A. Proof of Theorem 1

Here we prove Theorem 1 for information gain score.

Proof. $H(y)$ and $H(y|x^{(j)} < \eta)$ are defined as

$$H(y) = -\frac{|\mathcal{I}_0|}{|\mathcal{I}|}\log(\frac{|\mathcal{I}_0|}{|\mathcal{I}|}) - \frac{|\mathcal{I}_1|}{|\mathcal{I}|}\log(\frac{|\mathcal{I}_1|}{|\mathcal{I}|}),$$

and

$$\begin{aligned}&H(y|x^{(j)} < \eta) = \\ &-\frac{|\mathcal{I}_L|}{|\mathcal{I}|}\left[\frac{|\mathcal{I}_L\cap\mathcal{I}_0|}{|\mathcal{I}_L|}\log(\frac{|\mathcal{I}_L\cap\mathcal{I}_0|}{|\mathcal{I}_L|}) + \frac{|\mathcal{I}_L\cap\mathcal{I}_1|}{|\mathcal{I}_L|}\log(\frac{|\mathcal{I}_L\cap\mathcal{I}_1|}{|\mathcal{I}_L|})\right] \\ &-\frac{|\mathcal{I}_R|}{|\mathcal{I}|}\left[\frac{|\mathcal{I}_R\cap\mathcal{I}_0|}{|\mathcal{I}_R|}\log(\frac{|\mathcal{I}_R\cap\mathcal{I}_0|}{|\mathcal{I}_R|}) + \frac{|\mathcal{I}_R\cap\mathcal{I}_1|}{|\mathcal{I}_R|}\log(\frac{|\mathcal{I}_R\cap\mathcal{I}_1|}{|\mathcal{I}_R|})\right].\end{aligned}$$

For simplicity, we denote $N_0 := |\mathcal{I}_0|$, $N_1 := |\mathcal{I}_1|$, $n_0 := |\mathcal{I}_L \cap \mathcal{I}_0|$ and $n_1 := |\mathcal{I}_L \cap \mathcal{I}_1|$. The information gain of this split can be written as a function of n_0 and n_1:

$$\begin{aligned}IG = C_1[&n_0\log(\frac{n_0}{N_0(n_1+n_0)}) + n_1\log(\frac{n_1}{N_1(n_1+n_0)}) \\ &+ (N_0-n_0)\log(\frac{N_0-n_0}{N_0(N_1+N_0-n_1-n_0)}) \\ &+ (N_1-n_1)\log(\frac{N_1-n_1}{N_1(N_1+N_0-n_1-n_0)})] + C_2,\end{aligned} \tag{5}$$

where $C_1 > 0$ and C_2 are constants with respect to n_0. Taking n_0 as a continuous variable, we have

$$\frac{\partial IG}{\partial n_0} = C_1 \cdot \log(1 + \frac{n_0N_1 - N_0n_1}{(N_0-n_0)(n_1+n_0)}) \tag{6}$$

When $\frac{\partial IG}{\partial n_0} < 0$, perturbing one example in $\Delta\mathcal{I}_R$ with label 0 to $\mathcal{I}_L$ will increase n_0 and decrease the information gain. It is easy to see that $\frac{\partial IG}{\partial n_0} < 0$ if and only if $\frac{n_0}{N_0} < \frac{n_1}{N_1}$. This indicates that when $\frac{n_0}{N_0} < \frac{n_1}{N_1}$ and $\frac{n_0+1}{N_0} \leq \frac{n_1}{N_1}$, perturbing one example with label 0 to $\mathcal{I}_L$ will always decrease the information gain. □

Similarly, if $\frac{n_1}{N_1} < \frac{n_0}{N_0}$ and $\frac{n_1+1}{N_1} \leq \frac{n_0}{N_0}$, perturbing one example in $\Delta\mathcal{I}_R$ with label 1 to $\mathcal{I}_L$ will decrease the information gain. As mentioned in the main text, to decrease the information gain score in Algorithm 1, the adversary needs to perturb examples in $\Delta\mathcal{I}$ such that $\frac{n_0}{N_0}$ and $\frac{n_1}{N_1}$ are close to each other. Algorithm 3 gives an $O(|\Delta\mathcal{I}|)$ method to find Δn_0^* and Δn_1^*, the optimal number of points in $\Delta\mathcal{I}$ with label 0 and 1 to be added to the left.

