GTX: Ground Truth eXplanation Dataset

Xiayan Ji **Anton Xue** **Rajeev Alur** **Oleg Sokolsky** **Insup Lee** **Eric Wong**
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104
{xjiae,antonxue,alur,sokolsky,lee,exwong}@seas.upenn.edu

Abstract

Evaluating the quality of explainability methods is challenging due to the lack of ground truth explanations, and often rely on hand-crafted heuristics. We curate the Ground Truth eXplanation dataset (GTX) to evaluate the alignment of feature attributions with human annotations. These annotations are carefully selected to be directly causal to the ground truth label, which provides an unambiguous goal for human-aligned explainable models. GTX is a diverse benchmark spanning multiple real-world and high dimensional data types (time-series, image, and text). In these settings, the actual explanatory features constitute only a small fraction of the entire feature space. Our analysis finds that common explanation methods overlook the ground truth explanatory features with a worryingly high false negative rate. Our dataset provides a quantitative goal for the future development of feature attribution algorithms: re-aligning explainable models with human explanations. GTX datasets and data loaders publicly available at `https://github.com/xjiae/HDDDS`.

1 Introduction

The size of modern deep networks can easily exceed millions of parameters and hidden units, making it challenging for humans to understand [1]. However, decision makers need to comprehend the model's reasoning and determine if and when they should rely on these predictions. In higher stakes settings, there are severe repercussions for naively deploying models without fully understanding its reasoning and limitations, such as in diagnosis systems in medicine [2] or legal briefs in judiciaries [3]. To provide some degree of accountability, many post-hoc techniques [4, 5, 6] have been proposed to explain the reasoning behind individual predictions of machine learning models.

One popular class of explanation techniques is feature attributions [7, 8, 9], where given an input, the objective is to assign a score for each feature as it relates to the model's prediction. Intuitively, the score of a feature is intended to measure the "importance" of said feature towards the model prediction, where larger scores indicate the feature was highly important for making the prediction. Feature attributions have applications in classic machine learning settings such as vision [10, 11], language [12, 13], and reinforcement learning [14, 15], as well as more recent use-cases in industry [16] and law [17].

However, feature attributions rarely come with formal guarantees of behavior [18]. While various metrics have been proposed to evaluate feature attributions, each metric only provides a plausible, partial view into the underlying model's behavior that need not be accurate. For instance, some metrics progressively remove features with the highest attribution scores and assessing the resulting change [19, 20]. Many of these metrics arose out of necessity because ground-truth explanations for

Submitted to the 37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks. Do not distribute.

comparison were unclear or simply not available. As a result, ensuring that feature attributions are of high quality remains a significant challenge despite their widespread usage.

Hence, our work puts forth a measurable and human-aligned target for feature attributions. In order for models to effectively assist humans, it is essential for their decision-making to be aligned with human judgment [21]. Specifically, a feature attribution for an explainable model should identify the ground truth causal features in the *data*. [1] Therefore, we seek to quantify to what degree are models and their feature attributions aligned with human judgment. If explainable models exhibit a high degree of alignment with humans, then their explanations are more *usable* as a proxy for human experts. For example, a doctor could use such a human-aligned attribution to explain a medical diagnoses in lieu of asking a specialist, freeing up the specialist to pursue more challenging tasks.

However, evaluating feature attributions with human judgment is challenging due to the absence of a ground truth explanation [22]. Several benchmarks for evaluating feature attribution methods, such as Captum [23] and OpenXAI [24], have been established using synthetic datasets where the ground truth can be carefully controlled and specified. However, there remains a need to complement these benchmarks with high-dimensional, and real-world datasets that offer a diverse and rich perspective for evaluating of feature attribution algorithms in natural settings. To this end, we have curated a set of real-world datasets that possess ground truth human annotations in domains such as industrial controls [25, 26, 27], artifact evaluation [28], and machine comprehension [29].

Our benchmark, called the Ground Truth eXplanation dataset (GTX), is specifically designed to comprehensively evaluate the human-alignment of feature attribution methods in challenging, real-world settings. Our contribution can be summarized as follows:

- We meticulously clean and process the human annotations to create the Ground Truth Explanation (GTX) dataset. The resulting benchmark spans three prominent data domains: time-series, image, and text, as depicted in Figure 1, but has a standardized and measurable human-alignment goal across all tasks.
- We establish a baseline for the alignment of common feature attribution methods and models with the human annotations using our GTX benchmark.
- In our analysis, we show that existing feature attribution algorithms have a high false negative rate and tend to overlook the true explanatory features. This misalignment highlights the need for future research to achieve more usable explanations.

The remainder of this paper is organized as follows: Section 2 provides a review of existing feature attribution methods and explores relevant XAI benchmarks and datasets. In Section 3, we introduce our dataset, highlighting its unique attributes and characteristics. To showcase potential usage of feature attribution methods on our dataset, we present baseline experiments in Section 4. Subsequently, in Section 5, we discuss the limitations of our dataset and conclude with a summary of our findings.

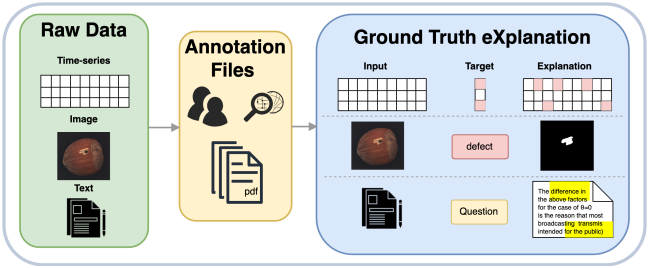


Figure 1: GTX overview. We consolidate raw data of time-series, image and text format. Then, we process the annotation files to obtain the ground truth explanations in column, pixel and clause levels.

[1] We note that this criteria (explaining the prediction from patterns in the data) differs from explaining predictions from patterns learned in the model. In the latter, the goal is different—these explanations aim to uncover properties learned in the model which need not be aligned with the ground truth or be usable for humans.

2 Related Work

2.1 XAI methods on Feature Attribution

The existing literature encompasses various taxonomies of Explainable Artificial Intelligence (XAI) methods, each tailored to address specific problems and aspects. In this study, our primary emphasis is on the post-hoc method branch, with a specific focus on feature attribution [5, 7] or feature relevance explanation [6]. Feature attribution refers to the process of determining the importance or contribution of individual features within a dataset or input data to the predictions or output of a machine learning model [7, 8, 9, 10, 30, 31, 32]. Specifically, we reproduce the code of two standard explanation methods for evaluation: Vanilla Gradient [31] and Integrated Gradients [32].

2.2 XAI Benchmarks and Datasets

Many open-source library implement a handful of feature attribution algorithms, for example, Captum [33] and SHAP bechmarks [34]. However, they do not perform ground-truth based evaluation. Several studies acknowledge the limitation of XAI due to the absence of ground-truth for evaluating explanations [20, 35, 36]. To address this issue, researchers have started introducing ground truth annotations to assess XAI methods. For instance, Amiri et al. [37] propose the use of canonical equations as representations of explanations for evaluating their accuracy. Furthermore, Arras et al. [38] introduce CLEVR-XA, a visual question answering dataset designed specifically for evaluating neural network explanations in computer vision tasks. OpenXAI [24] offers a transparent evaluation of post hoc model explanations using tabular data and perform faithfulness evaluation with ground truth of the synthetic data. Different from previous works, we present five real-world datasets of different data types, i.e., time-series, image, and text, that can be used to perform evaluation based on human-annotated ground truth explanations.

3 Ground Truth eXplanation Dataset

In our GTX dataset we consolidate three common types of data in time-series, image, and text, for evaluating feature attribution methods. For time-series data the features correspond to periodic samples of a plant state; for image data the features correspond to pixels of the image; for text data the features correspond to tokens of the text. By analyzing the importance of these features, we can gain insights into the decision-making process of the model from the input level.

3.1 Time-series

The time-series component of GTX consists of three different datasets from various industrial control settings involving real-world or simulated plants. All three datasets were generated by sampling the plant at a fixed frequency, where for each datapoint the feature values denote either a sensor reading or a controller output.

Hardware-In-the-Loop-based Augmented ICS Security Dataset (HAI) [25]: The HAI dataset was collected from a realistic industrial control system (ICS) testbed, augmented with a Hardware-In-the-Loop (HIL) simulator for 379.3 hours. The HIL simulator emulates two crucial components of the power generation domain: steam-turbine power generation and pumped-storage hydropower generation, with a total of $m = 86$ features.

