and architecture	76.3 ± 14.5	73.9 ± 16.1	−2.4 ± 12.9
Social and legal sciences	72.2 ± 13.0	69.5 ± 15.8	−2.7 ± 13.0
Natural sciences	70.9 ± 14.7	75.4 ± 12.8	4.5 ± 13.5
Arts and humanities	68.7 ± 17.7	67.8 ± 14.8	−0.9 ± 19.0
Health sciences	72.0 ± 14.2	71.3 ± 10.6	−0.7 ± 10.3

techniques has not been very remarkable. The high standard deviation suggests a high variability between the different study participants, both in the initial (Pre) and subsequent (Post) classifications and therefore in the explanation effect. The initial difference between the groups in the Pre-accuracy and the closeness of the three techniques in the explanation effect does not allow us to opt for any of the three options.

In Table 6 we can see the results of the test based on the area of knowledge of the participants. The results obtained using the different explainability techniques have been aggregated. The purpose of this comparison is to analyze whether there is any relationship between the technical background of the respondents and their performance on the test. As can be seen, there are slight differences between the values of pre and post accuracy, with the group of respondents belonging to the area of natural sciences standing out, whose explanation effect was the only positive one (4.5%).

Analyzing the results we can affirm that the initial accuracy (pre-accuracy) is quite high, which indicates that the respondents tend to successfully reproduce the behavior of the model even if they do not have the explanations. This could be because, in this case, both the input data to the AD model (in natural language) and the problem it solves are easily understandable to a human, which means that the respondents can solve the classification problem by themselves. The standard deviation accompanying the pre-accuracy results is notable but does not become too high, which reaffirms the previous argument.

After supplying the respondents with the explanations associated with the reviews, the post-accuracy obtained by them presents values very similar to the previous scores (pre-accuracy). We can therefore affirm that in general terms the effect of the explanations, in this case, has not been beneficial for the users during this test.

The non-improvement may be due to several reasons. One of them may be the presence of reviews that show a certain degree of ambiguity, which not only makes their classification difficult for the respondents but also for the AD models. However, in real scenarios the occurrence of reviews whose normality score is around the threshold value would be something to be expected, not all events are easily classifiable. Another possible reason may be that the tendency of some users throughout the survey has been to classify the reviews using their personal criteria, rather than trying to simulate the behavior of the anomaly detection model.

Table 7
Personal utility tests. Average ranking (± standard deviation) for each explainability technique.

Technique	Position
Terms frequency	1.6 ± 0.4
SHAP	2.1 ± 0.5
GPT-3	1.7 ± 0.5

Table 8
Personal utility tests. Average ranking (± standard deviation) for each technique and area of knowledge.

Area of knowledge	Technique	Position
Engineering and architecture	Terms frequency	1.6 ± 0.4
	SHAP	2.2 ± 0.5
	GPT-3	1.7 ± 0.5
Social and legal sciences	Terms frequency	1.7 ± 0.4
	SHAP	1.9 ± 0.4
	GPT-3	1.6 ± 0.5
Natural sciences	Terms frequency	1.6 ± 0.4
	SHAP	2.0 ± 0.5
	GPT-3	1.7 ± 0.5
Arts and humanities	Terms frequency	1.5 ± 0.4
	SHAP	1.9 ± 0.5
	GPT-3	1.8 ± 0.5
Health sciences	Terms frequency	1.8 ± 0.3
	SHAP	2.1 ± 0.5
	GPT-3	1.6 ± 0.5

5.2.5. Personal utility

After completing the first test, the participants were given a second exercise, common to all participants, regardless of the group to which they were assigned in the previous phase. This consists of a subjective evaluation of the three explainability techniques. The idea is to present the participant with a review, its classification by the model, and an explanation generated by each of the three explainability techniques. The participant must order the explanations based on how useful it is to understand the reasoning behind the model's decision (ties were allowed between explanations). This process was repeated with a total of eight reviews. Tables 7 and 8 show the final average rankings of the explainability techniques.

As can be seen in Table 7, the explanations based on Term frequency (1.6) and GPT-3 (1.7) are in very close positions, both being above the third method SHAP (2.1). The preference of respondents for the first two methods may be due to the fact that the format of their explanations is more accessible and descriptive for a larger part of the population. Grouping the results by areas of knowledge (Table 8), we can see that the general trend is maintained for most areas. We can highlight the case of Social and Legal Sciences and Health Sciences, areas in which the positions in the ranking of Terms frequency and GPT-3 techniques are slightly inverted, with GPT-3 being the preferred option.