B. Gini Impurity Score

We also have a theorem for Gini impurity score similar to Theorem 1.

Algorithm 3 Finding Δn_0^* and Δn_1^* to Minimize Information Gain or Gini Impurity

Input: N_0 and N_1, number of instances with label 0 and 1. n_0^o and n_1^o, number of instances with label 0 and 1 that are certainly on the left.
Input: $|\Delta\mathcal{I} \cap \mathcal{I}_0|$ and $|\Delta\mathcal{I} \cap \mathcal{I}_1|$, number of instances with label 0 and 1 that can be perturbed.
Output: Δn_0^*, Δn_1^*, optimal number of points with label 0 and 1 in $\Delta\mathcal{I}$ to be place on the left.
$\Delta n_0^* \leftarrow 0,\ \Delta n_1^* \leftarrow 0,\ \text{min\_diff} \leftarrow |\frac{n_0^o}{N_0} - \frac{n_1^o}{N_1}|$;
for $\Delta n_0 \leftarrow 0$ **to** $|\Delta\mathcal{I} \cap \mathcal{I}_0|$ **do**
  ceil $\leftarrow \lceil\frac{N_1(n_0^o+\Delta n_0)}{N_0}\rceil - n_1^o$;
  floor $\leftarrow \lfloor\frac{N_1(n_0^o+\Delta n_0)}{N_0}\rfloor - n_1^o$;
  for $\Delta n_1'$ **in** $\{$ceil, floor$\}$ **do**
    $\Delta n_1 \leftarrow \max\{\min\{\Delta n_1',\ |\Delta\mathcal{I} \cap \mathcal{I}_1|\},\ 0\}$;
    if min_diff $> |\frac{\Delta n_0 + n_0^o}{N_0} - \frac{\Delta n_1 + n_1^o}{N_1}|$ **then**
      $\Delta n_0^* \leftarrow \Delta n_0,\ \Delta n_1^* \leftarrow \Delta n_1,\ \text{min\_diff} \leftarrow |\frac{\Delta n_0 + n_0^o}{N_0} - \frac{\Delta n_1 + n_1^o}{N_1}|$;
    end if
  end for
end for
Return Δn_0^* and Δn_1^*;

Theorem B.1. *If $\frac{n_0}{N_0} < \frac{n_1}{N_1}$ and $\frac{n_0+1}{N_0} \leq \frac{n_1}{N_1}$, perturbing one example in $\Delta\mathcal{I}_R$ with label 0 to $\mathcal{I}_L$ will decrease the Gini impurity.*

Proof. The Gini impurity score of a split with threshold η on feature j is

$$\begin{aligned}Gini &= (1 - \frac{|\mathcal{I}_0|^2}{|\mathcal{I}|^2} - \frac{|\mathcal{I}_1|^2}{|\mathcal{I}|^2}) \\ &\quad - \frac{|\mathcal{I}_L|}{|\mathcal{I}|}(1 - \frac{|\mathcal{I}_0\cap\mathcal{I}_L|^2}{|\mathcal{I}_L|^2} - \frac{|\mathcal{I}_1\cap\mathcal{I}_L|^2}{|\mathcal{I}_L|^2}) \\ &\quad - \frac{|\mathcal{I}_R|}{|\mathcal{I}|}(1 - \frac{|\mathcal{I}_0\cap\mathcal{I}_R|^2}{|\mathcal{I}_R|^2} - \frac{|\mathcal{I}_1\cap\mathcal{I}_R|^2}{|\mathcal{I}_R|^2}) \\ &= C_3[\frac{n_0^2+n_1^2}{n_1+n_0} + \frac{(N_0-n_0)^2+(N_1-n_1)^2}{(N_0+N_1-n_0-n_1)}] + C_4,\end{aligned} \tag{7}$$

where we use the same notation as in (5). $C_3 > 0$ and C_4 are constants with respect to n_0. Taking n_0 as a continuous variable, we have

$$\frac{\partial\, Gini}{\partial n_0} = 2C_3\frac{m_1m_0(n_0m_1 + n_1m_0 + 2n_1m_1)}{(n_0+n_1)^2(m_0+m_1)^2}(\frac{n_0}{m_0} - \frac{n_1}{m_1}), \tag{8}$$

where $m_0 := N_0 - n_0$ and $m_1 := N_1 - n_1$. Then $\frac{\partial\, Gini}{\partial n_0} < 0$ holds if $\frac{n_0}{m_0} < \frac{n_1}{m_1}$, which is equivalent to $\frac{n_0}{N_0} < \frac{n_1}{N_1}$. □

Since the conditions of Theorem 1 and Theorem B.1 are the same, Algorithm 1 and Algorithm 3 also work for tree-based models using Gini impurity score.