Secure Water Treatment Dataset (SWaT) [26]: The Secure Water Treatment testbed serves as a scaled-down replica of a real-world industrial water treatment plant. It operates at a reduced capacity, producing five gallons per minute of water for over 11 days. The treatment process involves the utilization of membrane-based ultrafiltration and reverse osmosis units for effective water filtration, comprising of $m = 51$ features in total.

Water Distribution Dataset (WADI) [27]: WADI is an extension of the SWaT testbed featuring additional components and functionalities such as chemical dosing systems, booster pumps and valves, as well as instrumentation and analyzers. It is collected over 16 days with $m = 127$ dimensions.

At various time points a cyber attack (e.g. altering sensor readings) or a physical attack (e.g. altering water flow) is performed, which allows one to obtain a ground truth of whether a plant state is to be considered "normal" or "attacked". The attacks target specific sets of equipment and have precise start and end times, allowing us to obtain accurate ground truth explanations for the features during the attacks. We manually process annotation files in both PDF and Excel formats, which contain the start and end times of each attack. These annotations are then aligned with the timestamps of the raw data records.

Each dataset is a sequence of periodic samples $(x_1, y_1), \ldots, (x_T, y_T)$ where at each time step $t = 1, \ldots, T$ the observations $x_t \in \mathbb{R}^m$ denotes plant state while the label $y_t \in \{0, 1\}$ is an indicator of whether the plant was attacked ($y_t = 1$) or if the behavior is normal ($y_t = 0$). We use a binary mask $a_t \in \{0, 1\}^m$ as the ground truth explanation to denote which input feature is explanatory of an attack. If $y_t = 1$, then $(a_t)^i = 1$ implies that feautre i is involved in attack at time t — which we know from the annotations supplied with the original datasets. If $y_t = 0$, then we write $a_t = 0$, the zeros vector, to mean that no attack occurred. In summary, our dataset loader provides three key objects at each time step: the plant features x_t, the attack indicator y_t, and the explanation a_t.

To facilitate the use of our dataset with machine learning models in PyTorch [39], we wrap the raw data using the `torch.utils.data.Dataset` class. Below is an example code snippet demonstrating its usage with the HAI dataset:

```
bundle = get_data_bundle("hai", window_size=100, train_batch_size=32)
train_dataloader = bundle["train_dataloader"]
x, y, a = next(iter(train_dataloader))
# x.shape==(32,100,86), y.shape==(32,), a.shape==(32,100,86)
```

3.2 Image

MVTec-AD [28] is an industrial inspection dataset designed for benchmarking defects detection methods. It consists of a 15 categories with a total of more than 5000 high-resolution ($3 \times 1024 \times 1024$) images. Each category includes a set of defect-free training images and a test set containing images with different types of defects, as well as defect-free images. The dataset provides pixel-accurate ground truth annotations for the defect regions, which have been carefully annotated and reviewed by the authors to align with human interpretation of real-world defects.

We allow the user to specify an input size $d \leq 1024$ to down-sample an image to $m = 3 \times d \times d$ features, i.e. 3 color channels with a side-length of d pixels. Each image is correspondingly labeled with whether it has a defect ($y = 1$) or not ($y = 0$). The ground truth explanation is a bitmask $a \in \{0, 1\}^{d \times d}$ denoting which positions are defects; if $(a)^{ij} = 1$, then this means that the pixel at position (i, j) is part of the defect. If $y = 0$, then $a = 0$, the zeros matrix, indicating no defects. The objects returned by the dataset are the down-sampled image x, the defect label y, and the ground truth explanation a.

As there are 15 image categories in total, MVTec is in fact a collection of 15 different datasets. We implement the MVTec dataset with `torch.utils.data.Dataset`, and showcase its use below.

```
bundle = get_data_bundle("mvtec", input_size=256, train_batch_size=32)
train_dataloader = bundle["train_dataloader"]
x, y, a = next(iter(train_dataloader))
# x.shape==(32,3,256,256), y.shape==(32,), a.shape==(32,1,256,256)
```

Here `hazelnut` is one of 15 admissible classes among:

```
bottle, cable, capsule, carpet, grid, hazelnut, leather, metal_nut,
    pill, screw, tile, toothbrush, transistor, wood, zipper
```

Here `input_size` is the dimension d to which we downsample, and the last flag of `is_train=True` selects only images that are non-defect; if `is_train=False` then the selection is mixed.

3.3 Text

SQuAD (Stanford Question Answering Dataset) [29] is a widely used reading comprehension dataset that includes 107,785 question-answer pairs based on 536 Wikipedia articles. The dataset was generated by crowdworkers who formulated questions and provided specific text segments or spans as answers. The answers have undergone rigorous crowdworkers selection, additional answer collection, and manual crosscheck processes, making them reliable ground truth explanations for the corresponding questions.

The questions are concatenated with a context, such that model inputs have the form $x = (x_q, x_c)$, where x_q are the question tokens and x_c are the context tokens. The output of a question-answering model is to identify a range of indices to highlight in the conjoined input x that constitutes as the answer. As such, the output of a model on SQuAD is not a binary value as in the time-series and image data, but instead a start-index and an end-index that denotes which tokens to highlight. For a particular x in the dataset, the ground truth then consists of a pairing $a = (a_s, a_e)$, where a_s, a_e are integers that denote the highlight start and end indices, respectively, for x.

We implemented the SQuAD dataset with `torch.utils.data.Dataset` as follows, where we demonstrate tokenization with the RoBERTa [40] base tokenizer to sequence lengths of 384.

```
# Use the "roberta-base" tokenizer from Hugging Face
bundle = get_data_bundle("squad", tokenizer_or_name="roberta-base",
                                  train_batch_size=32)
train_dataloader = bundle["train_dataloader"]
input_ids, attn_mask, token_type_ids, start_pos, end_pos = next(iter(
                                  train_dataloader))
# input_ids.shape==(32,384), the default token sequence length
```

Each item within a SQuAD dataset contains a number of information relevant for a language-model transformer, among them: `item[0]` corresponds to x, which we emphasize is the concatenation of the question tokens and the context tokens; `item[3]` and `item[4]` correspond to the start and end position indices, respectively. We use the defaults supplied with `tensorflow_datasets` and `transformers` to determine the train-test split.

3.4 Dataset Statistics

In Table 1, we provide a summary of key statistics pertaining to the datasets. This includes information on the feature dimensions, the number of positive instances representing attacks or defects, the number of negative instances, and the corresponding positive ratio. Notably, the class distribution exhibits an imbalance, signifying a discrepancy in the distribution between negative and positive instances. We use "positive" to denote data for which $y = 1$, and "negative" to denote data for which $y = 0$.

However, it is important to note that the SQuAD dataset presents unique characteristics that distinguish it from the other datasets. The variable lengths of paragraphs contribute to the variability in feature dimensions. Additionally, the SQuAD dataset does not differentiate between positive and negative instances, making it unsuitable for inclusion in Table 1 for comparative purposes.

Data	Features Dimension	Positive Count	Negative Count	Total Count	Positive Ratio
HAI	86	12,030	1,353,572	1,365,602	0.88%
SWaT	51	54,621	892,098	946,719	5.77%
WADI	127	5,134	1,377,268	1,382,402	0.37%
MVTec	$3d^2$	1256	4094	5350	23.48%

Table 1: Basic statistics of HAI, SWaT, WADI, and MVTec. When $y = 1$ we say that the datapoint is positive; when $y = 0$ we say that it is negative.

Furthermore, we present several key statistics of the human annotations within our dataset. Specifically, we report the total number of human annotations conducted, the average number of explanatory features per input, and the ratio of explanatory features to the entire feature space. This ratio serves as a crucial metric for assessing the class imbalance between explanatory and non-explanatory features in the annotations. For the SQuAD dataset, we gather and estimate the summary statistics based on information provided in their papers [29, 41].

The presence of class imbalance poses significant challenges for feature attribution methods, as they aim to accurately identify and attribute the importance of each feature in the prediction process. When the number of explanatory features is significantly lower than the number of non-explanatory features, it can lead to biased attributions and potentially misleading interpretations of the model's behavior.

