Quantitative steganalysis using rich models

Jan Kodovský, Jessica Fridrich

February 7, 2013 / SPIE

BINGHAMTON
UNIVERSITY
STATE UNIVERSITY OF NEW YORK

Background

Quantitative steganalysis

- Estimates the message length (number of embedding changes)
- Structural attacks
  - Leverage knowledge of the embedding mechanism
  - Accurate, but limited to simple steganographic methods
  - LSBR: RS analysis, Sample Pairs, WS steganalysis
  - Difficult for more advanced schemes (HUGO, WOW)
- Machine-learning approach
  - Regression in a pre-defined feature space [Pevný, SPIE 2009]
  - Applicable to arbitrary steganographic method
  - Need for a training phase, cover-source dependence

Background

Rich models [Fridrich, 2012]

- High-dimensional spaces capturing complex statistical dependencies
- Improve classical (binary) steganalysis
- Require scalable machine learning algorithm
  - FLD-based ensemble classifier [Kodovský, 2012]
  - Ensemble average perceptron [Lubenko, 2012]

Contribution of this paper

- Extension of feature-based quantitative steganalysis to rich models
- Reasonable to expect similar improvement in the quantitative setting

Scalable regression framework

- Combination of simple regressors trained on random subspaces?
- Caveat: distortion is **not** linear in payload (coding, adaptivity)
- Ability to capture non-linear relationships is crucial

Regression framework

Objective: find regression function F minimizing the training error

Training error ... $\sum_{i=1}^{N} L\left(y_i, F(\mathbf{x}_i)\right)$

- $\mathbf{x}_i \in \mathcal{F}$ ... training feature vectors; $y_i \in [0, 1]$, relative payloads
- $F : \mathcal{F} \to \mathbb{R}$ ... regression function; $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ ... loss function

Generalized additive model (additive expansion)

$$F(\mathbf{x}; \{\mathbf{a}_m\}_1^M) = \sum_{m=1}^{M} h(\mathbf{x}; \mathbf{a}_m)$$

- $h(\mathbf{x}; \mathbf{a})$ ... base function (parametrized by $\mathbf{a}$)
- Optimization problem $\to$ find parameter vectors $\{\mathbf{a}_m\}_{m=1}^M$

Gradient boosting

Simplified algorithm (squared loss function) [Friedman, 2000]

- Incremental improvement of the solution
- In each stage, add base function minimizing the training error

Initialize $F_0(\mathbf{x})$

for $m = 1$ to M do

$e_i \leftarrow y_i - F_{m-1}(\mathbf{x}_i),\ i = 1, \ldots, N$ (update current error)

$\mathbf{a}_m = \arg\min_{\mathbf{a}} \sum_{i=1}^{N} \left[e_i - h(\mathbf{x}_i; \mathbf{a})\right]^2$ (least square regression)

$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \ h(\mathbf{x}; \mathbf{a}_m)$ (update regression function)

end

Gradient boosting

Simplified algorithm (squared loss function) [Friedman, 2000]

- Incremental improvement of the solution
- In each stage, add base function minimizing the training error

Initialize $F_0(\mathbf{x})$

for $m = 1$ to M do

$e_i \leftarrow y_i - F_{m-1}(\mathbf{x}_i),\ i = 1, \ldots, N$ (update current error)

$\mathbf{a}_m = \arg\min_{\mathbf{a}} \sum_{i=1}^{N} \left[e_i - h(\mathbf{x}_i; \mathbf{a})\right]^2$ (least square regression)

$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \eta h(\mathbf{x}; \mathbf{a}_m)$ (update regression function)

end

η: regularization parameter

Choice of base functions

Requirements

- Capture non-linear relationships, capable of local estimation
- Computationally inexpensive

Regression trees [Breiman,1984]

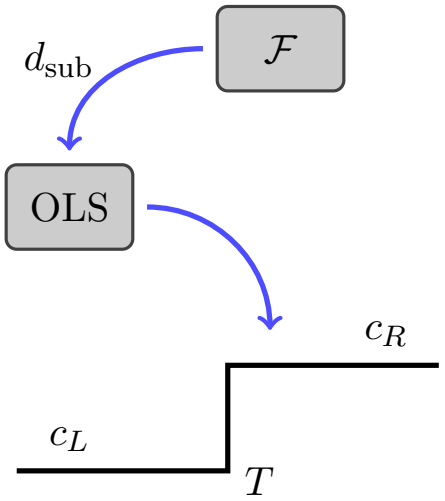
- Simple, yet powerful in ensemble systems
- Splitting decision based on a single feature

Modified splitting criterion

1. Form random subspace of the feature space $\mathcal{F}$
2. Train OLS in this subspace ... $B(\mathbf{x})$
3. Form a step function

$$B'(\mathbf{x}; T, c_L, c_R) = \begin{cases} c_L & \text{if } B(\mathbf{x}) \leq T, \\ c_R & \text{otherwise,} \end{cases}$$

4. Repeat recursively on both parts



Exploratory experiments

Image database

- BOSSbase images, 50/50 training/testing

Steganography

- Steganographic algorithm HUGO [Pevný, 2011]
- Relative payload uniformly distributed over [0,1] bpp

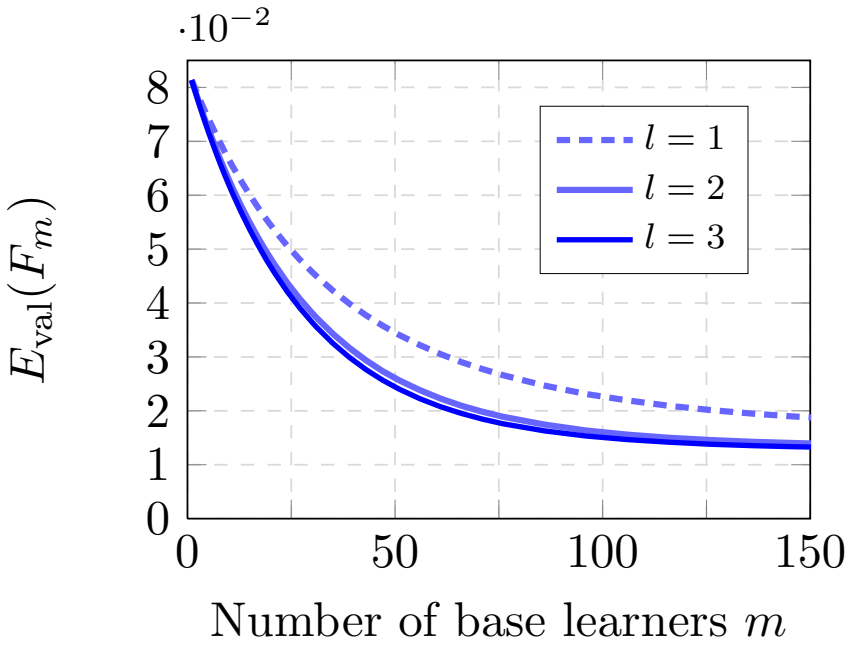
Steganalysis

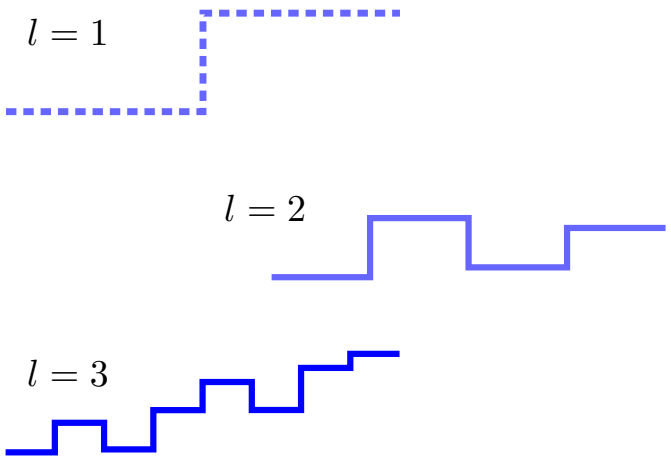
- Proposed gradient boosting system with different parameters
- Feature space $\mathcal{F}$ . . . spatial rich model SRMQ1 (dimension 12,753)