6. Conclusion

In this work, we have proposed a pipeline to detect anomalous reviews associated with Amazon products, which can be directly extrapolated to other online review platforms or scenarios with similar

characteristics. The representation of the reviews using MPNet embeddings has enabled the training of classical anomaly detection algorithms that have achieved a high performance in terms of F1-score. These have been evaluated using reviews from different products and categories, and the score they emit allows us to sort the reviews based on their normality.

A technique based on the occurrence of frequent terms has been proposed to generate explanations associated with the classifications of the reviews. This technique has been compared with SHAP, one of the reference post-hoc techniques in the field of explainability, and with GPT-3, due to its high power and versatility. To evaluate this aspect of the pipeline, we conducted a two-part survey in which 241 members of the university community participated.

From the first part of the explainability test we can conclude that, in general terms, the effect of the explanations has not been beneficial for the users. In any case, these tests allow us to reflect on the difficulty of using explainability and evaluation techniques in borderline scenarios where subjectivity plays an important role, such as the one presented in this article or in other fields of NLP, as well as in areas such as image or audio generation.

Regarding the second part of the explainability test, we have been able to conclude that respondents preferred explanations that presented a more natural and familiar appearance over more condensed and concise explanations such as those provided by SHAP, regardless of the explanation effect they provide. Explanations based on term frequency analysis have been preferred by respondents along with GPT-3, however, our approach presents significantly lower computational costs and both its use and the explanations produced are simpler for the users.

However, it is necessary to mention some of the limitations of the proposed architecture. The main constraint of the presented approach is that, given the nature of the problem, it is necessary to train and maintain an anomaly detection model for each product of the platform, which in some cases could involve problems such as system upscaling.

Moreover, in this type of situation, new reviews come in over time even after the model is already in production. This would mean retraining the algorithm from scratch if we want to incorporate them into the model. This problem is solved with the use of DAEF or OS-ELM as they are two of the few anomaly detection models that allow what is known as online or incremental training. The main difference between both is that OS-ELM employs autoencoders with a single hidden layer, while DAEF enables the use of deep architectures, which can be beneficial in certain scenarios.

Finally, the calculation of review embeddings using transformer models can be a slow operation if we are faced with a scenario of a certain magnitude, for example, millions of reviews for each product, which can become a limitation if sufficient hardware resources are not available.

In future work, it would be interesting to evaluate GPT-3 and other large language models carrying out the complete process followed by the pipeline proposed in this work, instead of being tested only in the explainability module. We have not performed this test due to the high computational cost that would be involved in processing the thousands of reviews to be evaluated. Since it is common for text reviews to be accompanied by images, it would also be of great interest to employ multimodal models for their analysis as a whole, which could be very beneficial to understanding user opinions at a deeper level (Pérez-Núñez et al., 2023). If we had information about users in relation to their purchases, it would also be interesting to test the use of recommender systems to discern between legitimate and illegitimate comments from the user's point of view.

Regarding explainability, another interesting possible line of future work would be to broaden the scope of the survey, both in terms of the number of products involved and the number of reviews, in order to clarify the conclusions reached at the forward simulation stage. It would be very useful to try to present the explanations issued by SHAP in a more familiar and natural format for the end user, so that we can see if their level of preference is increased for the general public. Finally, it would be interesting to develop an explainable-by-design anomaly detection algorithm, thus avoiding the need to use post-hoc explainability techniques.

CRediT authorship contribution statement

David Novoa-Paradela: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Oscar Fontenla-Romero:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Bertha Guijarro-Berdiñas:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by grant *Machine Learning on the Edge - Ayudas Fundación BBVA a Equipos de Investigación Científica 2019*; the Spanish National Plan for Scientific and Technical Research and Innovation (PID2019-109238GB-C22 and TED2021-130599A-I00); the Xunta de Galicia (ED431C 2022/44) and ERDF funds. CITIC, as a Research Center of the University System of Galicia, is funded by Consellería de Educación, Universidade e Formación Profesional of the Xunta de Galicia, Spain through the European Regional Development Fund (ERDF) and the Secretaría Xeral de Universidades (Ref. ED431C 2022/44). Funding for open access charge: Universidade da Coruña/CISUG.

Appendix. Hyperparameters used during training

This appendix contains the values of the hyperparameters finally chosen as the best for each method and dataset, listed in Tables A.9 and A.10.

