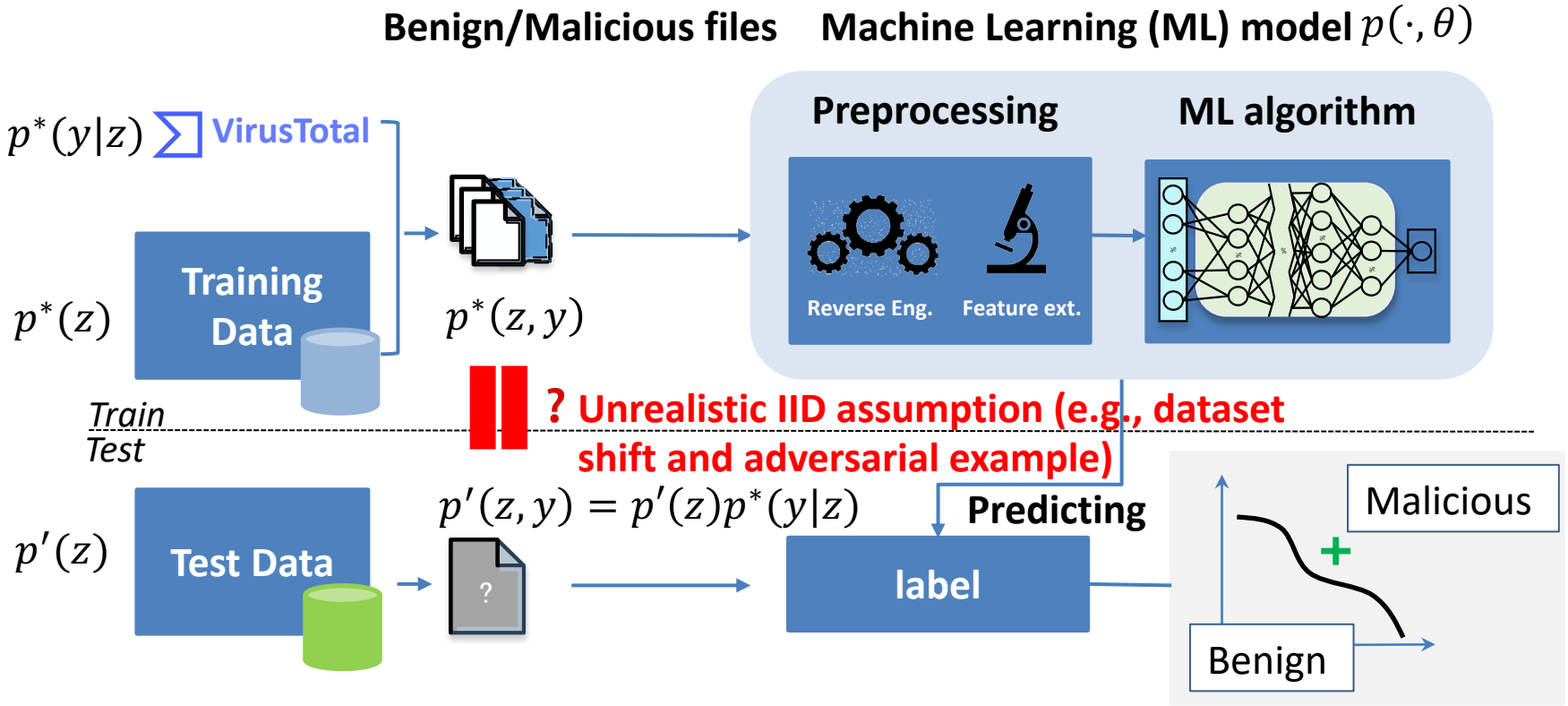
Can We Leverage Predictive Uncertainty to Detect Dataset Shift and Adversarial Examples in Android Malware Detection?

