ReX: A Framework for Incorporating Temporal Information in Model-Agnostic Local Explanation Techniques

Junhao Liu, Xin Zhang*

Key Lab of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing, China
School of Computer Science, Peking University, Beijing, China
liujunhao@pku.edu.cn, xin@pku.edu.cn

Abstract

Existing local model-agnostic explanation techniques are ineffective for machine learning models that consider inputs of variable lengths, as they do not consider temporal information embedded in these models. To address this limitation, we propose REX, a general framework for incorporating temporal information in these techniques. Our key insight is that these techniques typically learn a model surrogate by sampling model inputs and outputs, and we can incorporate temporal information in a uniform way by only changing the sampling process and the surrogate features. We instantiate our approach on three popular explanation techniques: Anchors, LIME, and Kernel SHAP. To evaluate the effectiveness of REX, we apply our approach to six models in three different tasks. Our evaluation results demonstrate that our approach 1) significantly improves the fidelity of explanations, making model-agnostic techniques outperform a state-of-the-art model-specific technique on its target model, and 2) helps end users better understand the models' behaviors.

Extended version — https://arxiv.org/abs/2209.03798

1 Introduction

As more critical applications employ machine learning models, how to explain the rationales behind these models has emerged as an important problem. Such explanations allow end users to 1) judge whether the results are trustworthy (Ribeiro, Singh, and Guestrin 2016; Doshi-Velez et al. 2017) and 2) understand knowledge embedded in the models, so they can use the knowledge to manipulate future events (Poyiadzi et al. 2020; Prosperi et al. 2020; Zhang, Solar-Lezama, and Singh 2018). This paper focuses on the problem of explaining deep models processing sequential data of variable lengths, such as Recurrent Neural Networks (RNNs) and Transformers (Vaswani et al. 2017; Wolf et al. 2020) including large language models (LLMs) (Touvron et al. 2023; Achiam et al. 2023).

To faithfully describe the behaviors of these models, it is important to consider the effect of temporal information, as the models care about not only the values of features but also their positions when making decisions. Unfortunately, existing techniques either consider temporal information but fail to produce faithful explanations that are understandable, or do not consider it at all and therefore produce explanations of low fidelity. All existing techniques that consider temporal information are global (Jacobsson 2005; Wang et al. 2018) (e.g. surrogate deterministic finite automatons (Omlin and Giles 1996; Weiss, Goldberg, and Yahav 2018; Dong et al. 2020)), which explain target models on the whole input domain (Dwivedi et al. 2023). However, faithful global explanations are complex for real-world models, which renders them hard for end users to understand, and limits their application in practice. In contrast, local techniques (Ribeiro, Singh, and Guestrin 2016, 2018; Zhang, Solar-Lezama, and Singh 2018; Arras et al. 2017; Wachter, Mittelstadt, and Russell 2017; Lundberg and Lee 2017) explain target models on a particular set of inputs (typically ones that are similar to a given input), so they can produce more tractable and understandable explanations (Zhang et al. 2021). However, none of them captures the effect of temporal information, which leads to low fidelity.

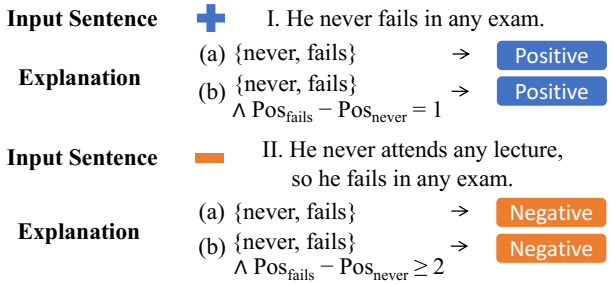

Figure 1: Example Anchors explanations (a) and REX-augmented Anchors explanations (b).

To bridge this gap, we plan to incorporate temporal information into various popular local explanations. Moreover, to ensure our method can explain a wide range of models, we focus on local model-agnostic techniques. Towards this end, we propose REX, a general framework for incorporating temporal information in various local model-agnostic explanation techniques.

We take two popular local model-agnostic explanation techniques, Anchors (Ribeiro, Singh, and Guestrin 2018) and LIME (Ribeiro, Singh, and Guestrin 2016), as examples to show how our framework improves existing techniques.

*Corresponding author.

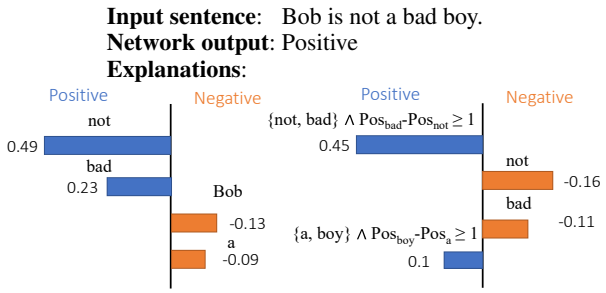

Figure 2: Example explanations generated by LIME (left) and REX-augmented LIME (right).

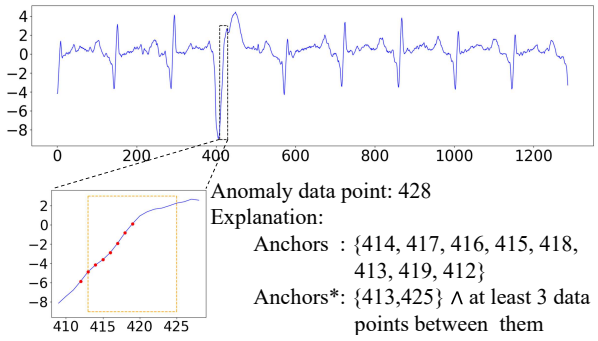

Figure 3: Anchors and REX-augmented Anchors (Denoted as Anchors*) explanations for an anomaly detection RNN. Anchors: if the values of x_{414}, x_{417}, x_{416}, x_{415}, x_{418}, x_{413}, x_{419}, and x_{412} remain unchanged, x_{428} will be classified as an anomaly. Anchors*: if the values of x_{413} and x_{425} remain unchanged, and there are at least 3 data points between them, x_{428} will be classified as an anomaly.