C. Decision Boundaries of Robust and Natural Models

Figure 4 shows the decision boundaries and test accuracy of natural trees as well as robust trees with different ϵ values on two dimensional synthetic datasets. All trees have depth 5 and we plot training examples in the figure. The results show that the decision boundaries of our robust decision trees are simpler than the decision boundaries in natural decision trees, agreeing with the regularization argument in the main text.

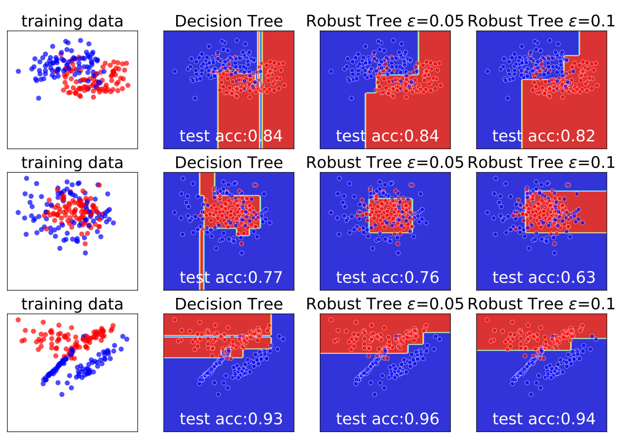

Figure 4. (Best viewed in color) The decision boundaries and test accuracy of natural decision trees and robust decision trees with depth 5 on synthetic datasets with two features.

D. Omitted Results on ℓ_1 and ℓ_2 distortion

In Tables 4 and 5 we present the ℓ_1 and ℓ_2 distortions of vanilla (information gain based) decision trees and GBDT models obtained by Kantchelian's ℓ_1 and ℓ_2 attacks. Again, only small or medium sized binary classification models can be evaluated by Kantchelian's attack. From the results we can see that although our robust decision tree training algorithm is designed for ℓ_∞ perturbations, it can also improve models ℓ_1 and ℓ_2 robustness significantly.

E. Omitted Results on Models with Different Number of Trees

Figure 5 shows the ℓ_∞ distortion and accuracy of Fashion-MNIST GBDT models with different number of trees. In Table 7 we present the test accuracy and ℓ_∞ distortion of models with different number of trees obtained by Cheng's ℓ_∞ attack. For each dataset, models are generated during a single boosting run. We can see that the robustness of robustly trained models consistently outperforms that of natural models with the same number of trees. Another interesting finding is that for MNIST and Fashion-MNIST datasets in Figures 3 (in the main text) and 5, models with more trees are generally more robust. This may not be true in other datasets; for example, results from Table 7 in the Appendix shows that on some other datasets, the natural GBDT models lose robustness when more trees are added.

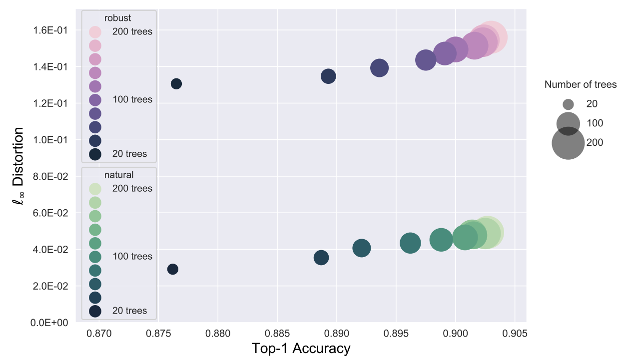

Figure 5. (Best viewed in color) ℓ_∞ distortion vs. classification accuracy of GBDT models on Fashion-MNIST datasets with different numbers of trees (circle size). The adversarial examples are found by Cheng's ℓ_∞ attack. The robust training parameter $\epsilon = 0.1$ for Fashion-MNIST. With robust training (purple) the distortion needed to fool a model increases dramatically with less than 1% accuracy loss.