Deqiang Li — Nanjing U. of Science and Technology
Tian Qiu — Nanjing U. of Science and Technology
Shuo Chen — RIKEN
Qianmu Li — Nanjing U. of Science and Technology
Shouhuai Xu — University of Colorado Colorado Springs

Outline

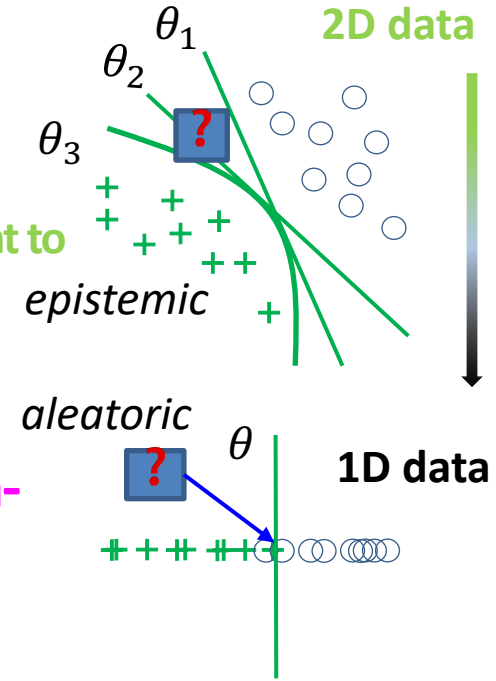
- ❑ Introduction
- ❑ Problem Statement
- ❑ Framework
- ❑ Case Study
- ❑ Conclusion

ML Classification for Malware Detection



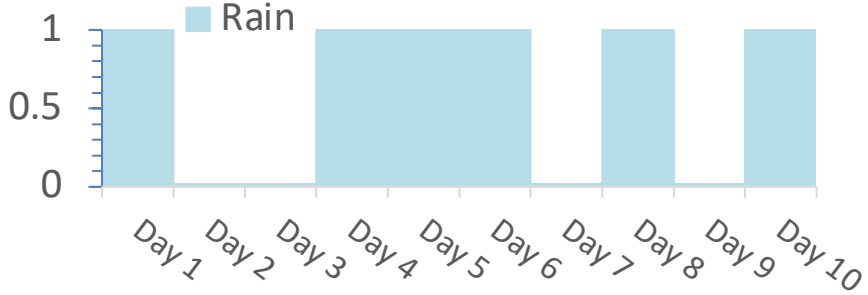
Uncertainty

- **Basic Idea: Output label along with its confidence in classification**
  - Benefiting informed decisions downstream
- **Source of uncertainty: *epistemic* vs. *aleatoric***
  - Epistemic uncertainty, known as model uncertainty (i.e., inherent to models), is induced by the inadequate knowledge
  - Aleatoric uncertainty is data uncertainty, induced by noises
- **They can be indistinguishable, e.g., because of non-robust feature extraction**



Calibration

- Goal: Turn the confidence score to be proper or well-calibrated
- A known example: A model predicts rain with 60% confidence of several days, what fraction do we observe indeed?
  - =60% implies the model is well-calibrated
  - <60% implies the model is over-confident
  - >60% implies the model is under-confident



- **Definition: Malware detector $p(\cdot,\theta): \mathcal{Z} \to [0,1]$ is well-calibrated if for each confidence $q \in [0,1]$ and $Z_q = \{z: p(y=1|z,\theta) = q, \forall z \in \mathcal{Z}\}$, it holds that $\Pr(y=1|Z_q) = q$ (proportion of positive samples in Z_q).**

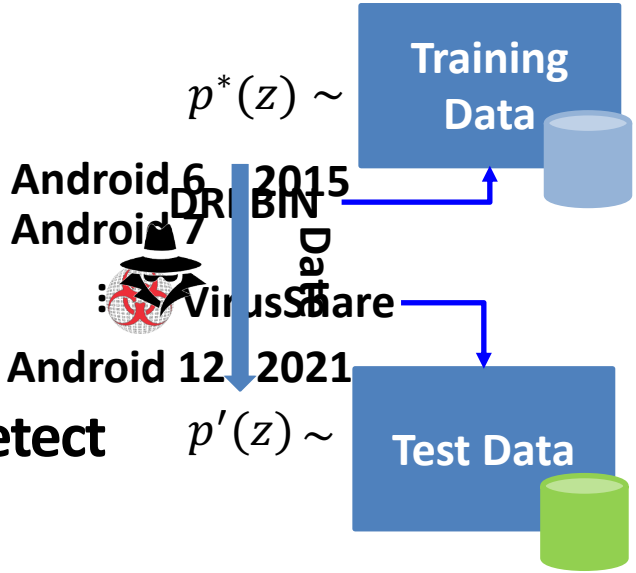
Outline

- ❑ Introduction
- ❑ **Problem Statement**
- ❑ Framework
- ❑ Case Study
- ❑ Conclusion

