Locating Steganographic Payload via WS Residuals

Andrew Ker

adk@comlab.ox.ac.uk

Royal Society University Research Fellow
Oxford University Computing Laboratory

ACM Multimedia & Security Workshop
Oxford, 22 September 2008

Locating Steganographic Payload via WS Residuals

Outline

- *The WS method*
- *Per-pixel residuals; experimental results*
- *Improved WS residuals; experimental results*
- *Conclusions*

WS steganalysis

Suppose a cover of n samples has payload embedded, by LSB replacement, giving a stego object $\boldsymbol{s} = (s_1, s_2, \ldots, s_n)$.

1. Estimate cover from stego object by filtering:

$$\hat{\boldsymbol{c}} = \boldsymbol{s} \ * \ \begin{pmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{pmatrix} \qquad \text{(2-D convolution)}$$

2. Estimate of number of flipped pixels by:

$$\sum_{i=1}^{n} (s_i - \hat{c}_i)\mathbf{par}(s_i) \qquad \text{where } \mathbf{par}(x) = \begin{cases} -1, & x \text{ even} \\ +1, & x \text{ odd} \end{cases}$$

J. Fridrich & M. Goljan. *On Estimation of Secret Message Length in LSB Steganography in Spatial Domain.* In Proc. Electronic Imaging 2004, SPIE.

WS residuals

Suppose a cover of n samples has payload embedded, by LSB replacement, giving a stego object $\boldsymbol{s} = (s_1, s_2, \ldots, s_n)$.

1. Estimate cover from stego object by filtering:

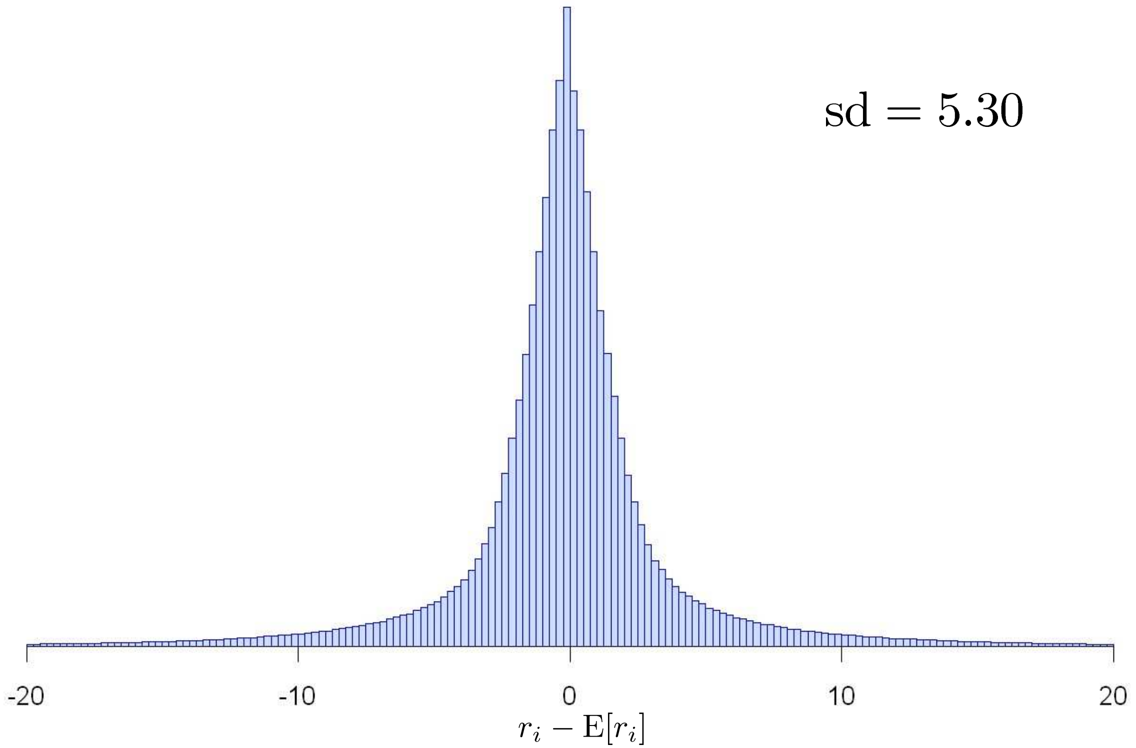
$$\hat{\boldsymbol{c}} = \boldsymbol{s} \ * \begin{pmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{pmatrix} \qquad \text{(2-D convolution)}$$

2. Consider the (appropriately weighted) ***residuals***:

$$r_i = (s_i - \hat{c}_i)\mathbf{par}(s_i) \qquad \text{where } \mathbf{par}(x) = \begin{cases} -1, & x \text{ even} \\ +1, & x \text{ odd} \end{cases}$$

When location i contains payload $\mathrm{E}[r_i] = 0.5,$ otherwise $\mathrm{E}[r_i] = 0.$

WS residuals



When location i contains payload $\mathrm{E}[r_i] = 0.5$, otherwise $\mathrm{E}[r_i] = 0$.

Pooled steganalysis

Suppose the steganalyst has access to N stego objects which contain ***different payloads*** placed in the ***same locations*** in ***different covers***. This is plausible if there have been many communications and:

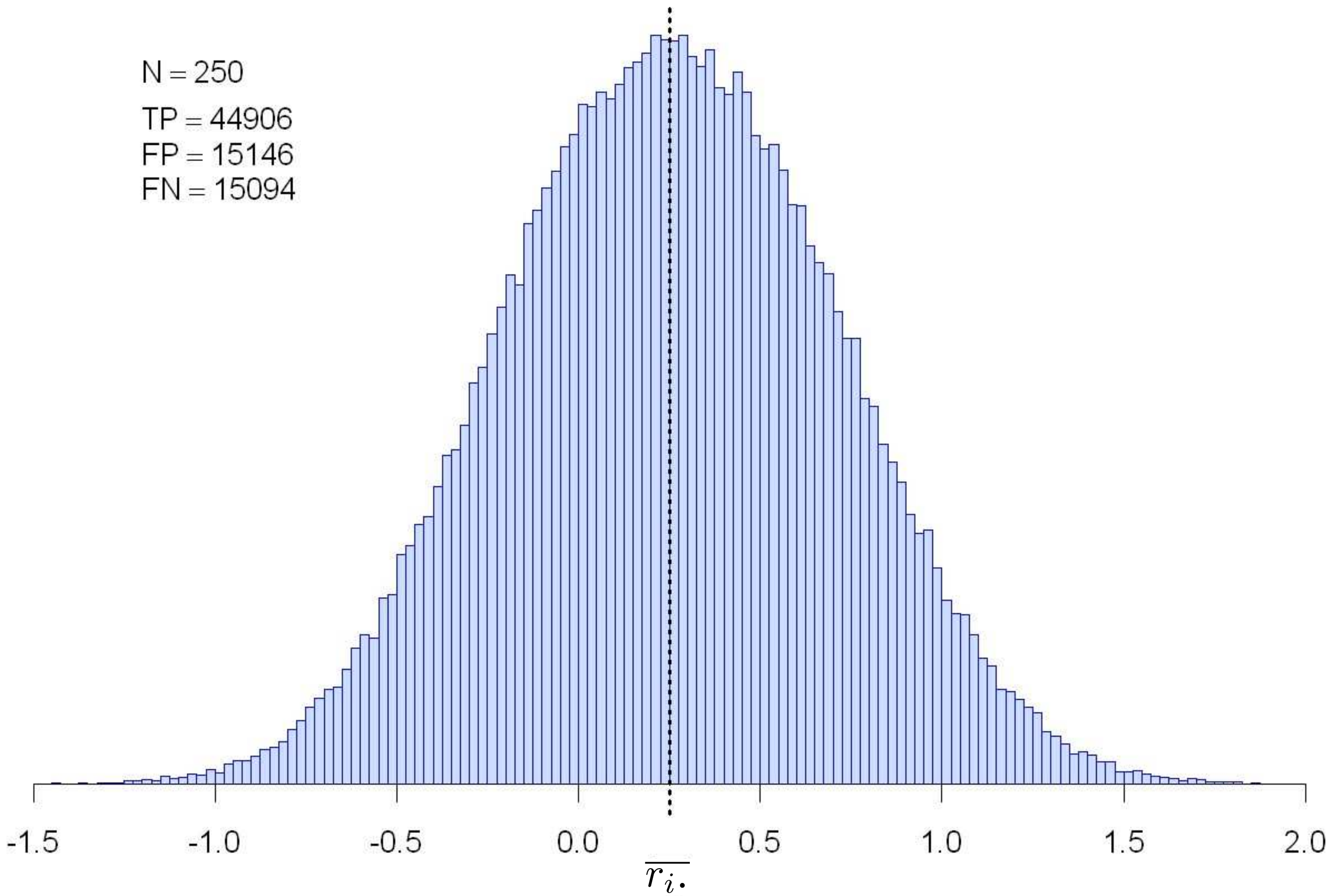
- the embedding method placed payload sequentially**, or**
- the embedding method omitted to randomise the location, **or**
- the embedder re-used the same stego-key for each embedding.

We compute each r_{ij}, the residual at location i in image j, and estimate the proportion of flipped pixels at location i by

$$\overline{r_{i\cdot}} = \frac{1}{N}\sum_{j=1}^{N} r_{ij}$$

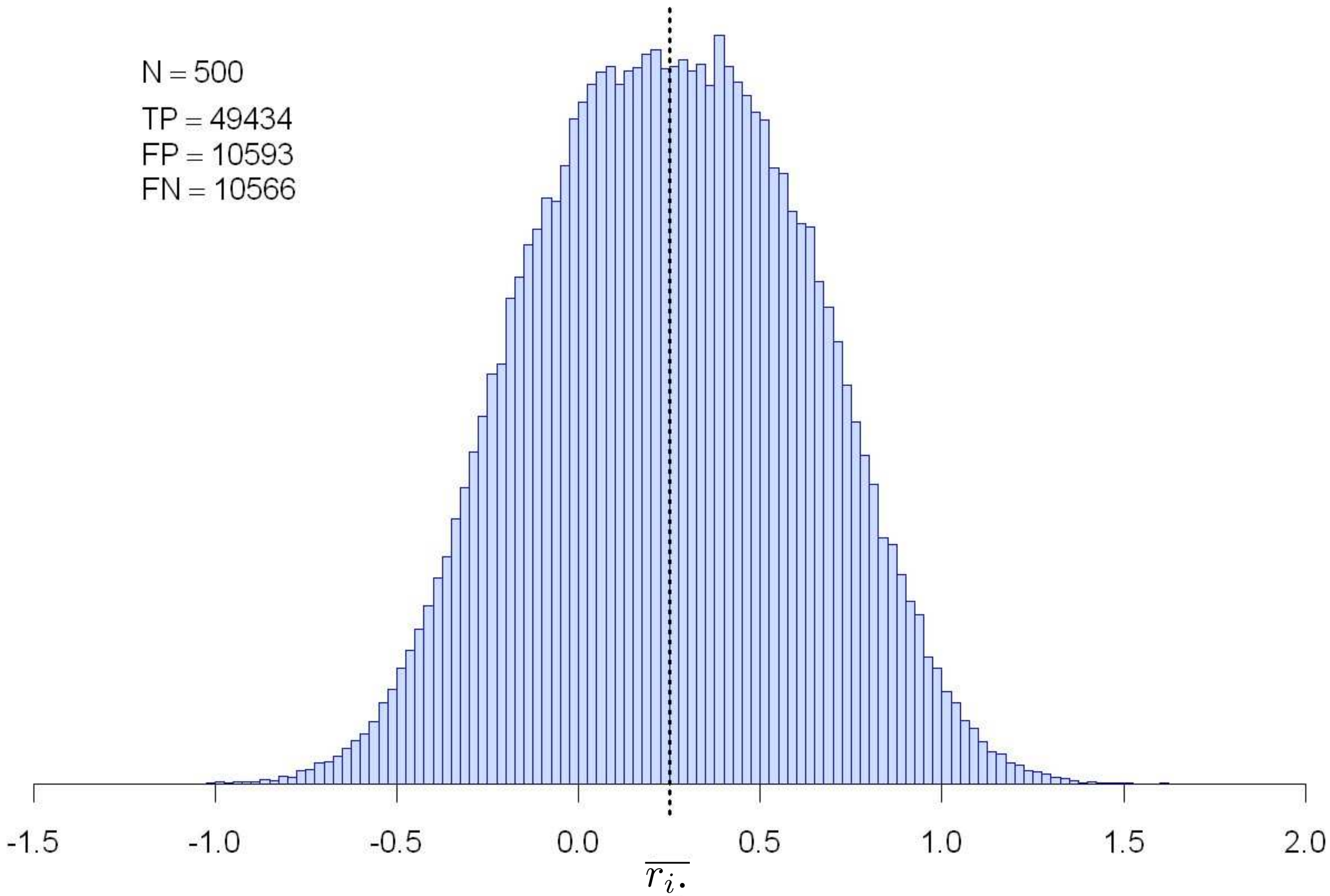
(guessing that this location contains payload if e.g. $\overline{r_{i\cdot}} > 0.25$)

Experimental results



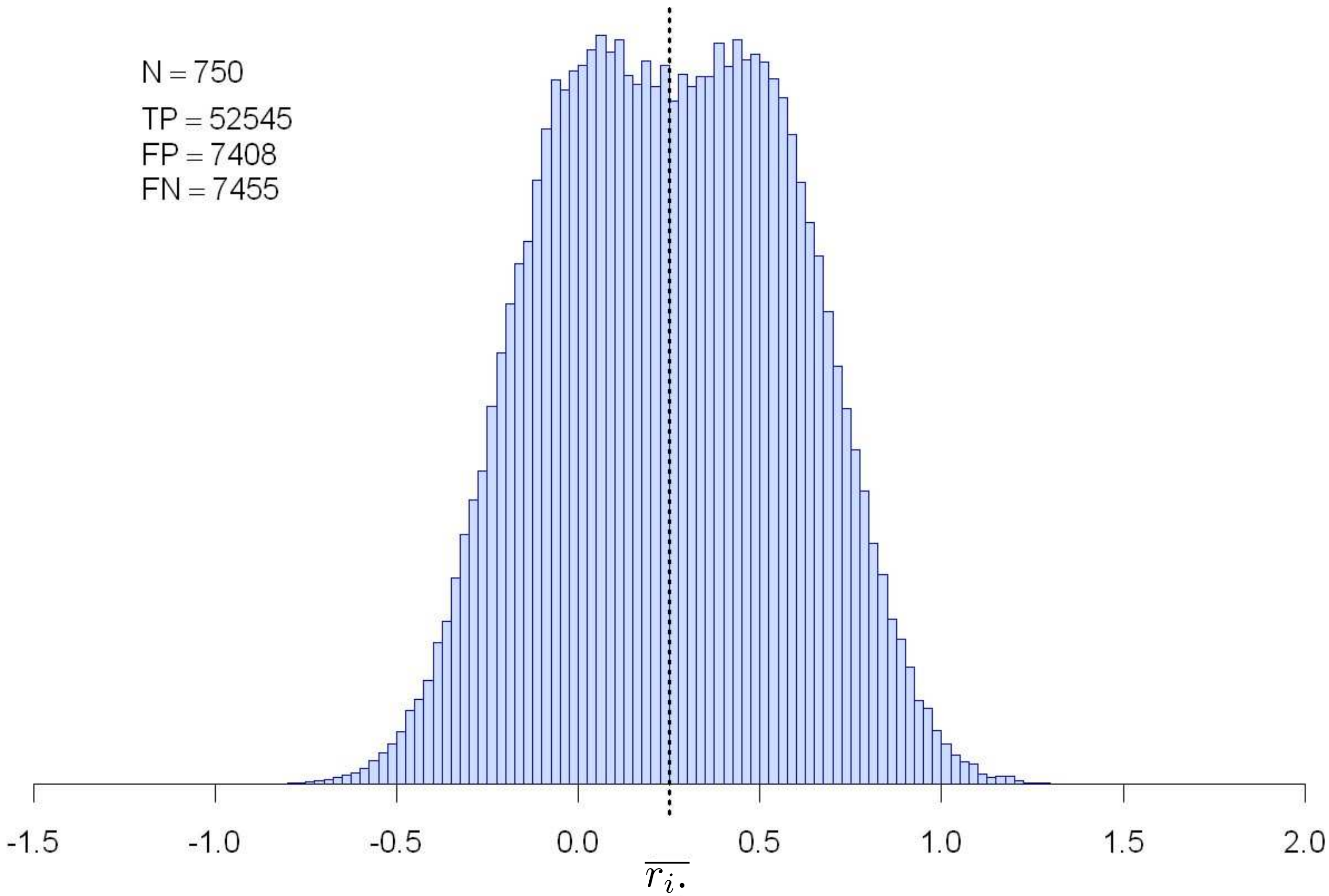
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