F. Reducing Depth Does Not Improve Robustness

One might hope that one can simply reduce the depth of trees to improve robustness since shallower trees provide stronger regularization effects. Unfortunately, this is not true. As demonstrated in Figure 6, the robustness of naturally trained GBDT models are much worse when compared to robust models, no matter how shallow they are or how many trees are in the ensemble. Also, when the number of trees in the ensemble model is limited, reducing tree depth will significantly lower the model accuracy.

G. Random Forest Model Results

We test our robust training framework on random forest (RF) models and our results are in Table 6. In these experiments we build random forest models with 0.5 data sampling rate and 0.5 feature sampling rate. We test the robust and natural random forest model on three datasets and in each dataset, we tested 100 points using Cheng's and Kantchelian's ℓ_∞ attacks. From the results we can see that our robust decision tree training framework can also significantly improve random forest model robustness.

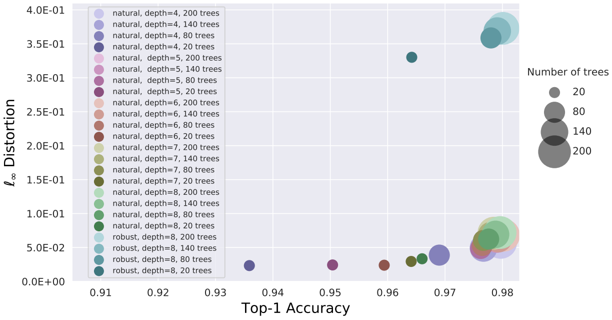

Figure 6. (Best viewed in color) Robustness vs. classification accuracy plot of GBDT models on MNIST dataset with different depth and different numbers of trees. The adversarial examples are found by Cheng's ℓ_∞ attack. The robust training parameter $\epsilon = 0.3$. Reducing the model depth cannot improve robustness effectively compared to our proposed robust training procedure.

H. More MNIST and Fashion-MNIST Adversarial Examples

In Figure 7 we present more adversarial examples for MNIST and Fashion-MNIST datasets using GBDT models.

Dataset	training set size	test set size	# of features	# of classes	robust ϵ	depth		test acc.		**avg. ℓ_1 dist. by Kantchelian's ℓ_1 attack**		**avg. ℓ_2 dist. by Kantchelian's ℓ_2 attack**	
						robust	natural	robust	natural	robust	natural	robust	natural
breast-cancer	546	137	10	2	0.3	5	5	.948	.942	**.534**	.270	**.504**	.209
diabetes	614	154	8	2	0.2	5	5	.688	.747	**.204**	.075	**.204**	.065
ionosphere	281	70	34	2	0.2	4	4	.986	.929	**.358**	.127	**.358**	.106

Table 4. The test accuracy and robustness of information gain based single decision tree models. The robustness is evaluated by the average ℓ_1 and ℓ_2 distortions of adversarial examples found by Kantchelian's ℓ_1 and ℓ_2 attacks. Average ℓ_∞ distortions of robust decision tree models found by the two attack methods are consistently larger than those of the naturally trained ones.

Dataset	training set size	test set size	# of features	# of classes	# of trees	robust ϵ	depth		test acc.		**avg. ℓ_1 dist. by Kantchelian's ℓ_1 attack**		dist. improv.	**avg. ℓ_2 dist. by Kantchelian's ℓ_2 attack**		dist. improv.
							robust	natural	robust	natural	robust	natural		robust	natural	
breast-cancer	546	137	10	2	4	0.3	8	6	.978	.964	**.488**	.328	**1.49X**	**.431**	.251	**1.72X**
cod-rna	59,535	271,617	8	2	80	0.2	5	4	.880	.965	**.065**	.059	**1.10X**	**.062**	.047	**1.32X**
diabetes	614	154	8	2	20	0.2	5	5	.786	.773	**.150**	.081	**1.85X**	**.135**	.059	**2.29X**
ijcnn1	49,990	91,701	22	2	60	0.1	8	8	.959	.980	**.057**	.051	**1.12X**	**.048**	.042	**1.14X**
MNIST 2 vs. 6	11,876	1,990	784	2	1000	0.3	6	4	.997	.998	**1.843**	.721	**2.56X**	**.781**	.182	**4.29X**

Table 5. The test accuracy and robustness of GBDT models. Average ℓ_1 and ℓ_2 distortions of robust GBDT models are consistently larger than those of the naturally trained models. The robustness is evaluated by the average ℓ_1 and ℓ_2 distortions of adversarial examples found by Kantchelian's ℓ_1 and ℓ_2 attacks.