Problem Statement

- **Three scenarios of $p^*(z) \neq p'(z)$**
  - **Out of source: $p^*(z)$ and $p'(z)$ come from different sources**
  - **Temporal covariate shift: test data evolves over time**
  - **Adversarial attack: test data is manipulated adversarially,**

    e.g., an attacker intentionally modifies malware samples



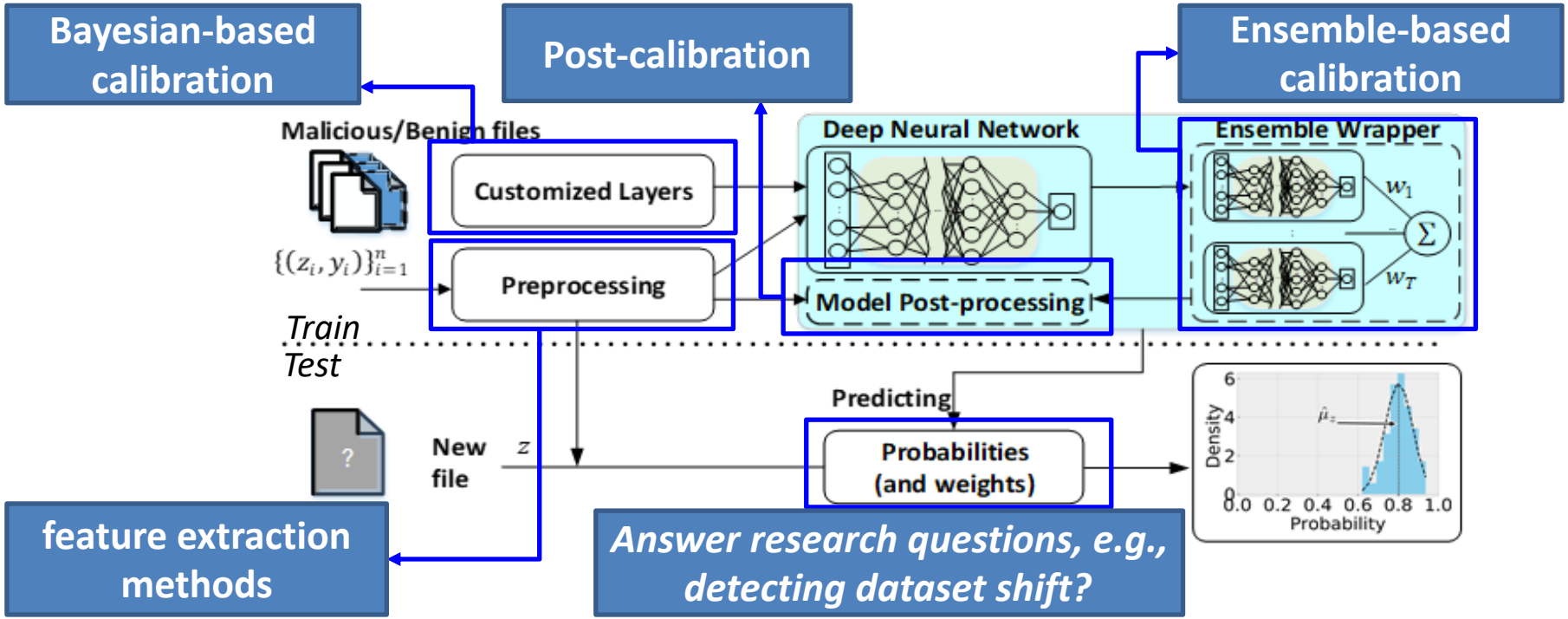
- **Q: Can we Leverage predictive uncertainty to detect $p^*(z) \neq p'(z)$ in Android malware detection?**
  - **How to cope with different types of malware detectors?**
  - **How to cope with different types of calibration methods?**
  - **How to measure uncertainty?**

Outline

- ❑ Introduction
- ❑ Problem Statement
- ❑ **Framework**
- ❑ Case Study
- ❑ Conclusion