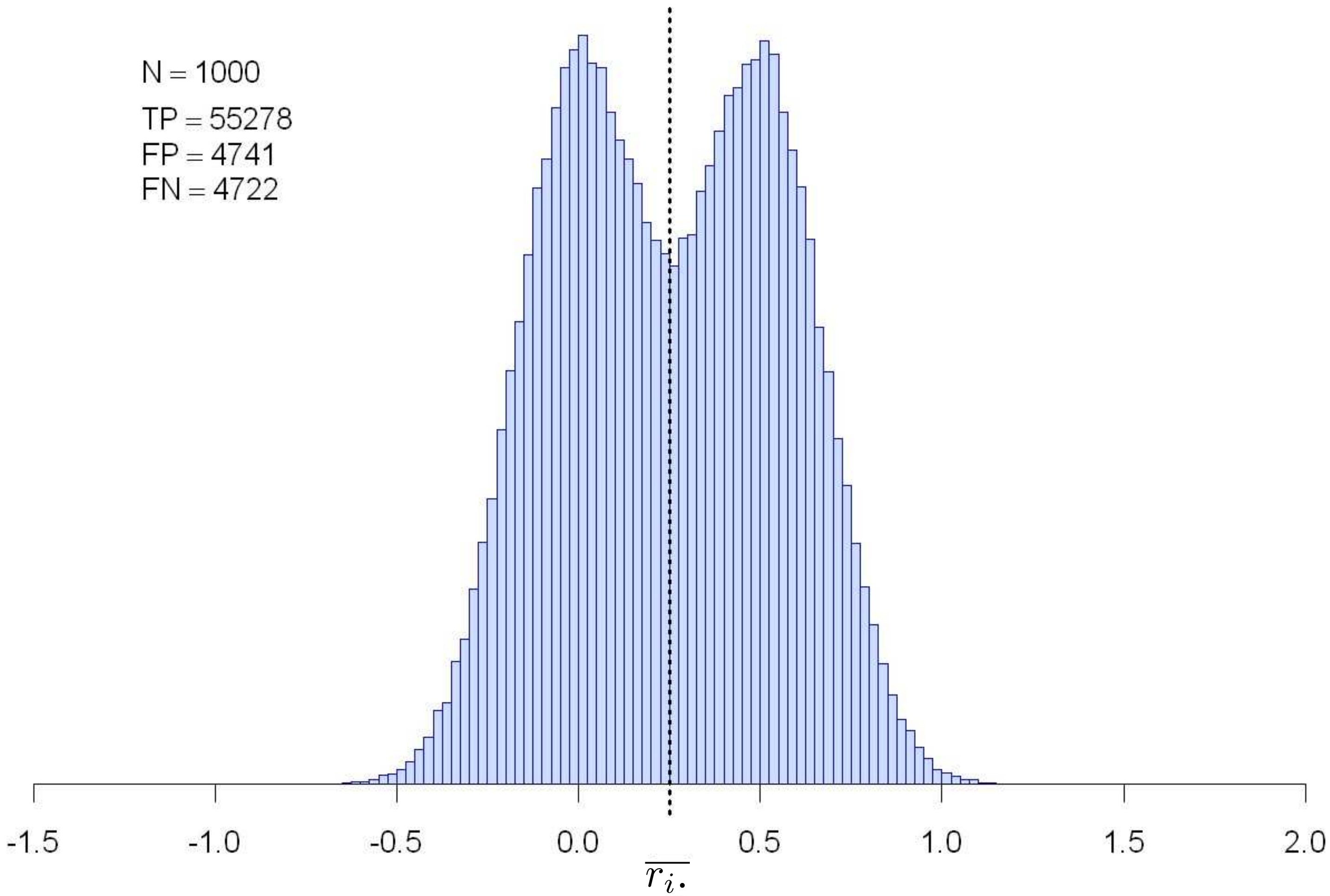
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



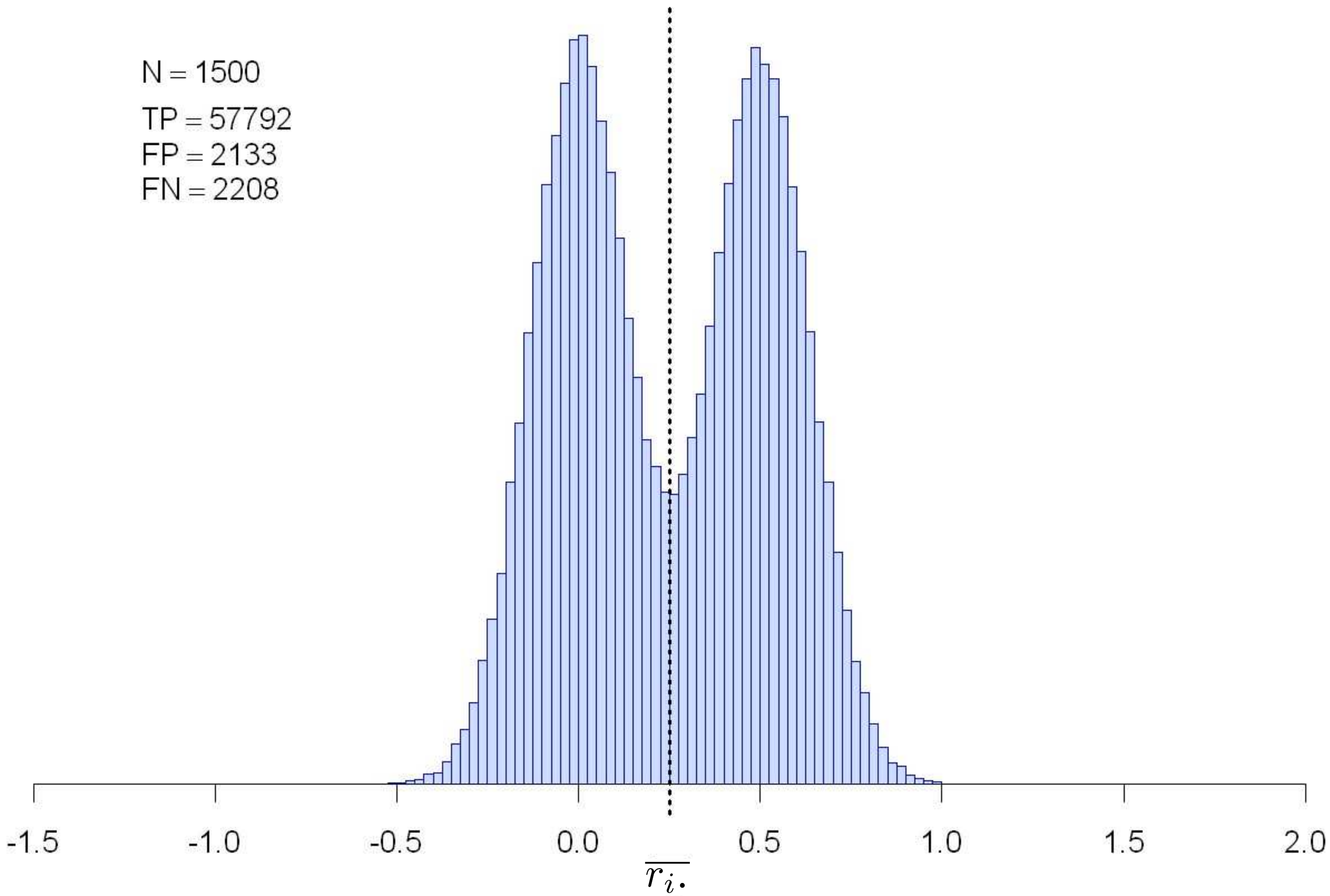
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



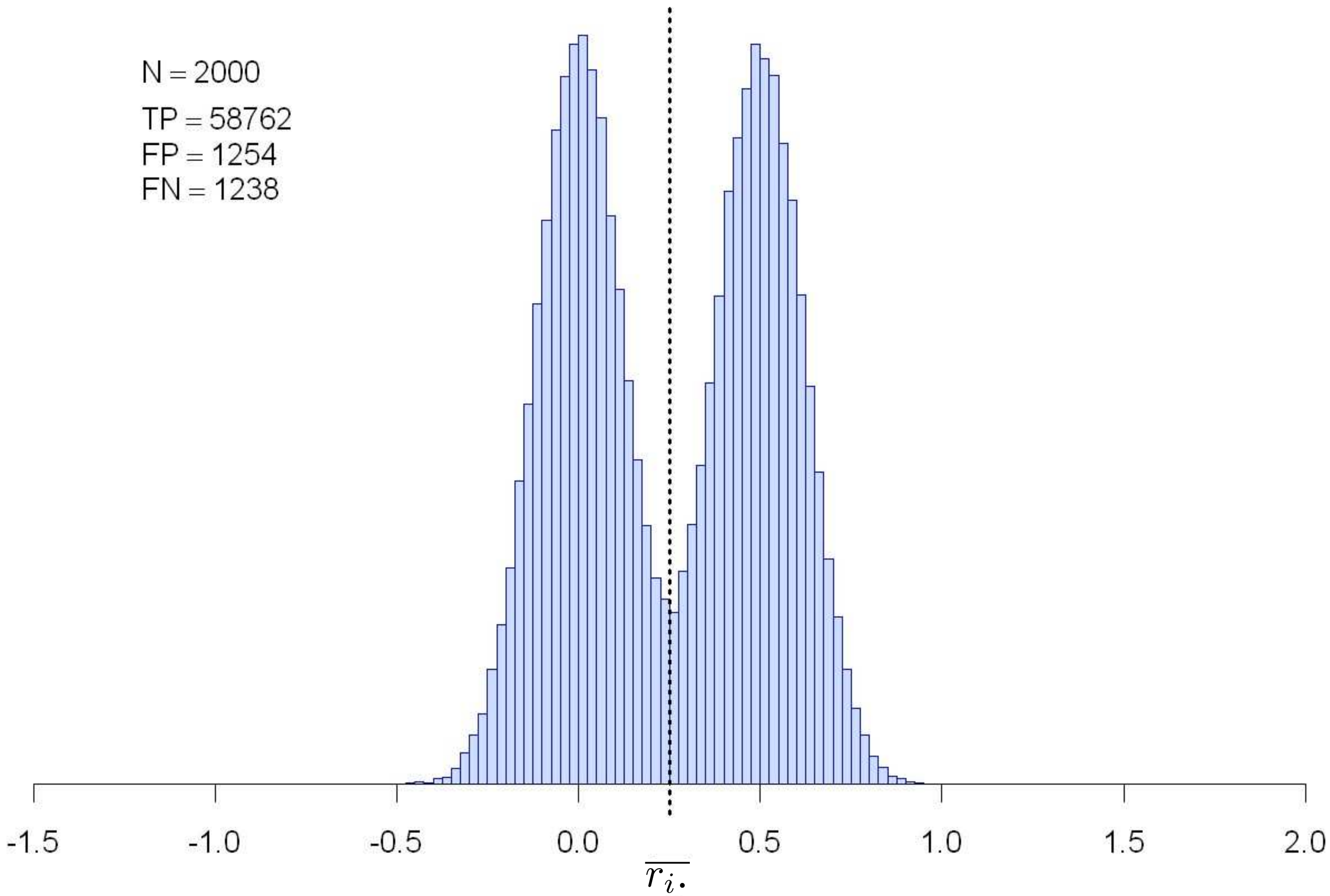
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



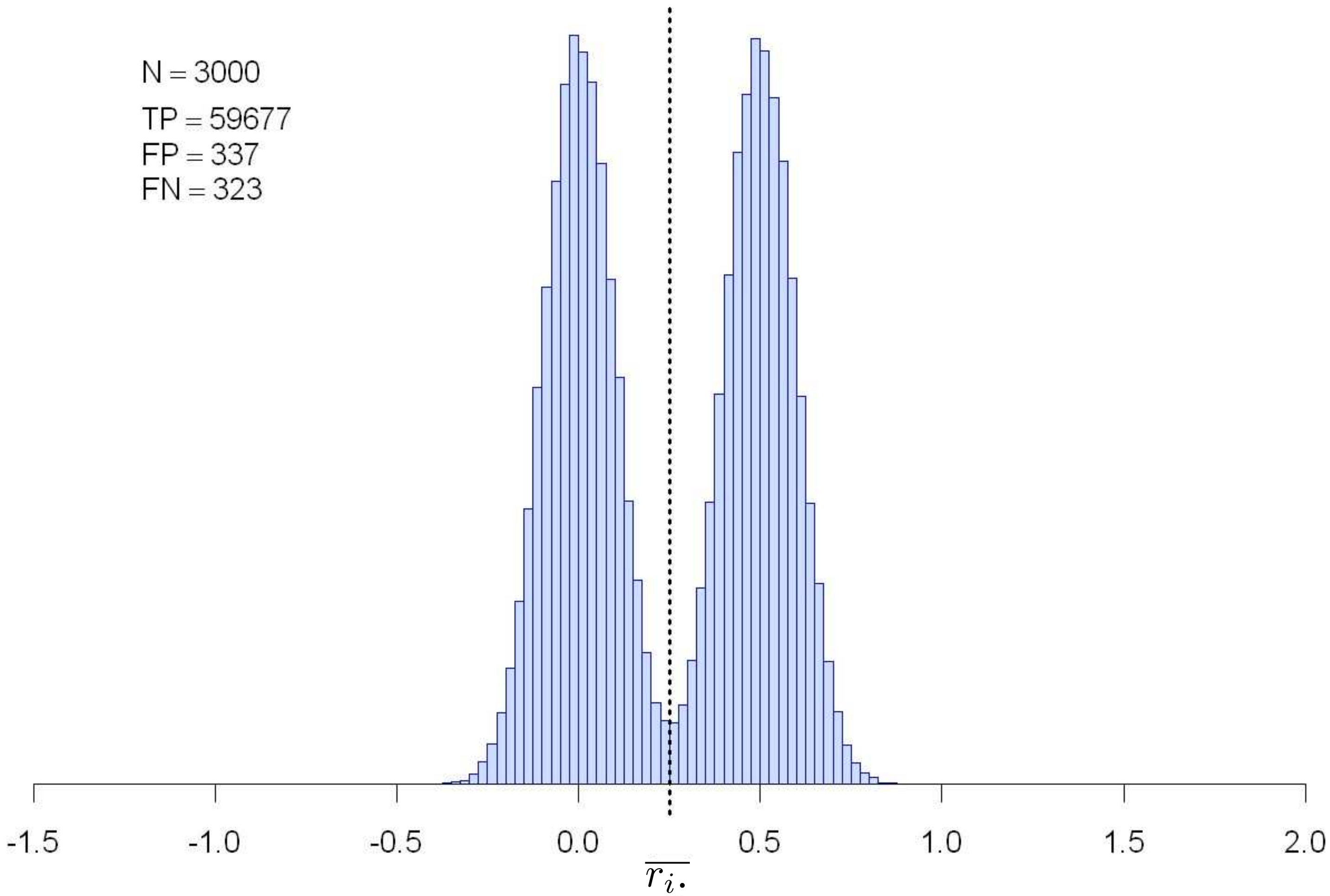
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