Figure 1 and 2 show explanations for two sentiment analysis models. Figure 1 shows the Anchors explanations (referred to as *anchors*) of an LSTM on two sentences. Anchors provides rule-based local sufficient conditions for model predictions. For Sentence I, the anchor states that *the presence of "never" and "fails" guarantees a positive prediction*. For Sentence II, the anchor is the same, but the prediction is negative. The key difference is that the words "never" and "fails" form a phrase in Sentence I, whereas they are separated in Sentence II. The anchors fail to capture the difference, and their infidelity leads to confusing results. Explanations with temporal information address this issue. REX-augmented explanations use Pos_w to denote the position of a word w in the sentence, i.e., w is the Pos_w-th word in the sequence. For Sentence I, the REX-augmented anchor states that *the presence of "never" and "fails" with "never" right before "fails" guarantees a positive prediction*. For Sentence II, the REX-augmented explanation is *the presence of "never" and "fails" with "never"* ***not*** *right before "fails" guarantees a negative prediction*. The REX-augmented anchors faithfully capture the behaviors of the LSTM.

Similar issues exist in LIME, which provides feature attributions. Figure 2 shows a LIME explanation of a BERT-based sentiment analysis model (Devlin et al. 2018) on a sentence. LIME assigns high positive scores to the words "not" and "bad". It indicates that either "not" or "bad" can make the sentence more positive, which is unfaithful to the BERT model. In this way, users can predict "Bob is a bad boy" as a positive sentence, as the word "bad" should have a strong positive effect. However, the model prediction is negative. The key is that "not bad" together is a positive phrase, whereas "not" or "bad" alone is a negative word. Incorporating temporal information in LIME addresses this issue. REX-augmented LIME gives *"not" before "bad"* the highest positive score, whereas *"not"* and *"bad"* both get negative scores, which 1) associates the two words, and 2) captures that "not" comes before "bad."

Figure 3 shows another example of explaining an anomaly detection RNN that takes a time series data $x = x_1x_2...x_n$ as input. After reading $x_1x_2...x_i$, the RNN outputs a binary label y_i to indicate whether x_i is an anomaly. An anchor is that *the anomaly is detected because of the presence of several separated data points*. The REX-augmented anchor states that *the anomaly is detected because of the presence of data points 413 and 425 with at least 3 points between them.* The REX-augmented explanation is more faithful and reveals more meaningful information to end users.

The preceding examples indicate that existing local explanations can be unfaithful and confusing for model processing inputs of variable lengths, because existing techniques only use the values of **features** (e.g. words for text data, data points for time-series data) as the components to build explanations. To address this issue, REX adds temporal information by showing the effect of the absolute positions of features and the relative positions between features. The examples show that incorporating temporal information improves the fidelity, making users better understand the behaviors of the models.

To incorporate temporal information into various local model-agnostic explanation techniques in a uniform and lightweight manner, we have made two key observations: 1) these techniques use a perturbation model to generate samples that are similar to the original input, and capture the local behavior of the model via these samples; 2) these techniques use feature predicates as the basic language components to construct explanations. Therefore, by only extending the perturbation model and the language components of explanations, REX enables these techniques to incorporate temporal information automatically without changing their core algorithms.

We demonstrate the effectiveness of REX by evaluating the explanations of an LSTM, four transformer models (BERT, T5 (Raffel et al. 2020), GPT-2 (Radford et al. 2019), Llama 2 (Touvron et al. 2023)) on a sentiment analysis task, an RNN on an anomaly detection task, and Llama 2 on a text generation task, after applying REX to Anchors, LIME, and Kernel SHAP (Lundberg and Lee 2017). On average, REX helps improve the fidelity of Anchors, LIME, and Kernel SHAP explanations by 218.7%, 41.2%, and 36.0% respectively. Moreover, REX-augmented LIME and SHAP outperform DecompX (Modarressi et al. 2023), a state-of-the-art explanation method for text models on its target model. We

also run a user study, which shows that REX helps end users better understand and predict the behaviors of target models.

2 Preliminaries

In this section, we introduce the background of our approach. Without loss of generality, we assume the target model is a black-box function from a sequence of real numbers to a real number, $f : \mathbb{R}^* \to \mathbb{R}$, where $\mathbb{R}^* = \bigcup_{T\in\mathbb{N}} \mathbb{R}^T$. For an input $x = (x_1, x_2, \ldots, x_n)$, let $|x|$ denote the length of x, and x_i represent the i-th feature value of x. We limit our discussion to classifiers and regressors.

Given an input $x \in \mathbb{R}^*$ and a model f, a local model-agnostic explanation technique t generates a local explanation $g_{f,x} := t(f, x)$. This explanation, denoted as g for simplicity, is a self-interpretable expression that describes the behaviors of f around x formed with **predicates**. Each predicate p maps an input to a binary value, i.e., $p : \mathbb{R}^* \to \{0, 1\}$.

All existing local model-agnostic explanation techniques generate explanations in a similar workflow, as shown in Figure 4, which involves three steps:

1. **Producing Predicates**: These techniques produce a set of predicates based on the input x, denoted as P.
2. **Generating Samples**: An underlying perturbation model t_{per} generates a set of samples that are similar to the input x, denoted as X_s.
3. **Learning Explanations**: These techniques learn a local explanation g consisting of predicates in P, using X_s and its corresponding model outputs $f(X_s)$.

Popular techniques such as LIME, Anchors, and Kernel SHAP all follow this workflow. Specifically, they use the same kinds of predicate sets and perturbation models. In the following, we introduce their predicate sets, perturbation models, and learning algorithms in detail.

Predicate Sets. Given an input x, and letting $[\![1, n]\!] := \{1, 2, \ldots, n\}$, a predicate set is defined as follows:

$$P := \{p_i | i \in [\![1, |x|]\!]\}. \tag{1}$$

Here, each p_i is a **feature predicate** defined by $p_i(z) := 1_{\mathrm{ran}(x,i)}(z_i)$, where $\mathrm{ran}(x, i)$ is a set containing x_i. Specifically, p_i is an indicator function that checks if the i-th feature of a sample z has a similar value to x_i (i.e., $z_i \in \mathrm{ran}(x, i)$). For example, we can use $\mathrm{ran}(x, i) = (x_i - \epsilon, x_i + \epsilon)$ for a real number x_i, and use $\mathrm{ran}(x, i) = \{x_i\}$ for a categorical value x_i.