Dataset	training set size	test set size	# of features	# of classes	# of trees	robust ϵ	depth		test acc.		**avg. ℓ_∞ dist. by Cheng's ℓ_∞ attack**		dist. improv.	**avg. ℓ_∞ dist. by Kantchelian's ℓ_∞ attack**		dist. improv.
							robust	natural	robust	natural	robust	natural		robust	natural	
breast-cancer	546	137	10	2	60	0.3	8	6	.993	.993	**.406**	.297	**1.37X**	**.396**	.244	**1.62X**
diabetes	614	154	8	2	60	0.2	5	5	.753	.760	**.185**	.093	**1.99X**	**.154**	.072	**2.14X**
MNIST 2 vs. 6	11,876	1,990	784	2	1000	0.3	6	4	.986	.983	**.445**	.180	**2.47X**	**.341**	.121	**2.82X**

Table 6. The test accuracy and robustness of random forest models. Average ℓ_∞ distortion of our robust GBDT models are consistently larger than those of the naturally trained models. The robustness is evaluated by the average ℓ_∞ distortion of adversarial examples found by Cheng's and Kantchelian's attacks.

breast-cancer (2)	train	test	feat.	# of trees	1		2		3		4		5		6		7		8		9		10	
$\epsilon = 0.3$				model	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.
depth$_r = 8,$ **depth**$_n = 6$	546	137	10	tst. acc.	.985	.942	.971	.964	.978	.956	.978	.964	.985	.964	.985	.964	.985	.971	.993	.971	.993	.971	1.00	.971
				ℓ_∞ dist.	**.383**	.215	**.396**	.229	**.411**	.216	**.411**	.215	**.406**	.226	**.407**	.229	**.406**	.248	**.439**	.234	**.439**	.238	**.437**	.241
covtype (7)	train	test	feat.	# of trees	20		40		60		80		100		120		140		160		180		200	
$\epsilon = 0.2$				model	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.
depth$_r =$ **depth**$_n = 8$	400,000	181,000	54	tst. acc.	.775	.828	.809	.850	.832	.865	.847	.877	.858	.891	.867	.902	.875	.912	.882	.921	.889	.926	.894	.930
				ℓ_∞ dist.	**.125**	.066	**.103**	.064	**.087**	.062	**.081**	.061	**.079**	.060	**.077**	.059	**.077**	.058	**.075**	.056	**.075**	.056	**.073**	.055
cod-rna (2)	train	test	feat.	# of trees	20		40		60		80		100		120		140		160		180		200	
$\epsilon = 0.2$				model	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.
depth$_r = 5,$ **depth**$_n = 4$	59,535	271,617	8	tst. acc.	.810	.947	.861	.959	.874	.963	.880	.965	.892	.966	.900	.967	.903	.967	.915	.967	.922	.967	.925	.968
				ℓ_∞ dist.	**.077**	.057	**.066**	.055	**.063**	.054	**.062**	.053	**.059**	.053	**.057**	.052	**.056**	.052	**.056**	.052	**.056**	.052	**.058**	.052
diabetes (2)	train	test	feat.	# of trees	2		4		6		8		10		12		14		16		18		20	
$\epsilon = 0.2$				model	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.
depth$_r =$ **depth**$_n = 5$	614	154	8	tst. acc.	.760	.753	.760	.753	.766	.753	.773	.753	.773	.734	.779	.727	.779	.747	.779	.760	.779	.773	.786	.773
				ℓ_∞ dist.	**.163**	.066	**.163**	.065	**.154**	.071	**.151**	.071	**.152**	.073	**.148**	.072	**.146**	.067	**.144**	.062	**.138**	.062	**.139**	.060
Fashion-MNIST (10)	train	test	feat.	# of trees	20		40		60		80		100		120		140		160		180		200	
$\epsilon = 0.1$				model	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.
depth$_r =$ **depth**$_n = 8$	60,000	10,000	784	tst. acc.	.877	.876	.889	.889	.894	.892	.898	.896	.899	.899	.900	.901	.902	.902	.902	.901	.902	.903	.903	.903