- Deep Autoencoder for Federated learning (DAEF) (Novoa-Paradela et al., 2023).
  - Architecture: Neurons per layer.
  - λ_{hid}: Regularization hyperparameter of the hidden layer.
  - λ_{last}: Regularization hyperparameter of the last layer.
  - μ: Anomaly threshold.
- Online Sequential Extreme Learning Machine (OS-ELM) (Liang et al., 2006).
  - Architecture: Neurons per layer.
  - μ: Anomaly threshold.
- One-Class Support Vector Machine (OC-SVM) (Wang et al., 2004).
  - An upper bound on the fraction of training errors and a lower bound of the fraction of support vectors (ν).

Table A.9
Hyperparameters used during the 1 vs. 6 experimentation.

Normal class	DAEF	OS-ELM	OC-SVM	IF	LOF
Chocolate bars	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: outlier IQR	Arch: [768, 400, 768], μ: extreme IQR	ν: 0.1, kernel: linear, γ: scale	Estimators: 1000, c: 0.05	Neighbors: 2000, c: 0.05
Colored pencils	Arch: [768, 550, 650, 768], λ_{hid}: 0.1, λ_{last}: 0.1, μ: outlier IQR	Arch: [768, 500, 768], μ: extreme IQR	ν: 0.1, kernel: linear, γ: scale	Estimators: 300, c: 0.1	Neighbors: 4000, c: 0.05
Gaming mouse	Arch: [768, 550, 650, 768], λ_{hid}: 0.1, λ_{last}: 0.1, μ: outlier IQR	Arch: [768, 400, 768], μ: extreme IQR	ν: 0.1, kernel: poly, degree: 2, γ: scale	Estimators: 1000, c: 0.1	Neighbors: 2000, c: 0.05
Bluetooth speaker	Arch: [768, 550, 650, 768], λ_{hid}: 0.75, λ_{last}: 0.1, μ: outlier IQR	Arch: [768, 300, 768], μ: outlier IQR	ν: 0.1, kernel: poly, degree: 3, γ: scale	Estimators: 1000, c: 0.1	Neighbors: 4000, c: 0.1
Foot insoles	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: outlier IQR	Arch: [768, 300, 768], μ: extreme IQR	ν: 0.1, kernel: poly, degree: 2, γ: scale	Estimators: 1000, c: 0.05	Neighbors: 2000, c: 0.05
Hamilton album	Arch: [768, 550, 650, 768], λ_{hid}: 0.75, λ_{last}: 0.1, μ: outlier IQR	Arch: [768, 100, 768], μ: outlier IQR	ν: 0.1, kernel: poly, degree: 2, γ: scale	Estimators: 200, c: 0.1	Neighbors: 1000, c: 0.1
Hygrometer	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: outlier IQR	Arch: [768, 200, 768], μ: outlier IQR	ν: 0.1, kernel: poly, degree: 2, γ: scale	Estimators: 1000, c: 0.1	Neighbors: 4000, c: 0.1

Table A.10
Hyperparameters used during the 1 vs. 1 experimentation.