Perturbation Models. The perturbation model t_{per} generates samples by changing the feature values of the input x. Given a parameter n, t_{per} generates n samples from the domain D defined as follows:

$$D = \{z \in \mathbb{R}^{|x|} \mid \forall i \in [\![1, |x|]\!], z_i \in \mathrm{per}(x, i)\} \tag{2}$$

where $\mathrm{per}(x, i)$ is the perturbation range of x_i, whose definition depends on the type of x_i. Specifically, $\mathrm{per}(x, i)$ contains values similar to x_i and $\mathrm{ran}(x, i) \subset \mathrm{per}(x, i)$.

Learning Algorithms and Explanations. This step is to learn an understandable expression g. In Anchors, g is a conjunction of predicates that provides a sufficient condition for f to produce the same output as $f(x)$, i.e., $f(z) = f(x)$ if $g(z) = 1$. Specifically, $g(z) = \bigwedge_{p\in Q} p(z)$, where Q is selected from P by a greedy algorithm based on the KL-LUCB algorithm (Kaufmann and Kalyanakrishnan 2013). In LIME and kernel SHAP, g is a linear expression that serves as a local surrogate model of f, i.e., $g(z) = \Sigma_{p\in P}\, \omega_p p(z) + \omega_0$, where ω_p is the weight of p and ω_0 is a constant. LIME and kernel SHAP use different linear regression algorithms to learn ω_p.

Due to the limitations of the preceding predicates and perturbation models, existing local explanation techniques can only capture the behavior of target models on samples of the same lengths as the original input[1] , and produce explanations with only constraints of feature values, which limits their effectiveness on models processing sequential data of different lengths.

3 The REX (tempoRal eXplanation) Framework

We propose REX to provide a general approach to incorporate temporal information in explanations, without requiring significantly modifying existing techniques. In this section, we introduce REX in three steps: 1) defining local explanations with temporal information, 2) showing how to augment existing techniques to generate these explanations, and 3) outlining the REX-augmented workflow.

Local Explanations with Temporal Information

Our key observation is that although the form of explanation expressions varies, the expressions are all built from the predicate set P. If we can use predicates that reflect temporal information to build explanations, temporal information is inherent in the explanations.

Our temporal predicates describe the temporal relationship between a set of features. We limit the number of features in a temporal predicate up to two because 1) in most cases, the temporal relationship between two features suffices to cover a large range of inputs of different lengths, and 2) humans have difficulty understanding high-dimensional information. Their definitions are as below:

Definition 1 (1-D Temporal Predicate) *Given an input x, a 1-D temporal predicate takes the form of*

$$p^{1D}_{k,d,op}(z) := \exists i \in \mathbb{Z}^+,\ (z_i \in \mathrm{ran}(x, k) \wedge i\ op\ d) \tag{3}$$

where $d \in \mathbb{Z}^+$, and op is a binary operator, like $=, \leq$, and $\geq$.

Definition 2 (2-D Temporal Predicate) *Given an input x, a 2-D temporal predicate takes the form of*

$$p^{2D}_{k,l,d,op}(z) := \exists i, j \in \mathbb{Z}^+, \\ (z_i \in \mathrm{ran}(x, k) \wedge z_j \in \mathrm{ran}(x, l) \wedge j - i\ op\ d) \tag{4}$$

where $d \in \mathbb{Z}$, and op is a binary operator.

We use 1-D temporal predicates to illustrate the effect of a single feature's absolute position, and 2-D to illustrate the

[1]When explaining NLP models, some perturbation models allow replacing a word with an empty string. This enables the explanations to cover inputs of shorter lengths to some extent.

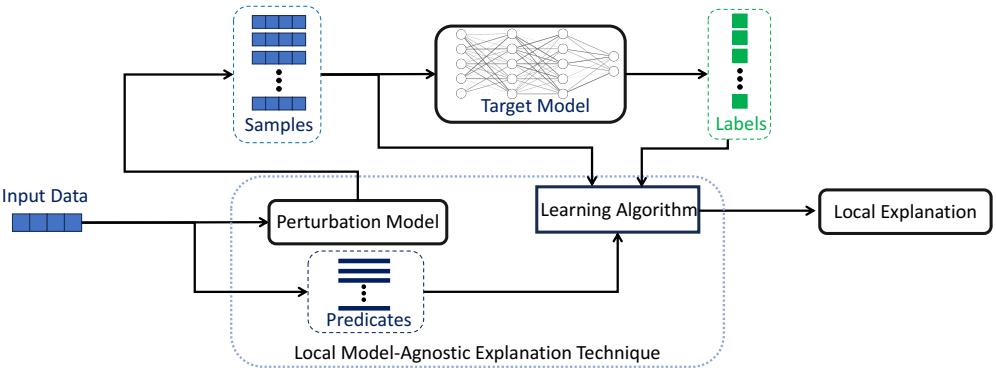

Figure 4: The workflow of generating explanations by a local model-agnostic explanation technique.

effect of the relative position between two features. Moreover, 2-D temporal predicates also apply to the case where the presence of two features together is important, but their order is not. We then introduce the definition of local explanations with temporal information:

Definition 3 (Explanation with Temporal Information) *A local explanation with temporal information is a local explanation constructed from feature predicates, 1-D temporal predicates, and 2-D temporal predicates.*

Examples. Given Input Sentence I in Figure 1, a 2-D temporal predicate is $\exists i, j \in \mathbb{Z}^+$, $(z_i =$ "$never$" $\wedge\, z_j =$ "$fails$" $\wedge\, j - i = 1)$. The REX-augmented anchor in Figure 1 is a conjunction consisting of only the preceding 2-D predicate. For another sentence "He could solve only the problem," which is judged as negative, the REX-augmented anchor is $\exists i \in \mathbb{Z}^+$, $(z_i =$ "$only$" $\wedge\, i \geq 2)$. This 1-D predicate indicates that similar sentences with "only" not at the beginning are judged as negative, such as "She only could solve the problem." However, sentences like "Only he could solve the problem" are neutral or positive.