Data	Annotation Count	Average Count	Explanatory Ratio
HAI	1,034,580	1.00	1.17%
SWaT	2,785,671	1.07	2.10%
WADI	652,018	1.93	1.52%
MVTec	1,317,011,456	45950.49	4.38%
SQuAD	107,785	4.64	3.10%

Table 2: Feature statistics of HAI, SWaT, WADI, MVTec and SQuAD.

3.5 Task Definition

We formulate the task as predicting which features of an input $x \in \mathcal{X}, \mathcal{X} \subseteq \mathbb{R}^m$ related to the target $y \in \mathcal{Y}$. Specifically, a feature attribution model $A : \mathcal{X} \to [0,1]^m$ maps an input x to an m-dimensional vector $\hat{a} = \hat{A}(x) \in [0,1]^m$, where each element is a score representing the degree that the corresponding feature is explanatory of y. For each input x and target y, our dataset has a m-bit vector that encodes the ground truth annotation function $A : \mathcal{X} \to \{0,1\}^m$. It maps each input x to the human annotation $a = A(x) \in \{0,1\}^m$. By comparing $\hat{a}$ and a, we can directly evaluate the performance of the feature attribution model.

4 Experiments

4.1 Predictive models

In our experiments involving time-series data, we utilize the standard implementation of well-established logistic regression model (LR) [42] and Long Short-term Memory networks (LSTM). In our analysis of the MVTec dataset, we utilize Fastflow [43], a CNN that employs ResNet18 as its backbone for image feature extraction. The SQuAD dataset is processed using the widely-used RoBERTa model [40]. These models are chosen to facilitate the application of diverse feature attribution methods.

4.2 Feature Attribution methods

In our study, we employ several widely used feature attribution techniques. Specifically, we apply the vanilla gradient (GRAD) [31] and integrated gradient (INTG) [32] to evalute the feature attribution performance. The objective of this analysis is to assess the importance of individual features in the decision-making process of the machine learning models. To quantify the effectiveness of the feature attribution techniques in a binary manner, we establish a threshold on the feature attribution scores. This threshold is determined by maximizing the F1-score, a widely utilized metric that balances precision and recall.

4.3 Training

In our time-series data study, we employed two distinct training strategies. Firstly, we randomly divided 70% of the dataset as the training set and train the Logistic Regression (LR) model. Secondly,

we preserved the temporal order by training a three-layered Long Short-Term Memory (LSTM) model on the sliding window version of the dataset. The training ratio of 70% was maintained, and a window size of 100 with a stride of one was used. For the image data, the FastFlow model was trained on 70% of the MVTec dataset. We sampled both negative and positive instances for the time-series and image training sets. The text data was trained using the RoBERTa model on the default training set, which accounted for 82% of the SQuAD dataset.

All models were trained for five epochs with a learning rate of 10^{-6}. The model with the best validation accuracy was selected. All experiments are run with 4 Nvidia 2080Ti GPU, 80 vCPUs, a processor Intel(R) % Xeon(R) Gold 6148 @ 2.4 GHz and 768GiB of RAM. Further details on the model architecture can be found in the Appendix. It is important to note that hyperparameter tuning was not the main focus of our study, as our primary objective was to showcase the utility of our dataset.

4.4 Metrics

By employing thresholds that optimize the F1-score on feature attribution, we obtain binary predictions for individual features. In this context, we consider a prediction of '1' a positive outcome (explanatory), while '0' denotes a negative outcome (not explanatory). Let $a = A(x) = \{a^i \mid i = 1, \ldots, m\} \in \{0,1\}^m$ represent the ground truth annotation for $x \in \mathbb{R}^m$. Let $\hat{a} = \hat{A}(x) = \{\hat{a}^i \mid i = 1, \ldots, m\} \in \{0,1\}^m$ denote the prediction generated by the attribution models and the threshold. We then concatenate all the features of all the inputs and compute several evaluation metrics, including False Positive Rate (FPR), False Negative Rate (FNR), Accuracy (ACC), and F1-score. The computation formulas for these metrics are presented below:

$$\text{FPR} = \frac{\{\hat{a}^i = 1 \mid a^i = 0\}}{\{a^i = 0\}}, \ \text{FNR} = \frac{\{\hat{a}^i = 0 \mid a^i = 1\}}{\{a^i = 1\}} \tag{1}$$

$$\text{ACC} = \frac{\{\hat{a}^i = 0 \mid a^i = 0\} + \{\hat{a}^i = 1 \mid a^i = 1\}}{\{a^i = 0\} + \{a^i = 1\}} \tag{2}$$

$$\text{F1-score} = \frac{2\{\hat{a}^i = 1 \mid a^i = 1\}}{2\{\hat{a}^i = 1 \mid a^i = 1\} + \{\hat{a}^i = 1 \mid a^i = 0\} + \{\hat{a}^i = 0 \mid a^i = 1\}}. \tag{3}$$

4.5 Results

We randomly sampled 100 instances on individual datasets and conducted the experiments using 20 random seeds. The obtained results were then analyzed by reporting the mean and standard error metric.

Time-series The results obtained from our experiments on time-series data have revealed that GRAD and INTG miss-classify the explanatory feature as non-explanatory, which leads to a higher FNR (+42.23%) than FPR on average, as shown in Table 3, 4, 5. In addition, we observe that GRAD has a better performance in general, with higher accuracy (+17.71%) and F1-score (+11.10%) performance than that of the INTG. This presents a serious problem as it undermines the reliability

Models	Attribution	FPR	FNR	ACC	F1-score
LR	GRAD	0.02 ± 0.02	0.89 ± 0.09	0.97 ± 0.02	0.98 ± 0.01
	INTG	0.49 ± 0.03	0.99 ± 0.01	0.51 ± 0.03	0.67 ± 0.03
LSTM	GRAD	0.06 ± 0.10	$\mathbf{0.87 \pm 0.19}$	0.94 ± 0.10	0.96 ± 0.06
	INTG	$\mathbf{0.01 \pm 0.00}$	0.89 ± 0.02	$\mathbf{0.98 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$

Table 3: Results for HAI dataset.

and effectiveness of the attribution methods in correctly identifying the features that contribute to the model's decision-making process. Consequently, it highlights the need for further improvement and development of attribution algorithms to address this challenge and enhance their capability to accurately identify and attribute the explanatory features.

Models	Attribution	FPR	FNR	ACC	F1-score
LR	GRAD	**0.03 ± 0.00**	**0.50 ± 0.05**	**0.96 ± 0.00**	**0.97 ± 0.00**
	INTG	0.52 ± 0.03	0.90 ± 0.03	0.48 ± 0.03	0.64 ± 0.03
LSTM	GRAD	0.31 ± 0.36	0.66 ± 0.35	0.69 ± 0.36	0.73 ± 0.37
	INTG	0.53 ± 0.01	0.66 ± 0.04	0.47 ± 0.01	0.63 ± 0.01

Table 4: Results for SWaT dataset.

Models	Attribution	FPR	FNR	ACC	F1-score
LR	GRAD	0.58 ± 0.07	**0.28 ± 0.07**	0.42 ± 0.07	0.58 ± 0.07
	INTG	0.50 ± 0.03	0.94 ± 0.03	0.49 ± 0.03	0.66 ± 0.03
LSTM	GRAD	**0.31 ± 0.11**	0.46 ± 0.14	**0.69 ± 0.11**	**0.80 ± 0.09**
	INTG	0.33 ± 0.24	0.71 ± 0.20	0.67 ± 0.24	0.77 ± 0.16

Table 5: Results for WADI dataset.

Image For the image data, Table 11 demonstrates that a higher FNR (+69.97%) than FPR is also observed, suggesting that the attribution methods fail to capture all the explanatory pixels. Different from the time-series dataset, INTG has a better performance than GRAD method, with a higher accuracy (+3.29%) and F1-score (+2.15%).

