CONNECTION SCIENCE
2022, VOL. 34, NO. 1, 1719–1735
https://doi.org/10.1080/09540091.2022.2078792

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Automation of reversible steganographic coding with nonlinear discrete optimisation

Ching-Chun Chang

Department of Computer Science, University of Warwick, Coventry, UK

ABSTRACT

Authentication mechanisms are at the forefront of defending the world from various types of cybercrime. Steganography can serve as an authentication solution through the use of a digital signature embedded in a carrier object to ensure the integrity of the object and simultaneously lighten the burden of metadata management. Nevertheless, despite being generally imperceptible to human sensory systems, any degree of steganographic distortion might be inadmissible in fidelity-sensitive situations such as forensic science, legal proceedings, medical diagnosis and military reconnaissance. This has led to the development of reversible steganography. A fundamental element of reversible steganography is predictive analytics, for which powerful neural network models have been effectively deployed. Another core element is reversible steganographic coding. Contemporary coding is based primarily on heuristics, which offers a shortcut towards sufficient, but not necessarily optimal, capacity–distortion performance. While attempts have been made to realise automatic coding with neural networks, perfect reversibility is unattainable via such learning machinery. Instead of relying on heuristics and machine learning, we aim to derive optimal coding by means of mathematical optimisation. In this study, we formulate reversible steganographic coding as a nonlinear discrete optimisation problem with a logarithmic capacity constraint and a quadratic distortion objective. Linearisation techniques are developed to enable iterative mixed-integer linear programming. Experimental results validate the near-optimality of the proposed optimisation algorithm when benchmarked against a brute-force method.

ARTICLE HISTORY
Received 7 April 2022
Accepted 6 May 2022

KEYWORDS
Automatic coding; mathematical optimisation; reversible steganography

1. Introduction

Steganography is the art and science of concealing information within a carrier object (Anderson & Petitcolas, 1998). The term encompasses a wide range of techniques and applications, including but not limited to covert communications (Fridrich et al., 2005), ownership identification (Cox et al., 1997), copyright protection (Barni et al., 1998), broadcast monitoring (Depovere et al., 1999) and traitor tracing (He & Wu, 2006). An important application of steganography is data authentication, which plays a vital role in cybersecurity. The advent of data-centric artificial intelligence has been accompanied by

CONTACT Ching-Chun Chang c.c.chang@warwickgrad.net

cybersecurity concerns (Boden et al., 2017). It has been reported that intelligent systems are vulnerable to adversarial attacks such as poisonous data collected for re-training during deployment (Muñoz-González et al., 2017), malware codes hidden in neural network parameters (Liu et al., 2020) and invisible perturbations crafted to cause erroneous decisions (Goodfellow et al., 2015). A proper authentication mechanism must ensure that the integrity of data has not been undermined and that the identity of users has not been forged, and thereby protect against these insidious threats.

Digital signatures are a type of authentication technology that is based upon modern cryptography (Rivest et al., 1978). This technology can be incorporated into a trustworthy surveillance camera in such a way that photographs are taken and stored along with digital signatures (Friedman, 1993). However, storing such auxiliary metadata as a separate file entails the risk of accidental loss and mismanagement during the data lifecycle. Steganography can allow auxiliary information about the data to be embedded invisibly within the data itself. Nevertheless, although generally imperceptible to human sensory systems, any degree of steganographic distortion might not be admissible in some fidelity-sensitive situations such as forensic science, legal proceedings, medical diagnosis and military reconnaissance. This is where the notion of reversible computing comes into play (Alattar, 2004; Chang et al., 2018; Coatrieux et al., 2013; De Vleeschouwer et al., 2003; Fridrich et al., 2001; Lee et al., 2007; Wu & Zhang, 2020).

A fundamental element of reversible steganography, in common with lossless compression, is predictive modelling (Rissanen, 1984; Shannon, 1948; Weinberger & Seroussi, 1997). Prediction error modulation is a cutting-edge reversible steganographic technique composed of a *predictive analytics* module and a *reversible coding* module (Celik et al., 2005; Dragoi & Coltuc, 2014; Fallahpour, 2008; Hwang et al., 2016; Li et al., 2011; Sachnev et al., 2009; Thodi & Rodriguez, 2007). The recent development of deep learning has advanced the frontier of reversible steganography. It has been shown that deep neural networks can be applied as powerful predictive models (Chang, 2020, 2021, 2022; Hu & Xiang, 2021). Despite inspiring progress in the analytics module, the design of the coding module is still based largely on heuristics. While there are studies on *end-to-end* deep learning that use neural networks for automatic reversible computing, perfect reversibility cannot be guaranteed (Duan et al., 2019; Lu et al., 2021; Z. Zhang et al., 2019). From a certain point of view, it is hard for a neural network, as a monolithic black box, to follow the intricate procedures of reversible computing (Castelvecchi, 2016). While deep learning is adept at handling the complex nature of the real world (LeCun et al., 2015), reversible computing is more of a mechanical process in which procedures have to be conducted in accordance with rigorous algorithms. Therefore, at the time of writing, it seems advisable to follow a *modular* framework.