Performance evaluation

- Mean square error achieved on the validation set

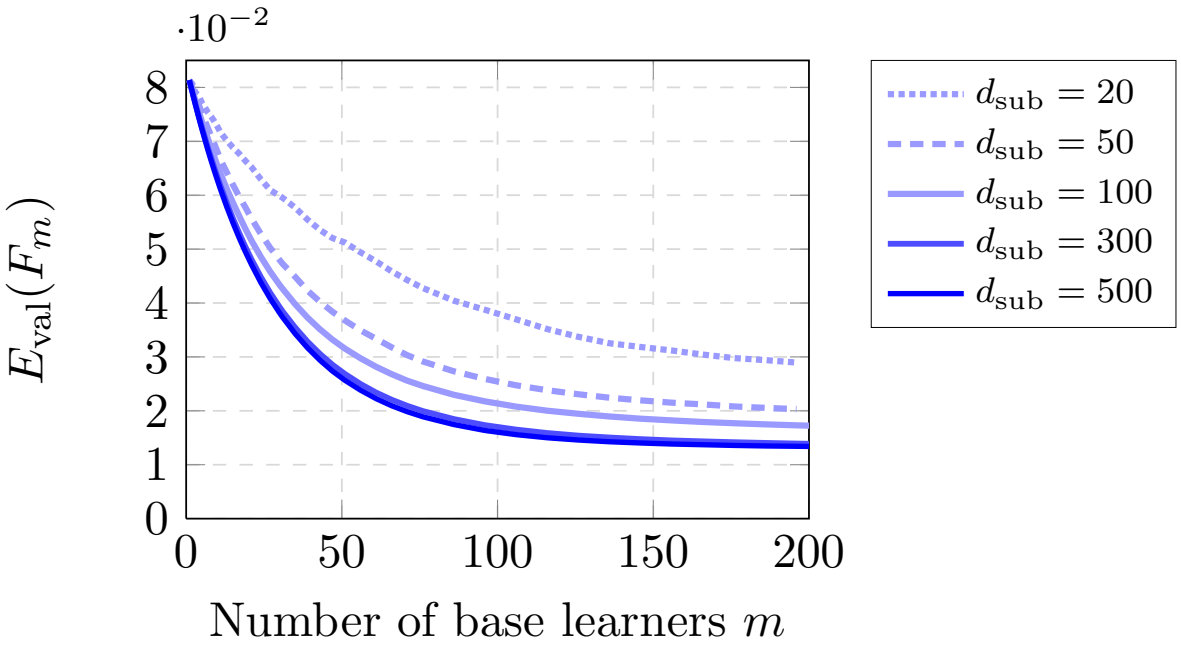
Effects of the tree depth l





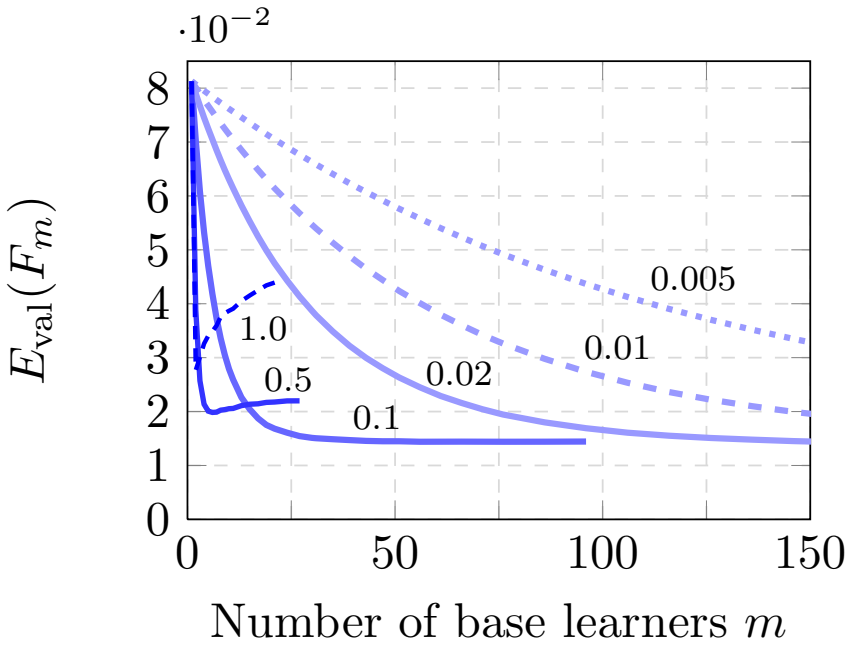
- Exponentially growing number of OLS regressions
- $l = 2$ ... gained ability of **local** estimation
- Further increase of l does not bring any additional improvement

Effects of the subspace dimension d_{sub}



- Quick improvement and saturation around $d_{\text{sub}} = 500$
- Can be automatized through an additional 1D search
- Note: complexity grows quadratically with d_{sub}

Effects of the regularization η



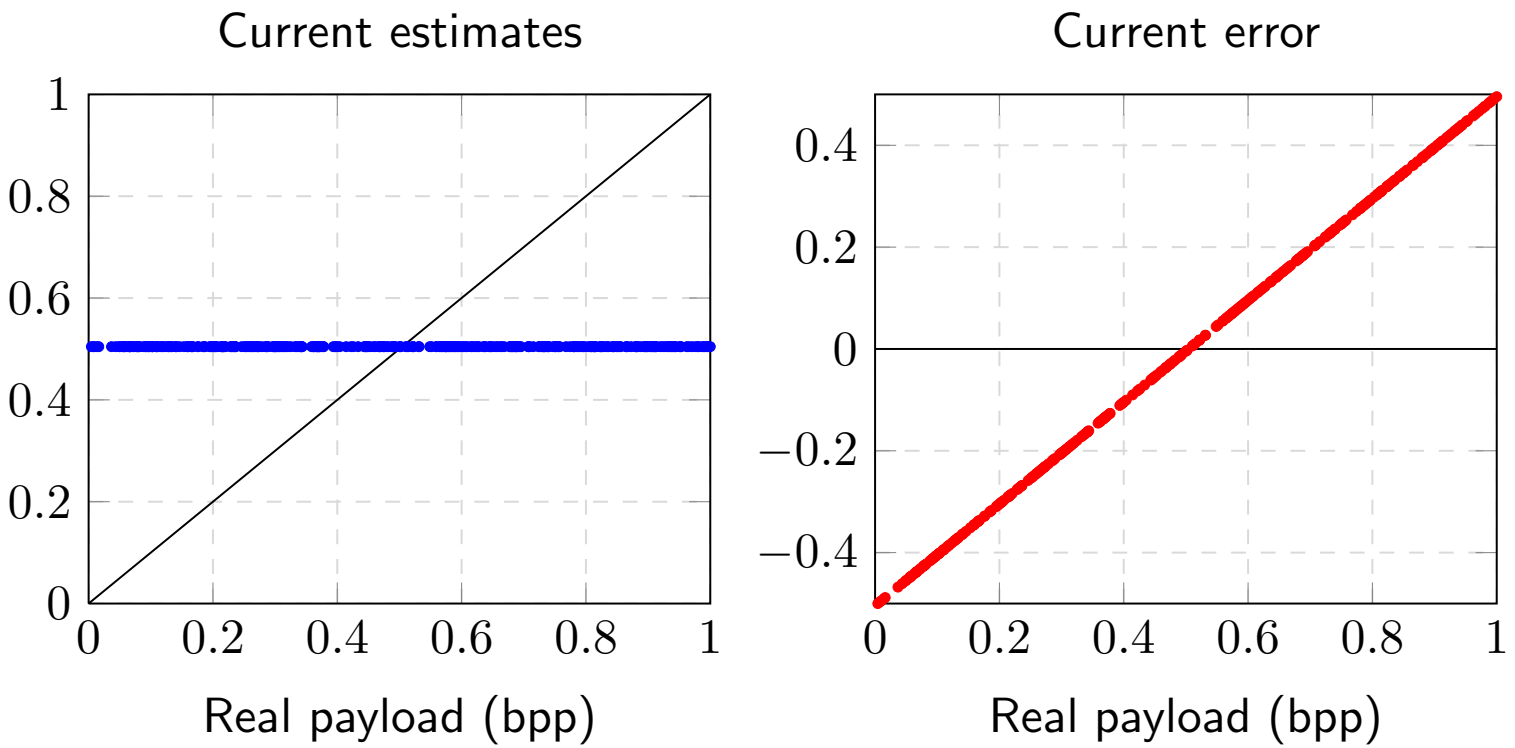
η	M	$E_{\text{val}}(F_M)$	Time (sec)
1.0	2	$2.77 \cdot 10^{-2}$	5
0.5	6	$1.98 \cdot 10^{-2}$	8
0.1	75	$1.44 \cdot 10^{-2}$	60
0.02	335	$1.35 \cdot 10^{-2}$	255
0.01	636	$1.33 \cdot 10^{-2}$	476
0.005	1,104	$1.32 \cdot 10^{-2}$	828

- Lower η prevents overtraining (regularization)
- More base learners (and thus more random subspaces)