Augmenting Generation Techniques

Consider the workflow of existing local model-agnostic techniques in Figure 4. While their core algorithms differ, they are not different from standard machine learning algorithms at a high level. To make them incorporate temporal information without modifying their underlying design, we only need to change their features and the data they learn from: To incorporate the temporal predicates into explanations, we need to extend the predicate set P; to capture the effect of temporal information, we need to extend the perturbation model t_{per}.

Extending Predicate Sets. Definition 1 and 2 provide the forms of 1-D and 2-D temporal predicates. Given an input x, we add the corresponding 1-D temporal predicates

$$P_{1D} := \{p^{1D}_{k,d,op} | k, d \in [\![1, |x|]\!] \wedge op \in \{=, >, <, \geq, \leq\}\} \tag{5}$$

and 2-D temporal predicates

$$P_{2D} := \{p^{2D}_{k,l,d,op} | k, l \in [\![1, |x|]\!] \wedge k < l \wedge d \in [\![-|x|, |x|]\!] \wedge op \in \{=, >, <, \geq, \leq\}\} \tag{6}$$

to the predicate set. For a specific usage scenario, users can further restrict the range of k,l,d, and op. For example, users can set a window w to limit $|d - k| \leq w$ for P_{1D}, and $|d| \leq w$ and $l - k \leq w$ for P_{2D}. We define the REX extended predicate set as $P^R := P \cup P_{1D} \cup P_{2D}$.

Extending Perturbation Models. To generate samples of different lengths, we add a postprocessor to the perturbation models of existing techniques. The postprocessor can generate samples of different lengths by removing or switching features. For an input x, the postprocessor does feature removal and swap on it in sequence: 1) $rf(x)$ returns a set of all subsequences of x; 2) $sf(x, i, j)$ switches the i-th and j-th features of x. We denote the REX-augmented perturbation model as t^R_{per}, and $t^R_{per}(x, n)$ takes n samples from the domain D^R defined as follows:

$$D^R = \{\text{sf}(z, i, j) \mid z \in \bigcup_{z' \in t_{\text{per}}(x)} \text{rf}(z') \wedge i, j \in [\![1, |z|]\!] \wedge i < j\}. \tag{7}$$

REX-Augmented Workflow

Compared to the vanilla workflows shown in Figure 4, the REX-augmented techniques use similar workflows, but replace the predicate set P with P^R and the perturbation model t_{per} with t^R_{per}. As a result, they can capture target models' behaviors on variable-length inputs by t^R_{per}-generated samples, and present the effect of temporal information with temporal predicates in P^R.

4 Empirical Evaluation

In this section, we demonstrate the generality of REX and its effectiveness in improving the explanation fidelity and helping end users understand and predict the behaviors of the target models through empirical evaluation. To show the generality of REX, we instantiated it on three different techniques: Anchors, LIME, and Kernel SHAP (KSHAP for short). They were applied to explain various models of two classification tasks (sentiment analysis and anomaly detection), and a text-generation task. To show the fidelity improvement by REX, we compared explanation fidelity of the REX-augmented techniques with the vanillas and a state-of-the-art model-specific technique, DecompX (Modarressi

Method	Precision (%)						Coverage (%)					
	LSTM	BERT	T5	GPT-2	Llama 2	Anom.	LSTM	BERT	T5	GPT-2	Llama 2	Anom.
Anchor	84.74	82.64	81.74	82.03	79.15	**90.40**	1.87	2.46	1.29	8.08	1.75	4.60
Anchor*	**86.48**	**84.04**	**84.04**	**82.54**	**80.20**	89.10	**10.77**	**12.07**	**11.78**	**10.26**	**10.35**	**8.70**

Table 1: Average precision and coverage of anchors and ReX-augmented anchors (denoted as anchors*) for sentiment analysis models and the anomaly detection RNN (Anom.).

.

Method	Accuracy (%)						AUROC					
	LSTM	BERT	T5	GPT-2	Llama-2	Anom.	LSTM	BERT	T5	GPT-2	Llama-2	Anom.
LIME	58.47	55.78	66.66	51.46	54.35	62.30	0.604	0.584	0.603	0.533	0.521	0.575
LIME*	**75.09**	**68.20**	**86.99**	**69.03**	**62.68**	**80.10**	**0.887**	**0.927**	**0.924**	**0.759**	**0.728**	**0.763**
KSHAP	63.64	58.02	66.24	56.06	51.06	62.90	0.613	0.593	0.590	0.578	0.536	0.557
KSHAP*	**86.10**	**83.75**	**73.62**	**69.34**	**61.81**	**77.40**	**0.879**	**0.890**	**0.909**	**0.711**	**0.680**	**0.716**
DecompX	–	60.80	–	–	–	–	–	0.601	–	–	–	–

Table 2: Average accuracy and AUROC of the explanations generated by LIME, KSHAP, their REX-augmented versions, and DecompX for sentiment analysis models and the anomaly detection RNN (Anom.).

et al. 2023). To show how much REX helps end users understand and predict the behaviors of the target model, we conducted a user study. Finally, we discuss the time efficiency of REX-augmented techniques.

Target Models, Datasets and Experimental Setup

Sentiment Analysis. Sentiment analysis models take a text sequence as input and return a binary value indicating positive or negative sentiment. We used an LSTM, BERT, GPT-2, and T5, along with Llama 2 as the target models, and used the Stanford Sentiment Treebank dataset (Socher et al. 2013) with its original train/validation/test split. We explained the target models on the test set, which contains 1821 sentences. For the text data, we set $\mathrm{ran}(x, i) = \{x_i\}$, and defined $\mathrm{per}(x, i)$ as the set of words that BERT predicts can appear in the context of x_i. In other words, all the vanilla feature predicates indicate whether the i-th word matches the i-th word in the target input and the vanilla perturbation model replaces words using BERT. As long-distance temporal information exerts little influence on models, we set a window $w = 5$, which further limits $|d-k| \leq 5$ for P_{1D}, and $|d| \leq 5$ and $l - k \leq 5$ for P_{2D}.