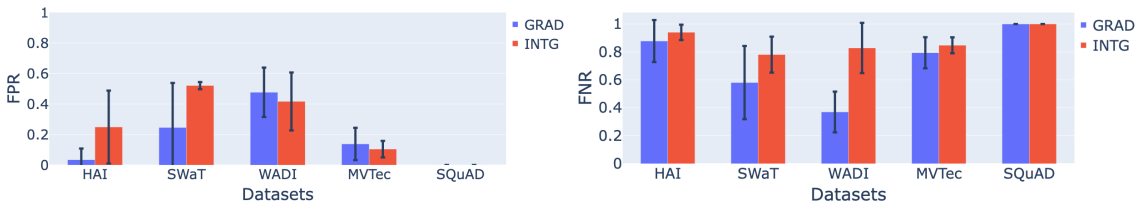
Models	Attribution	FPR	FNR	ACC	F1-score
FastFlow	GRAD	0.14 ± 0.11	**0.79 ± 0.11**	0.86 ± 0.10	0.91 ± 0.07
	INTG	**0.10 ± 0.05**	0.85 ± 0.06	**0.89 ± 0.05**	**0.93 ± 0.03**

Table 6: Results for MVTec dataset.

Text The result of the SQuAD dataset can be found in Table 12. As with previous datasets, a notable observation is the presence of a high FNR compared with the image dataset. The same result for INTG and GRAD could be due to the similar gradient computations by the RoBERTa model. However, the results may vary depending on the predictive model architecture, and the complexity of the explanation task in practice.

Overall, the GRAD method demonstrates a slightly better performance than INTG, exhibiting a higher accuracy (+9.97%) and F1-score (+6.23%) on average. In Figure 2, we present a comparative analysis of average FPRs and FNRs for different attribution methods and datasets. The figure highlights that SQuAD exhibits the highest FNR while WADI showcases the highest FPR on average.

More figures and results on other attribution methods (e.g. SHAP [7] and LIME [8]) can be found in the Appendix. Our experiments are illustrative in nature, running with different machines or configurations may yield slightly different results. However, the overall trends and patterns observed in the data should remain similar and consistent.



(a) The average False Negative Rate (FPR).

(b) The average False Negative Rate (FNR)

Figure 2: Average error rates comparison for INTG and GRAD across all datasets.

Models	Attribution	FPR	FNR	ACC	F1-score
RoBERTa	GRAD	0.00 ± 0.00	1.00 ± 0.00	0.87 ± 0.00	0.81 ± 0.00
	INTG	0.00 ± 0.00	1.00 ± 0.00	0.87 ± 0.00	0.81 ± 0.00

Table 7: Results for SQuAD dataset.

5 Conclusion

One limitation of our dataset is the absence of real-world graph data. However, we are actively searching and we will update our repository once we find suitable datasets. In summary, our GTX dataset includes time-series, image, and text data, along with detailed feature-wise ground truth explanations. We have established a baseline for aligning common feature attribution algorithms with human annotation of the actual explanatory features, which takes up a relatively small proportion in real-world datasets. Our experiments have revealed a significant challenge posed by a higher FNR than FPR in existing feature attribution methods, emphasizing the need for improvements to accurately identify the true explanatory features. With its comprehensive collection and diverse data types, our dataset is a valuable resource for the XAI community, facilitating quantitative evaluation and advancements in feature attribution algorithms.

References

[1] Zhengxuan Wu, Atticus Geiger, Christopher Potts, and Noah D Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. *arXiv preprint arXiv:2305.08809*, 2023.

[2] Julia Amann, Alessandro Blasimme, Effy Vayena, Dietmar Frey, Vince I Madai, and Precise4Q Consortium. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. *BMC medical informatics and decision making*, 20:1–9, 2020.

[3] Ashley Deeks. The judicial demand for explainable artificial intelligence. *Columbia Law Review*, 119(7):1829–1850, 2019.

[4] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.

[5] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020.

[6] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

[7] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[8] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

[9] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[10] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise, 2017.

[11] Vanessa Buhrmester, David Münch, and Michael Arens. Analysis of explainers of black box deep neural networks for computer vision: A survey. *Machine Learning and Knowledge Extraction*, 3(4):966–989, 2021.

[12] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92, 2014.

[13] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. A survey of the state of explainable ai for natural language processing. *arXiv preprint arXiv:2010.00711*, 2020.

[14] Nikaash Puri, Sukriti Verma, Piyush Gupta, Dhruv Kayastha, Shripad Deshmukh, Balaji Krishnamurthy, and Sameer Singh. Explain your move: Understanding agent actions using specific and relevant feature attribution. *arXiv preprint arXiv:1912.12191*, 2019.

[15] Lei He, Nabil Aouf, and Bifeng Song. Explainable deep reinforcement learning for uav autonomous path planning. *Aerospace science and technology*, 118:107052, 2021.

[16] Wenqian Liu, Runze Li, Meng Zheng, Srikrishna Karanam, Ziyan Wu, Bir Bhanu, Richard J. Radke, and Octavia Camps. Towards visually explaining variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[17] Katie Atkinson, Trevor Bench-Capon, and Danushka Bollegala. Explanation in ai and law: Past, present and future. *Artificial Intelligence*, 289:103387, 2020.

[18] Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *arXiv preprint arXiv:2201.08164*, 2022.

[19] Aya Abdelsalam Ismail, Mohamed Gunady, Hector Corrada Bravo, and Soheil Feizi. Benchmarking deep learning interpretability in time series predictions. *Advances in neural information processing systems*, 33:6441–6452, 2020.

[20] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, 2019.

[21] Giang Nguyen, Daeyoung Kim, and Anh Nguyen. The effectiveness of feature attribution methods and its correlation with automatic evaluation scores. *Advances in Neural Information Processing Systems*, 34:26422–26436, 2021.

[22] Yilun Zhou, Serena Booth, Marco Tulio Ribeiro, and Julie Shah. Do feature attribution methods correctly attribute features? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9623–9633, 2022.

[23] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*, 2020.

[24] Chirag Agarwal, Satyapriya Krishna, Eshika Saxena, Martin Pawelczyk, Nari Johnson, Isha Puri, Marinka Zitnik, and Himabindu Lakkaraju. Openxai: Towards a transparent evaluation of model explanations. *Advances in Neural Information Processing Systems*, 35:15784–15799, 2022.

[25] Hyeok-Ki Shin, Woomyo Lee, Jeong-Han Yun, and Byung-Gil Min. Two ics security datasets and anomaly detection contest on the hil-based augmented ics testbed. In *Cyber Security Experimentation and Test Workshop*, CSET '21, page 36–40, New York, NY, USA, 2021. Association for Computing Machinery.

[26] Aditya P Mathur and Nils Ole Tippenhauer. Swat: A water treatment testbed for research and training on ics security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*, pages 31–36. IEEE, 2016.

[27] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P Mathur. Wadi: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd international workshop on cyber-physical systems for smart water networks*, pages 25–28, 2017.

[28] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019.

[29] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250, 2016.

[30] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences, 2017.

[31] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[32] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[33] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for pytorch, 2020.

[34] Shap benchmark. `https://shap.readthedocs.io/en/latest/index.html`. [Accessed 17-May-2023].

[35] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning, 2017.

[36] Richard Tomsett, Dan Harborne, Supriyo Chakraborty, Prudhvi Gurram, and Alun Preece. Sanity checks for saliency metrics, 2019.

[37] Shideh Shams Amiri, Rosina O. Weber, Prateek Goel, Owen Brooks, Archer Gandley, Brian Kitchell, and Aaron Zehm. Data representing ground-truth explanations to evaluate xai methods, 2020.

[38] Leila Arras, Ahmed Osman, and Wojciech Samek. Clevr-xai: a benchmark dataset for the ground truth evaluation of neural network explanations. *Information Fusion*, 81:14–40, 2022.

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[40] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[41] Jason Freeman and Raine Hoover. Question answering with squad : Variations on multi-perspective context matching. 2017.

[42] sklearn logistic regression. `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`.

[43] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows, 2021.

[44] Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. Cuad: An expert-annotated nlp dataset for legal contract review. *arXiv preprint arXiv:2103.06268*, 2021.

A Appendix

A.1 Model Architecture

Three-layered LSTM model

```
 SimpleLSTM(
(lstm1):  LSTM(num_features, 128)
(lstm2):  LSTM(128, 128)
(lstm3):  LSTM(128, 128)
(linear):  Linear(128, 2))
```

We used the FastFlow with ResNet18 backbone and RoBERTa-base model. For details of model configuration, please refer to their papers [43, 40].

A.2 More Experiment Results

In this section, we present additional experimental results for the LIME method [8] and the SHAP method [7] applied to all the datasets, utilizing their respective predictive models. It is worth noting that the issue persists in both methods, whereby the false negative rate (FNR) remains greater than the false positive rate (FPR).