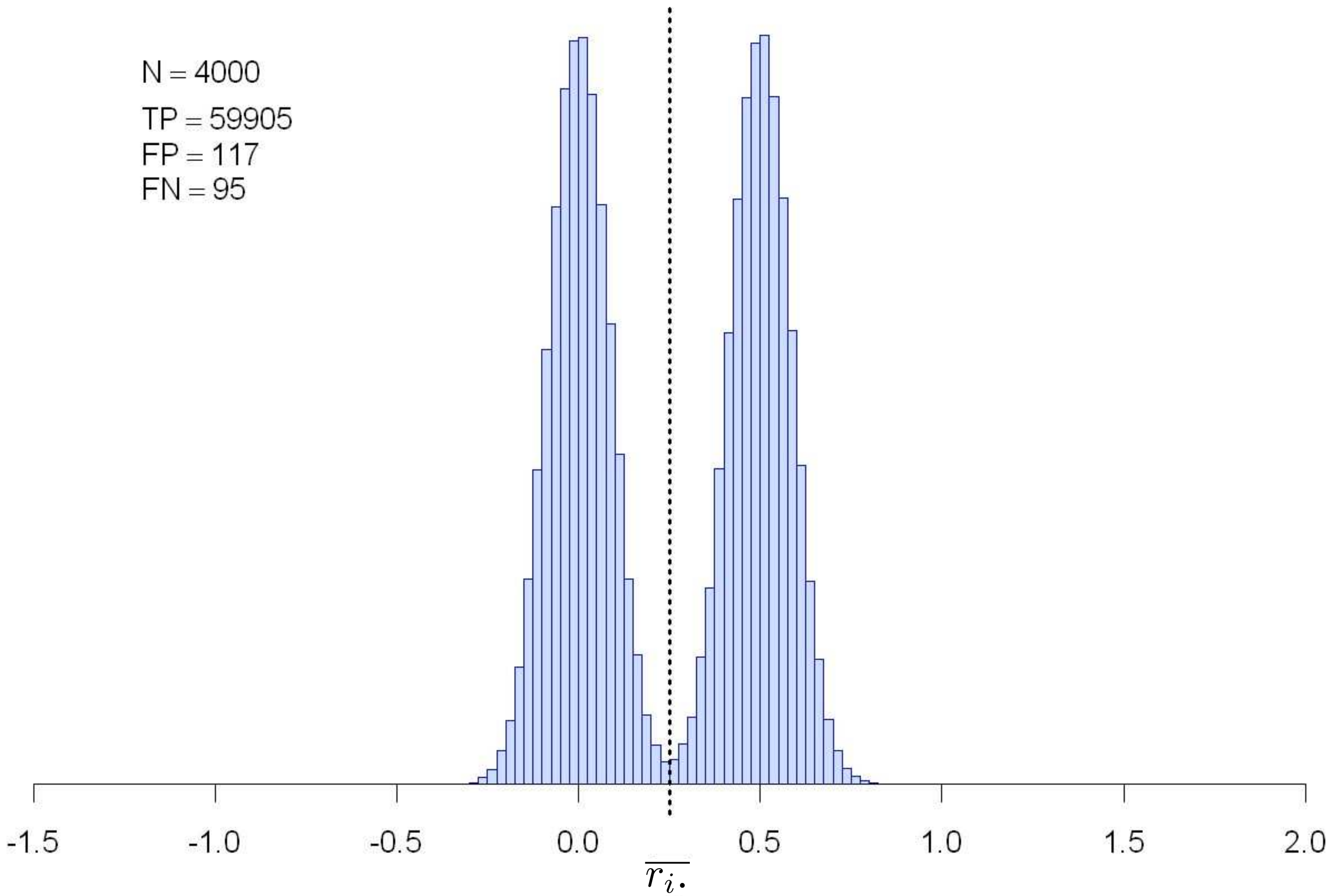
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



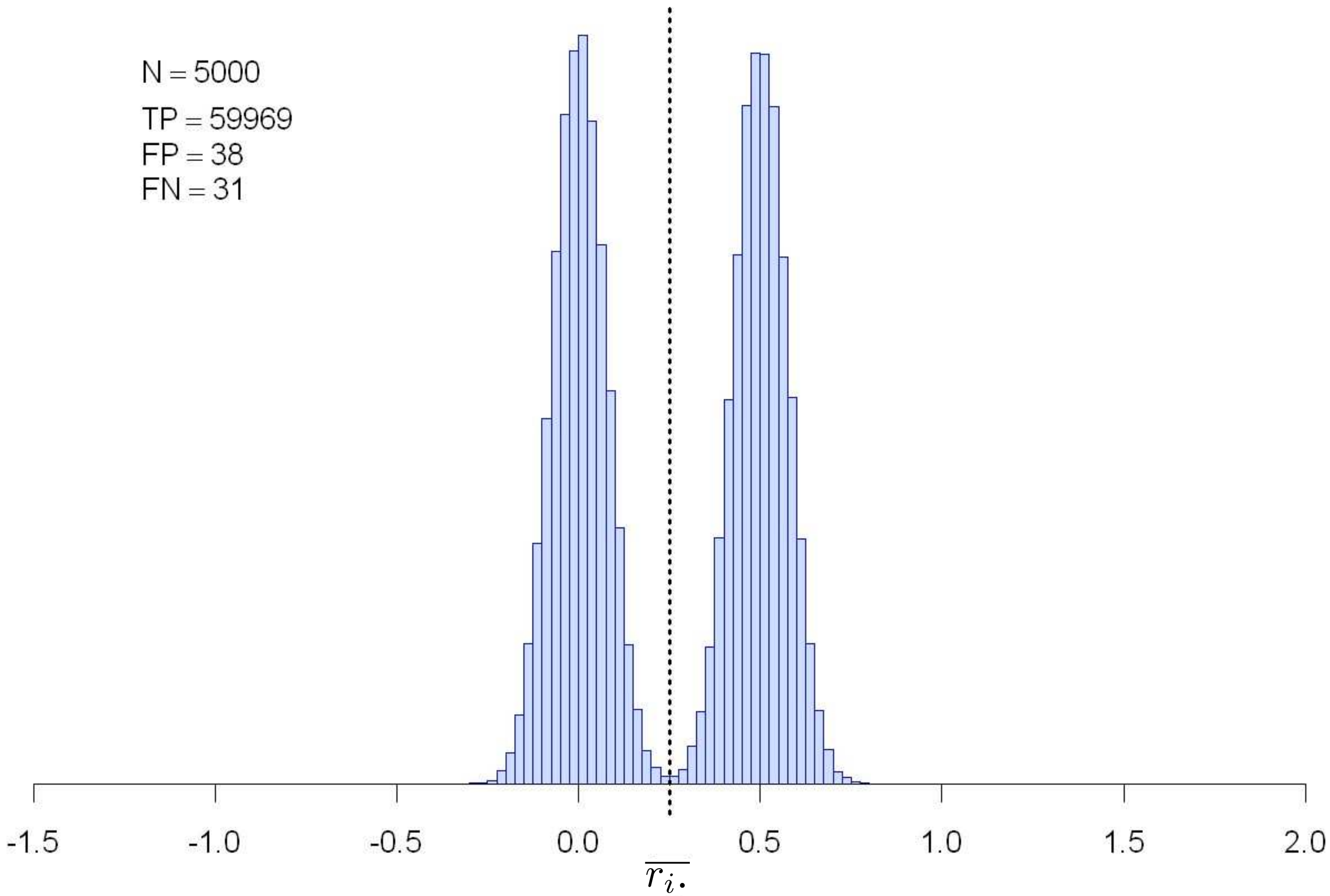
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



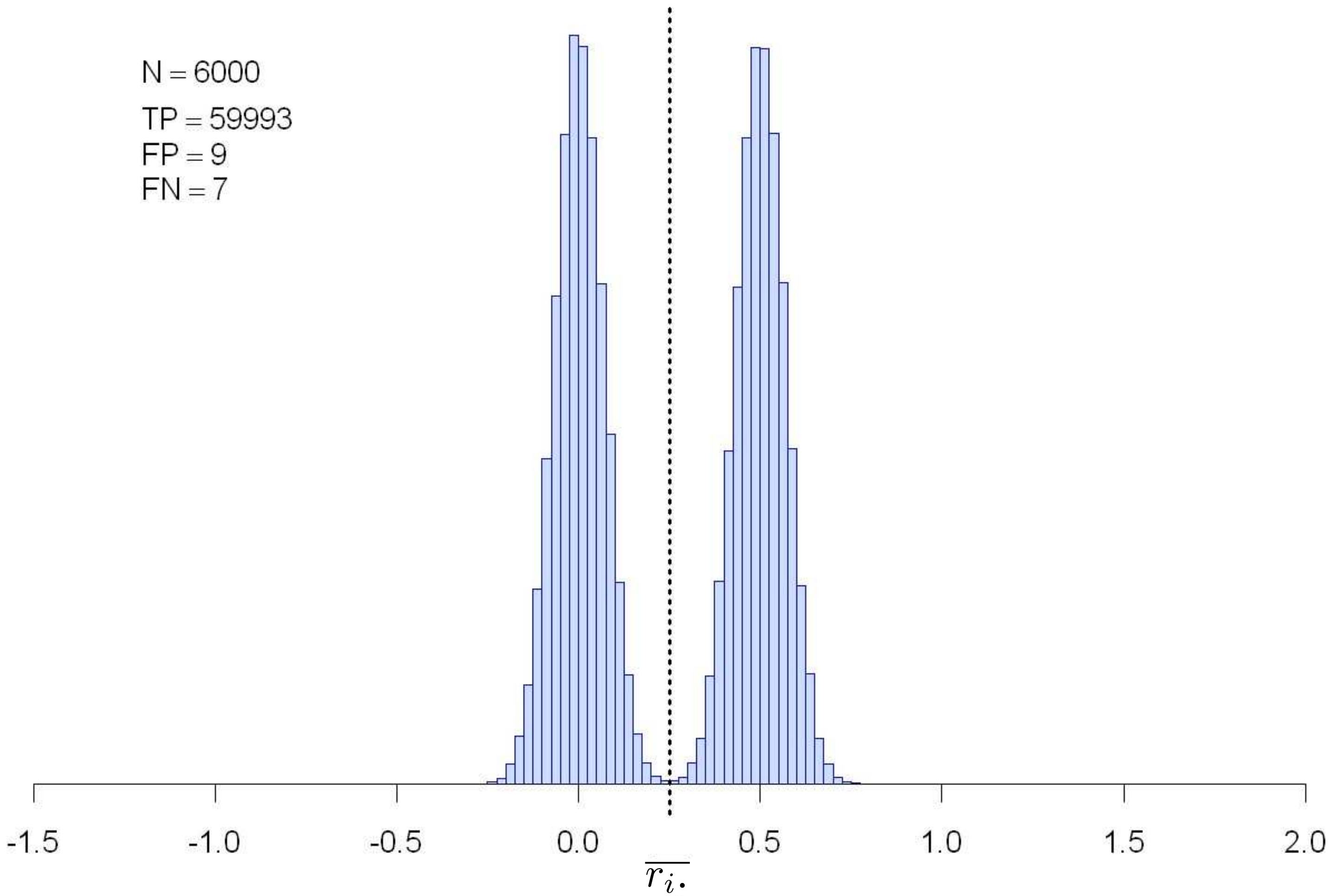
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



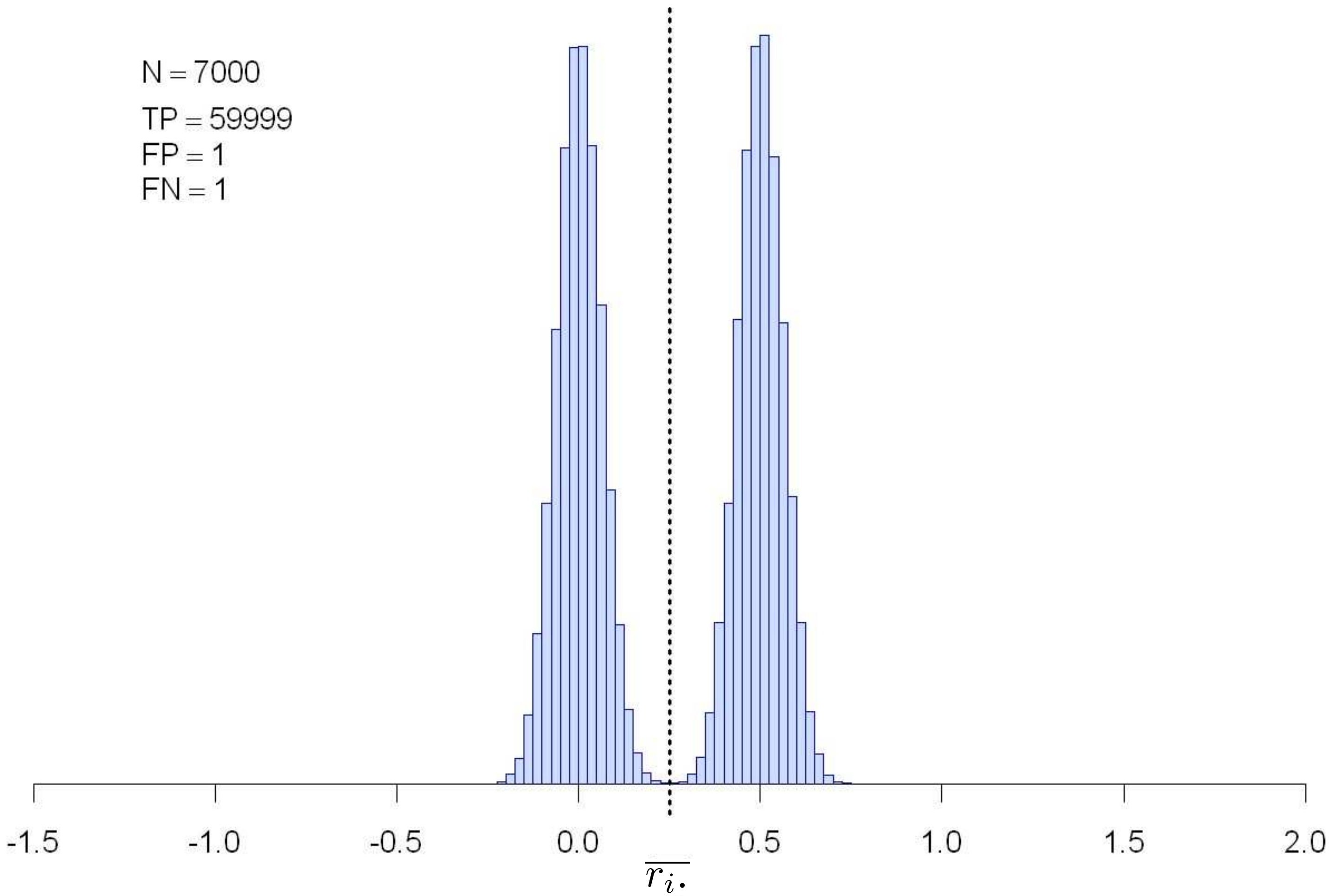
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