Anomaly Detection. Anomaly detection models take a real number sequence $x = (x_1, x_2, \ldots, x_n)$ as the input. After reading $(x_1, x_2, \ldots, x_i)$, the model outputs $y_i \in \{0, 1\}$ that indicates if x_i is an anomaly. We trained an Anomaly Detection RNN (Park 2018) on an ECG dataset (Dau et al. 2018) with its original train/validation/test split, and explained the target model on the 9 anomalous inputs in the test set. For the real number sequence data, we set $\mathrm{ran}(x, i) = \{x_i\}$, defined $\mathrm{per}(x, i)$ as the real numbers sampled from a normal distribution $\mathcal{N}(x_i, 1)$. In other words, all the non-temporal features indicate whether the i-th number matches the i-th number in the target input, and the vanilla perturbation model samples numbers from a normal distribution for each feature. For tractability, we limited the explanations to only consist of data points that are at most 20 steps before the detected anomalous point in a time series.

Text Generation. Text generation models take a text sequence as the input and return another text sequence. Anchors, LIME, and KSHAP are not originally designed for text generation models, while MExGen (Paes et al. 2024) introduces a method to adapt LIME and KSHAP to these models by converting the generation task to a regression task. We followed this approach to adapt LIME and KSHAP. We also consider two additional baselines, C-LIME and L-SHAP, which are instances of MExGen that are designed specifically for text generation models. We used LLama 2 as the target model, and explained it on 100 randomly chosen sentences from Google Natural Questions Dataset (Kwiatkowski et al. 2019). We used the same REX setting as the sentiment analysis task.

Fidelity Evaluation

Fidelity reflects how faithfully an explanation describes the target model. As Anchors provides rule-based explanations while LIME and KSHAP provide attribution-based surrogates, we employ different metrics.

Considering that anchors are rule-based sufficient conditions (Lakkaraju, Bach, and Leskovec 2016; Ribeiro, Singh, and Guestrin 2018; Craven and Shavlik 1995), we used **coverage** and **precision** as fidelity metrics. Given a target model f, an input x, their corresponding anchor g, and the distribution D from the perturbation model, we define **coverage** as $cov(x; f, g) := \mathbb{E}_{z \sim D(x)}[g(z)]$, i.e., the proportion of inputs in the perturbation domain that **match the rules**; we define **precision** as $prec(x; f, g) := \mathbb{E}_{z \sim D(x)}[\mathbf{1}_{\mathrm{f(x)=f(z)}} | g(z) = 1]$, i.e., the proportion of covered data that have **the same model output** as the original input.

For LIME and KSHAP, we use the metrics for surrogate models (Balagopalan et al. 2022; Yeh et al. 2019; Ismail, Corrada Bravo, and Feizi 2021). Given a target model f, an input x, their explanation surrogate model g, the distribution D, and a performance metric L (e.g., accuracy, area

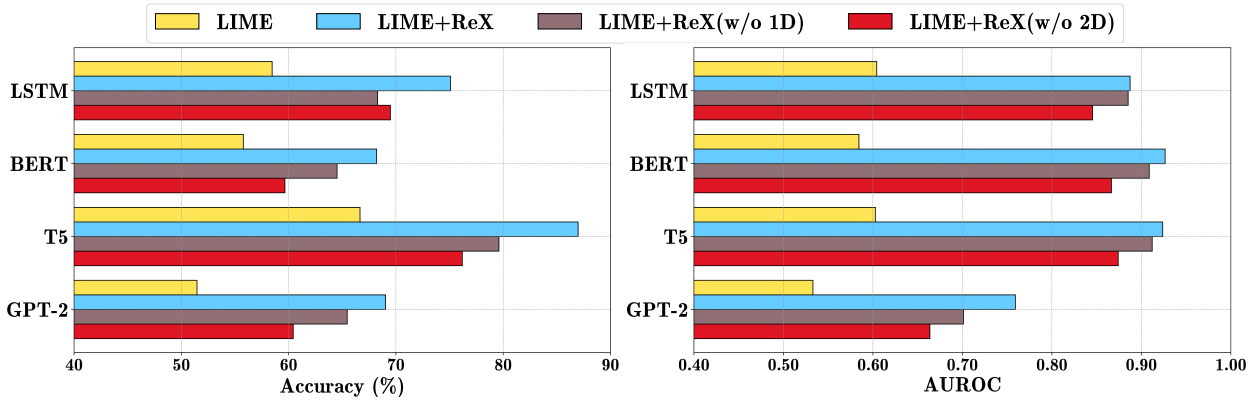

Figure 5: Average accuracy and AUROC of explanations for the four sentiment analysis models under different settings. The explanations are generated by LIME and its three augmented versions, which are augmented by REX, REX without 1D-predicates, and REX without 2D-predicates.

Method	LIME	C-LIME	LIME*	KSHAP	L-SHAP	KSHAP*
MSE	0.187	0.137	**0.054**	0.069	0.065	**0.028**

Table 3: Average mean square error (MSE) of explanations for LLama 2 on the text generation task.

under the receiver operating characteristic curve (AUROC), or mean squared error (MSE)), the (in)fidelity is defined as $\mathbb{E}_{z\sim D(x)}L(f(z),g(z))$. In our evaluation, we used accuracy and AUROC for the sentiment analysis task and anomaly detection task, and MSE for the text generation task.

Table 1 shows the fidelity of anchors and REX-augmented anchors. REX improves the average coverage by 218.7% relative to the vanilla anchors, while maintaining roughly the same level of precision or slightly improving it. Table 2 and 3 show the explanation fidelity of LIME, KSHAP, and the REX-augmented versions. On the sentiment analysis and anomaly detection task, relative to the vanillas, REX improves the average accuracy of explanations by 26.6% and 26.3%, and the AUROC by 45.8% and 38.0% respectively; on the text generation task, REX decreases the average MSE by 0.133 and 0.041 respectively, and both REX-augmented techniques outperform C-LIME and L-SHAP. For these 20 setup pairs that are only different in whether REX is applied, our **paired t-tests** indicate that with over 99% confidence, REX significantly improves the explanation fidelity.