Models	Attribution	FPR	FNR	ACC	F1-score
LSTM	LIME	$\mathbf{0.19 \pm 0.13}$	0.74 ± 0.13	$\mathbf{0.81 \pm 0.13}$	$\mathbf{0.88 \pm 0.09}$
	SHAP	0.53 ± 0.04	$\mathbf{0.48 \pm 0.03}$	0.47 ± 0.04	0.64 ± 0.03

Table 8: Results on LIME and SHAP for HAI dataset.

Models	Attribution	FPR	FNR	ACC	F1-score
LSTM	LIME	$\mathbf{0.37 \pm 0.20}$	0.63 ± 0.19	$\mathbf{0.63 \pm 0.20}$	$\mathbf{0.75 \pm 0.14}$
	SHAP	0.53 ± 0.02	$\mathbf{0.51 \pm 0.02}$	0.47 ± 0.02	0.63 ± 0.02

Table 9: Results on LIME and SHAP for SWaT dataset.

Models	Attribution	FPR	FNR	ACC	F1-score
LSTM	LIME	0.44 ± 0.15	$\mathbf{0.55 \pm 0.15}$	0.56 ± 0.15	0.70 ± 0.10
	SHAP	$\mathbf{0.18 \pm 0.16}$	0.79 ± 0.15	$\mathbf{0.82 \pm 0.16}$	$\mathbf{0.88 \pm 0.11}$

Table 10: Results on LIME and SHAP for WADI dataset.

Models	Attribution	FPR	FNR	ACC	F1-score
FastFlow	LIME	0.28 ± 0.25	$\mathbf{0.57 \pm 0.23}$	0.72 ± 0.25	0.80 ± 0.20
	SHAP	$\mathbf{0.18 \pm 0.21}$	0.73 ± 0.18	$\mathbf{0.82 \pm 0.21}$	$\mathbf{0.87 \pm 0.18}$

Table 11: Results on LIME and SHAP for MVTec dataset.

Models	Attribution	FPR	FNR	ACC	F1-score
RoBERTa	LIME	0.01 ± 0.03	1.00 ± 0.02	0.99 ± 0.03	0.98 ± 0.01
	SHAP	$\mathbf{0.00 \pm 0.00}$	1.00 ± 0.00	0.99 ± 0.00	$\mathbf{0.99 \pm 0.00}$

Table 12: Results on LIME and SHAP for SQuAD dataset.

Users are free to apply our dataset to modern models such as the tabular version of Transformer, Vision Transformers, and GPT-2. Since we provide PyTorch compatibility, running these models is straightforward.

A.3 URL to website

URL: `https://github.com/xjiae/HDDDS`.

A.4 Author statement

We bear all responsibility in case of violation of rights, and confirm that we will use the MIT License.

A.5 Dataset documentation, intended use and metadata

We adopt the framework `data cards`, please find the requested information below. For better visual display, please visit this link: `https://github.com/xjiae/HDDDS/blob/main/description.md`.

The Ground Truth eXplanation (GTX) dataset is a curated collection that addresses the challenge of evaluating the quality of explainability methods. Existing approaches often lack ground truth explanations and heavily rely on hand-crafted heuristics. In response, the GTX dataset has been created to assess the alignment of feature attributions with human annotations. It contains time-series data (HAI, SWaT, WADI) from the industrial control domain, image data (MVTec) from the defect inspection domain, and text data (SQuAD) from the machine comprehension domain.

Dataset Link Dataset Link: HAI, SWaT, WADI, MVTec, SQuAD.

Data Card Author(s)

- **Xiayan Ji, University of Pennsylvania:** (Manager)
- **Anton Xue, University of Pennsylvania:** (Manager)

A.5.1 Authorship

Dataset Owners

Team(s) University of Pennsylvania

Author(s)

- Xiayan Ji, Ph.D. Student, University of Pennsylvania, 2023
- Anton Xue, Ph.D. Student, University of Pennsylvania, 2023
- Rajeev Alur, Professor, University of Pennsylvania, 2023
- Oleg Sokolsky, Professor, University of Pennsylvania, 2023
- Insup Lee, Professor, University of Pennsylvania, 2023
- Eric Wong, Assistant Professor, University of Pennsylvania, 2023

Contact Detail(s)

- **Point of Contact:** Xiayan Ji
- **Affiliation:** University of Pennsylvania
- **Contact:** xjiae@seas.upenn.edu

A.5.2 Dataset Overview

- Data about places and objects
- Synthetically generated data
- Data about systems or products and their behaviors

Dataset Snapshot

Category	Data
Size of Dataset	12 GB
Number of Instances	3,798,242
Number of Labels (explanation)	5,951,278,880
Average Labeles Per Instance	1566.85
Algorithmic Labels	4,629,687,370
Human Labels	1,321,591,510

Dataset Summary: time-series, image and text data with ground truth explanation labels.

Content Description Each content contains an input data (x), a target label (y) and an explanation (a).

Additional Notes: for SQuAD, the format is slightly different, the input and target are combined together to better be fitted to a language model. In addition, the explanation is in the form of a start and end position.

Risk Type(s)

- No Known Risks

A.5.3 Dataset Version and Maintenance

Maintenance Status **Regularly Updated** - New versions of the dataset have been or will continue to be made available.

Version Details **Current Version:** 1.0

Last Updated: 06/2023

Release Date: 06/2023

Maintenance Plan In our maintenance plan, our primary focus will be on preserving and leveraging the existing data that we have collected. This involves ensuring the integrity and security of the data through regular backups, implementing robust data storage practices, and conducting periodic audits to identify any potential issues or anomalies. Additionally, we recognize the growing importance of graph datasets in various domains. To capitalize on this, we will actively explore and evaluate potential graph datasets that align with our needs and objectives. This includes seeking out reliable

sources, assessing the quality and relevance of the data, and integrating suitable graph datasets into our existing infrastructure. By incorporating graph datasets, we aim to enhance the depth and breadth of our analysis, uncover hidden patterns and relationships, and gain valuable insights that can drive informed decision-making and optimize our operations. In addition, we are aware that the SQuAD dataset does not have a clear classification task and may not align well with the remaining dataset. We are also exploring the Contract Understanding Atticus Dataset (CUAD) [44] to see if we can algin the document classification task with the ground truth explanation they provide.

Our maintenance plan thus combines the preservation of existing data with the exploration of new graph and text datasets, ensuring a comprehensive and forward-looking approach to data management and utilization.

Versioning: The dataset is versioned based on several criteria. This includes significant updates or changes in the data collection process, methodology, or data sources. Corrections or improvements to enhance data accuracy or reliability also warrant a new version. Substantial additions or expansions, such as new data points or variables, are considered for versioning. User feedback and requests for specific modifications are also taken into account. The versioning process ensures transparency, traceability, and reproducibility, keeping the dataset relevant and adaptable to evolving needs.

Updates: The dataset is refreshed or updated based on regular time-based updates, changes in data sources or collection methodologies, user feedback, and advancements in technology or analytical techniques. This ensures the dataset remains relevant, accurate, and valuable for users in making informed decisions.

Errors: Error handling for the dataset involves systematic procedures to identify and correct errors, maintaining data integrity through documentation and tracking, and implementing measures to prevent future errors. These criteria ensure data quality, transparency, and reliability for users.

Feedback: The dataset incorporates criteria for feedback by actively seeking input from users and stakeholders. Feedback on the dataset's content, quality, and usability is welcomed and considered for future updates and improvements. This iterative feedback process ensures that the dataset meets the needs and expectations of its users, enhancing its relevance and value.

Next Planned Update(s) **Version affected:** 1.0

Next data update: 08/2023

Next version: 1.1

Next version update: 08/2023

Expected Change(s) **Updates to Data:** Next version of the dataset will possibly include suitable graph dataset and modification to the text dataset so that it has a clear classification task. We are currently investigating at the CUAD dataset [44].

A.5.4 Example of Data Points

Primary Data Modality

- Image Data
- Text Data
- Time Series

Sampling of Data Points

- Demo Link

Data Fields

Field Name	Field Value	Description
x	input data	The input data, time-series or image or pagraph.
y	target label (0/1)	The target label of attacked/defect/answerable.
a	explanation	The ground truth feature to explain the target label.