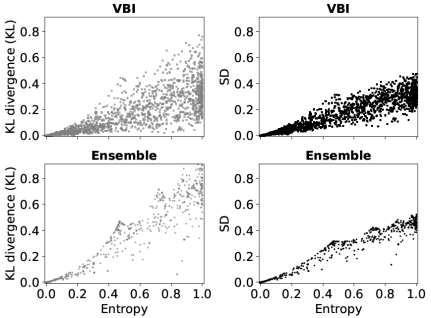
Framework: Calibrating Malware Detectors

Basic idea: Calibrating DNN-based malware detectors



Measuring uncertainty

- Uncertainty when D_{test} has ground-truth labels (proper scoring rules)
  - Negative Log-Likelihood (NLL): The loss function is used for training model
  - Brier Score Error (BSE): Measuring accuracy of predictive probability by $\mathbb{E}_{(z,y)\in D_{test}}\big(y - p(y=1|z,\theta)\big)^2$
  - Expected Calibration Error (ECE): alike to BSE yet in a bin-wise manner and each bin weighted by # of malware examples; eliminating the bin weights leads to un-weighted ECE (uECE).
  - Balanced NLL (bNLL): Averaging NLLs that are computed upon test examples of per-class; bBSE
- Uncertainty when D_{test} has no ground-truth labels given (ood detection)
  - Entropy: measuring the state of disorder in a physical system
  - Standard Deviation (SD): measuring inconsistencies between models
  - KL divergence: An alternative to SD



Outline

- ❑ Introduction
- ❑ Problem Statement
- ❑ Framework
- ❑ Case Study
- ❑ Conclusion

Case Study

- **Malware detectors considered:** Representative, reproducible, and different DNN-based malware detectors
  - Two Multiple Layer Perceptron (MLP)-based models: ***DeepDrebin*** and ***MultimodalNN***;
  - A text *Convolutional Neural Network* (CNN) based model: ***DeepDroid***;
  - A *Long Short Term Memory* (LSTM)-based model: ***Droidetec.***
  - Note: These Malware detectors use different static features.
- **Calibration methods considered:** Effective, scalable, and applicable
  - No effort made for calibration: ***Vanilla***;
  - A post-processing method: ***Temperature scaling*** (Temp scaling);
  - Two approximate Bayesian inference methods: ***Monte Carlo dropout*** (MC dropout) and ***Variational Bayesian Inference*** (VBI);
  - Two ensemble learning based methods: ***Deep Ensemble*** (Ensemble) and ***Weighted Deep Ensemble*** (wEnsemble).

Datasets

- **Drebin**
  - 5,560 malicious APKs and 42,333 benign ones; 60% for training, 20% for validation, the reminded 20% for in-distribution test
  - Adversarial examples are generated by perturbing malware in the testing dataset
- **VirusShare**
  - 12,383 malicious APKs and 340 benign ones; serving as the out of source data to Drebin;
- **Androzoo**
  - 12,735 malicious APKs and 116,993 benign ones; spanning from Jan. 2014 to Dec. 2016; 83.4% APKs in 2014 for training, 8.33% for validation and 8.33% for in-distribution testing;
  - The reminded data (i.e., Jan. 2015 to Dec.2016) for the temporal testing purpose