To illustrate the effect of 1-D and 2-D predicates, we conducted an ablation study on LIME. Figure 5 shows the results, which indicate that both 1-D and 2-D temporal predicates can improve the fidelity separately, and bring about more significant improvement together.

We also compare the fidelity of REX with one of the state-of-the-art attribution-based techniques: DecompX (Modarressi et al. 2023), which is a white-box method designed for BERT. Since it is model-specific, it cannot be augmented with REX. However, by applying REX, the explanation fidelity of LIME and KSHAP, two relatively old model-agnostic techniques, surpasses that of DecompX.

Methods	$\text{Precision}_u(\%)$					$\text{Coverage}_u(\%)$				
	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5
Anchors	70.6	47.4	18.1	47.4	57.8	58.0	44.0	37.0	37.0	43.5
Anchors*	**81.2**	**99.4**	**73.9**	**84.1**	**97.7**	**61.9**	**69.5**	**80.5**	**71.5**	**60.5**

Table 4: Average Precision_u and Coverage_u of each test in the user study.

User Study

To assess how REX helps end users understand target models and predict their behaviors, we conducted a user study by comparing anchors and REX-augmented anchors (denoted as anchors*) on the preceding sentiment analysis LSTM. Similarly, we used coverage and precision as metrics, but now they describe how well a human's predictions match the model's after consuming explanations. These metrics are denoted as precision_u and coverage_u. We employed 19 CS undergraduates with machine learning backgrounds but no experience with explanation techniques, to study how much REX improves Anchors. The questionnaire contains five tests. Each test first presents a sentence, the network's output on the sentence, and their corresponding anchor and anchor*. We randomly chose the sentences from the test set. Then we asked each user to predict the RNN's output on 10 new sentences, which are produced using our perturbation model (with BERT). They could answer "positive", "negative", or "I don't know". Coverage_u for a sentence is the percentage of users that do not answer "I don't know". Precision_u for a sentence is the percentage of users that give a prediction that matches the model output among the users that do not answer "I don't know".

Table 4 shows the average coverage_u and precision_u across the 19 users and 10 sentences for each test. The anchors* outperform the anchors on all tests in terms of both coverage_u and precision_u. Among these tests, the anchors* yield an average precision_u of 87.3% and an average coverage_u of 68.8%, while the anchors yield only 48.3% and 43.9%. The relative improvements are 80.9% and 56.7%

	LSTM		BERT		T5		GPT-2		Llama 2 (senti.)		RNN (Anom.)		Llama 2 (gene.)	
		*		*		*		*		*		*		*
Anchor	**0.43**	0.76	**1.87**	1.92	11.06	**10.03**	14.05	**8.72**	612.78	**276.54**	460.30	**371.40**	–	–
LIME	**0.15**	3.98	**1.23**	5.04	**2.82**	7.41	**4.12**	8.27	**239.55**	241.76	**245.20**	263.30	**398.77**	404.08
KSHAP	**0.13**	3.97	**1.23**	5.45	**3.50**	7.93	**4.02**	8.08	**289.38**	297.90	**267.40**	291.40	**424.55**	429.61

Table 5: Average execution time (in seconds) of Anchors, LIME, and KSHAP and their REX-augmented version (denoted as "*") to explain the models in our experiments.

respectively. We did **paired t-tests** on these paired data. With more than 99% confidence, REX significantly helps users predict more instances more precisely, i.e., REX helps users better understand the target model's behavior.

Runtime Overhead

Table 5 shows the execution time of the fidelity experiments. For Anchors, REX increases the execution time of explaining the LSTM and BERT by 0.19 seconds on average, but reduces the time to explain other target models by 107.8 seconds on average; for LIME and KSHAP, REX slightly increases the average execution time by 6.87 seconds.

How REX affects the explanation time depends on the underlying explanation technique. The execution time of Anchors heavily depends on the underlying KL-LUCB algorithm. REX can often accelerate the KL-LUCB algorithm. The execution times of LIME and KSHAP equal the sum of the model's predicting time and the regression time. REX keeps the same predicting time and increases the regression time from $O(n^2|X_s|)$ to $O(n^2w^4|X_s|)$ in the sentiment analysis and text generation tasks, and from $O(20^2|X_s|)$ to $O(20^6|X_s|)$ in the anomaly detection task, where n is the input length, $|X_s|$ is the number of samples, and $w = 5$ is a small constant. For small models, such an increase is acceptable as the original explanation techniques already run fast. For large models like LLama 2, the extra overhead is negligible as the explanation time is dominated by running the model.

5 Related Work

Our work is related to explanation techniques capturing temporal information and (model-agnostic or model-specific) local explanation techniques.

Within our knowledge, existing techniques that capture temporal information provide global explanations. These techniques mainly provide DFAs (Jacobsson 2005; Weiss, Goldberg, and Yahav 2018; Wang et al. 2023) and their variants (Ayache, Eyraud, and Goudian 2018; Du et al. 2019; Dong et al. 2020) as global surrogates. However, as the complexity of practical target models increases, faithful global explanations are hard for users to understand, which limits these techniques to explaining relatively simple models.

In contrast, local explanation techniques generate explanations that are easier to understand, as they describe target models' behaviors on a subset of inputs. Existing local explanations describe model behaviors by presenting the effect of each input feature value, e.g., feature attribution (Ribeiro, Singh, and Guestrin 2016; Lundberg and Lee 2017; Strumbelj and Kononenko 2014; Tan, Tian, and Li 2023; Arras et al. 2017; Vinayavekhin et al. 2018; Arras et al. 2019; Schlegel et al. 2019; Kokalj et al. 2021; Denil, Demiraj, and de Freitas 2014; Murdoch, Liu, and Yu 2018), decision rules (Ribeiro, Singh, and Guestrin 2018; Guidotti et al. 2018), counterfactuals, (Wachter, Mittelstadt, and Russell 2017; Dandl et al. 2020; Zhang, Solar-Lezama, and Singh 2018), or visualization (Goldstein et al. 2015; Li et al. 2016; Ding et al. 2017). For models processing variable-length inputs, such explanations cannot faithfully capture models' behavior. Therefore, a few model-specific techniques consider the effect of multiple features together (Chen, Zheng, and Ji 2020; Singh, Murdoch, and Yu 2018; Sivill and Flach 2022; Tsang, Rambhatla, and Liu 2020; Nayebi et al. 2023; Ferrando, Gállego, and Costa-jussà 2022; Mohebbi et al. 2023), but these techniques are designed for specific models or domains, and still ignore temporal information, thus limiting their fidelity.