Typical Data Point This is a typical data point:


```
{'x': tensor([[0.6273, 0.2893, 0.2775,  ..., 0.4198, 0.3439, 0.5313],
        [0.6273, 0.2985, 0.2775,  ..., 0.4198, 0.3401, 0.5330],
        [0.6273, 0.3055, 0.2775,  ..., 0.4198, 0.3439, 0.5292],
        ...,
        [0.6273, 0.3265, 0.2775,  ..., 0.4198, 0.3467, 0.4995],
        [0.6273, 0.3341, 0.2775,  ..., 0.4198, 0.3467, 0.5019],
        [0.6273, 0.3444, 0.2775,  ..., 0.4198, 0.3467, 0.5022]]),
  tensor([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0]),
  tensor([[0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
        ...,
        [0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.]], dtype=torch.float64)}
```


A.5.5 Motivations & Intentions

Motivations

Purpose(s)

- Research

Domain(s) of Application `Machine Learning`, `Explainability`, `XAI`, `Anomaly Detection`.

Motivating Factor(s)

- Evaluating the quality of explainability methods is challenging due to the lack of ground truth explanations, and often rely on hand-crafted heuristics.
- Re-aligning explainable models with human explanations

A.5.6 Intended Use

Dataset Use(s)

- Safe for research use

Suitable Use Case(s) **Suitable Use Case:** One suitable use case for the dataset is in the field of explainable artificial intelligence (AI). The dataset, Ground Truth eXplanation (GTX), provides a valuable resource for evaluating and improving feature attribution methods. Researchers and practitioners in the field can utilize the dataset to benchmark and compare different algorithms, assess their alignment with human annotations, and identify areas for improvement. The diverse

nature of the dataset, spanning various data types such as time-series, images, and text, allows for comprehensive evaluation in different real-world scenarios.

Unsuitable Use Case(s) **Unsuitable Use Case: Suitable Use Case:** One suitable use case for the dataset is in the field of explainable artificial intelligence (AI). The dataset, Ground Truth eXplanation (GTX), provides a valuable resource for evaluating and improving feature attribution methods. Researchers and practitioners in the field can utilize the dataset to benchmark and compare different algorithms, assess their alignment with human annotations, and identify areas for improvement. The diverse nature of the dataset, spanning various data types such as time-series, images, and text, allows for comprehensive evaluation in different real-world scenarios.

Research and Problem Space(s) The specific problem space that the Ground Truth eXplanation (GTX) dataset aims to address is the evaluation and improvement of feature attribution methods in explainable artificial intelligence (AI). The dataset seeks to tackle the challenge of assessing the alignment between feature attributions and human annotations, providing a quantitative benchmark for evaluating the quality of these methods.

Citation Guidelines **Guidelines & Steps:** Please cite our work as follows (to be updated later):

BiBTeX:

```
@article{snp2023,
  title={Ground Truth eXplanation Datset},
  author={../},
  journal={...},
  year={2023}
}
```

A.5.7 Access, Rentention, & Wipeout

Access

Access Type

- External - Open Access

Documentation Link(s)

- GitHub URL

Policy Link(s)

- Direct download URL: link

Code to download data: `https://github.com/xjiae/HDDDS/blob/main/setup.sh`

Retention

Duration Infinite duration.

A.5.8 Provenance

Collection

Method(s) Used

- Taken from other existing datasets

Methodology Detail(s) Collection Type

Source: HAI, SWaT, WADI, MVTec, SQuAD.

Is this source considered sensitive or high-risk? [No]

Dates of Collection: [05 2023 - 06 2023]

Primary modality of collection data:

- Image Data
- Text Data
- Time Series

Update Frequency for collected data:

- Static

Source Description(s)

- **Source:** Hardware-In-the-Loop-based Augmented ICS Security Dataset (HAI) The HAI dataset was collected from a realistic industrial control system (ICS) testbed, augmented with a Hardware-In-the-Loop (HIL) simulator for 379.3 hours. The HIL simulator emulates two crucial components of the power generation domain: steam-turbine power generation and pumped-storage hydropower generation, with a total of m = 86 features.
- **Source:** SWaT, WADI. The Secure Water Treatment testbed serves as a scaled-down replica of a real-world industrial water treatment plant. It operates at a reduced capacity, producing five gallons per minute of water for over 11 days. The treatment process involves the utilization of membrane-based ultrafiltration and reverse osmosis units for effective water filtration, comprising of (m = 51) features in total. WADI is an extension of the SWaT testbed featuring additional components and functionalities such as chemical dosing systems, booster pumps and valves, as well as instrumentation and analyzers. It is collected over 16 days with (m = 127) dimensions.
- **Source:** MVTec is an industrial inspection dataset designed for benchmarking defects detection methods. It consists of a 15 categories with a total of more than 5000 high-resolution (3,1024, 1024) images. Each category includes a set of defect-free training images and a test set containing images with different types of defects, as well as defect-free images. The dataset provides pixel-accurate ground truth annotations for the defect regions, which have been carefully annotated and reviewed by the authors to align with human interpretation of real-world defects.
- **Source:** SQuAD is a widely used reading comprehension dataset that includes 107,785 question-answer pairs based on 536 Wikipedia articles. The dataset was generated by crowdworkers who formulated questions and provided specific text segments or spans as answers. The answers have undergone rigorous crowdworkers selection, additional answer collection, and manual crosscheck processes, making them reliable ground truth explanations for the corresponding questions.

Collection Cadence Static: Data was collected once from single or multiple sources.

Data Integration

Source Included Fields

Data fields of each datasets were collected and are included in the dataset. We found the detailed description for HAI (Table 15 and 16) and SWaT (Table 17) and consolidate them to the tables below. For WADI, we did not find any detailed description. It is an extension of SWaT hence

they share similar features. We attach the testbed information `https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_wadi/`.

Data Processing Collection Method or Source

Description: In our data processing pipeline, we employ different techniques based on the data type. For timeseries data, we apply normalization to ensure it falls within the range of [0, 1], enabling better comparison and analysis across different variables. On the other hand, we do not perform any additional processing for image and text data, as they are inherently suitable for analysis without preprocessing steps.

When it comes to annotations, we have a dedicated process to handle them. For ground truth annotation files, which are typically stored in formats such as Excel or PDF, we extract the relevant information such as start time, end time, and the sensors involved in the attack. We then align this information with the raw data to ensure accurate labeling of explanations. This process allows us to establish a clear link between the annotated events and the underlying data, facilitating the evaluation and analysis of the explanations provided by our models.

By leveraging these data processing techniques, we ensure that the data is appropriately prepared and annotated for further analysis and evaluation. This enables us to derive valuable insights and make informed decisions based on the processed and labeled data.

Methods employed: Normalization.

Tools or libraries: Min-Max scaling.

A.5.9 Collection Criteria

Data Selection

- **Collection Method of Source:** We select the dataset based on availability of ground truth of explanations.

Data Inclusion

- **Collection Method of Source:** Same as above.

Data Exclusion

- **Collection Method of Source:** We exclude data that does not have ground truth for explanation.

A.5.10 Relationship to Source

Use & Utility(ies)

- **Dataset:** The resulting Ground Truth eXplanation (GTX) dataset is closely aligned with the purposes, motivations, and intended use of the upstream sources (HAI, WADI, SWaT, MVTec, and SQuAD). Through meticulous cleaning and preprocessing of annotation files, the dataset provides accurate ground truth information for feature attribution evaluation in explainable AI. This alignment ensures that the GTX dataset is a valuable resource for benchmarking, model development, and educational purposes, enabling advancements in transparency, interpretability, and trustworthiness of AI systems across domains.

Benefit and Value(s)

- **Dataset:** The Ground Truth eXplanation (GTX) dataset provides consumers with curated and cleaned annotations, consolidating data from multiple sources. Compared to the upstream sources, it offers enhanced data quality, convenience, and relevance for evaluating and improving feature attribution methods in explainable AI.

Limitation(s) and Trade-Off(s)

- **Dataset:** While the resulting Ground Truth eXplanation (GTX) dataset offers benefits, it also has certain limitations compared to the upstream sources. Firstly, the GTX dataset may have reduced granularity compared to the original upstream sources, as it involves cleaning and preprocessing steps that can result in some loss of detailed information. Secondly, the dataset's scope and coverage may be limited to specific features or attributes relevant to feature attribution evaluation, potentially excluding certain aspects present in the upstream sources. Additionally, the GTX dataset's generalizability may be constrained by the specific contexts and domains of the upstream sources, which may not fully represent the diverse range of applications and scenarios. It is important for consumers to consider these limitations and assess whether the available data adequately meets their specific needs and requirements.