				ℓ_∞ dist.	**.131**	.029	**.135**	.035	**.139**	.041	**.144**	.043	**.147**	.045	**.149**	.047	**.151**	.048	**.153**	.048	**.154**	.049	**.156**	.049
HIGGS (2)	train	test	feat.	# of trees	50		100		150		200		250		300		350		400		450		500	
$\epsilon = 0.05$				model	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.
depth$_r =$ **depth**$_n = 8$	10,500,000	500,000	28	tst. acc.	.676	.747	.688	.753	.700	.755	.702	.758	.705	.759	.709	.760	.711	.762	.712	.764	.716	.763	.718	.764
				ℓ_∞ dist.	**.023**	.013	**.023**	.014	**.022**	.014	**.022**	.014	**.022**	.014	**.022**	.014	**.021**	.015	**.021**	.015	**.021**	.015	**.021**	.015
ijcnn1 (2)	train	test	feat.	# of trees	10		20		30		40		50		60		70		80		90		100	
$\epsilon = 0.1$				model	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.
depth$_r =$ **depth**$_n = 8$	49,990	91,701	22	tst. acc.	.933	.973	.942	.977	.947	.977	.952	.979	.958	.980	.959	.980	.962	.980	.964	.980	.967	.980	.968	.980
				ℓ_∞ dist.	**.065**	.048	**.061**	.047	**.058**	.048	**.057**	.047	**.054**	.048	**.054**	.047	**.054**	.047	**.053**	.047	**.052**	.047	**.052**	.047
MNIST (10)	train	test	feat.	# of trees	20		40		60		80		100		120		140		160		180		200	
$\epsilon = 0.3$				model	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.
depth$_r =$ **depth**$_n = 8$	$60,000$	$10,000$	784	tst. acc.	.964	.966	.973	.975	.977	.977	.978	.978	.978	.978	.979	.979	.979	.979	.980	.979	.980	.979	.980	.980
				ℓ_∞ dist.	**.330**	.033	**.343**	.049	**.352**	.057	**.359**	.062	**.363**	.065	**.367**	.067	**.369**	.069	**.370**	.071	**.371**	.072	**.373**	.072
Sensorless (11)	train	test	feat.	# of trees	3		6		9		12		15		18		21		24		27		30	
$\epsilon = 0.05$				model	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.
depth$_r =$ **depth**$_n = 6$	48,509	10,000	48	tst. acc.	.834	.977	.867	.983	.902	.987	.923	.991	.945	.992	.958	.994	.966	.996	.971	.996	.974	.997	.978	.997
				ℓ_∞ dist.	**.037**	.022	**.036**	.022	**.035**	.023	**.035**	.023	**.035**	.023	**.035**	.023	**.035**	.023	**.035**	.023	**.035**	.023	**.035**	.023
webspam (2)	train	test	feat.	# of trees	10		20		30		40		50		60		70		80		90		100	
$\epsilon = 0.05$				model	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.	rob.	nat.
depth$_r =$ **depth**$_n = 8$	$300,000$	$50,000$	254	tst. acc.	.950	.976	.964	.983	.970	.986	.973	.989	.976	.990	.978	.990	.980	.991	.981	.991	.982	.992	.983	.992
				ℓ_∞ dist.	**.049**	.010	**.048**	.015	**.049**	.019	**.049**	.021	**.049**	.023	**.049**	.024	**.049**	.024	**.049**	.024	**.048**	.024	**.049**	.024

Table 7. The test accuracy and robustness of GBDT models. Here depth$_n$ is the depth of natural trees and depth$_r$ is the depth of robust trees. Robustness is evaluated by the average ℓ_∞ distortion of adversarial examples found by Cheng's attack (Cheng et al., 2019). The number in the parentheses after each dataset name is the number of classes. Models are generated during a single boosting run. We can see that the robustness of our robust models consistently outperforms that of natural models with the same number of trees.