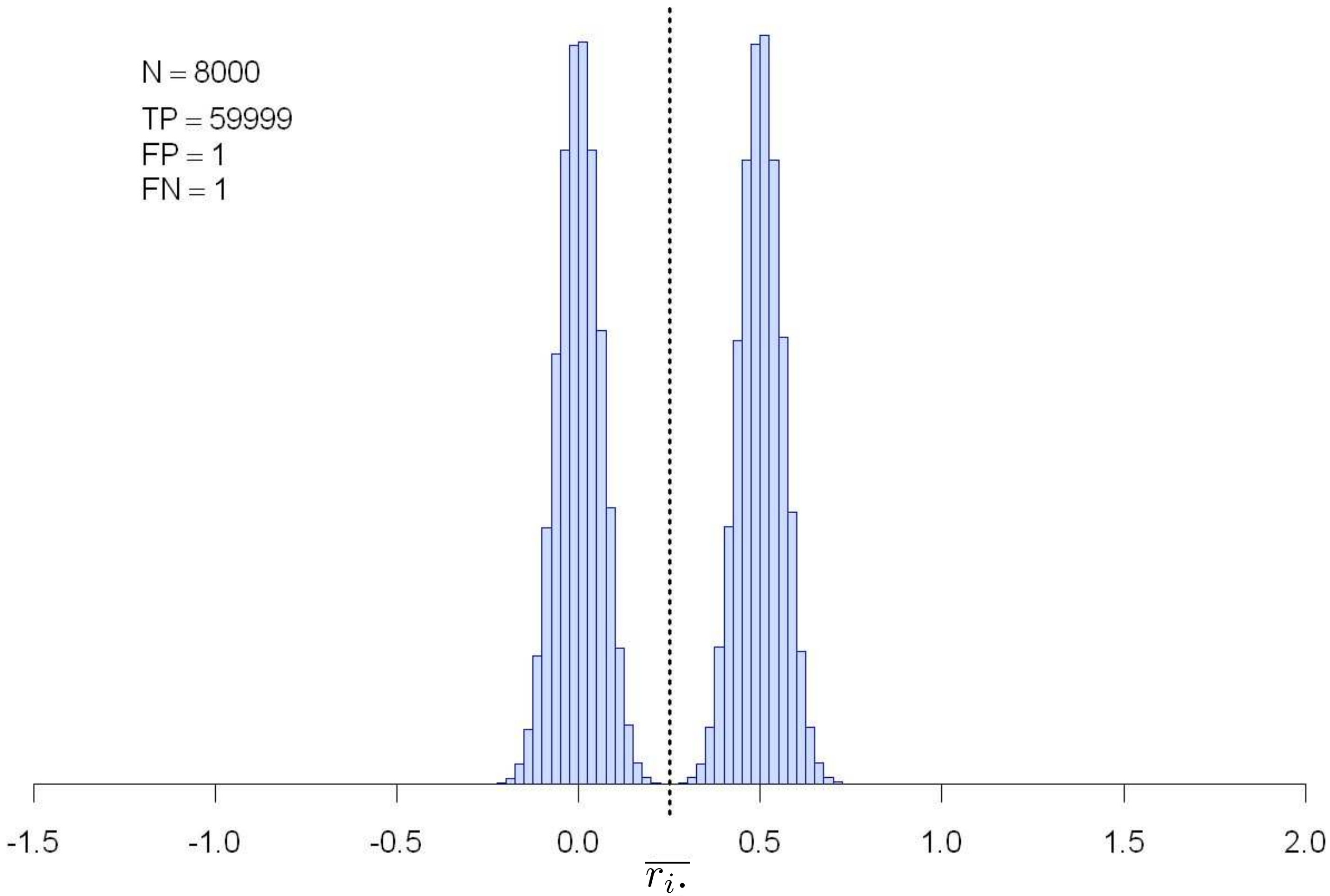
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



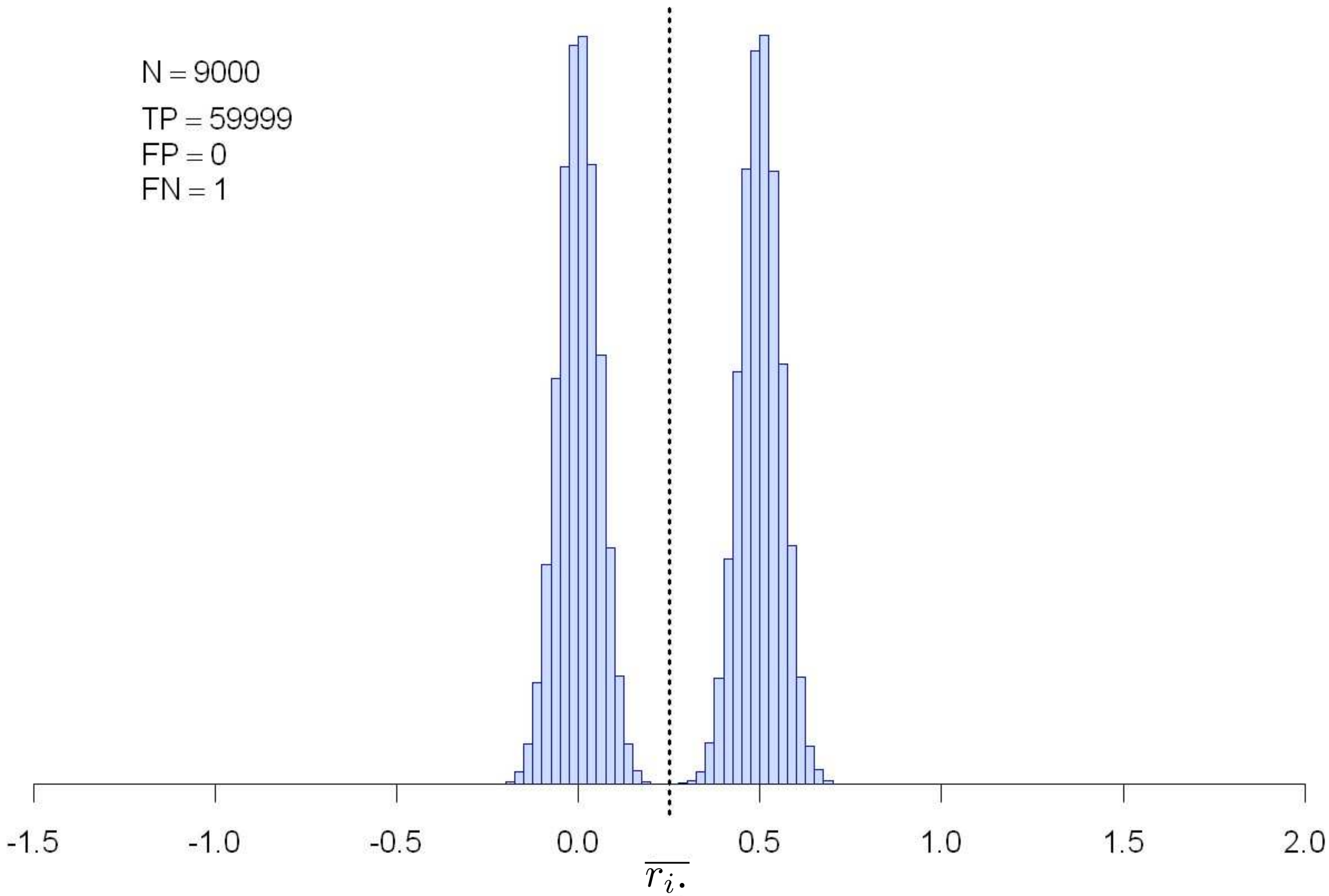
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



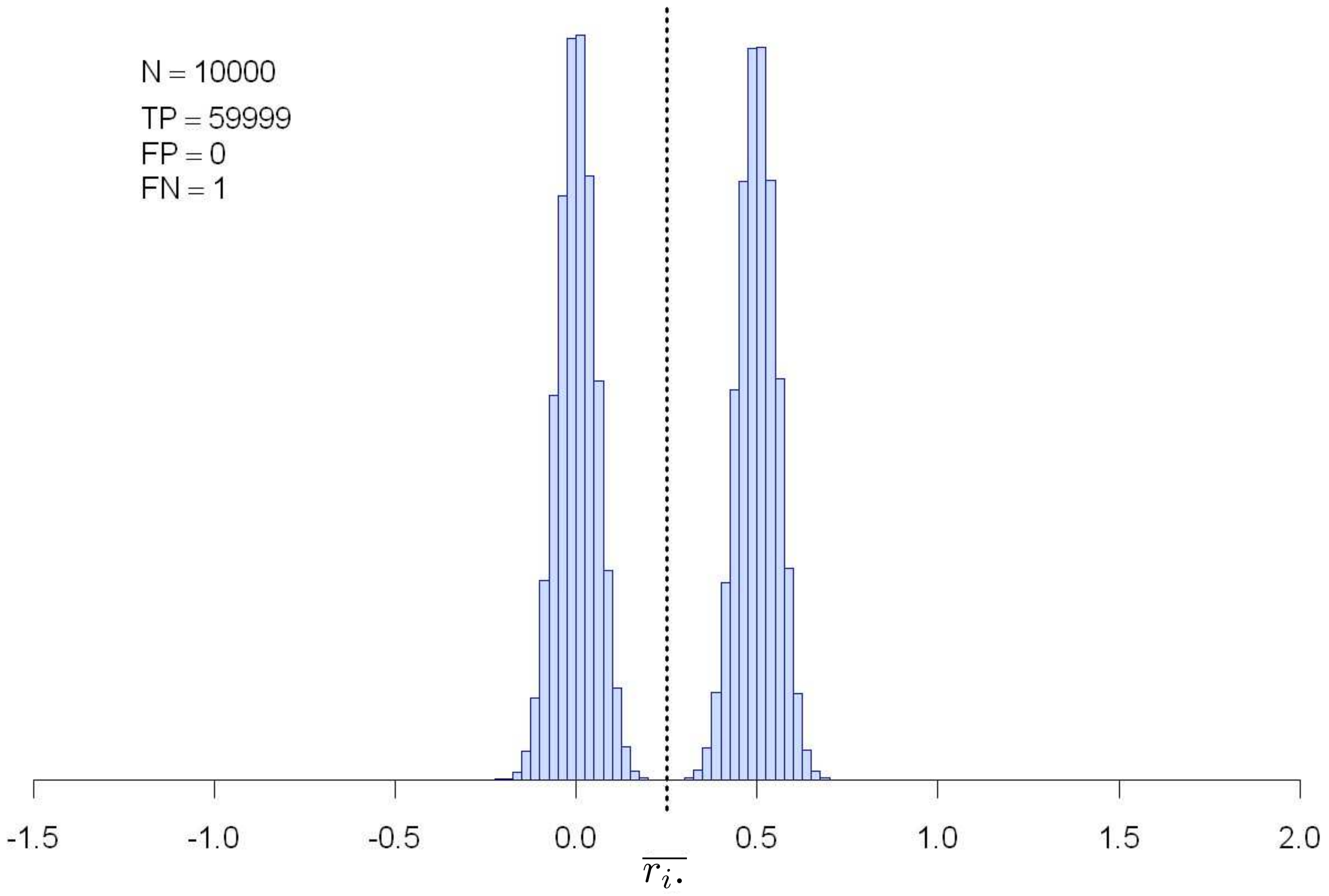
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Improved WS

WS can be re-engineered to improve its accuracy as a payload-size estimator.

1. Estimate cover from stego object using a ***trained*** filter:

$$\hat{\boldsymbol{c}} = \boldsymbol{s} * \mathcal{F} \quad \text{where } \mathcal{F} \text{ minimizes } \|\boldsymbol{s} - \boldsymbol{s} * \mathcal{F}\|.$$

2. ***Weight*** the estimate for number of flipped pixels:

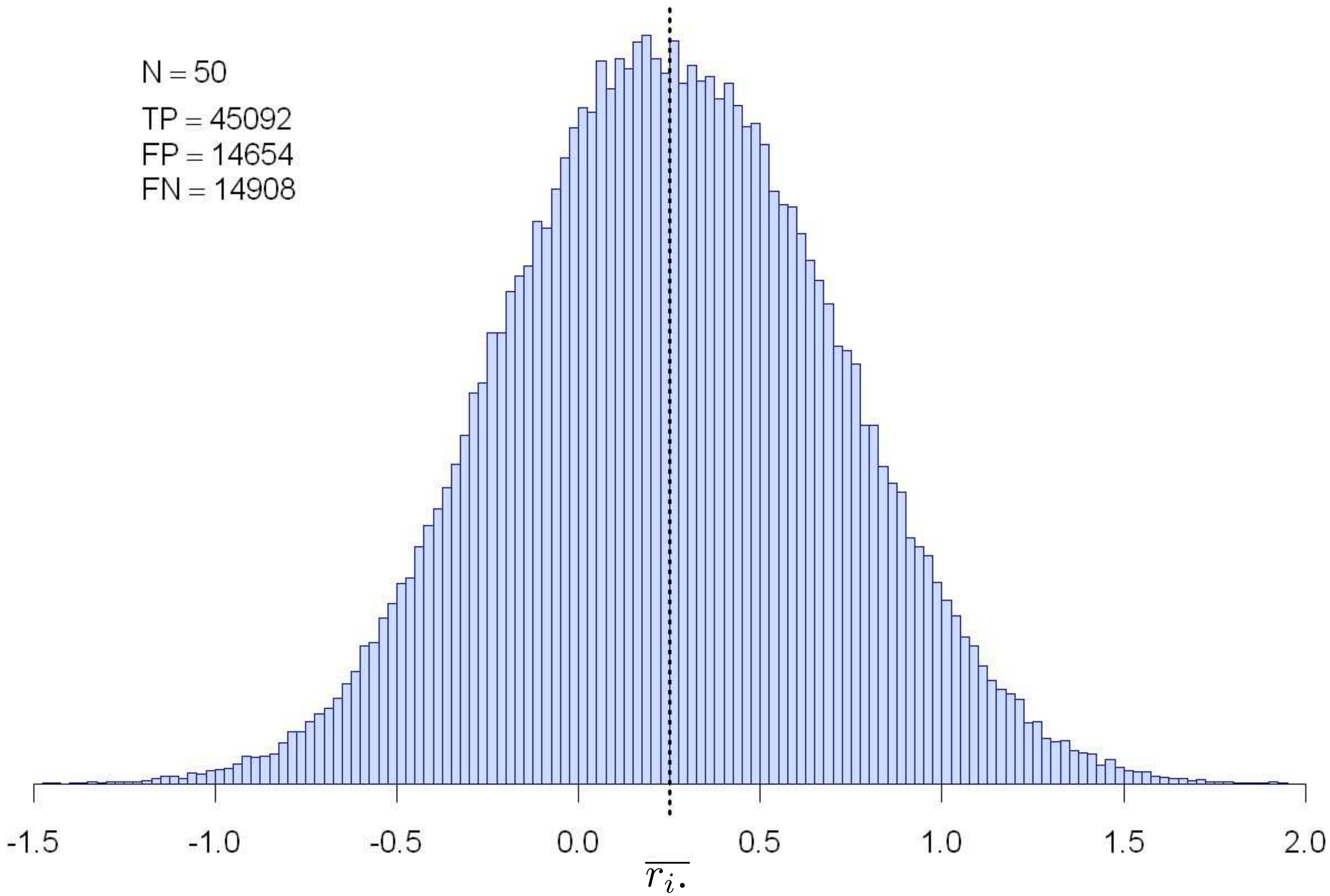
$$\sum_{i=1}^{n} w_i(s_i - \hat{c}_i)\mathbf{par}(s_i) \qquad \text{where } w_i \propto \frac{1}{5+\sigma_i^2}$$

σ_i^2 is a local variance measure for location i.

(other improvements exist but are not needed for our purposes.)