A.5.11 Version and Maintenance

- **Release date:** 06/2023
- **Link to dataset:** GTX + 1.0
- **Status:** [Actively Maintained]
- **Size of Dataset:** 12 GB
- **Number of Instances:** 3,798,242

Note(s) and Caveat(s) We may update the dataset content if we find suitable graph dataset, but it will not affect the exitsing datasets.

Cadence

- Static

Last and Next Update(s)

- **Date of last update:** 14/06/2023
- **Total data points affected:** 3,798,242
- **Data points updated:** 3,798,242
- **Data points added:** 3,798,242
- **Data points removed:** 0
- **Date of next update:** 08/08/2023

Changes on Update(s)

- **Dataset:** Update five real-world datasets.

A.5.12 Extended Use

Use with Other Data

Safety Level

- Safe to use with other data

Known Safe Dataset(s) or Data Type(s) **Data Type:** time-series, image, and text.

Best Practices When using the Ground Truth eXplanation (GTX) dataset with other datasets or data types, it is important to ensure data compatibility, identify common features, validate and cross-reference the data, consider contextual relevance, document assumptions and limitations, and perform exploratory analysis for insights.

Known Unsafe Dataset(s) or Data Type(s) N/A

A.5.13 Forking & Sampling

Safety Level

- Safe to form and/or sample

Acceptable Sampling Method(s)

- Cluster Sampling
- Haphazard Sampling
- Multi-stage sampling
- Random Sampling
- Retrospective Sampling
- Stratified Sampling
- Systematic Sampling
- Weighted Sampling
- Unknown
- Unsampled

Best Practice(s) When forking or sampling the GTX dataset, best practices include clearly defining sampling criteria, maintaining representative samples, documenting the sampling methodology, considering sample size and statistical power, and validating the sample.

Risk(s) and Mitigation(s) No known risk for sampling.

A.5.14 Use in ML or AI Systems

Dataset Use(s)

- Training
- Testing
- Validation
- Development or Production Use
- Fine Tuning

Notable Feature(s) The GTX dataset exhibits notable feature distributions and explicit relationships between individual instances. Through careful curation, the dataset captures diverse real-world data types such as time-series, image, and text, each with its distinct feature distributions. These distributions may reveal patterns, trends, or variations in the data, providing valuable insights into the characteristics of different instances. Additionally, explicit relationships between individual instances can be identified through the ground truth annotations, which establish causal connections between features and the corresponding labels. These relationships help to elucidate the impact and importance of specific features in explaining the ground truth, contributing to the evaluation and improvement of feature attribution methods in explainable AI. By leveraging the feature distributions and explicit relationships within the dataset, researchers, practitioners, and educators can gain a deeper understanding of the data and make informed decisions in their respective domains.

Usage Guideline(s) **Usage Guidelines:** When using the GTX dataset, consumers should comply with licensing and terms of use, provide proper attribution and citation, aim for reproducibility and transparency, practice responsible and ethical use, and foster communication and collaboration within the community.

Approval Steps: N/A.

Reviewer: Provide the name of a reviewer for publications referencing this dataset.

Distribution(s)

Set	Number of data points
Train	70%
Test	20%
Validation	10%

Splits: Recommand splts.

Known Correlation(s) All the features are correlated with each other in a given instance. Hence, user should treat them as a complete data point when process them.

A.5.15 Transformations

Synopsis

Transformation(s) Applied

- Cleaning Missing Values
- Normalization

Field(s) Transformed **Transformation Type**

All features in time-series dataset are preprocessed. But user can also specified "raw" for contents to get the original dataset.

Library(ies) and Method(s) Used **Transformation Type**

Method: For timeseries data, we apply normalization to ensure it falls within the range of [0, 1], enabling better comparison and analysis across different variables.

Platforms, tools, or libraries: - Platform, tool, or library: sklearn.preprocessing.MinMaxScaler.

Transformation Results: All time-series values falls within the range of [0, 1].

A.5.16 Breakdown of Transformations

Cleaning Missing Value(s) We fill missing sensor values with mean of the corresponding column.

Method(s) Used To handle missing sensor values, we replace them with the mean value of the corresponding column.

Platforms, tools, or libraries

- Platform, tool, or library: pandas.DataFrame.fillna

A.5.17 Annotations & Labeling

Annotation Workforce Type

- Annotation Target in Data
- Machine-Generated
- Annotations

A.5.18 Annotation Characteristic(s)

Annotation	Number
Total number of annotations	1,321,591,510
Average annotations per example	17,962

Annotation Description(s) The annotations applied to the dataset were manually performed by the author. The author meticulously reviewed the annotation file, ensuring precise alignment of the start and end times of each attack/defect. They annotated the affected features, indicating the specific features impacted during each attack. The annotation process involved a thorough analysis and interpretation of the data to ensure accuracy and consistency. For non-attacked/defect instances, an all zeroes annotation is generated automatically. No specific platforms, tools, or libraries were mentioned in the provided information.

Annotation Distribution(s) There are two classes of annotations, 1 for explanatory feature and 0 otherwise. We report the ratio for class 1.

Annotation Type	Number
HAI, column-wise	1,034,580 (1.17%)
SWaT, column-wise	2,785,671 (2.10%)
WADI, column-wise	652,018 (1.52%)
MVTec, pixel-wise	1,317,011,456 (4.38%)
SQuAD, start-end position pair	107,785 (3.10%)

Annotation summary: We summarize the explanatory feature count and ratio.

A.5.19 Terms of Art

Concepts and Definitions referenced in this Data Card

Term of Art Definition: feature attribution

Interpretation: Feature attributions indicate how much each feature in your model contributed to the predictions for each given instance.