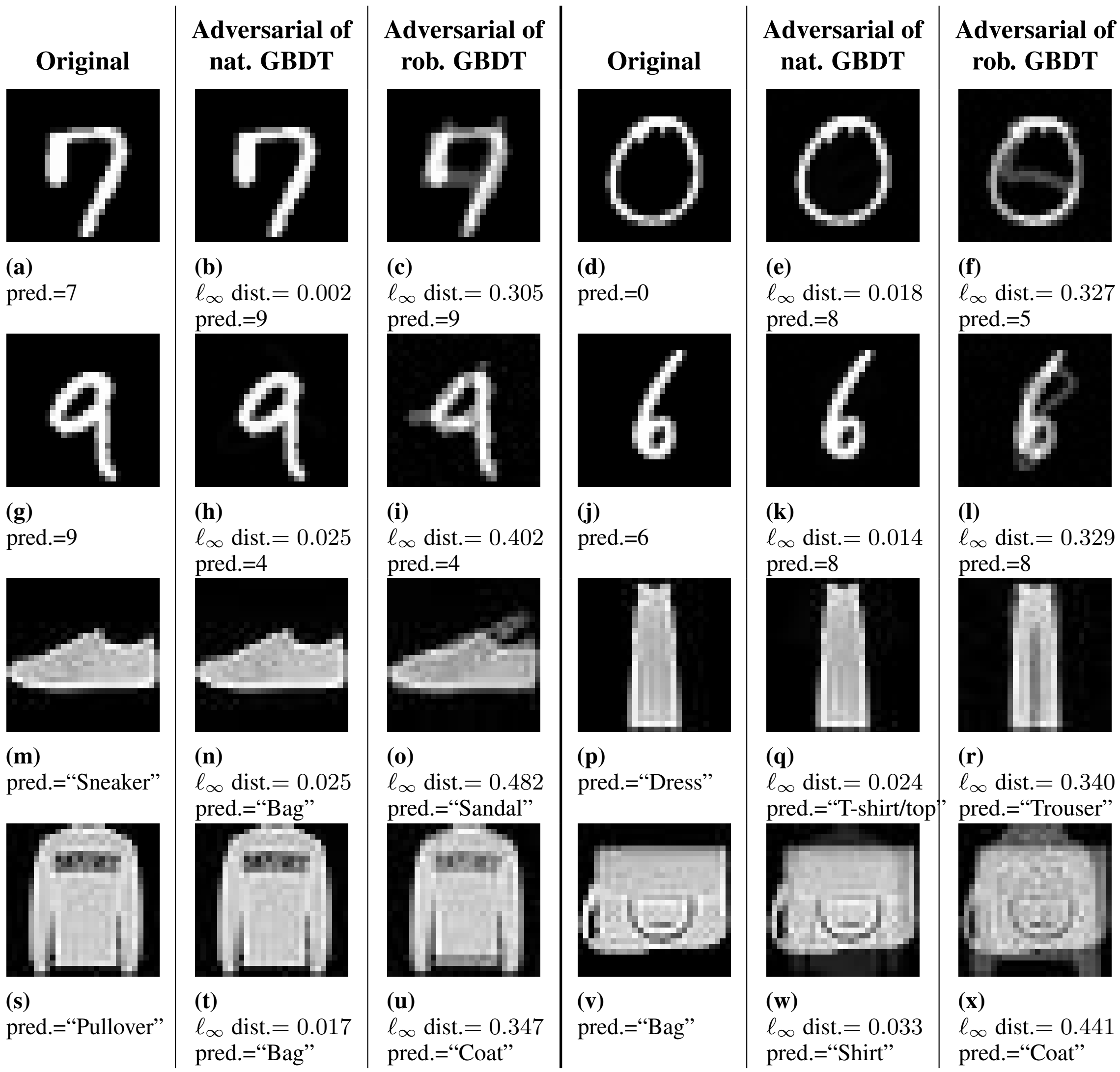

Figure 7. MNIST and Fashion-MNIST examples and their adversarial examples found using the untargeted Cheng's ℓ_∞ attack (Cheng et al., 2019) on 200-tree gradient boosted decision tree (GBDT) models trained using XGBoost with depth=8. For both MNIST and Fashion-MNIST robust models, we use $\epsilon = 0.3$.