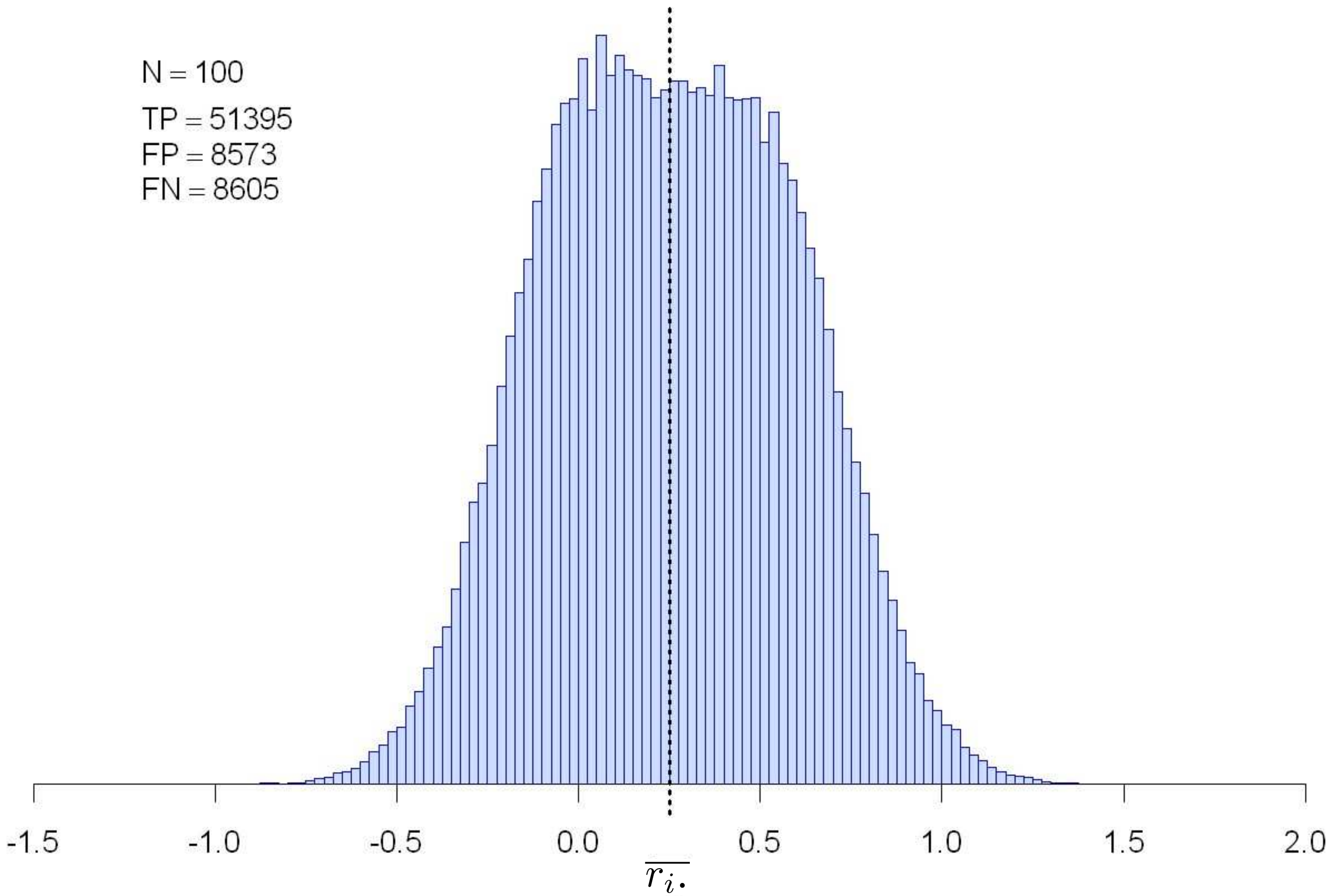
A. Ker & R.Böhme. *Revisiting Weighted Stego Image Steganalysis.* In Proc. Electronic Imaging 2008, SPIE.

Experimental results



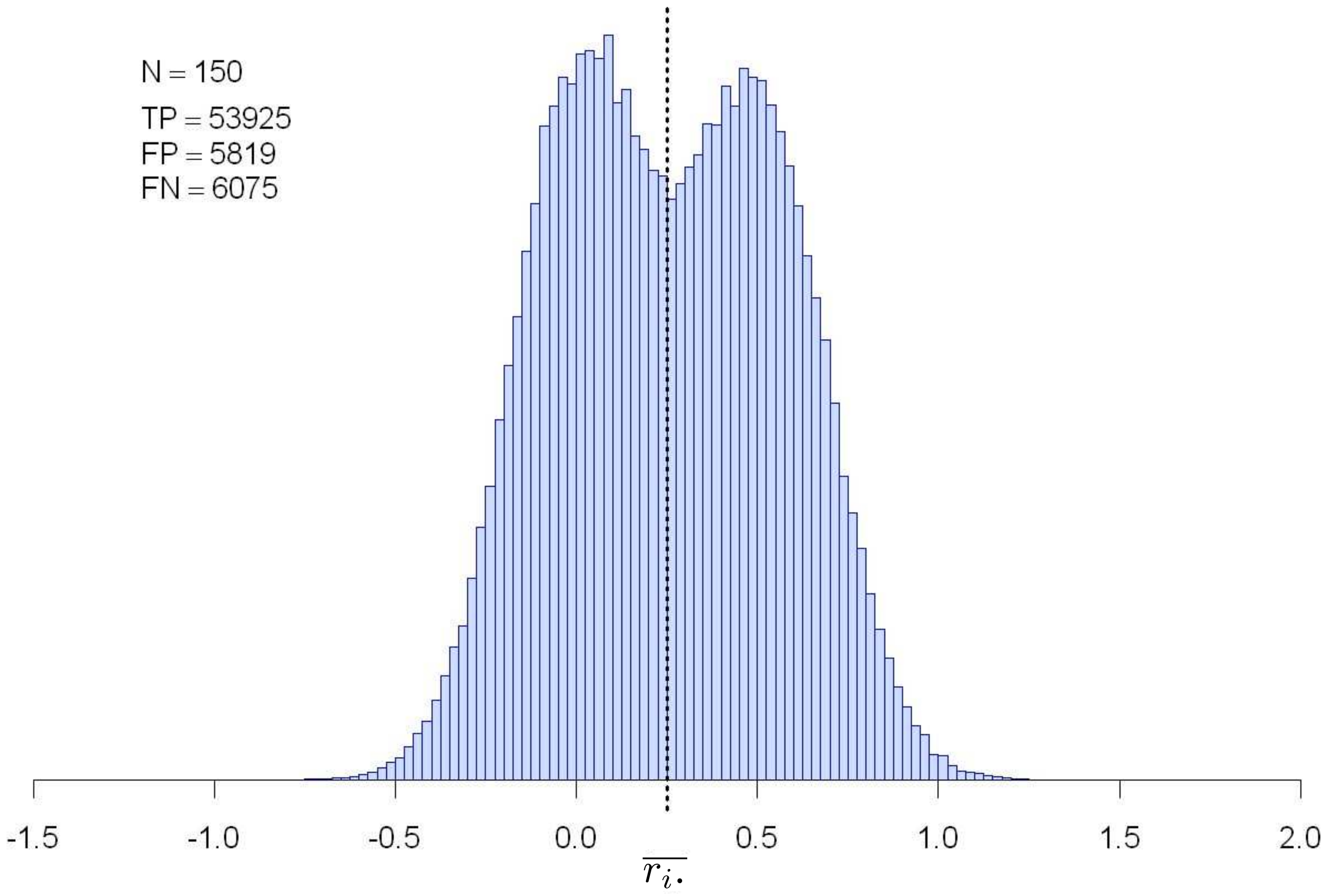
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



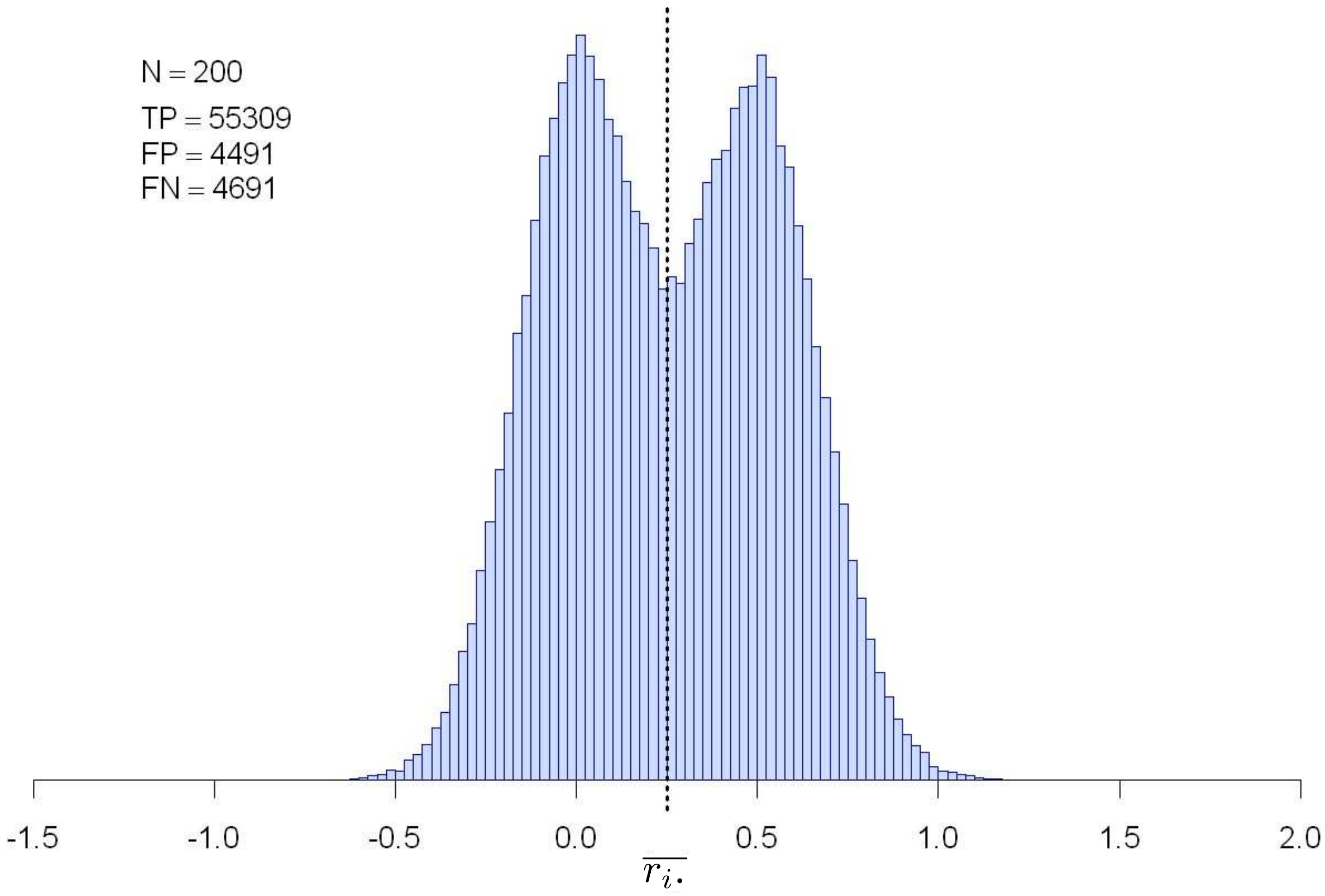
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



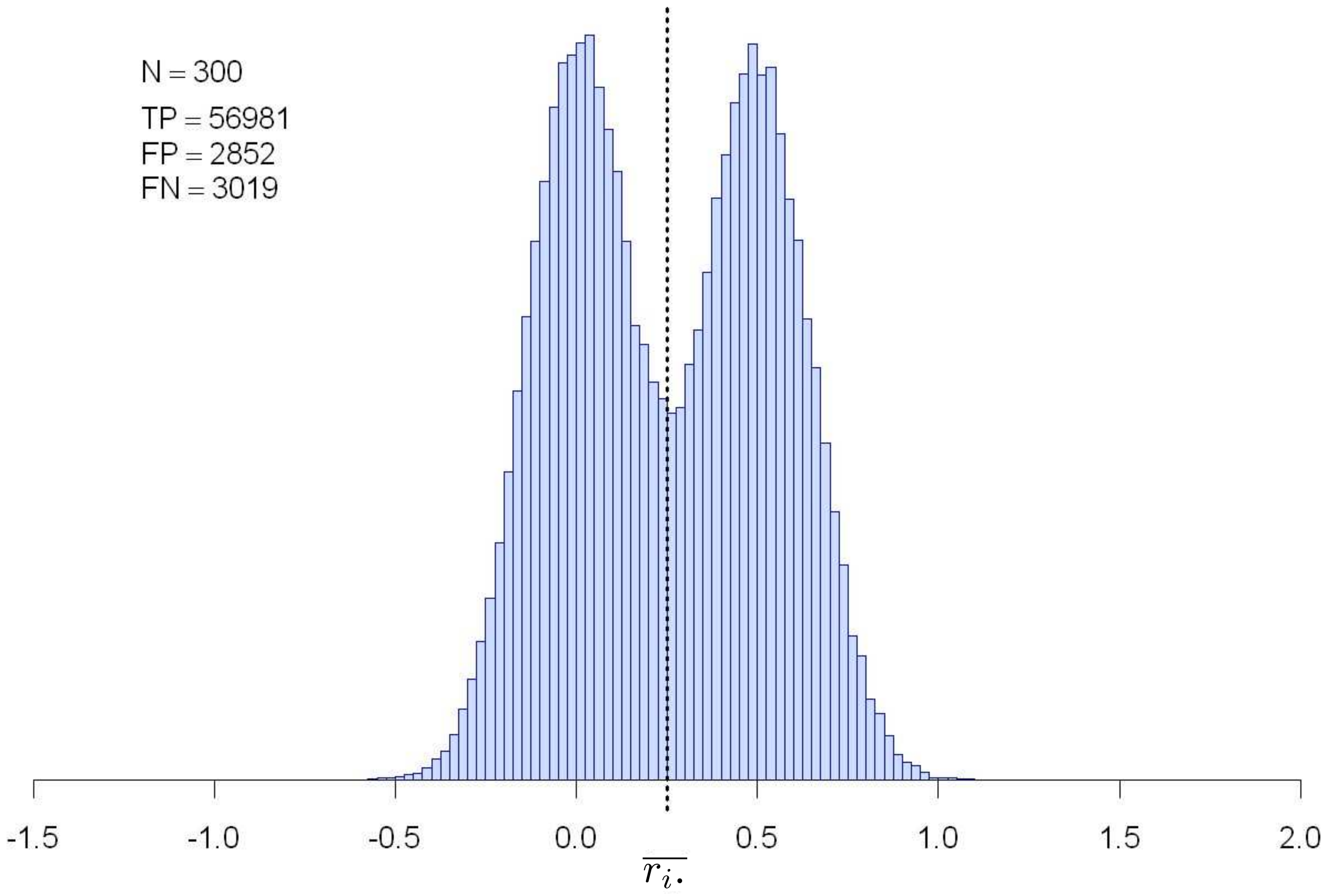
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