  $\Rightarrow$ extracts more information from the feature space $\mathcal{F}$

Short simulation

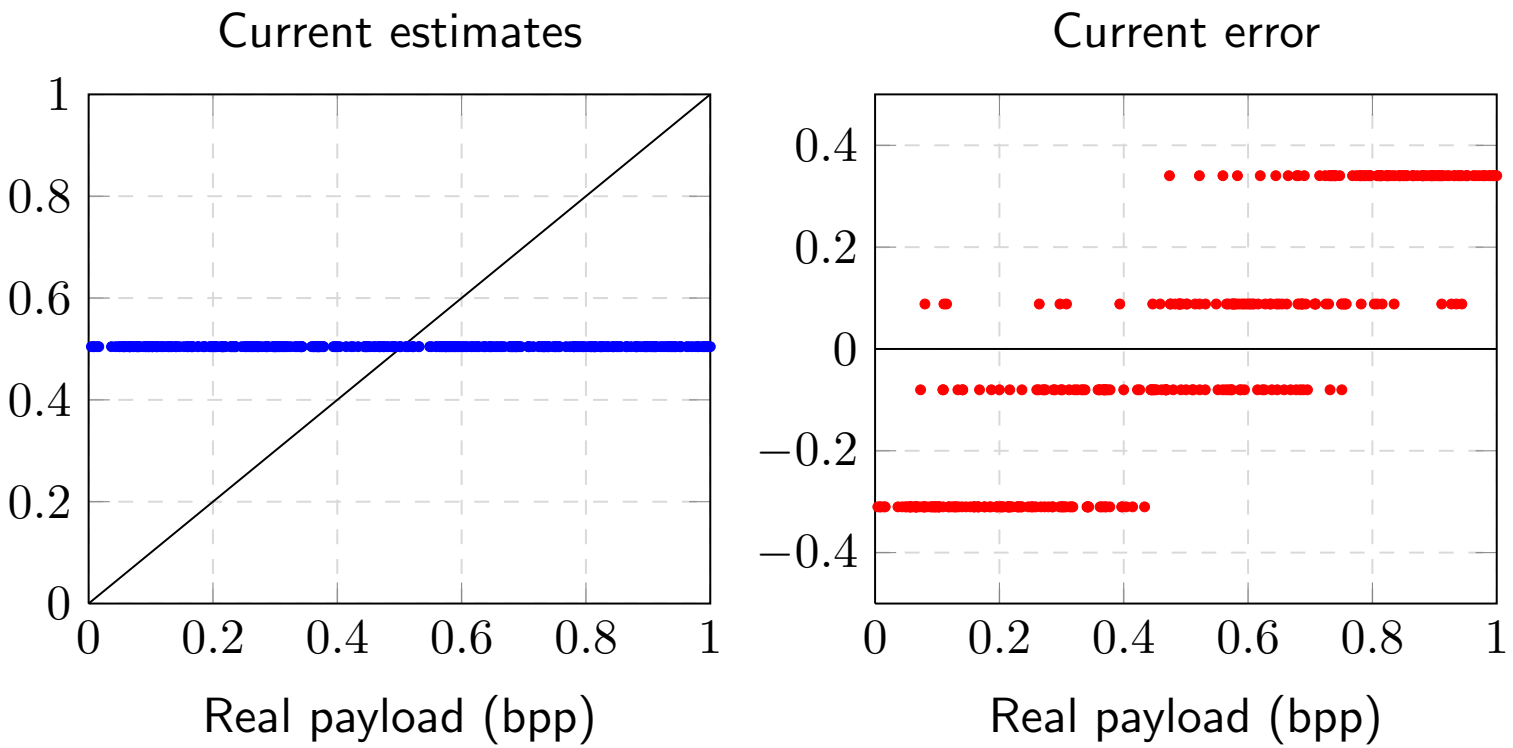
stage 1



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

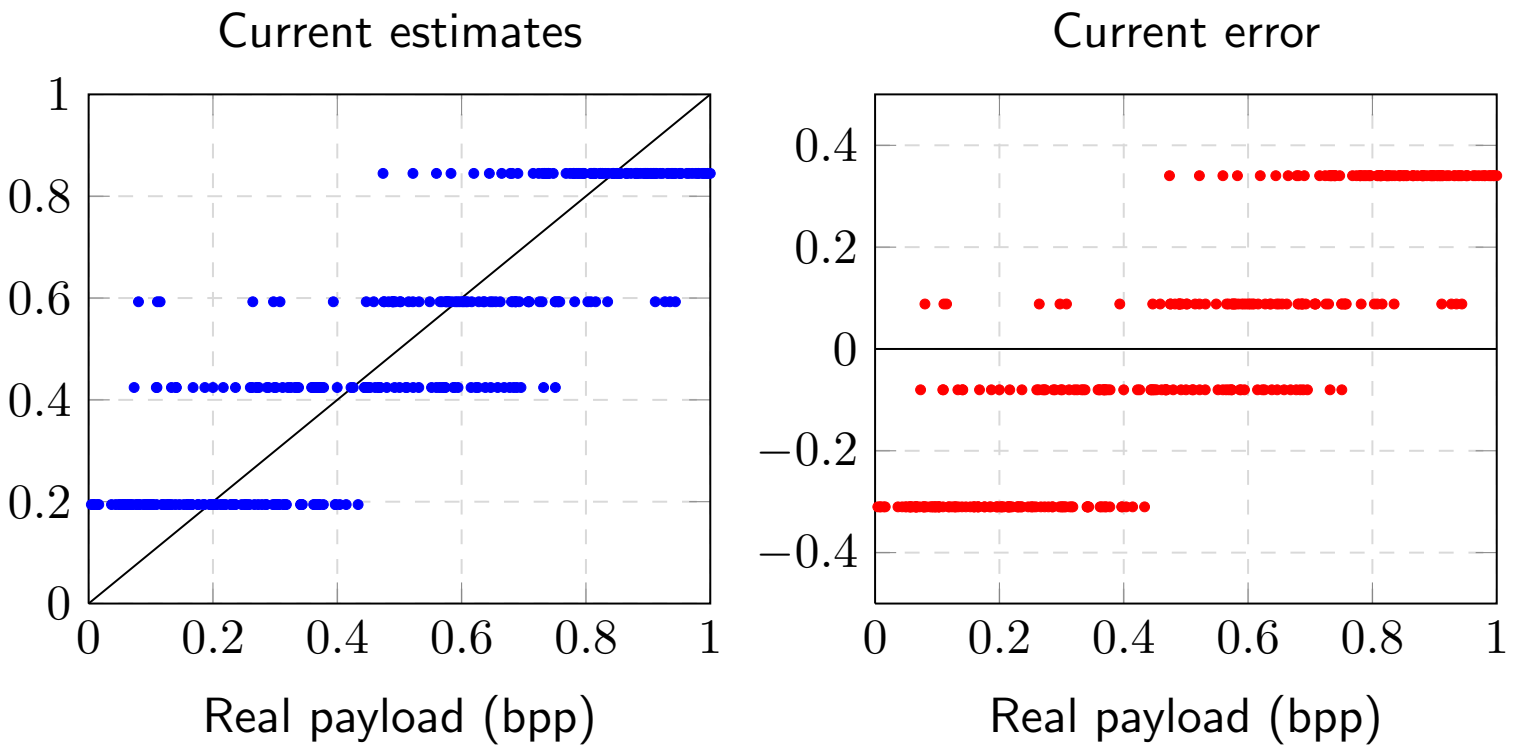
stage 1



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

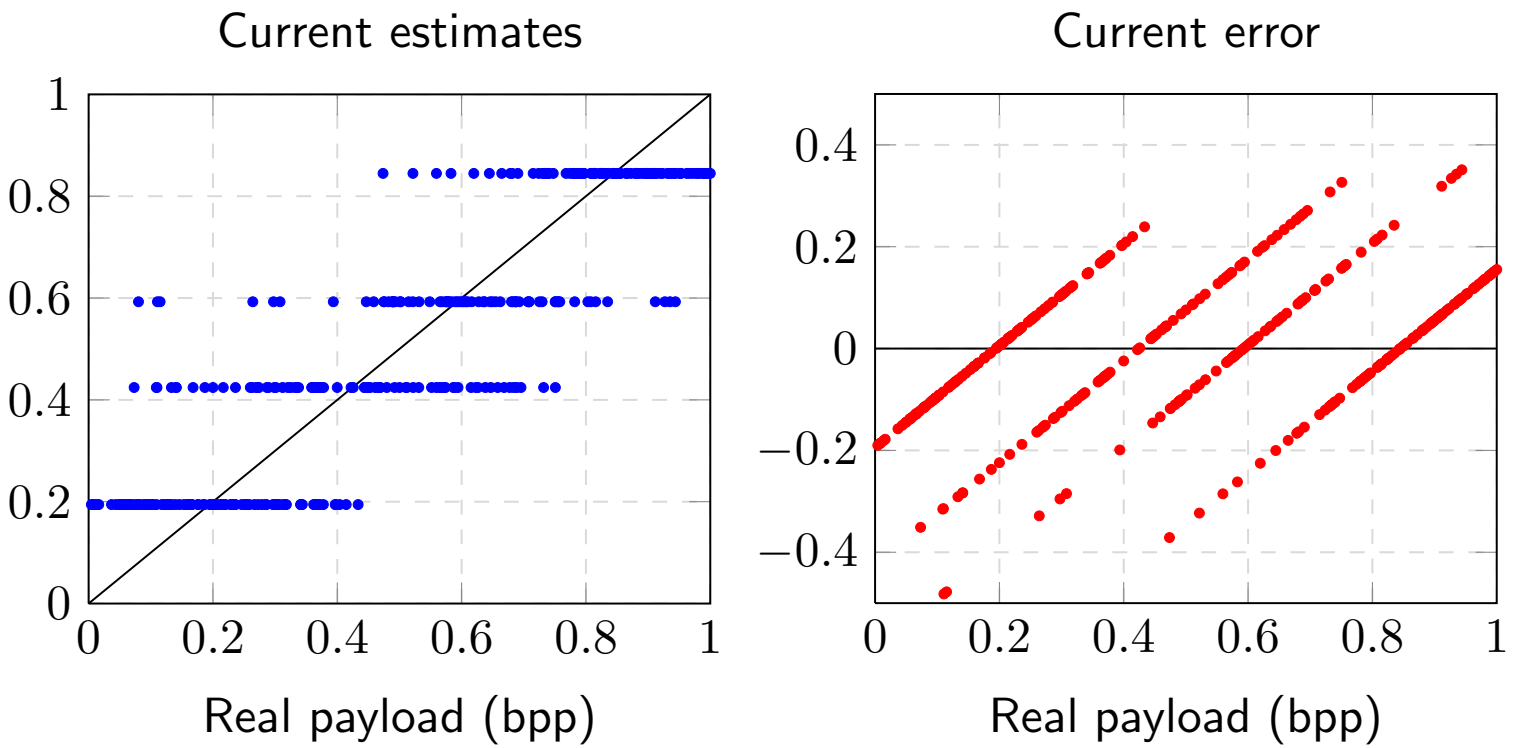
stage 1



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

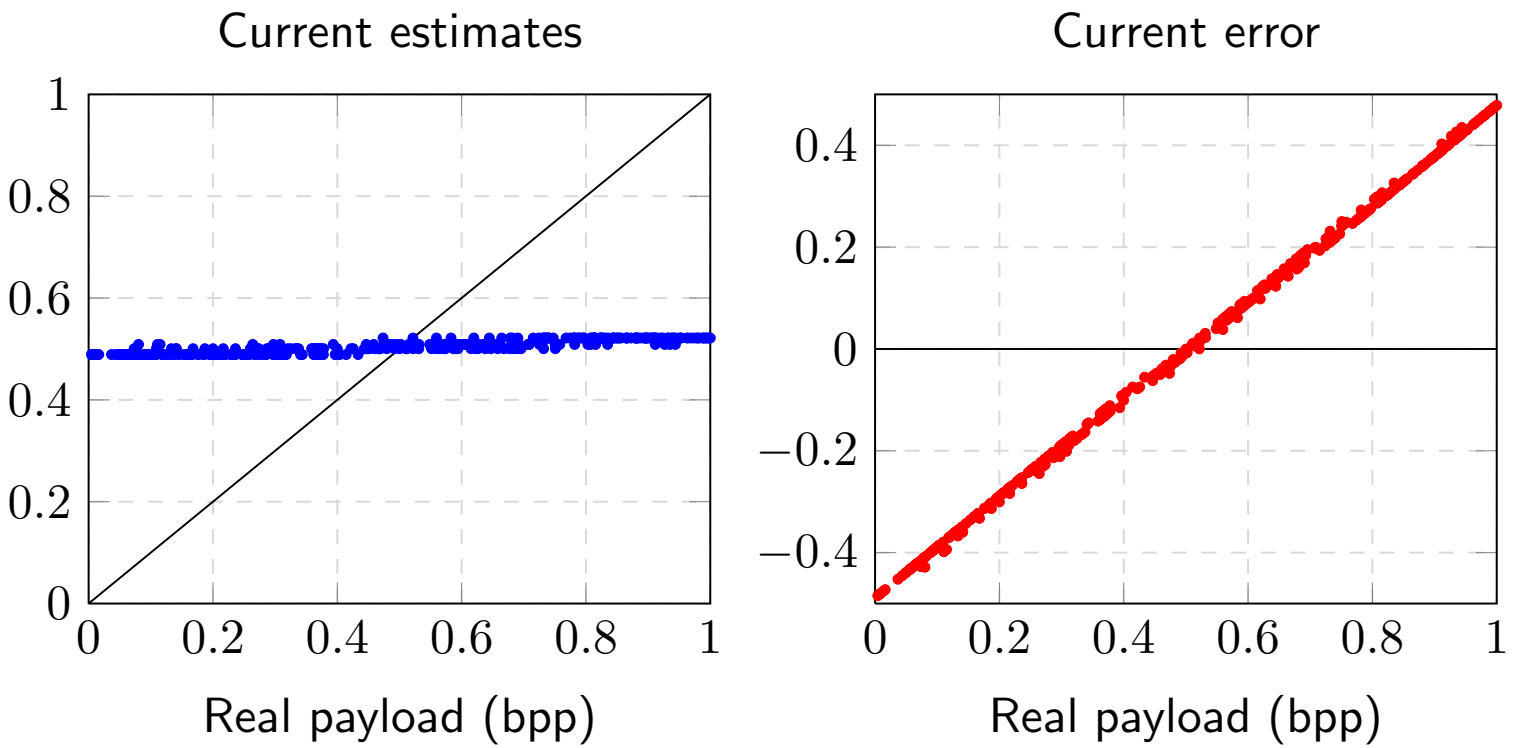
stage 1



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

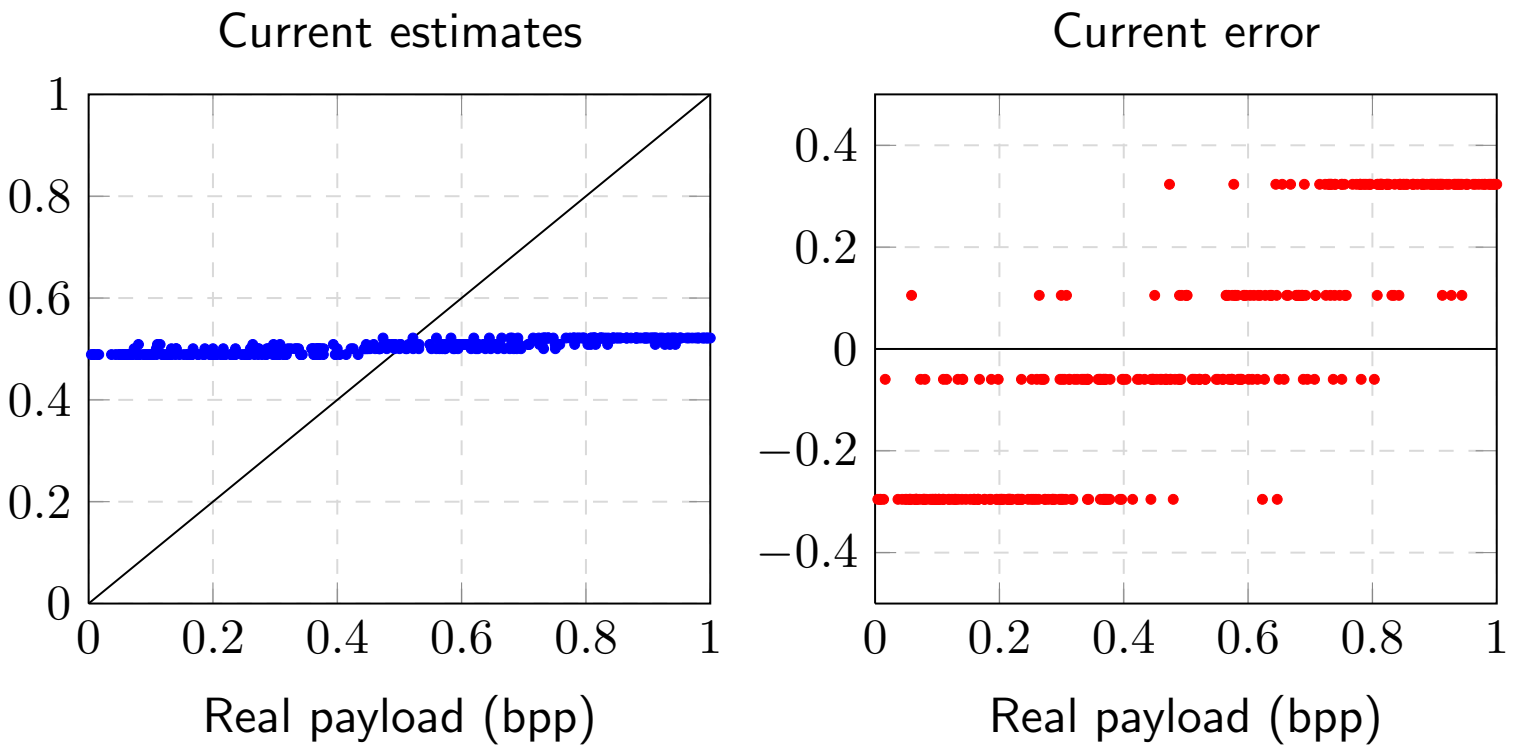
stage 1



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

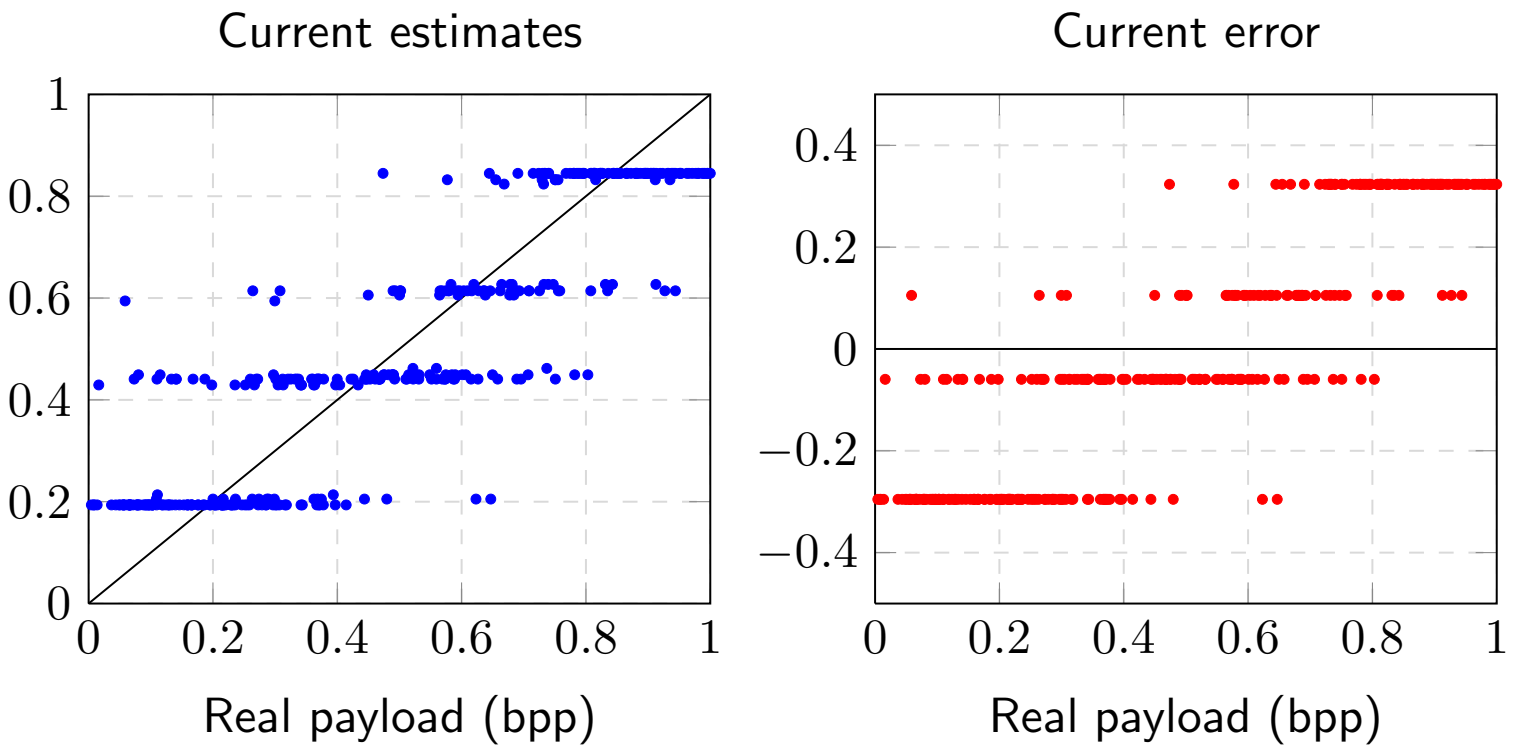
stage 2



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