Answering RQ1

❑ RQ1: What's the predictive uncertainty *in absence of dataset shift?*

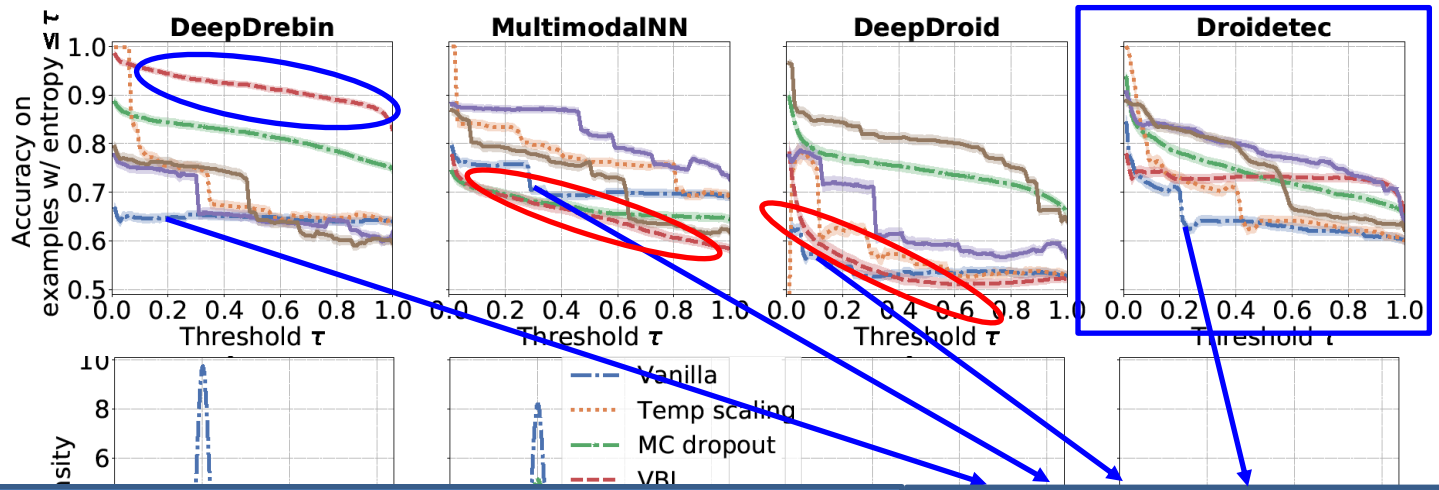
- ❖ Malware detectors are trained on Drebin training and tested on Drebin test dataset

Detector	Calibration	Accuracy (%)	NLL	bNLL	BSE	bBSE	ECE	uECE
DeepDrebin	Vanilla	99.28	**0.100**	**0.329**	0.007	**0.020**	0.006	0.104
	Temp scaling	99.28	0.052	0.109	0.006	0.018	0.007	0.062
	MC dropout	99.32	**0.033**	**0.094**	0.006	**0.015**	**0.002**	**0.056**
	VBI	**98.88**	0.054	**0.094**	**0.009**	0.016	**0.012**	0.102
	Ensemble	**99.37**	0.063	0.211	**0.005**	0.018	0.005	**0.160**
	wEnsemble	99.36	0.058	0.190	**0.005**	0.018	0.004	0.095

Observation: In terms of consistency results, NLL and BSE suffer from the imbalanced dataset and their balanced versions relieve this issue

Answering RQ2

- **RQ2: What's the predictive uncertainty w.r.t. out-of-source examples?**
  - Malware detectors are trained on Drebin dataset and tested on VirusShare dataset



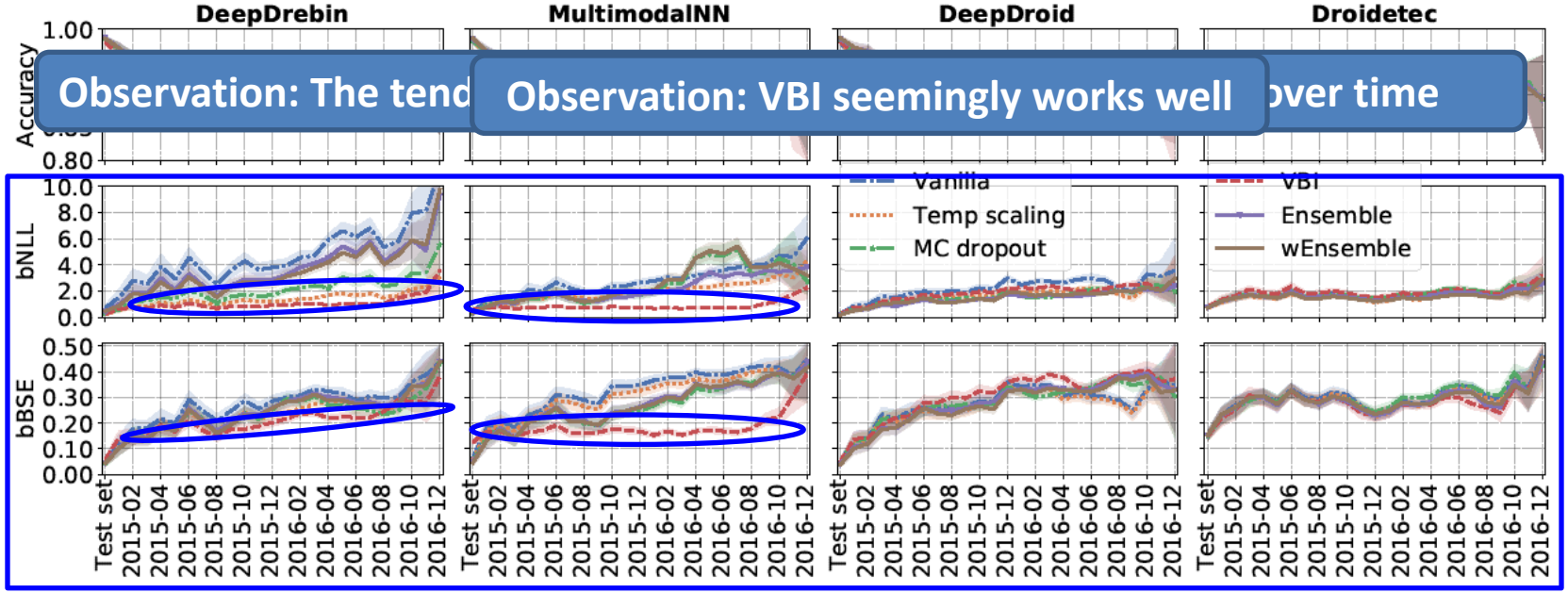
Observation: VBI can achieve the notably quality of uncertainty when applied to calibrating the simple n