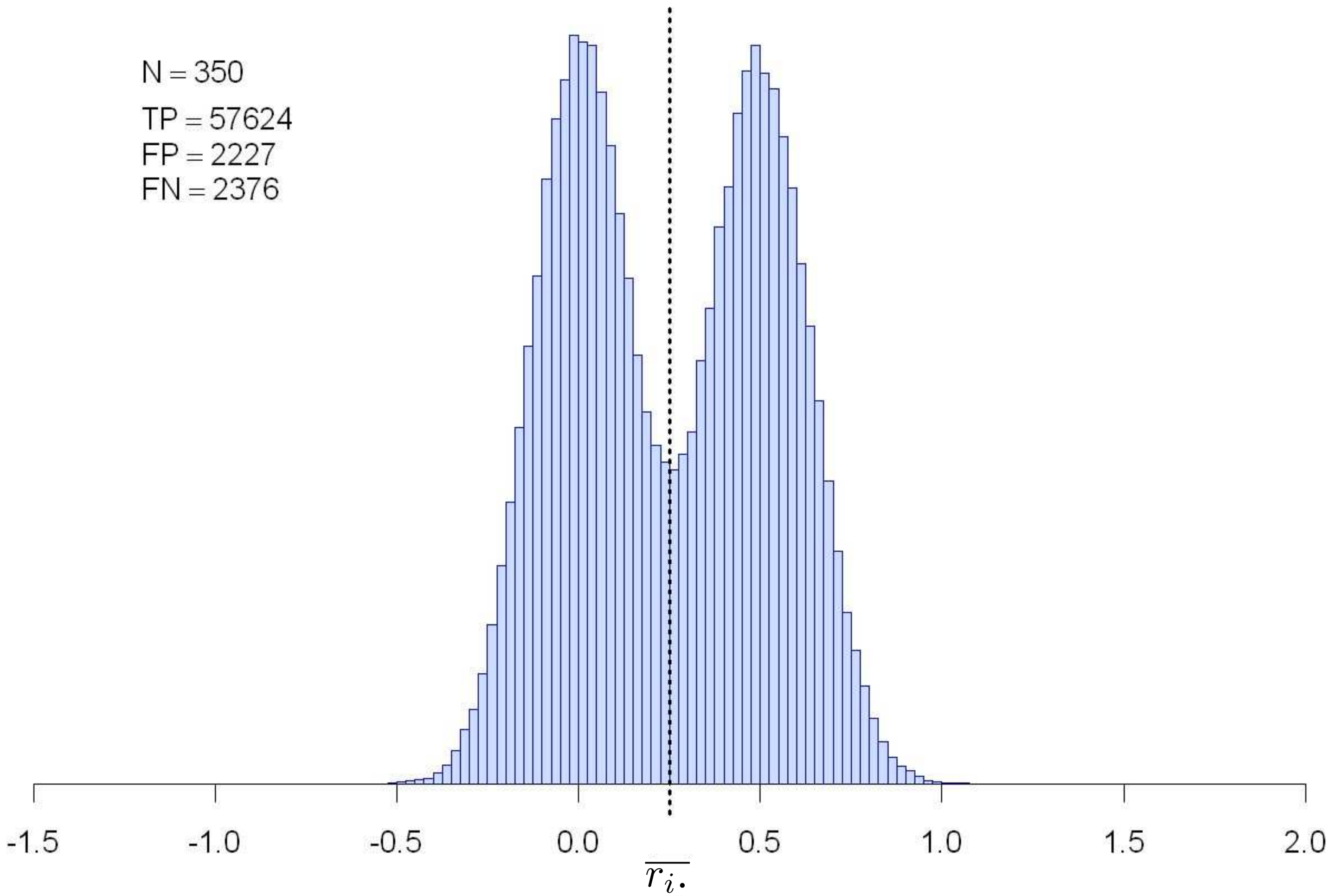
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



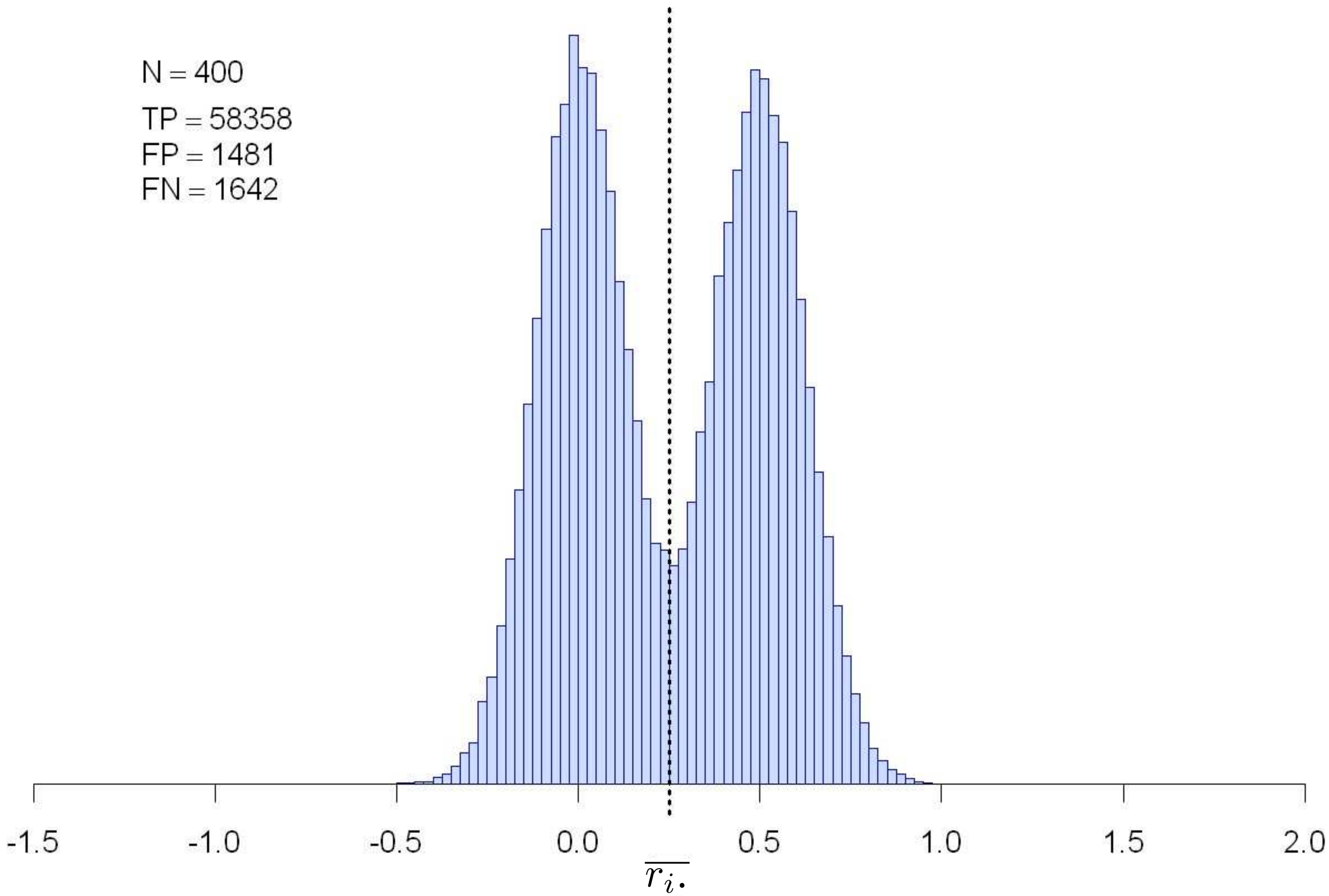
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



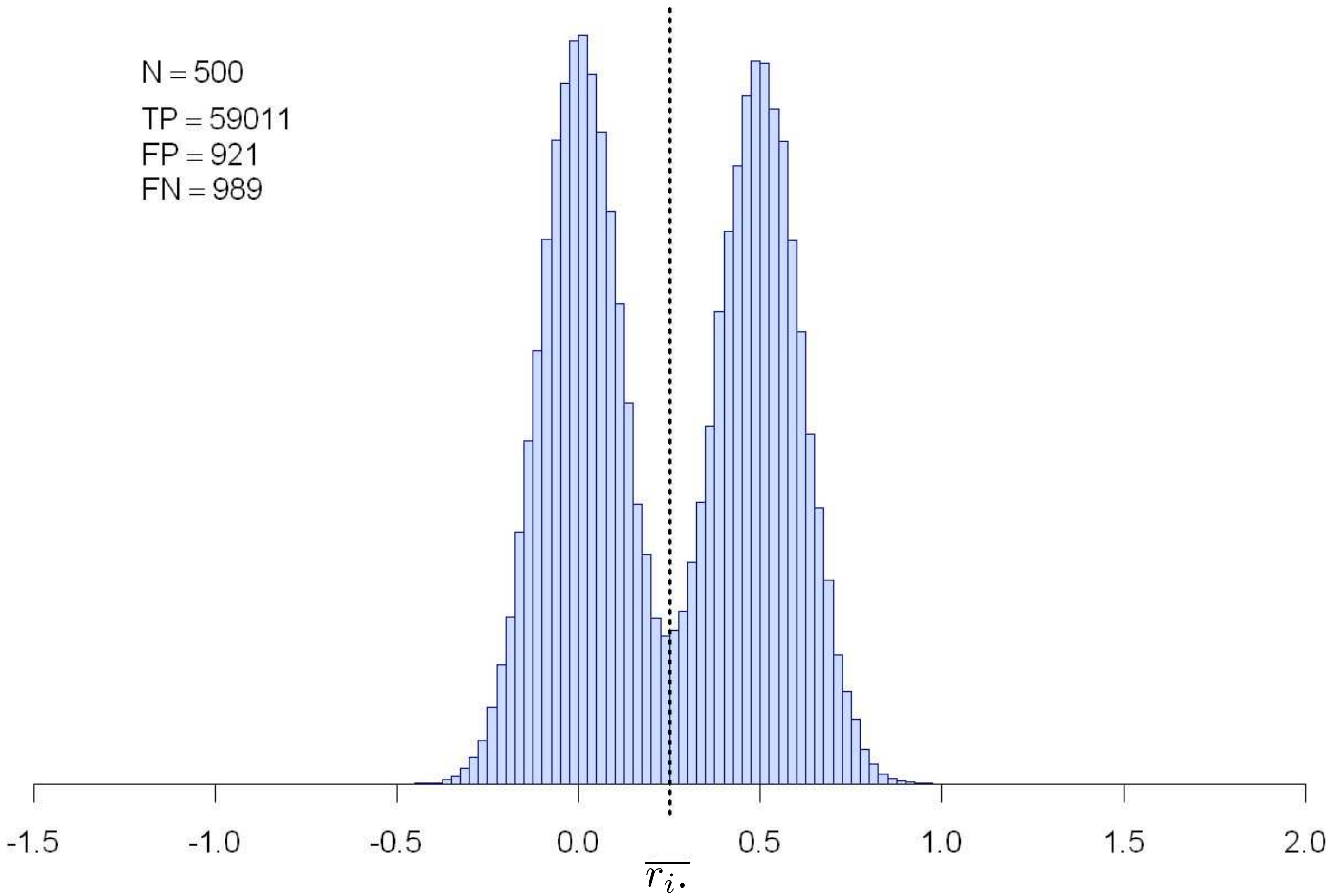
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



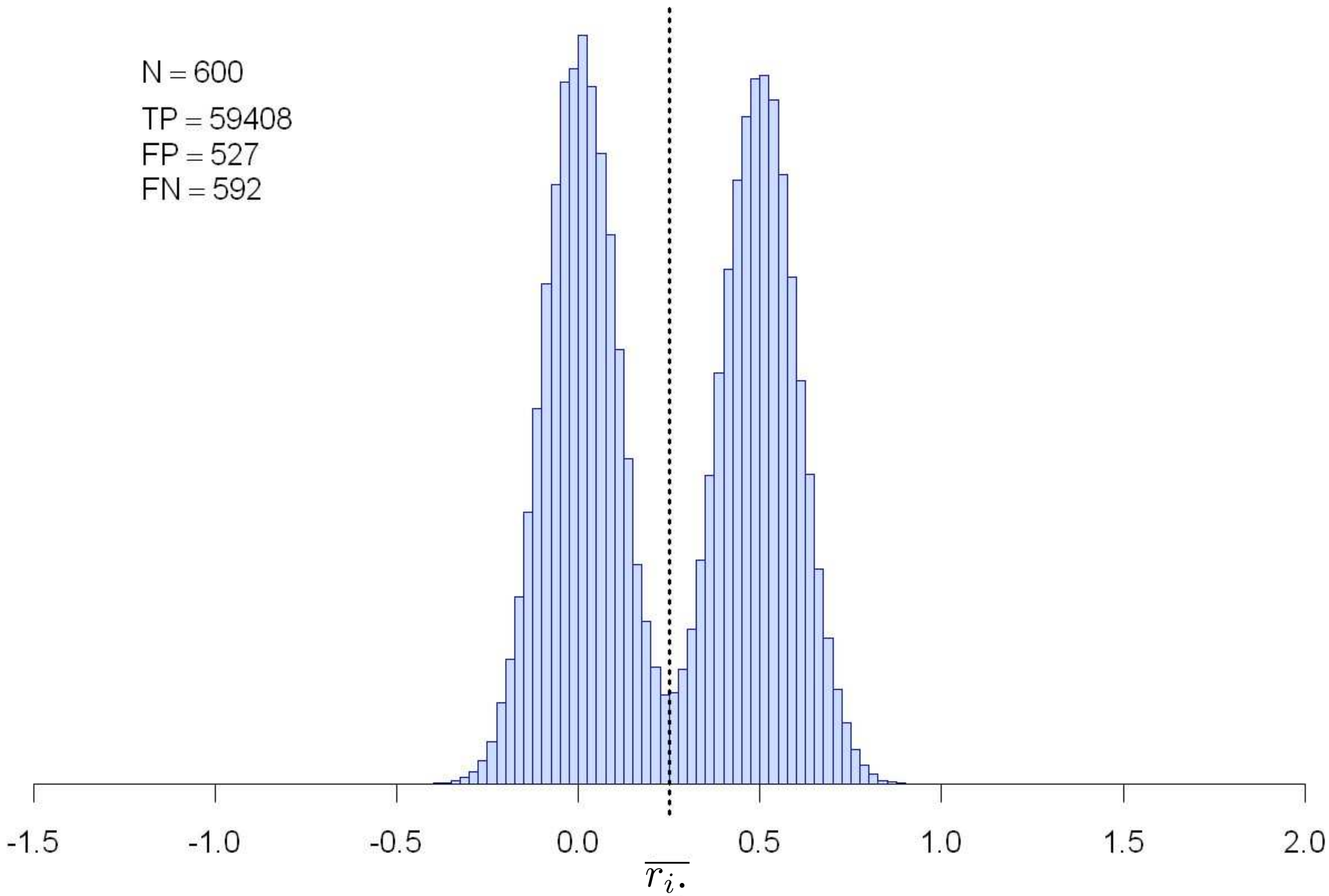
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



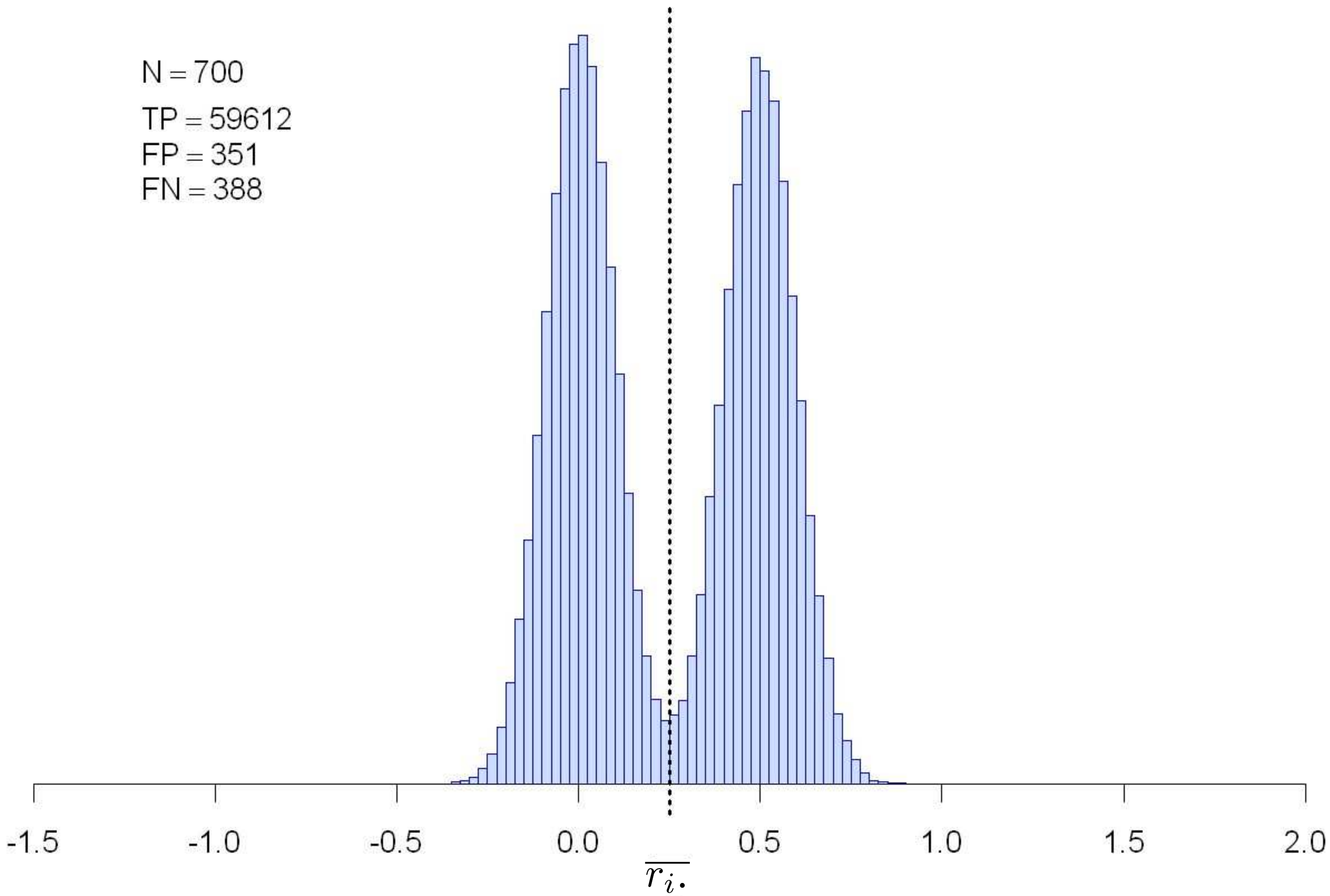
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