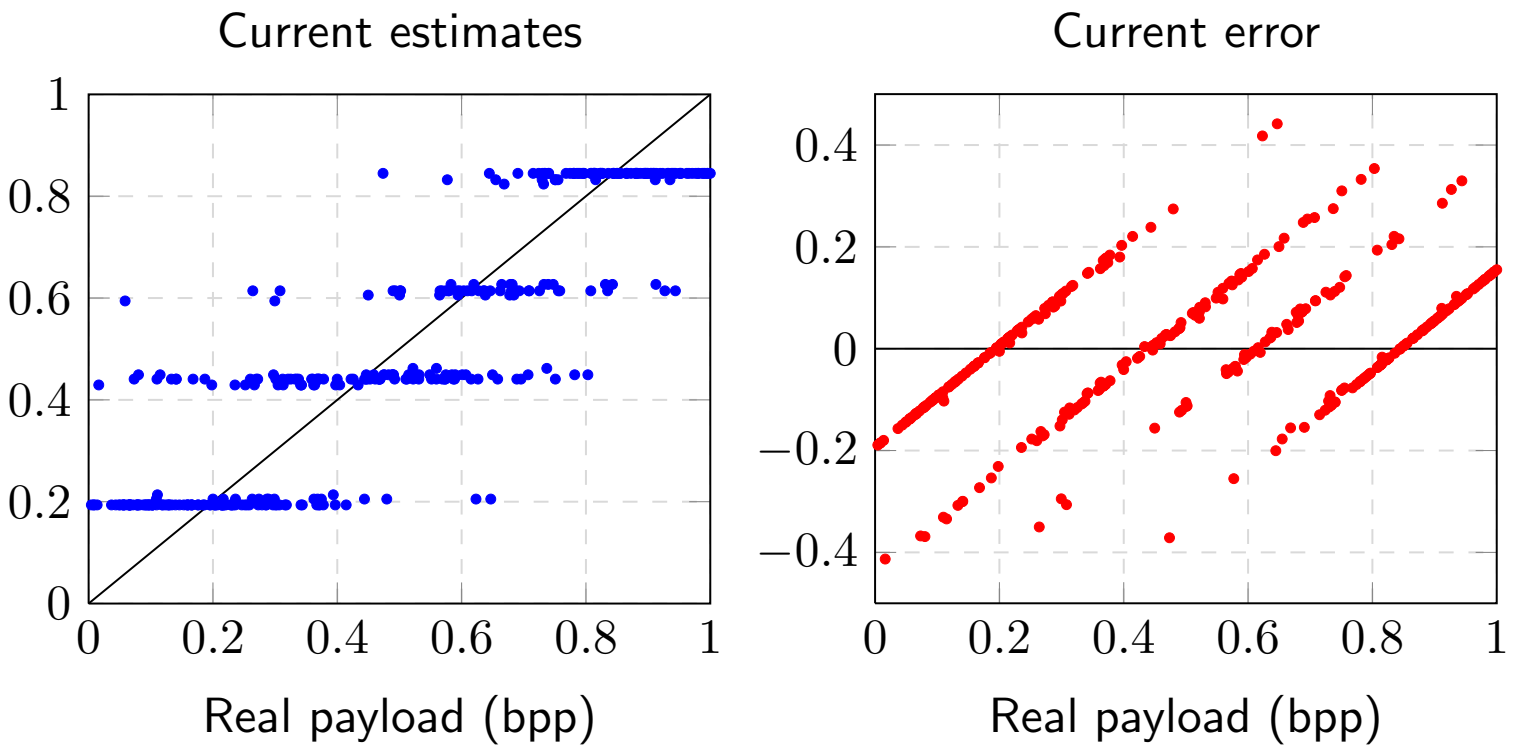
stage 2



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

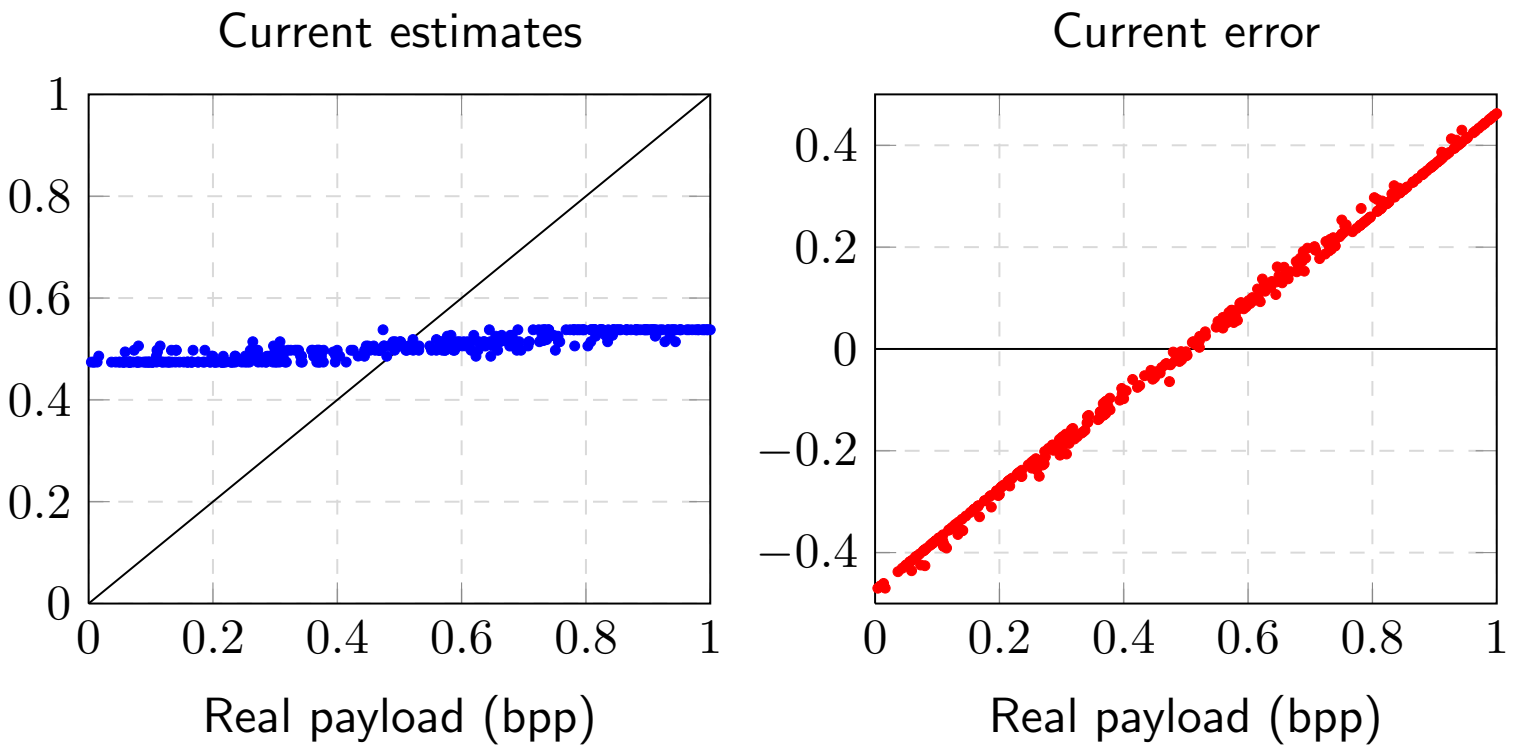
stage 2



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

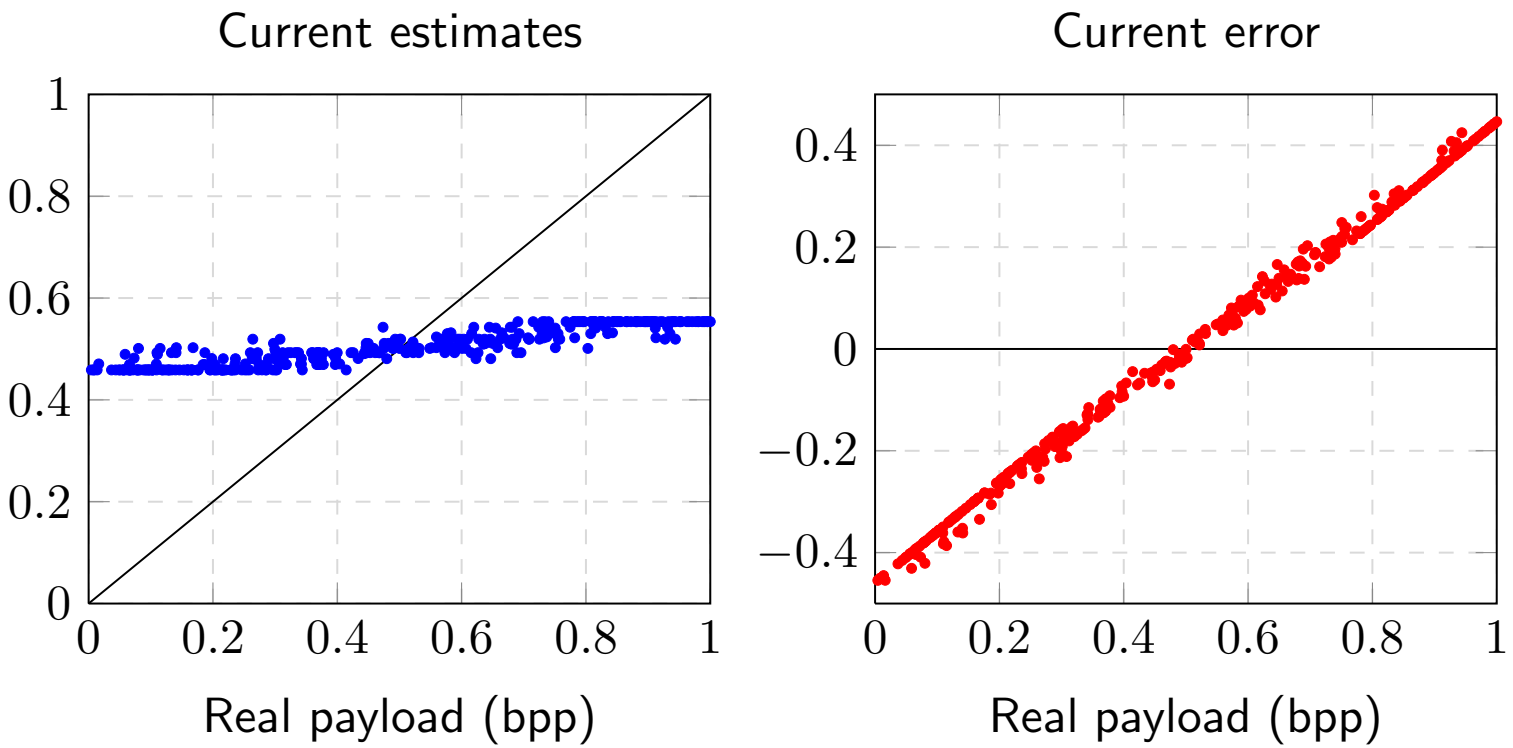
stage 2



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

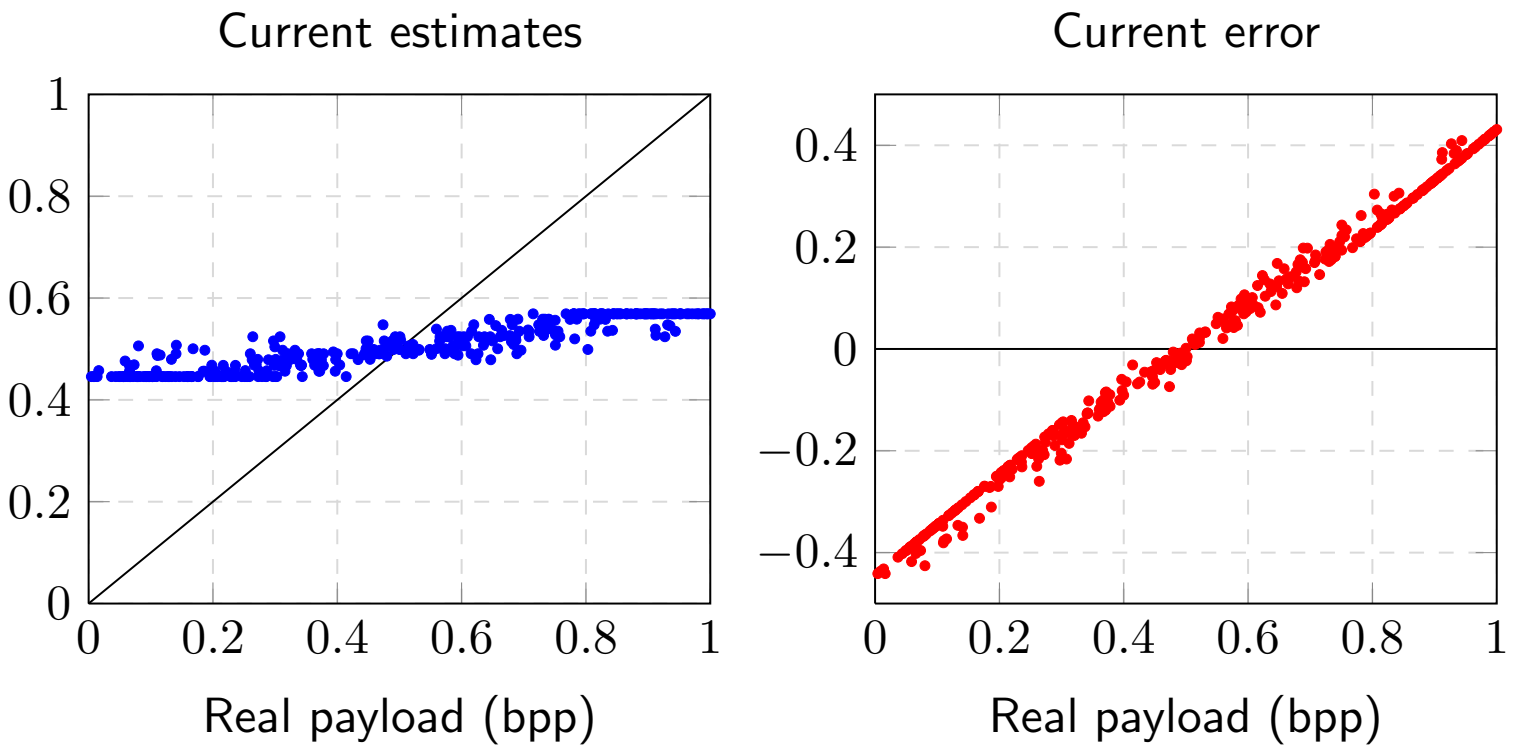
stage 3



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

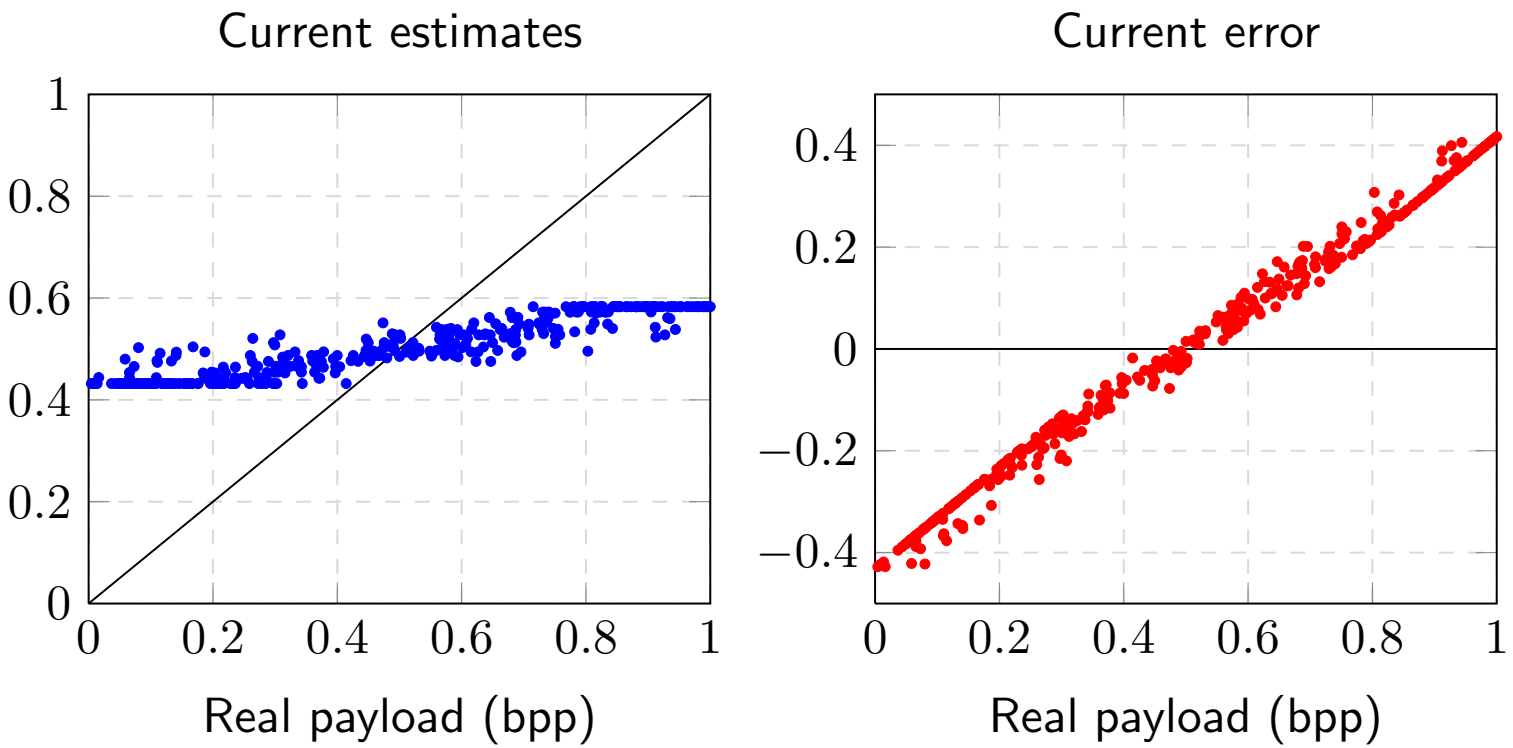
stage 4



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

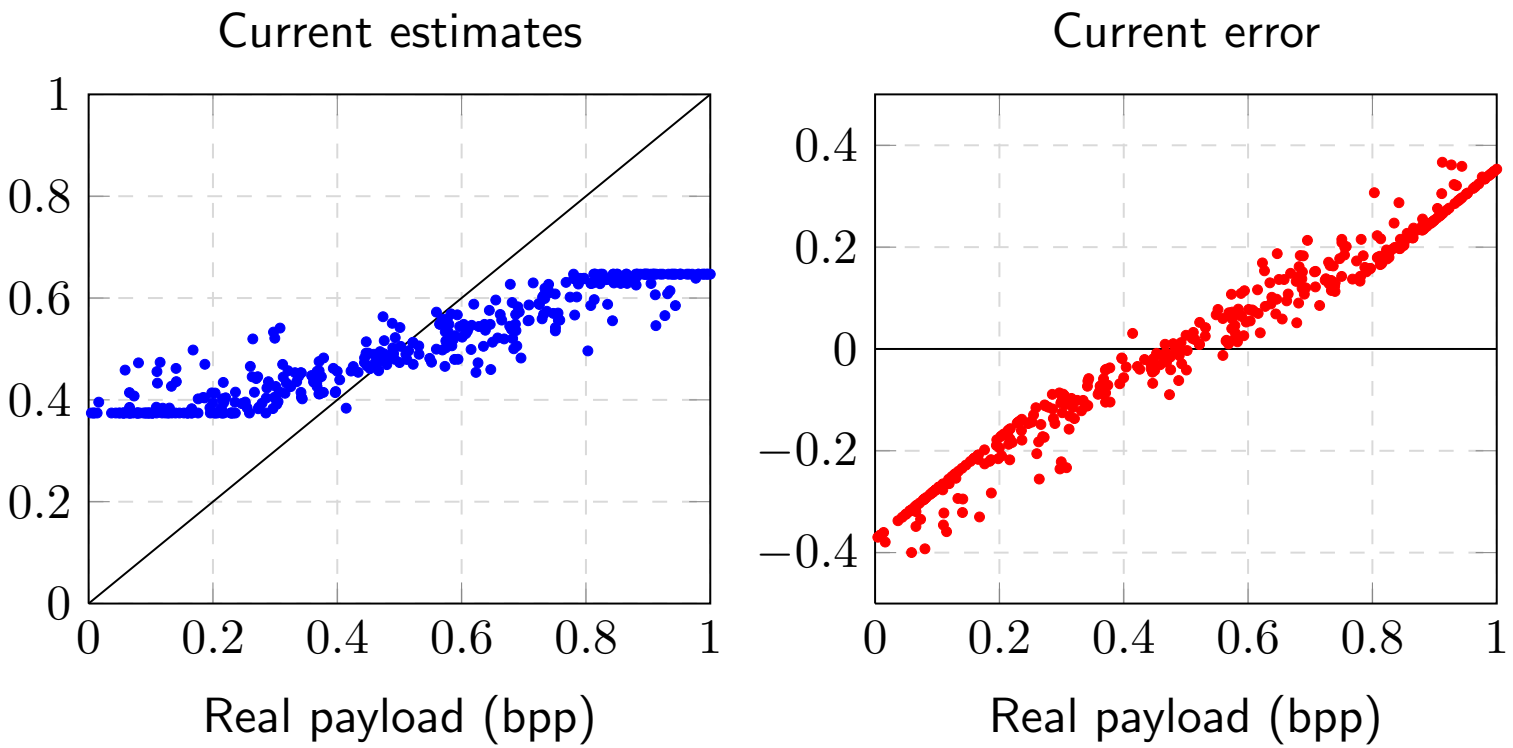
stage 5



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

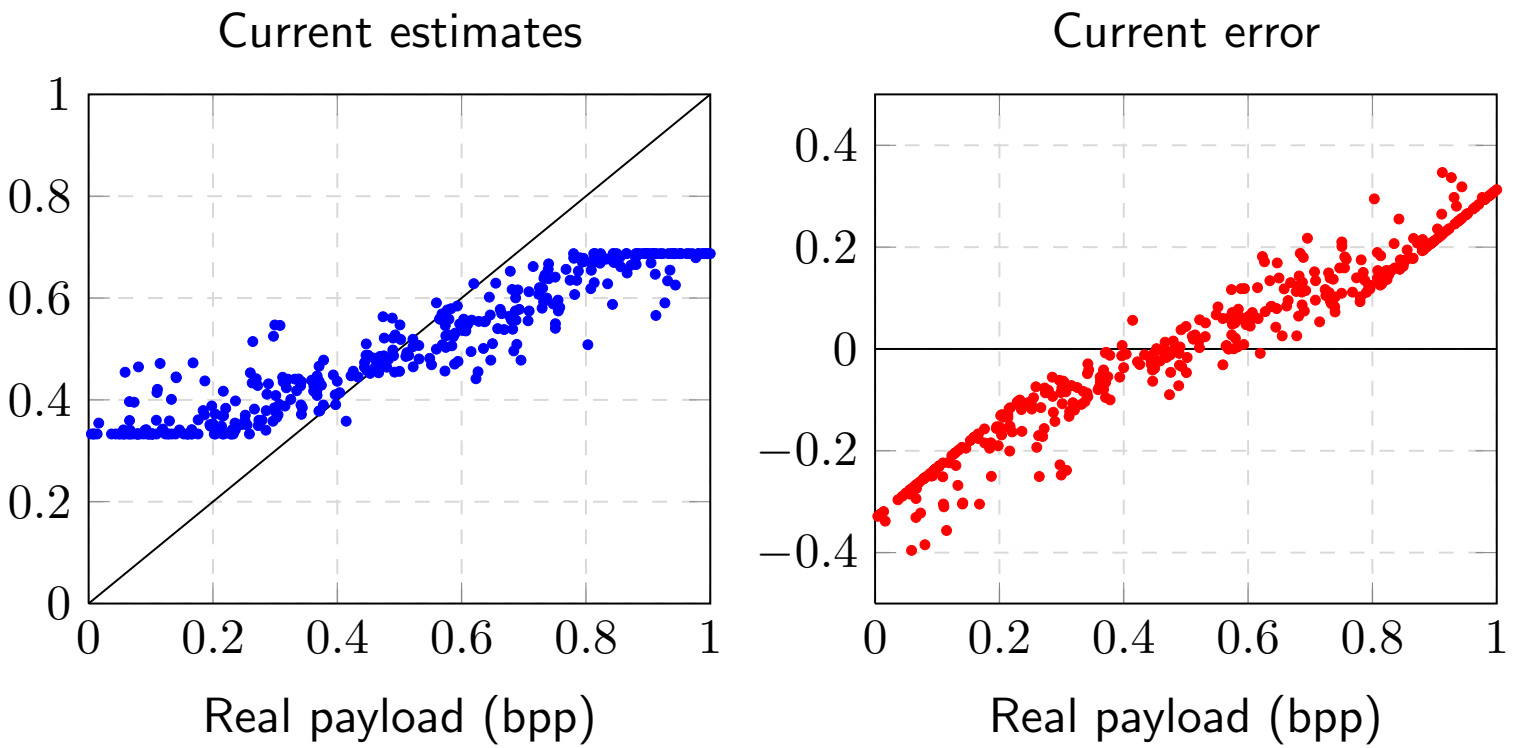
stage 10