Observation: Vanilla models are poorly-calibrated

Observation: Non-robust feature extraction may contribute to poor calibration

Answering RQ3

- **RQ3: What's the predictive uncertainty under temporal covariate shift?**
  - Malware detectors are trained on Androzoo dataset from 2014 and tested on Androzoo dataset from 2015 and 2016 monthly



Answering RQ4

❑ RQ4: What's the predictive uncertainty under adversarial attacks?

- ❖ **Malware detectors are trained on Drebin dataset and tested on 1,100 adversarial malware examples (generated from a *surrogate DeepDrebin* model)**

Detector	Calibration	No attack	"Max" PGDs+GDKDE attack			Mimicry		
		Acc. (%)	Acc. (%)	NLL	BSE	Acc. (%)	NLL	BSE
DeepDrebin	Vanilla	96.09	**0.00**	33.22	1.000	66.09	4.778	0.317
	Temp scaling	96.09	**0.00**	**7.015**	**0.985**	66.09	1.427	0.266
	MC dropout	**96.55**	**0.00**	33.22	1.000	69.18	1.639	0.245
	VBI	96.27	**0.00**	33.22	1.000	**69.91**	**1.034**	**0.211**
	Ensemble	96.00	**0.00**	33.22	1.000	64.82	3.296	0.295
	wEnsemble	96.00	**0.00**	33.22	1.000	64.64	2.944	0.296

Answering RQ4

❑ RQ4: What's the predictive uncertainty under adversarial attacks?

Observation: Adversarial attacks can make the uncertainty quantification useless; Robust features may be prerequisite to uncertainty quantification

Detector	Calibration	No attack	"Max" PGDs+GDKDE attack			Mimicry		
		Acc. (%)	Acc. (%)	NLL	BSE	Acc. (%)	NLL	BSE
DeepDroid	Vanilla	91.55	85.45	0.773	0.116	86.09	0.786	0.110
	Temp scaling	91.55	85.45	0.536	0.105	86.09	0.538	0.101
	MC dropout	92.55	**93.55**	**0.273**	**0.048**	90.18	0.529	0.083
	VBI	87.27	84.00	0.592	0.117	82.00	0.705	0.136
	Ensemble	92.55	90.64	0.366	0.068	89.55	0.433	0.079
	wEnsemble	**95.00**	93.00	0.309	0.057	**92.82**	**0.348**	**0.06**

Outline

- ❑ Introduction
- ❑ Problem Statement
- ❑ Framework
- ❑ Case Study
- ❑ Conclusion

Conclusion: Initial Insights

- VBI is promising to calibrate and generalize *simple* malware detectors to deal with dataset shift
- Adversarial evasion attacks can render calibration methods useless (i.e., malware detectors return incorrect label with high confidence)
- Uncertainty quantification can be leveraged to detect dataset shift but may not be able to cope with adversarial examples
  - Possible cause: non-robust features and/or adversarial examples cause worst-case dataset shift

Thank you!

E-mail: lideqiang@njust.edu.cn

Codes can be found: https://www.github.com/deqangss/malware-uncertainty