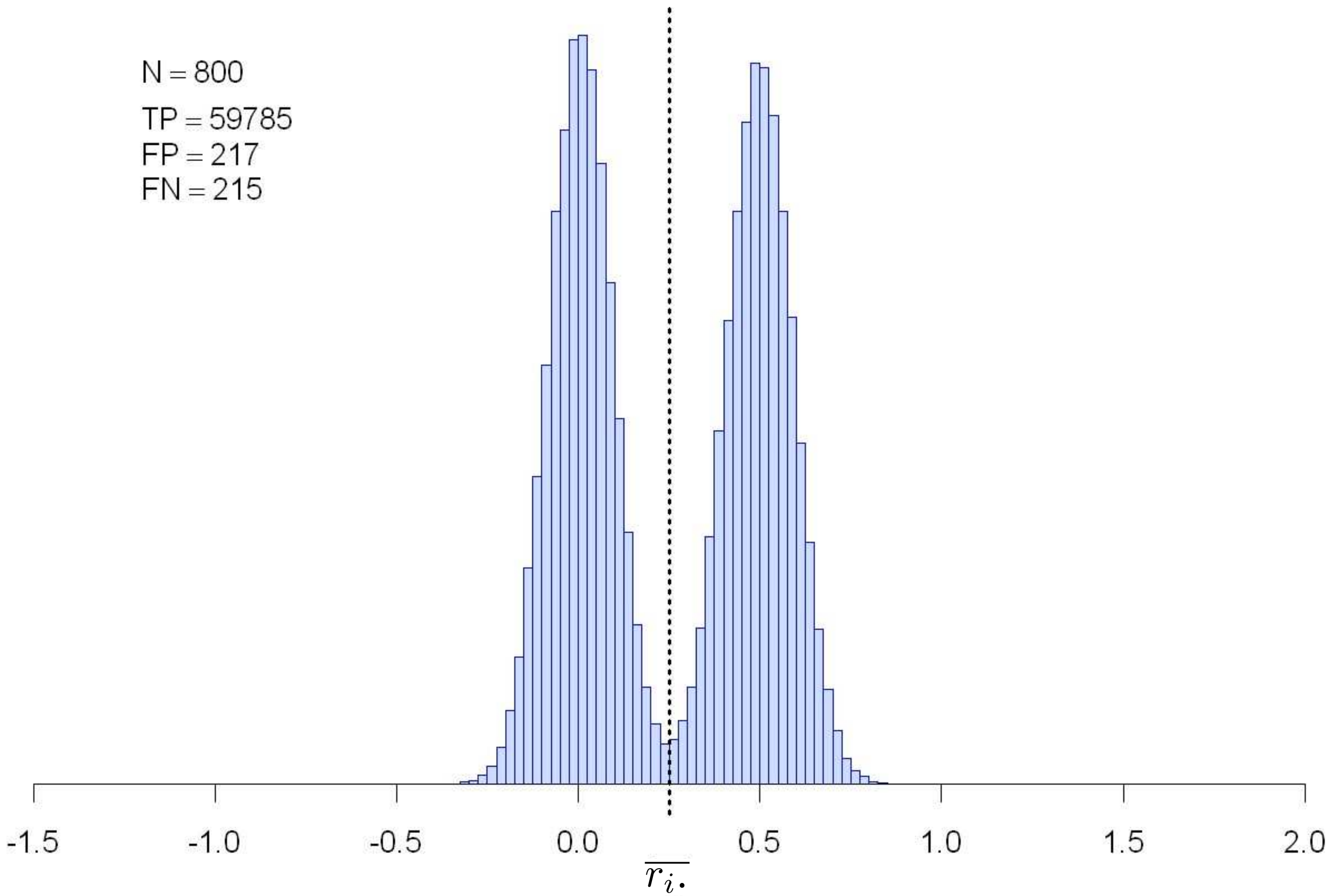
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



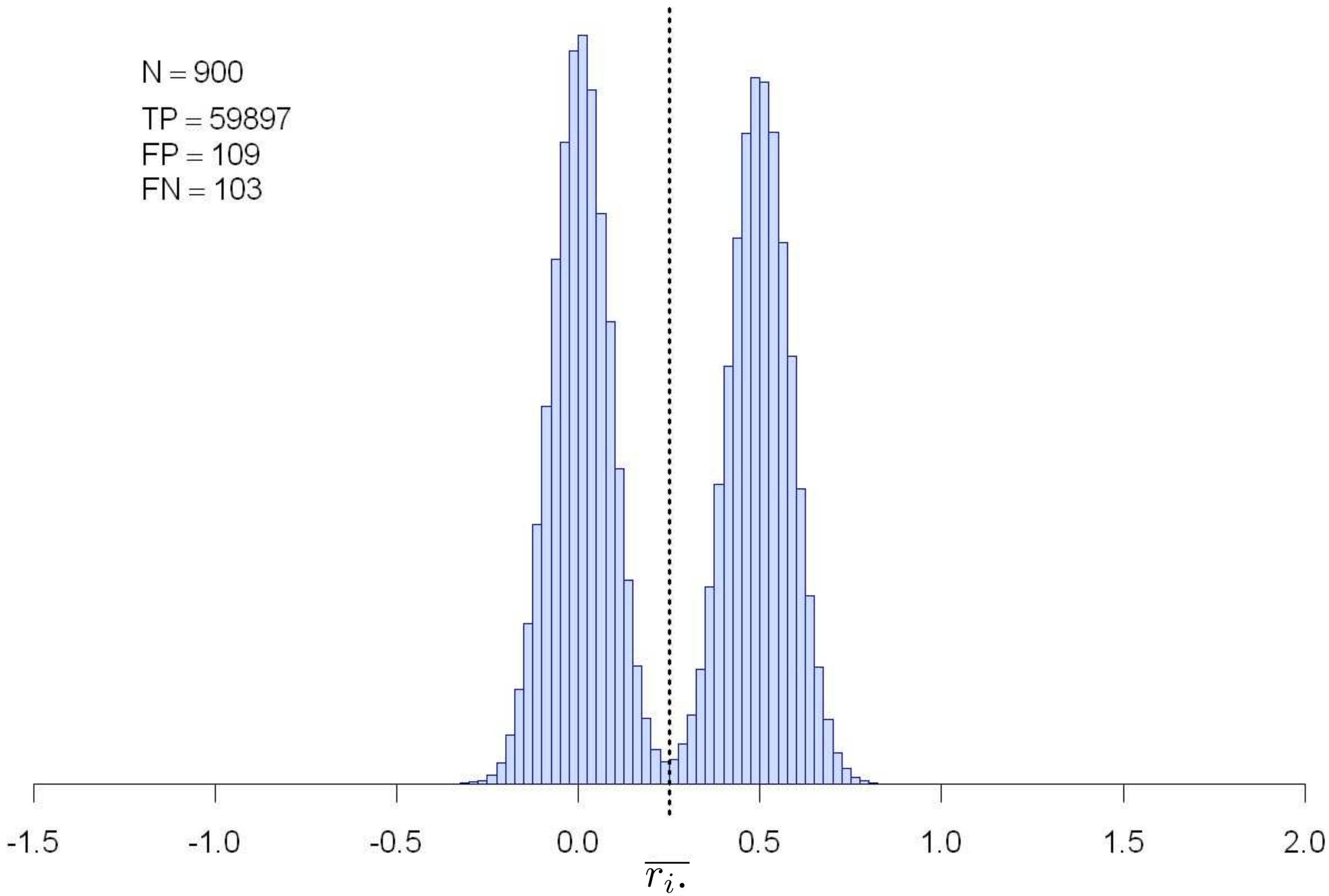
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



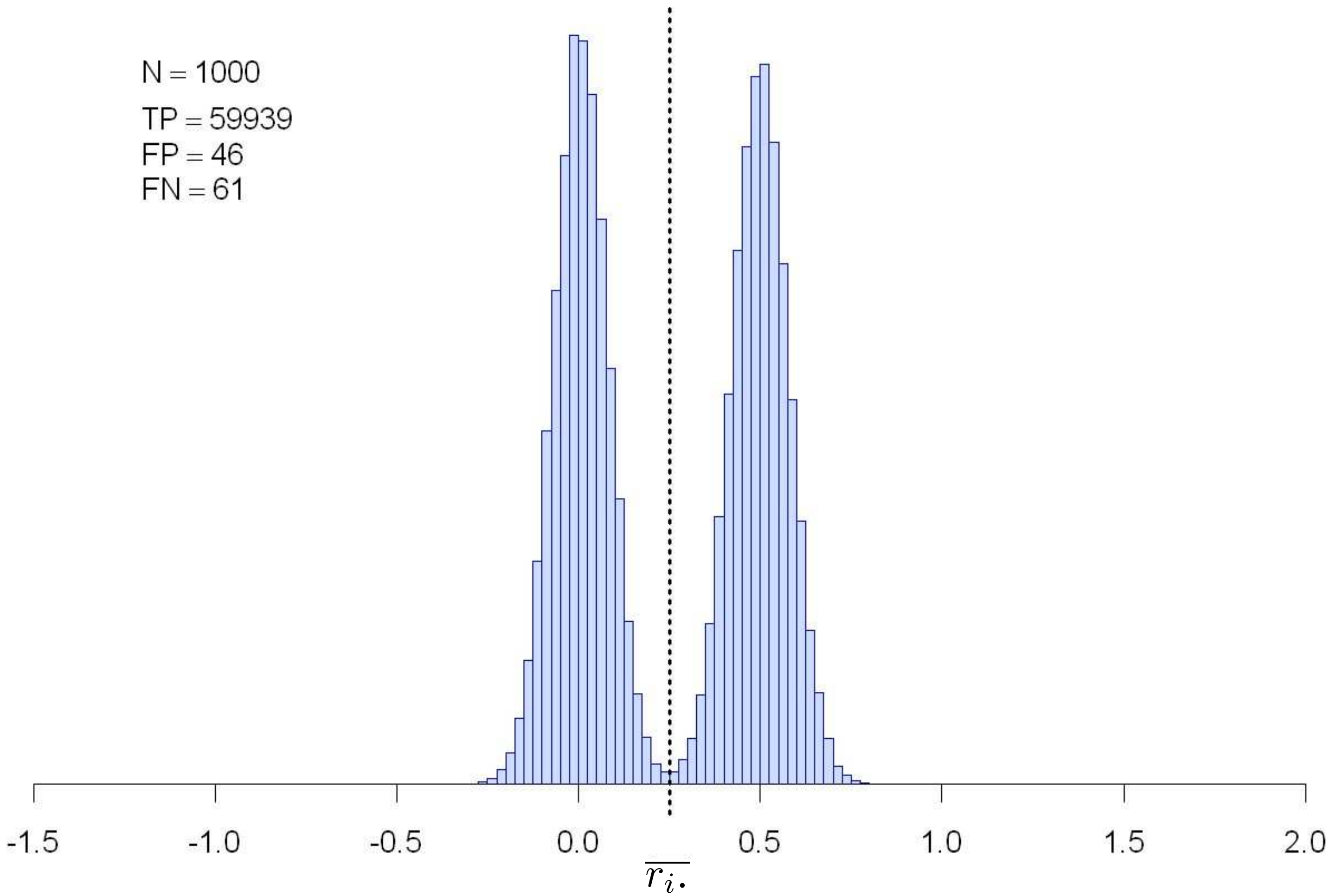
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



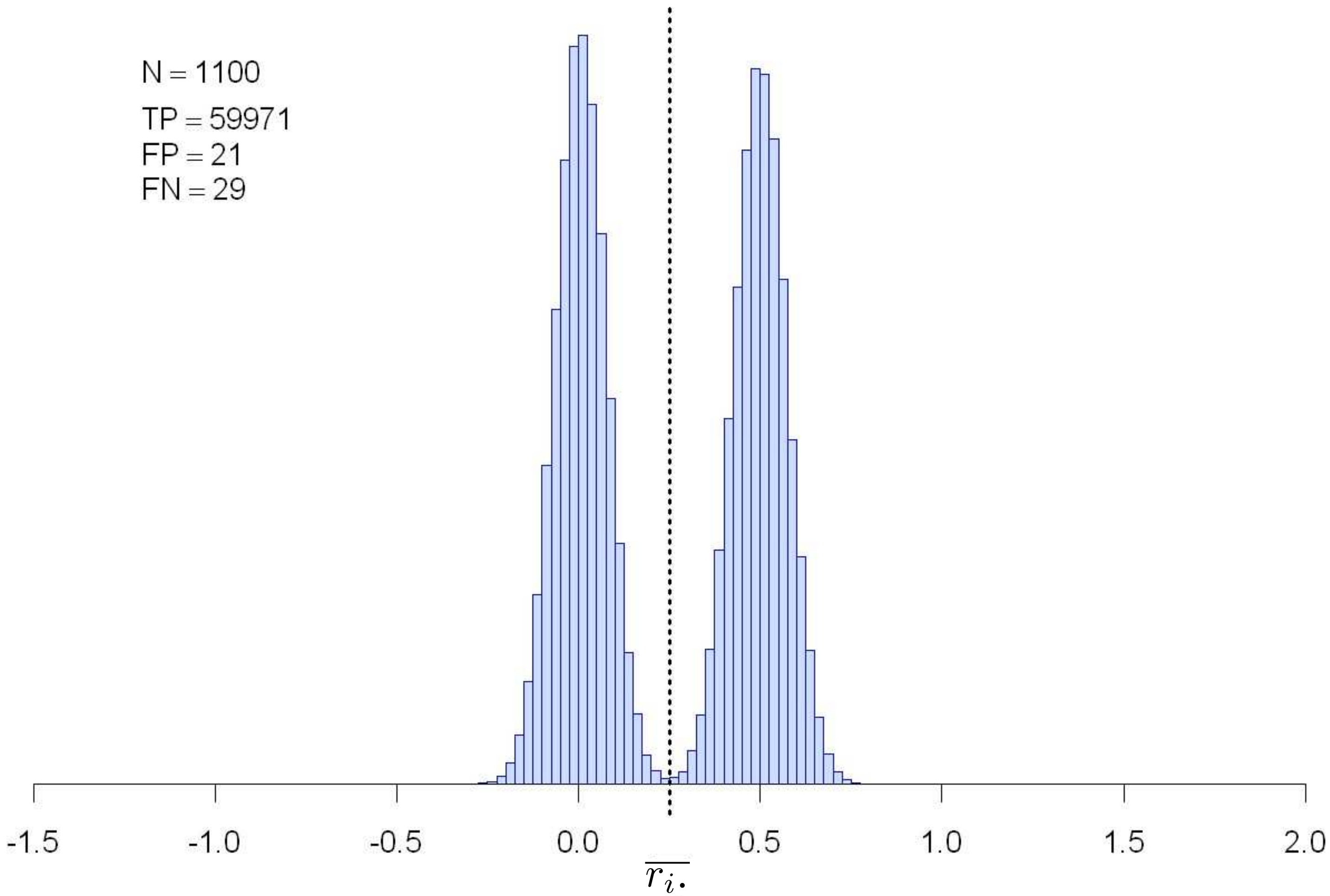
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