Normal class	DAEF	OS-ELM	OC-SVM	IF	LOF
Chocolate bars	Arch: [768, 550, 650, 768], λ_{hid}: 0.1, λ_{last}: 0.1, μ: Q_{90}	Arch: [768, 500, 768], μ: outlier IQR	ν: 0.1, kernel: poly, degree: 4, γ: scale	Estimators: 1000, c: 0.1	Neighbors: 4000, c: 0.1
Anise seeds	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: Q_{80}	Arch: [768, 400, 768], μ: outlier IQR	ν: 0.1, kernel: poly, degree: 4, γ: scale	Estimators: 1000, c: 0.2	Neighbors: 3000, c: 0.2
Colored pencils	Arch: [768, 550, 650, 768], λ_{hid}: 0.25, λ_{last}: 0.25, μ: outlier IQR	Arch: [768, 500, 768], μ: extreme IQR	ν: 0.1, kernel: linear, γ: scale	Estimators: 500, c: 0.05	Neighbors: 4000, c: 0.05
Ergonomic cushion	Arch: [768, 550, 650, 768], λ_{hid}: 0.5, λ_{last}: 0.1, μ: outlier IQR	Arch: [768, 200, 768], μ: extreme IQR	ν: 0.1, kernel: sigmoid, γ: scale	Estimators: 500, c: 0.05	Neighbors: 3000, c: 0.05
Gaming mouse	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: Q_{90}	Arch: [768, 500, 768], μ: extreme IQR	ν: 0.1, kernel: poly, degree: 2, γ: scale	Estimators: 1000, c: 0.1	Neighbors: 3000, c: 0.1
PS4 membership	Arch: [768, 550, 650, 768], λ_{hid}: 0.1, λ_{last}: 0.1, μ: Q_{90}	Arch: [768, 400, 768], μ: extreme IQR	ν: 0.1, kernel: poly, degree: 2, γ: scale	Estimators: 1000, c: 0.2	Neighbors: 1000, c: 0.2
Bluetooth speaker	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: Q_{90}	Arch: [768, 20, 768], μ: Q_{90}	ν: 0.1, kernel: poly, degree: 2, γ: scale	Estimators: 1000, c: 0.1	Neighbors: 4000, c: 0.1
Wi-Fi range extender	Arch: [768, 550, 650, 768], λ_{hid}: 0.25, λ_{last}: 0.1, μ: Q_{90}	Arch: [768, 500, 768], μ: outlier IQR	ν: 0.1, kernel: poly, degree: 4, γ: scale	Estimators: 1000, c: 0.1	Neighbors: 4000, c: 0.1
Foot insoles	Arch: [768, 550, 650, 768], λ_{hid}: 0.75, λ_{last}: 0.1, μ: outlier IQR	Arch: [768, 20, 768], μ: extreme IQR	ν: 0.1, kernel: poly, degree: 4, γ: scale	Estimators: 1000, c: 0.2	Neighbors: 500, c: 0.2
Yoga leggings	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: Q_{90}	Arch: [768, 20, 768], μ: extreme IQR	ν: 0.1, kernel: poly, degree: 4, γ: scale	Estimators: 500, c: 0.2	Neighbors: 300, c: 0.2
Hamilton album	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: Q_{90}	Arch: [768, 100, 768], μ: Q_{80}	ν: 0.1, kernel: poly, degree: 4, γ: scale	Estimators: 200, c: 0.2	Neighbors: 50, c: 0.2
Partners album	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: Q_{90}	Arch: [768, 50, 768], μ: Q_{80}	ν: 0.2, kernel: poly, degree: 4, γ: scale	Estimators: 1000, c: 0.2	Neighbors: 300, c: 0.2
Hygrometer	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: Q_{90}	Arch: [768, 100, 768], μ: Q_{90}	ν: 0.1, kernel: poly, degree: 4, γ: scale	Estimators: 1000, c: 0.2	Neighbors: 4000, c: 0.2
Vacuum	Arch: [768, 550, 650, 768], λ_{hid}: 0.9, λ_{last}: 0.9, μ: Q_{90}	Arch: [768, 300, 768], μ: outlier IQR	ν: 0.1, kernel: poly, degree: 3, γ: scale	Estimators: 300, c: 0.1	Neighbors: 4000, c: 0.1

  - Kernel type: Linear, Polynomial or RBF.
  - Kernel coefficient γ (in the case of polynomial and RBF kernels).
  - Degree (in the case of polynomial kernel).

- Isolation Forest (IF) (Liu et al., 2008).
  - The number of base estimators in the ensemble.
  - Contamination of the dataset (c).

- Local Outlier Factor (LOF) (Breunig et al., 2000).
  - Number of neighbors.
  - Contamination of the dataset (c).

References

Amazon, Amazon customer reviews dataset, https://nijianmo.github.io/amazon/index.html.

Amazon, Amazon targets fake review fraudsters on social media, https://www.aboutamazon.com/news/policy-news-views/amazon-targets-fake-review-fraudsters-on-social-media.

Bird, S., Klein, E., Loper, E., 2009. Natural Language Processing with Python, first ed. O'Reilly Media, Inc..

Birim, Ş.Ö., Kazancoglu, I., Kumar Mangla, S., Kahraman, A., Kumar, S., Kazancoglu, Y., 2022. Detecting fake reviews through topic modelling. J. Bus. Res. 149, 884–900. http://dx.doi.org/10.1016/j.jbusres.2022.05.081.

Breunig, M., Kriegel, H.-P., Ng, R.T., Sander, J., 2000. LOF: Identifying density-based local outliers. In: ACM Sigmoid International Conference on Management of Data. ACM SIGMOD Record, pp. 93–104.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al., 2020. Language models are few-shot learners. Advances in neural information processing systems 33, 1877–1901.

Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: A survey. ACM Comput. Surv. 41 (3), http://dx.doi.org/10.1145/1541880.1541882.

Chernyavskiy, A., Ilvovsky, D., Nakov, P., 2021. Transformers:"the end of history" for natural language processing? In: Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21. Springer, pp. 677–693.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7 (1), 1–30.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. http://dx.doi.org/10.48550/ARXIV.1810.04805.

Eckart, C., Young, G., 1936. The approximation of one matrix by another of lower rank. Psychometrika 1 (3), 211–218. http://dx.doi.org/10.1007/BF02288367.

Fontenla-Romero, O., Guijarro-Berdiñas, B., Pérez-Sánchez, B., 2021. Regularized one-layer neural networks for distributed and incremental environments. In: IWANN. Vol. 12862, Springer, pp. 343–355. http://dx.doi.org/10.1007/978-3-030-85099-9_28.

Fontenla-Romero, O., Pérez-Sánchez, B., Guijarro-Berdiñas, B., 2021. DSVD-autoencoder: A scalable distributed privacy-preserving method for one-class classification. Int. J. Intell. Syst. 36 (1), 177–199. http://dx.doi.org/10.1002/int.22296.

García, S., Herrera, F., 2008. An extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. J. Mach. Learn. Res. 9 (89), 2677–2694.

Hase, P., Bansal, M., 2020. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Online, pp. 5540–5552. http://dx.doi.org/10.18653/v1/2020.acl-main.491.

Hilal, W., Gadsden, S.A., Yawney, J., 2022. Financial fraud: A review of anomaly detection techniques and recent advances. Expert Syst. Appl. 193, 116429. http://dx.doi.org/10.1016/j.eswa.2021.116429.

Hugging Face, 2023. Accessed: 2023-05-09, https://huggingface.co/.

Huong, T.T., Bac, T.P., Long, D.M., Luong, T.D., Dan, N.M., Quang, L.A., Cong, L.T., Thang, B.D., Tran, K.P., 2021. Detecting cyberattacks using anomaly detection in industrial control systems: A federated learning approach. Comput. Ind. 132, 103509. http://dx.doi.org/10.1016/j.compind.2021.103509.

Jindal, N., Liu, B., 2008. Opinion spam and analysis. In: Proceedings of the 2008 International Conference on Web Search and Data Mining. WSDM '08, Association for Computing Machinery, New York, NY, USA, pp. 219–230. http://dx.doi.org/10.1145/1341531.1341560.

Kasun, L., Zhou, H., Huang, G.-B., Vong, C.-M., 2013. Representational learning with ELMs for Big Data. IEEE Intell. Syst. 28, 31–34.

Khan, S.S., Madden, M.G., 2014. One-class classification: taxonomy of study and review of techniques. Knowl. Eng. Rev. 29 (3), 345–374.

Kim, Y., Bang, S., Sohn, J., Kim, H., 2022. Question answering method for infrastructure damage information retrieval from textual data using bidirectional encoder representations from transformers. Autom. Constr. 134, 104061. http://dx.doi.org/10.1016/j.autcon.2021.104061.

Liang, N.-y., Huang, G.-b., Saratchandran, P., Sundararajan, N., 2006. A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans. Neural Netw. 17 (6), 1411–1423. http://dx.doi.org/10.1109/TNN.2006.880583.

Liu, F.T., Ting, K.M., Zhou, Z., 2008. Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining. pp. 413–422.

Lundberg, S.M., Lee, S.-I., 2017. A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Vol. 30, Curran Associates, Inc., URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. http://dx.doi.org/10.48550/ARXIV.1301.3781.

Mohawesh, R., Xu, S., Tran, S.N., Ollington, R., Springer, M., Jararweh, Y., Maqsood, S., 2021. Fake reviews detection: A survey. IEEE Access 9, 65771–65802. http://dx.doi.org/10.1109/ACCESS.2021.3075573.

Mu, J., Zhang, X., Li, Y., Guo, J., 2021. Deep neural network for text anomaly detection in sIoT. Comput. Commun. 178, 286–296. http://dx.doi.org/10.1016/j.comcom.2021.08.016.

Nanda, A., Mahapatra, A., Mohapatra, B., mahapatra, a., 2021. Multiple comparison test by Tukey's honestly significant difference (HSD): Do the confident level control type I error. Int. J. Appl. Math. Stat. 6, 59–65. http://dx.doi.org/10.22271/maths.2021.v6.i1a.636.