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

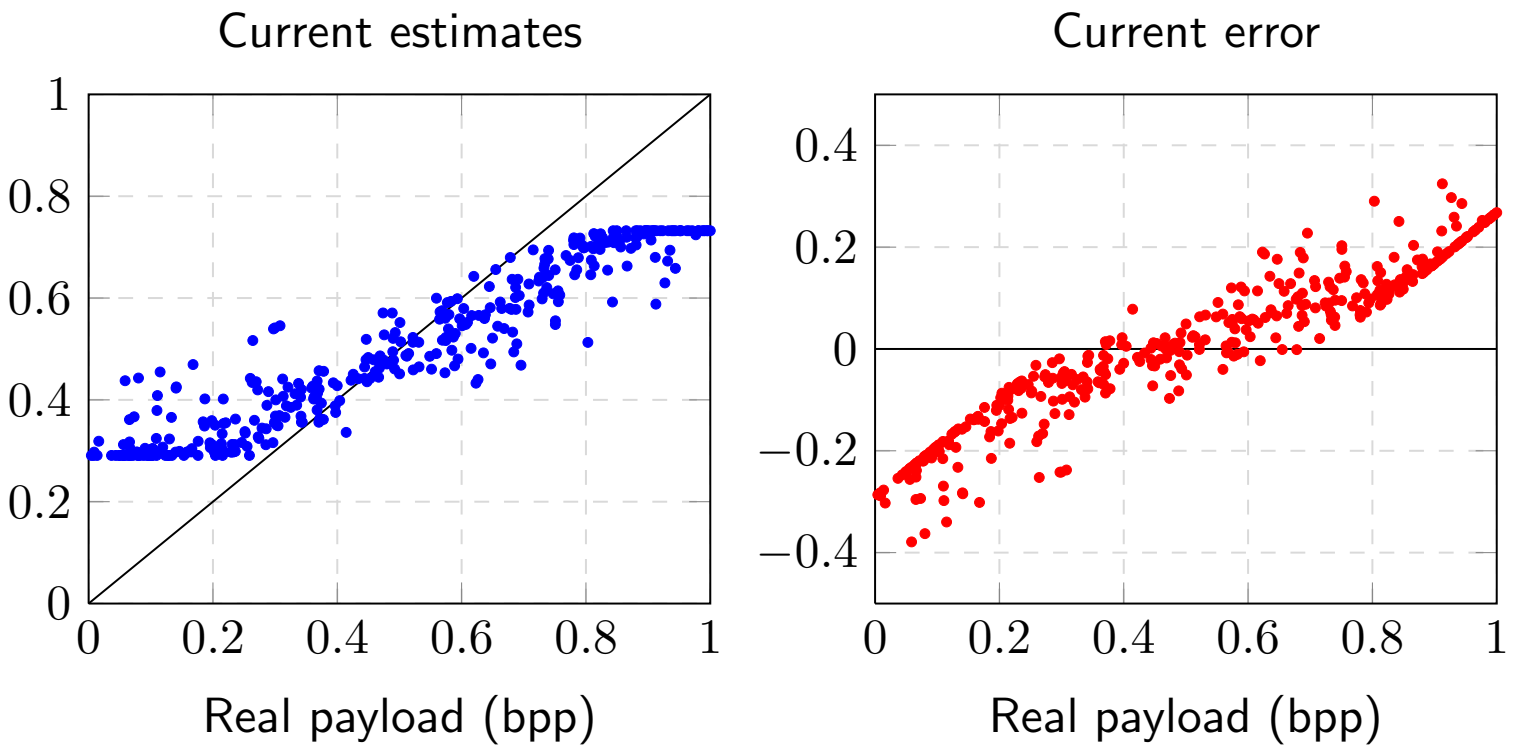
stage 15



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

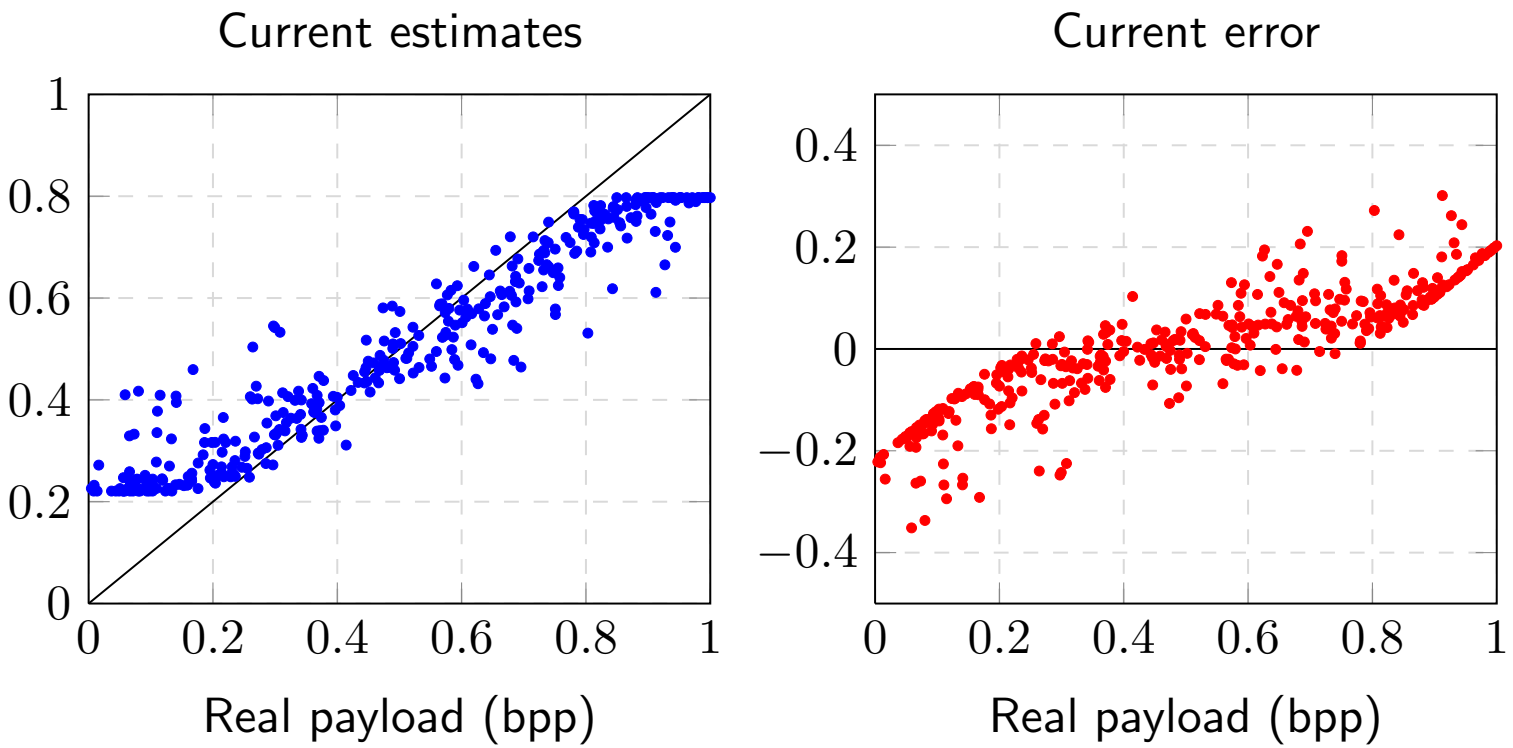
stage 20



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

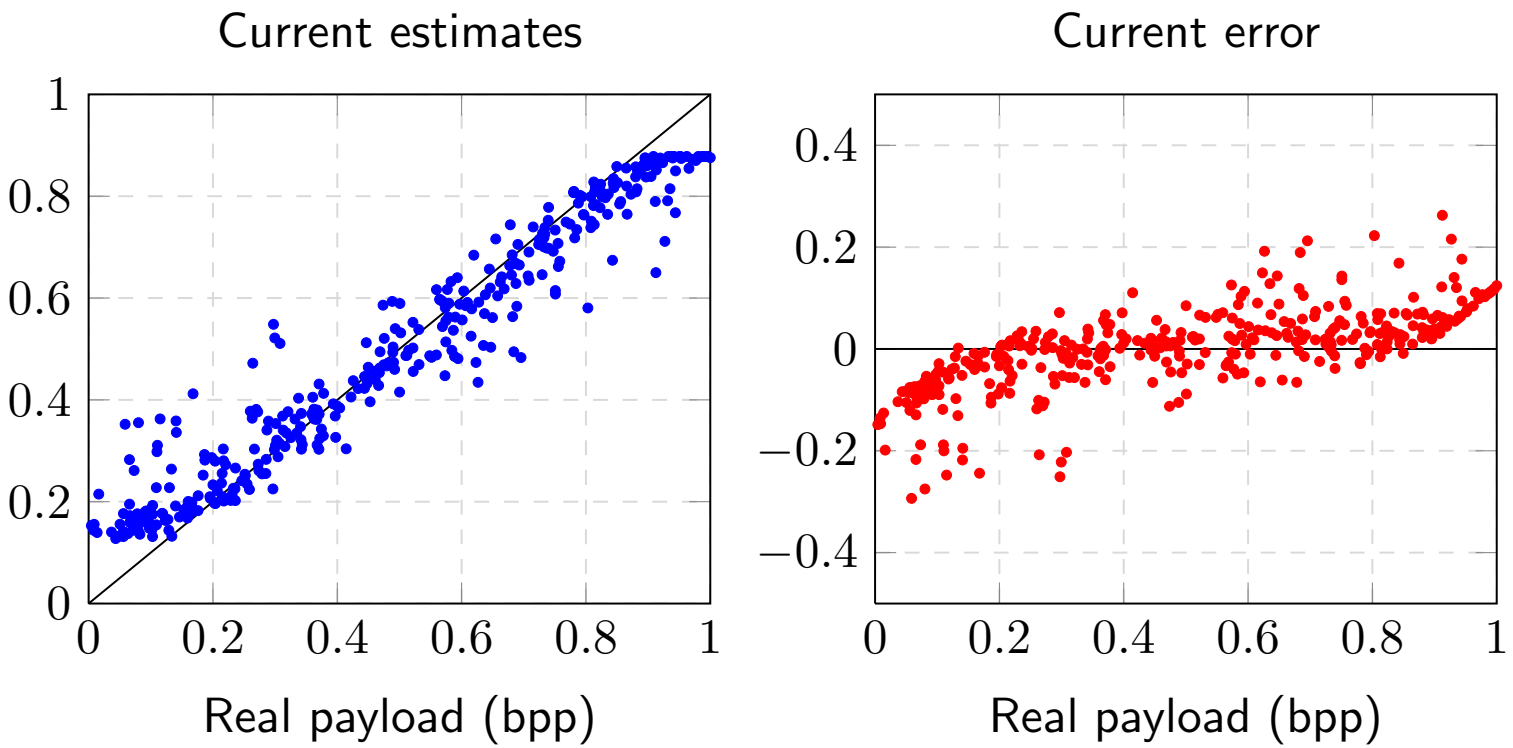
stage 30



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

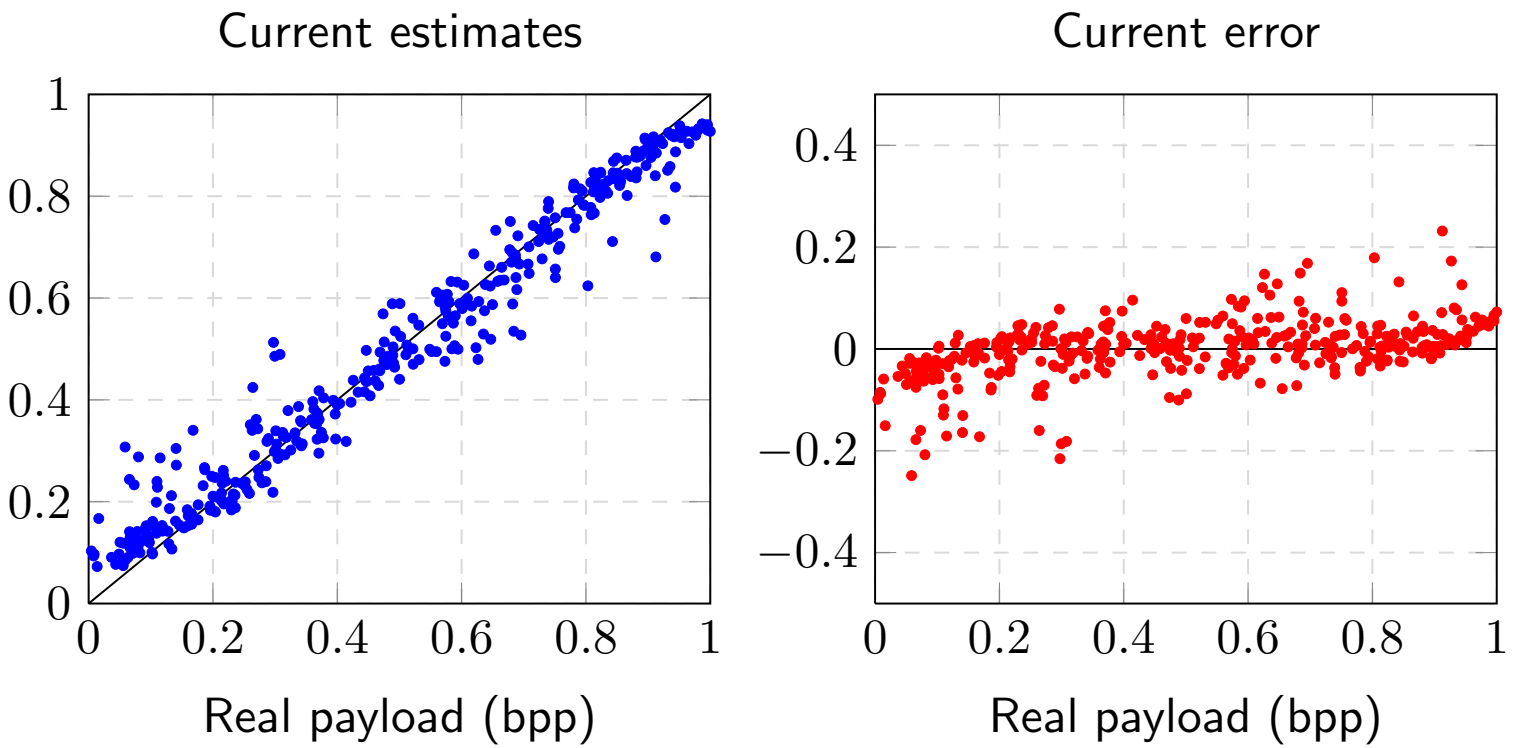
stage 50



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

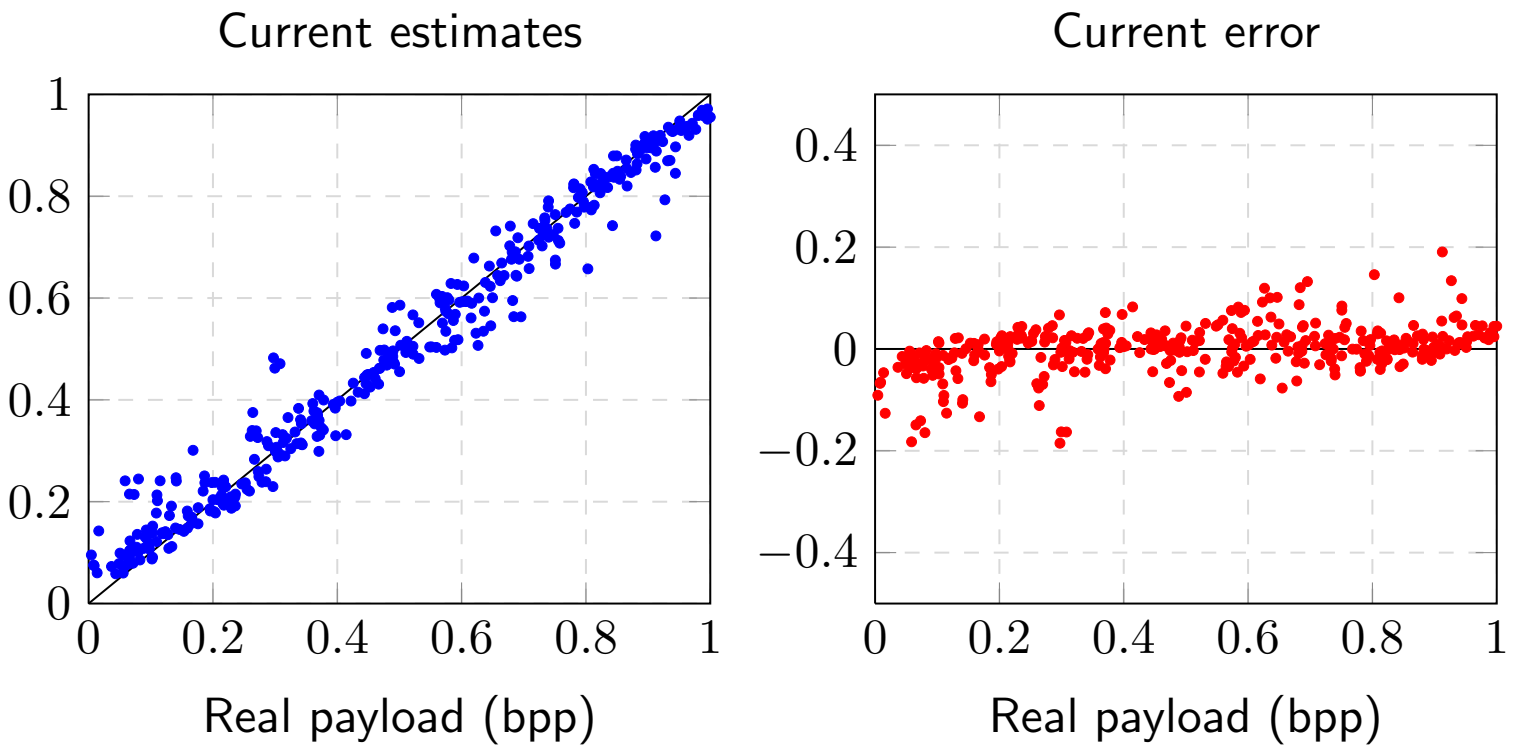
stage 80



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Short simulation

stage 107



- Steganalysis of HUGO, uniformly distributed payloads