6 Limitations and Future Work

Although we have demonstrated the effectiveness of REX, there are still some limitations remaining.

Realistic Perturbation Models. The perturbation model is a key component of model-agnostic explanation techniques. However, in some domains, finding a realistic perturbation model is challenging. REX also faces this challenge. For example, the perturbation for time series data like stock prices is not clear.

Efficiency. REX increases the number of predicates. The benefits of temporal predicates reduce the running time of rule-based methods, but not for attribution-based methods. REX still slightly increases the running time of LIME and KSHAP. If we can eliminate unimportant predicates, we can further reduce the running time. We plan to address this in our future work.

7 Conclusion

In conclusion, we have proposed REX, a general framework that adds temporal information to existing local model-agnostic explanation techniques. REX allows these methods to generate more useful explanations for models that handle inputs of variable lengths (e.g., RNNs and transformers). REX achieves this by extending language components of explanations with temporal predicates, and modifying perturbation models so they can generate different-length samples. We have instantiated REX on Anchors, LIME, and Kernel SHAP, and demonstrated the effectiveness empirically.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62172017.

References

Achiam, J.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Arras, L.; Montavon, G.; Müller, K.; and Samek, W. 2017. Explaining Recurrent Neural Network Predictions in Sentiment Analysis. *CoRR*, abs/1706.07206.

Arras, L.; Osman, A.; Müller, K.; and Samek, W. 2019. Evaluating Recurrent Neural Network Explanations. In Linzen, T.; Chrupala, G.; Belinkov, Y.; and Hupkes, D., eds., *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@ACL 2019, Florence, Italy, August 1, 2019*, 113–126. Association for Computational Linguistics.

Ayache, S.; Eyraud, R.; and Goudian, N. 2018. Explaining Black Boxes on Sequential Data using Weighted Automata. In Unold, O.; Dyrka, W.; and Wieczorek, W., eds., *Proceedings of the 14th International Conference on Grammatical Inference, ICGI 2018, Wrocław, Poland, September 5-7, 2018*, volume 93 of *Proceedings of Machine Learning Research*, 81–103. PMLR.

Balagopalan, A.; Zhang, H.; Hamidieh, K.; Hartvigsen, T.; Rudzicz, F.; and Ghassemi, M. 2022. The road to explainability is paved with bias: Measuring the fairness of explanations. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 1194–1206.

Chen, H.; Zheng, G.; and Ji, Y. 2020. Generating Hierarchical Explanations on Text Classification via Feature Interaction Detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5578–5593.

Craven, M.; and Shavlik, J. 1995. Extracting tree-structured representations of trained networks. *Advances in neural information processing systems*, 8.

Dandl, S.; Molnar, C.; Binder, M.; and Bischl, B. 2020. Multi-Objective Counterfactual Explanations. In Bäck, T.; Preuss, M.; Deutz, A. H.; Wang, H.; Doerr, C.; Emmerich, M. T. M.; and Trautmann, H., eds., *Parallel Problem Solving from Nature - PPSN XVI - 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part I*, volume 12269 of *Lecture Notes in Computer Science*, 448–469. Springer.

Dau, H. A.; et al. 2018. The UCR Time Series Classification Archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

Denil, M.; Demiraj, A.; and de Freitas, N. 2014. Extraction of Salient Sentences from Labelled Documents. *CoRR*, abs/1412.6815.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, Y.; Liu, Y.; Luan, H.; and Sun, M. 2017. Visualizing and Understanding Neural Machine Translation. In Barzilay, R.; and Kan, M., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 1150–1159. Association for Computational Linguistics.

Dong, G.; Wang, J.; Sun, J.; Zhang, Y.; Wang, X.; Dai, T.; Dong, J. S.; and Wang, X. 2020. Towards Interpreting Recurrent Neural Networks through Probabilistic Abstraction. In *35th IEEE/ACM International Conference on Automated Software Engineering, ASE 2020, Melbourne, Australia, September 21-25, 2020*, 499–510. IEEE.

Doshi-Velez, F.; et al. 2017. Accountability of AI Under the Law: The Role of Explanation. *CoRR*, abs/1711.01134.

Du, X.; Xie, X.; Li, Y.; Ma, L.; Liu, Y.; and Zhao, J. 2019. DeepStellar: model-based quantitative analysis of stateful deep learning systems. In Dumas, M.; Pfahl, D.; Apel, S.; and Russo, A., eds., *Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE 2019, Tallinn, Estonia, August 26-30, 2019*, 477–487. ACM.

Dwivedi, R.; et al. 2023. Explainable AI (XAI): Core ideas, techniques, and solutions. *ACM Computing Surveys*, 55(9): 1–33.

Ferrando, J.; Gállego, G. I.; and Costa-jussà, M. R. 2022. Measuring the Mixing of Contextual Information in the Transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 8698–8714.

Goldstein, A.; Kapelner, A.; Bleich, J.; and Pitkin, E. 2015. Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics*, 24(1): 44–65.

Guidotti, R.; Monreale, A.; Ruggieri, S.; Pedreschi, D.; Turini, F.; and Giannotti, F. 2018. Local Rule-Based Explanations of Black Box Decision Systems. *CoRR*, abs/1805.10820.

Ismail, A. A.; Corrada Bravo, H.; and Feizi, S. 2021. Improving deep learning interpretability by saliency guided training. *Advances in Neural Information Processing Systems*, 34: 26726–26739.

Jacobsson, H. 2005. Rule Extraction from Recurrent Neural Networks: A Taxonomy and Review. *Neural Comput.*, 17(6): 1223–1263.

Kaufmann, E.; and Kalyanakrishnan, S. 2013. Information complexity in bandit subset selection. In *Conference on Learning Theory*, 228–251. PMLR.