Features	Min Value	Max Value	Unit	Description
P1_B2004	0	10	bar	Heat-exchanger outlet pressure setpoint
P1_B2016	0	10	bar	Pressure demand for thermal power output control
P1_B3004	0	720	mm	Water level setpoint (return water tank)
P1_B3005	0	2500	l/h	Discharge flowrate setpoint (return water tank)
P1_B4002	0	100	°C	Heat-exchanger outlet temperature setpoint
P1_B4005	0	100	%	Temperature PID control output
P1_B400B	0	2500	l/h	Water outflow rate setpoint (heating water tank)
P1_B4022	0	40	°C	Temperature demand for thermal power output control
P1_FCV01D	0	100	%	Position command for the FCV01 valve
P1_FCV01Z	0	100	%	Current position of the FCV01 valve
P1_FCV02D	0	100	%	Position command for the FCV02 valve
P1_FCV02Z	0	100	%	Current position of the FCV02 valve
P1_FCV03D	0	100	%	Position command for the FCV03 valve
P1_FCV03Z	0	100	%	Current position of the FCV03 valve
P1_FT01	0	2500	mmH2O	Measured flowrate of the return water tank
P1_FT01Z	0	3190	l/h	Water inflow rate converted from P1_FT01
P1_FT02	0	2500	mmH2O	Measured flowrate of heating water tank
P1_FT02Z	0	3190	l/h	Water outflow rate conversion from P1_FT02
P1_FT03	0	2500	mmH2O	Measured flowrate of the return water tank
P1_FT03Z	0	3190	l/h	Water outflow rate converted from P1_FT03
P1_LCV01D	0	100	%	Position command for the LCV01 valve
P1_LCV01Z	0	100	%	Current position of the LCV01 valve
P1_LIT01	0	720	mm	Water level of the return water tank
P1_PCV01D	0	100	%	Position command for the PCV01 valve
P1_PCV01Z	0	100	%	Current position of the PCV01 valve
P1_PCV02D	0	100	%	Position command for the PCV2 valve
P1_PCV02Z	0	100	%	Current position of the PCV02 valve
P1_PIT01	0	10	bar	Heat-exchanger outlet pressure
P1_PIT01_HH	0	10	bar	Highest outlet pressure of the heat-exchanger
P1_PIT02	0	10	bar	Water supply pressure of the heating water pump
P1_PP01AD	0	1	Boolean	Start command of the main water pump PP01A
P1_PP01AR	0	1	Boolean	Running state of the main water pump PP01A
P1_PP01BD	0	1	Boolean	Start command of the main water pump PP01B
P1_PP01BR	0	1	Boolean	Running state of the main water pump PP01B
P1_PP02D	0	1	Boolean	Start command of the heating water pump PP02
P1_PP02R	0	1	Boolean	Running state of the heating water pump PP02
P1_PP04	0	100	%	Control out of the cooler pump
P1_PP04SP	0	100	°C	Cooler temperature setpoint
P1_SOL01D	0	1	Boolean	Open command of the main water tank supply valve
P1_SOL03D	0	1	Boolean	Open command of the main water tank drain valve
P1_STSP	0	1	Boolean	Start/stop command of the boiler DCS
P1_TIT01	-50	150	°C	Heat-exchanger outlet temperature
P1_TIT02	-50	150	°C	Temperature of the heating water tank
P1_TIT03	-50	150	°C	Temperature of the main water tank
P2_24Vdc	0	30	Voltage	DCS 24V Input Voltage
P2_ATSW_Lamp	0	1	Boolean	Lamp of the Auto SW
P2_AutoGo	0	1	Boolean	Auto start button
P2_AutoSD	0	3200	RPM	Auto speed demand
P2_Emerg	0	1	Boolean	Emergency button
P2_MASW	0	1	Boolean	Manual(1)/Auto(0) SW
P2_MASW_Lamp	0	1	Boolean	Lamp of Manual SW
P2_ManualGO	0	1	Boolean	Manual start button
P2_ManualSD	0	3200	RPM	Manual speed demand
P2_OnOff	0	1	Boolean	On/off switch of the turbine DCS
P2_RTR	0	2880	RPM	RPM trip rate
P2_SCO	0	100000	-	Control output value of the speed controller
P2_SCST	-100	100	RPM	Speed change proportional to frequency change of the STM
P2_SIT01	0	3200	RPM	Current turbine RPM measured by speed probe
P2_TripEx	0	1	Boolean	Trip emergency exit button
P2_VIBTR01	-10	10	μm	Shaft-vibration-related Y-axis displacement near the 1st mass wheel
P2_VIBTR02	-10	10	μm	Shaft-vibration-related X-axis displacement near the 1st mass wheel
P2_VIBTR03	-10	10	μm	Shaft-vibration-related Y-axis displacement near the 2nd mass wheel
P2_VIBTR04	-10	10	μm	Shaft-vibration-related X-axis displacement near the 2nd mass wheel
P2_VT01	11	12	rad/s	Phase lag signal of the key phasor probe
P2_VTR01	-10	10	μm	Preset vibration limit for the sensor P2_VIBTR01
P2_VTR02	-10	10	μm	Preset vibration limit for the sensor P2_VIBTR02
P2_VTR03	-10	10	μm	Preset vibration limit for the sensor P2_VIBTR03
P2_VTR04	-10	10	μm	Preset vibration limit for the sensor P2_VIBTR03

Table 15: HAI feature description.

P3_FIT01	0	27648	-	Flow rate of water flowing into the upper water tank
P3_LCP01D	0	27648	-	Speed command for the pump LCP01
P3_LCV01D	0	27648	-	Position command for the valve LCV01
P3_LH01	0	70	%	High water level set-point
P3_LIT01	0	90	%	Water level of the upper water tank
P3_LL01	0	70	%	Low water level set-point
P3_PIT01	0	27648	-	Pressure of water flowing into the upper water tank
P4_HT_FD	-0.02	0.02	mHz	Frequency deviation of HTM
P4_HT_PO	0	100	MW	Output power of HTM
P4_HT_PS	0	100	MW	Scheduled power demand of HTM
P4_LD	0	500	MW	Total electrical load demand
P4_ST_FD	-0.02	0.02	Hz	Frequency deviation of STM
P4_ST_GOV	0	27648	-	Gate opening rate of STM
P4_ST_LD	0	500	MW	Electrical load demand of STM
P4_ST_PO	0	500	MW	Output power of STM
P4_ST_PS	0	500	MW	Scheduled power demand of STM
P4_ST_PT01	0	27648	-	Digital value of steam pressure of STM
P4_ST_TT01	0	27648	-	Digital value of steam temperature of STM

Table 16: HAI feature description continued.

Feature	Type	Description
FIT-101	Sensor	Flow meter; Measures inflow into raw water tank.
LIT-101	Sensor	Level Transmitter; Raw water tank level.
MV-101	Actuator	Motorized valve; Controls water flow to the raw water tank.
P-101	Actuator	Pump; Pumps water from raw water tank to second stage.
P-102 (backup)	Actuator	Pump; Pumps water from raw water tank to second stage.
AIT-201	Sensor	Conductivity analyser; Measures NaCl level.
AIT-202	Sensor	pH analyser; Measures HCl level.
AIT-203	Sensor	ORP analyser; Measures NaOCl level.
FIT-201	Sensor	Flow Transmitter; Control dosing pumps.
MV-201	Actuator	Motorized valve; Controls water flow to the UF feed water tank.
P-201	Actuator	Dosing pump; NaCl dosing pump.
P-202 (backup)	Actuator	Dosing pump; NaCl dosing pump.
P-203	Actuator	Dosing pump; HCl dosing pump.
P-204 (backup)	Actuator	Dosing pump; HCl dosing pump.
P-205	Actuator	Dosing pump; NaOCl dosing pump.
P-206 (backup)	Actuator	Dosing pump; NaOCl dosing pump.
DPIT-301	Sensor	Differential pressure indicating transmitter; Controls the back-wash process.
FIT-301	Sensor	Flow meter; Measures the flow of water in the UF stage.
LIT-301	Sensor	Level Transmitter; UF feed water tank level.
MV-301	Actuator	Motorized Valve; Controls UF-Backwash process.
MV-302	Actuator	Motorized Valve; Controls water from UF process to De-Chlorination unit.
MV-303	Actuator	Motorized Valve; Controls UF-Backwash drain.
MV-304	Actuator	Motorized Valve; Controls UF drain.
P-301 (backup)	Actuator	UF feed Pump; Pumps water from UF feed water tank to RO feed water tank via UF filtration.
P-302	Actuator	UF feed Pump; Pumps water from UF feed water tank to RO feed water tank via UF filtration.
AIT-401	Sensor	RO hardness meter of water.
AIT-402	Sensor	ORP meter; Controls the NaHSO3dosing(P203), NaOCl dosing (P205).
FIT-401	Sensor	Flow Transmitter ; Controls the UV dechlorinator.
LIT-401	Actuator	Level Transmitter; RO feed water tank level.
P-401 (backup)	Actuator	Pump; Pumps water from RO feed tank to UV dechlorinator.
P-402	Actuator	Pump; Pumps water from RO feed tank to UV dechlorinator.
P-403	Actuator	Sodium bi-sulphate pump.
P-404 (backup)	Actuator	Sodium bi-sulphate pump.
UV-401	Actuator	Dechlorinator; Removes chlorine from water.
AIT-501	Sensor	RO pH analyser; Measures HCl level.
AIT-502	Sensor	RO feed ORP analyser; Measures NaOCl level.
AIT-503	Sensor	RO feed conductivity analyser; Measures NaCl level.
AIT-504	Sensor	RO permeate conductivity analyser; Measures NaCl level.
FIT-501	Sensor	Flow meter; RO membrane inlet flow meter.
FIT-502	Sensor	Flow meter; RO Permeate flow meter.
FIT-503	Sensor	Flow meter; RO Reject flow meter.
FIT-504	Sensor	Flow meter; RO re-circulation flow meter.
P-501	Actuator	Pump; Pumps dechlorinated water to RO.
P-502 (backup)	Actuator	Pump; Pumps dechlorinated water to RO.
PIT-501	Sensor	Pressure meter; RO feed pressure.
PIT-502	Sensor	Pressure meter; RO permeate pressure.
PIT-503	Sensor	Pressure meter;RO reject pressure.
FIT-601	Sensor	Flow meter; UF Backwash flow meter.
P-601	Actuator	Pump; Pumps water from RO permeate tank to raw water tank (not used for data collection).
P-602	Actuator	Pump; Pumps water from UF back wash tank to UF filter to clean the membrane.

Table 17: SWaT feature description.