- Fixed parameters $d_{\text{sub}} = 500$, $\eta = 0.1$, $l = 2$

Comparison to prior art

Spatial domain steganography

- LSB replacement, HUGO
- Payloads uniformly distributed on [0,1] bpp

JPEG domain steganography

- BCHopt [Sachnev, 2009], nsF5, MME3 [Kim, 2006]
- Payloads uniformly distributed on [0,0.3] bpac

Steganalysis

- Proposed framework with $d_{\text{sub}} = 500$, $\eta = 0.02$, $l = 2$
- Rich models: CCJRM, SRMQ1, parity-aware SRMQ1 [2011, 2012]

Benchmarks

- Kernelized SVR in low-dimensional feature spaces (SPAM,CCPEV)
- Optimized constant benchmark (0.5 bpp, 0.15 bpac)
- LSBR: WS steganalysis [Ker, 2008]

Comparison to prior art

	Const.	kSVR	WS	Proposed	Improvement
HUGO	$8.33 \cdot 10^{-2}$	$4.84 \cdot 10^{-2}$		$1.49 \cdot 10^{-2}$	69.2%
LSBR	$8.33 \cdot 10^{-2}$	$1.48 \cdot 10^{-2}$	$1.15 \cdot 10^{-3}$	$1.01 \cdot 10^{-3}$	12.2%
BCHopt	$7.50 \cdot 10^{-3}$	$6.46 \cdot 10^{-3}$		$5.45 \cdot 10^{-3}$	15.6%
MME3	$7.50 \cdot 10^{-3}$	$2.35 \cdot 10^{-3}$		$1.31 \cdot 10^{-3}$	44.3%
nsF5	$7.50 \cdot 10^{-3}$	$3.10 \cdot 10^{-3}$		$1.91 \cdot 10^{-3}$	38.4%

(reporting MSE calculated on the testing set)

- Improvement across all schemes, in both domains
- Most significant improvement observed for HUGO
  - consistent with binary steganalysis

Discussion

Payload distribution

- Knowledge of the payload distribution matters
- Uniform distribution $\Rightarrow$ training samples generated accordingly
- A priori: probability of encountering cover image is zero!

Quantitative steganalysis for practical applications

- Should consider probability of seeing cover/stego image as well
- Compound binary and quantitative system?

Questions

- What payload distribution should be used for classification?
- Should we minimize total error or fix the FA rate?
- Is squared loss the most appropriate measure?

Conclusion

Summary

- Natural adaptation of rich models in the quantitative setting
- Based on gradient boosting
- Customized variant of regression trees
- Improvement roughly mimics improvement in binary classification

Resources

- Implementation of the proposed system, rich model extractors:
  `http://dde.binghamton.edu/download`