Novoa-Paradela, D., Fontenla-Romero, O., Guijarro-Berdiñas, B., 2023. Fast deep autoencoder for federated learning. Pattern Recognit. 143, 109805. http://dx.doi.org/10.1016/j.patcog.2023.109805, URL https://www.sciencedirect.com/science/article/pii/S0031320323005034.

OpenAI, 2023. GPT-4 technical report. arXiv:2303.08774.

Ostertagova, E., Ostertag, O., Kováč, J., 2014. Methodology and application of the Kruskal-Wallis test. Appl. Mech. Mater. 611, 115–120.

Pennington, J., Socher, R., Manning, C., 2014. Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp. 1532–1543. http://dx.doi.org/10.3115/v1/D14-1162.

Pérez-Núñez, P., Díez, J., Luaces, O., Remeseiro, B., Bahamonde, A., 2023. Users' photos of items can reveal their tastes in a recommender system. Inform. Sci. 642, 119227. http://dx.doi.org/10.1016/j.ins.2023.119227, URL https://www.sciencedirect.com/science/article/pii/S0020025523008125.

Ruff, L., Zemlyanskiy, Y., Vandermeulen, R., Schnake, T., Kloft, M., 2019. Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, pp. 4061–4071. http://dx.doi.org/10.18653/v1/P19-1398.

Salminen, J., Kandpal, C., Kamel, A.M., gyo Jung, S., Jansen, B.J., 2022. Creating and detecting fake reviews of online products. J. Retail. Consum. Serv. 64, 102771. http://dx.doi.org/10.1016/j.jretconser.2021.102771.

Schneider, P., Xhafa, F., 2022. Chapter 9 - anomaly detection, classification and CEP with ML methods: Machine learning pipeline for medicine. In: Schneider, P., Xhafa, F. (Eds.), Anomaly Detection and Complex Event Processing over IoT Data Streams. Academic Press, pp. 193–233. http://dx.doi.org/10.1016/B978-0-12-823818-9.00020-1.

Seo, S., Seo, D., Jang, M., Jeong, J., Kang, P., 2020. Unusual customer response identification and visualization based on text mining and anomaly detection. Expert Syst. Appl. 144, 113111. http://dx.doi.org/10.1016/j.eswa.2019.113111.

Song, B., Suh, Y., 2019. Narrative texts-based anomaly detection using accident report documents: The case of chemical process safety. J. Loss Prev. Process Ind. 57, 47–54. http://dx.doi.org/10.1016/j.jlp.2018.08.010.

Song, K., Tan, X., Qin, T., Lu, J., Liu, T.-Y., 2020. Mpnet: Masked and permuted pre-training for language understanding. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA.

Tabinda Kokab, S., Asghar, S., Naz, S., 2022. Transformer-based deep learning models for the sentiment analysis of social media data. Array 14, 100157. http://dx.doi.org/10.1016/j.array.2022.100157.

Tripadvisor, 2021. Tripadvisor review transparency report 2021. https://www.tripadvisor.com/TransparencyReport2021.

Truong, H.T., Ta, B.P., Le, Q.A., Nguyen, D.M., Le, C.T., Nguyen, H.X., Do, H.T., Nguyen, H.T., Tran, K.P., 2022. Light-weight federated learning-based anomaly detection for time-series data in industrial control systems. Comput. Ind. 140, 103692. http://dx.doi.org/10.1016/j.compind.2022.103692.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Vidanagama, D., Silva, A., Karunananda, A., 2022. Ontology based sentiment analysis for fake review detection. Expert Syst. Appl. 206, 117869.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., Bottou, L., 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.. J. Mach. Learn. Res. 11 (12).

von Helversen, B., Abramczuk, K., Kopeć, W., Nielek, R., 2018. Influence of consumer reviews on online purchasing decisions in older and younger adults. Decis. Support Syst. 113, 1–10. http://dx.doi.org/10.1016/j.dss.2018.05.006.

Wang, Y., Wong, J., Miner, A., 2004. Anomaly intrusion detection using one class SVM. In: Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004. pp. 358–364. http://dx.doi.org/10.1109/IAW.2004.1437839.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V., 2019. Xlnet: Generalized autoregressive pretraining for language understanding. Adv. Neural Inf. Process. Syst. 32.