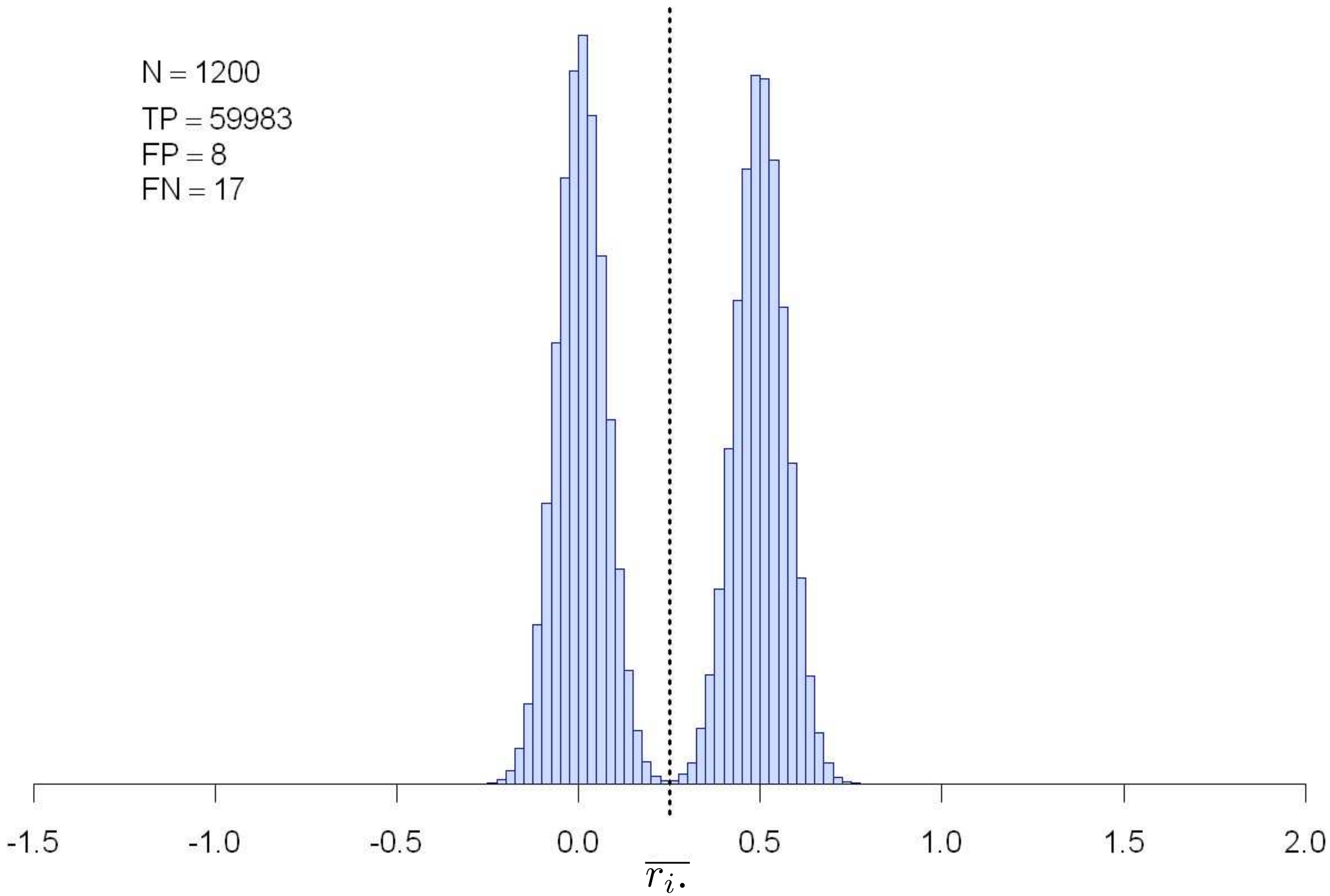
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



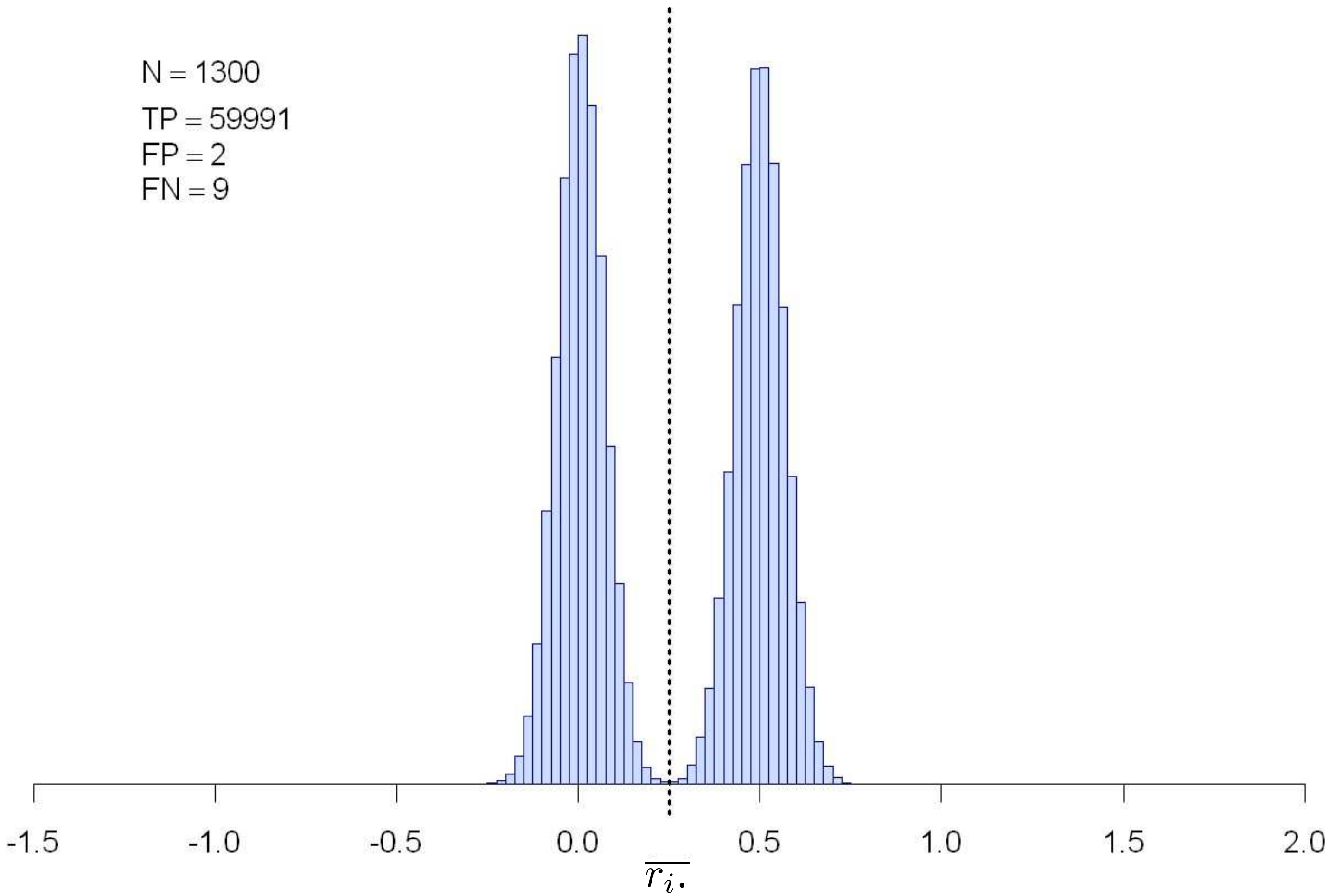
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



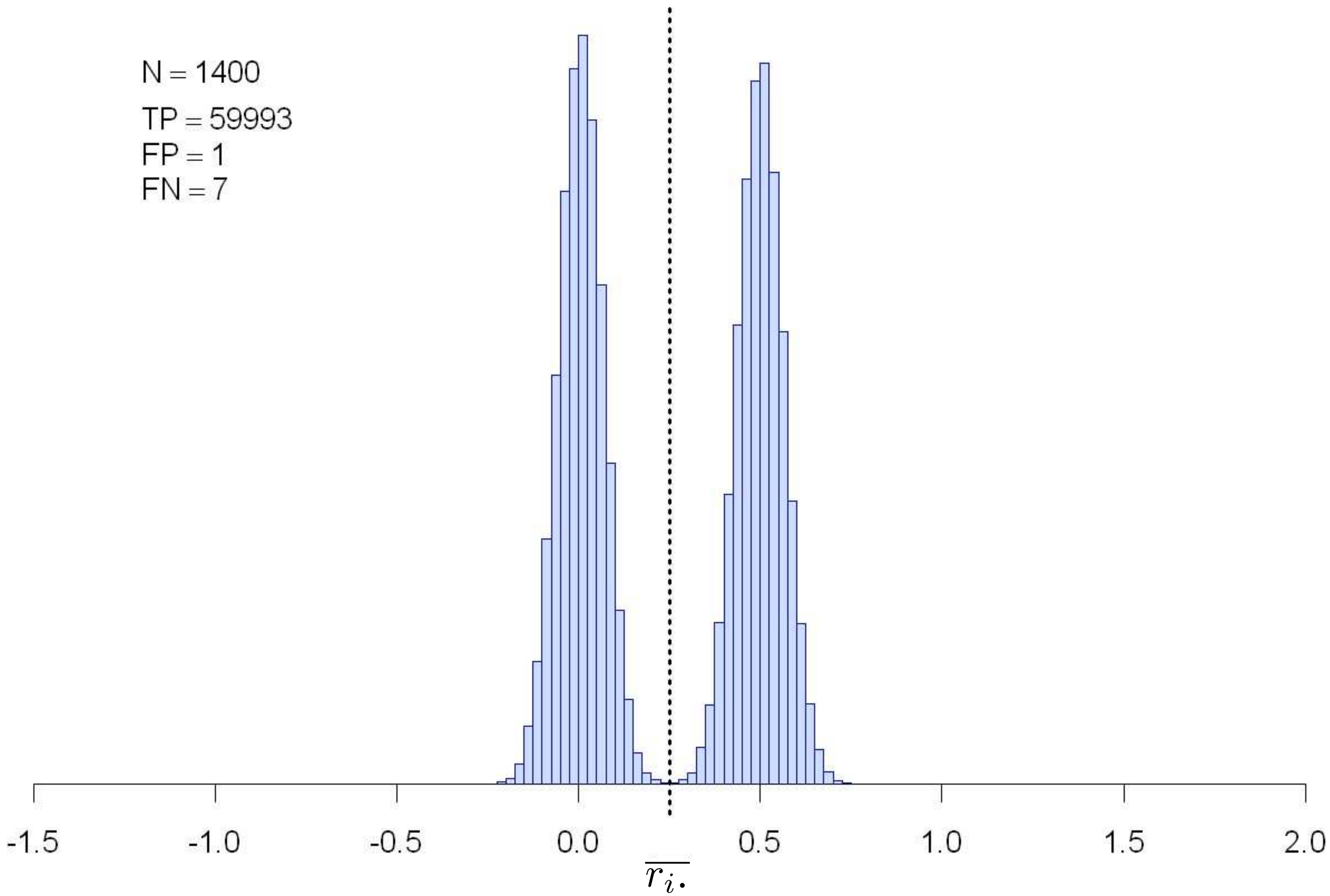
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



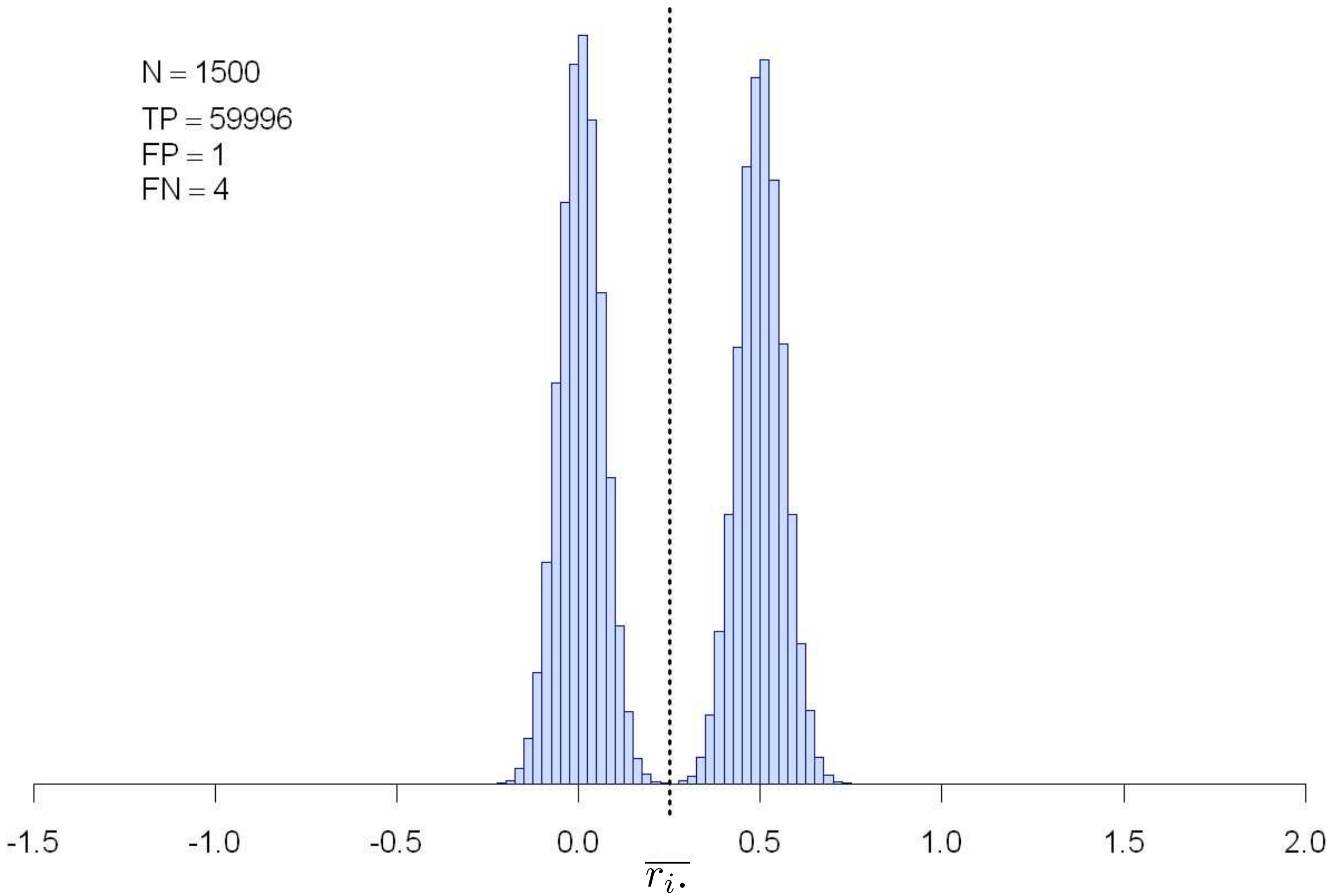
Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Experimental results



Cover images: 400×300, cropped from digital camera images. Random payloads embedded across 60000 locations.

Conclusions

- Considering the WS residuals allows us to estimate payload location, given stego images with payload in the same locations.

  This might identify the embedding software, or allow application of specialised steganalysis tools.

- A few hundred stego images are enough to locate the payloads exactly. Even a few dozen are enough to gain information about the payload.

  This demonstrates why steganographic embedding keys must not be re-used.

- The use of WS residuals is limited to LSB replacement embedding.

End

adk@comlab.ox.ac.uk