Kokalj, E.; Škrlj, B.; Lavrač, N.; Pollak, S.; and Robnik-Šikonja, M. 2021. BERT meets Shapley: Extending SHAP Explanations to Transformer-based Classifiers. In Toivonen, H.; and Boggia, M., eds., *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, 16–21. Online: Association for Computational Linguistics.

Kwiatkowski, T.; et al. 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics*. (Creative Commons Share-Alike 3.0).

Lakkaraju, H.; Bach, S. H.; and Leskovec, J. 2016. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1675–1684.

Li, J.; Chen, X.; Hovy, E. H.; and Jurafsky, D. 2016. Visualizing and Understanding Neural Models in NLP. In Knight, K.; Nenkova, A.; and Rambow, O., eds., *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, 681–691. The Association for Computational Linguistics.

Lundberg, S. M.; and Lee, S. 2017. A Unified Approach to Interpreting Model Predictions. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 4765–4774.

Modarressi, A.; Fayyaz, M.; Aghazadeh, E.; Yaghoobzadeh, Y.; and Pilehvar, M. T. 2023. DecompX: Explaining Transformers Decisions by Propagating Token Decomposition. In *Proceedings of*

the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2649–2664. Toronto, Canada: Association for Computational Linguistics.

Mohebbi, H.; Zuidema, W.; Chrupała, G.; and Alishahi, A. 2023. Quantifying Context Mixing in Transformers. In *In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 3378–3400. Association for Computational Linguistics.

Murdoch, W. J.; Liu, P. J.; and Yu, B. 2018. Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Nayebi, A.; Tipirneni, S.; Reddy, C. K.; Foreman, B.; and Subbian, V. 2023. WindowSHAP: An efficient framework for explaining time-series classifiers based on Shapley values. *Journal of Biomedical Informatics*, 144: 104438.

Omlin, C. W.; and Giles, C. L. 1996. Extraction of rules from discrete-time recurrent neural networks. *Neural Networks*, 9(1): 41–52.

Paes, L. M.; et al. 2024. Multi-Level Explanations for Generative Language Models. *arXiv preprint arXiv:2403.14459*.

Park, J. 2018. RNN based Time-series Anomaly Detector Model Implemented in Pytorch. https://github.com/chickenbestlover/RNN-Time-series-Anomaly-Detection. Accessed: 2022-06-01.

Poyiadzi, R.; Sokol, K.; Santos-Rodríguez, R.; Bie, T. D.; and Flach, P. A. 2020. FACE: Feasible and Actionable Counterfactual Explanations. In Markham, A. N.; Powles, J.; Walsh, T.; and Washington, A. L., eds., *AIES '20: AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, February 7-8, 2020*, 344–350. ACM.

Prosperi, M. C. F.; Guo, Y.; Sperrin, M.; Koopman, J. S.; Min, J. S.; He, X.; Rich, S. N.; Wang, M.; Buchan, I. E.; and Bian, J. 2020. Causal inference and counterfactual prediction in machine learning for actionable healthcare. *Nat. Mach. Intell.*, 2(7): 369–375.

Radford, A.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Raffel, C.; et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In Krishnapuram, B.; Shah, M.; Smola, A. J.; Aggarwal, C. C.; Shen, D.; and Rastogi, R., eds., *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 1135–1144. ACM.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2018. Anchors: High-Precision Model-Agnostic Explanations. In McIlraith, S. A.; and Weinberger, K. Q., eds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 1527–1535. AAAI Press.

Schlegel, U.; Arnout, H.; El-Assady, M.; Oelke, D.; and Keim, D. A. 2019. Towards a rigorous evaluation of xai methods on time series. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 4197–4201. IEEE.

Singh, C.; Murdoch, W. J.; and Yu, B. 2018. Hierarchical interpretations for neural network predictions. In *International Conference on Learning Representations*.

Sivill, T.; and Flach, P. 2022. LIMESegment: Meaningful, Realistic Time Series Explanations. In *International Conference on Artificial Intelligence and Statistics*, 3418–3433. PMLR.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Strumbelj, E.; and Kononenko, I. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.*, 41(3): 647–665. (CC0 1.0 Deed).

Tan, Z.; Tian, Y.; and Li, J. 2023. GLIME: General, Stable and Local LIME Explanation. arXiv:2311.15722.

Touvron, H.; et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Tsang, M.; Rambhatla, S.; and Liu, Y. 2020. How does this interaction affect me? interpretable attribution for feature interactions. *Advances in neural information processing systems*, 33: 6147–6159.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*.

Vinayavekhin, P.; et al. 2018. Focusing on What is Relevant: Time-Series Learning and Understanding using Attention. In *2018 24th International Conference on Pattern Recognition (ICPR)*, 2624–2629. IEEE Computer Society.

Wachter, S.; Mittelstadt, B. D.; and Russell, C. 2017. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *CoRR*, abs/1711.00399.

Wang, Q.; Zhang, K.; II, A. G. O.; Xing, X.; Liu, X.; and Giles, C. L. 2018. An Empirical Evaluation of Rule Extraction from Recurrent Neural Networks. *Neural Comput.*, 30(9).

Wang, Z.; Huang, Y.; Song, D.; Ma, L.; and Zhang, T. 2023. Deepseer: Interactive rnn explanation and debugging via state abstraction. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–20.

Weiss, G.; Goldberg, Y.; and Yahav, E. 2018. Extracting Automata from Recurrent Neural Networks Using Queries and Counterexamples. In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 5244–5253. PMLR.

Wolf, T.; et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Online: Association for Computational Linguistics.

Yeh, C.-K.; Hsieh, C.-Y.; Suggala, A.; Inouye, D. I.; and Ravikumar, P. K. 2019. On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems*, 32.

Zhang, X.; Solar-Lezama, A.; and Singh, R. 2018. Interpreting Neural Network Judgments via Minimal, Stable, and Symbolic Corrections. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 4879–4890.

Zhang, Y.; Tiňo, P.; Leonardis, A.; and Tang, K. 2021. A Survey on Neural Network Interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5): 726–742. ArXiv:2012.14261 [cs].