The essence of reversible steganographic coding is determining how values change to represent different message digits. Different solutions can lead to different trade-offs between capacity and distortion. Instead of relying on heuristics, this study pursues the development of optimal coding for reversible steganography in order to attain optimal capacity–distortion performance. We model reversible steganographic coding as a mathematical optimisation problem and propose an optimisation algorithm for addressing the nonlinearity of this problem. In particular, the task is to minimise steganographic distortion subject to a capacity constraint, where both objective and constraint are nonlinear functions. We propose linearisation techniques for addressing this nonlinear discrete optimisation problem. The remainder of this paper is organised as follows. Section 2 outlines the background regarding reversible steganography. Section 3 formulates the nonlinear discrete optimisation problem and discusses the complexity of a brute-force search algorithm. Section 4 presents linearisation techniques for tackling the nonlinear discrete optimisation problem. Section 5 analyses the optimality of solutions through simulation experiments. Section 6 provides concluding remarks.

2. Background

Prediction error modulation is a reversible steganographic technique that consists of an analytics module and a coding module. The analytics module begins by splitting a cover image into *context* and *query* sets, denoted by $\boldsymbol{c}$ and $\boldsymbol{q}$ respectively. A conventional method is to arrange pixels in two groups according to a chequered pattern. Then a predictive model is applied to predict the intensities of the query pixels from the intensities of the context pixels. A contemporary practice of predictive modelling is to employ an artificial neural network originally designed for computer vision tasks. The coding module embeds a message $\boldsymbol{\omega}$ into the cover image by modulating the prediction errors $\boldsymbol{\varepsilon} = \boldsymbol{q} - \tilde{\boldsymbol{q}}$. The modulated errors $\boldsymbol{\varepsilon}'$ are then added to the predicted intensities, causing distortion of the query pixels. The stego image is created by merging the context set $\boldsymbol{c}$ and the modulated query set $\boldsymbol{q}'$. The decoding procedure is similar to the encoding procedure. It begins by predicting the query pixel intensities. Since the context set is kept unchanged, the prediction in the decoding phase is guaranteed to be identical to that in the encoding phase given the same predictive model. The message is extracted and the query set is recovered by demodulating the prediction errors. The image is reversed to its original state by merging the context and recovered query sets. The procedures for encoding and decoding are depicted schematically in Figure 1 and also provided in Algorithms 1 and 2. We would like to note that the message may contain certain auxiliary information for handling pixel intensity overflow. This paper does not go into detail about every aspect of the stego-system; instead, our study focuses on the mathematical optimisation of reversible steganographic coding.

Algorithm 1 Encoding

```
Input: cover, $\boldsymbol{\omega}$
Output: stego

  ▷ analytics module
  $[\boldsymbol{c}, \boldsymbol{q}] \leftarrow \text{split}(\text{cover})$
  $[\tilde{\boldsymbol{c}}, \tilde{\boldsymbol{q}}] \leftarrow \text{predict}([\boldsymbol{c}, \mathbf{0}])$

  ▷ coding module
  $\boldsymbol{\varepsilon} \leftarrow \boldsymbol{q} - \tilde{\boldsymbol{q}}$
  $\boldsymbol{\varepsilon}' \leftarrow \text{modulate}(\boldsymbol{\varepsilon}, \boldsymbol{\omega})$
  $\boldsymbol{q}' \leftarrow \tilde{\boldsymbol{q}} + \boldsymbol{\varepsilon}'$
  stego $\leftarrow \text{merge}(\boldsymbol{c}, \boldsymbol{q}')$
```

Algorithm 2 Decoding

```
Input: stego
Output: cover, $\boldsymbol{\omega}$

  ▷ analytics module
  $[\boldsymbol{c}, \boldsymbol{q}'] = \text{split}(\text{stego})$
  $[\tilde{\boldsymbol{c}}, \tilde{\boldsymbol{q}}] = \text{predict}([\boldsymbol{c}, \mathbf{0}])$

  ▷ coding module
  $\boldsymbol{\varepsilon}' = \boldsymbol{q}' - \tilde{\boldsymbol{q}}$
  $[\boldsymbol{\varepsilon}, \boldsymbol{\omega}] = \text{demodulate}(\boldsymbol{\varepsilon}')$
  $\boldsymbol{q} = \tilde{\boldsymbol{q}} + \boldsymbol{\varepsilon}$
  cover $= \text{merge}(\boldsymbol{c}, \boldsymbol{q})$
```

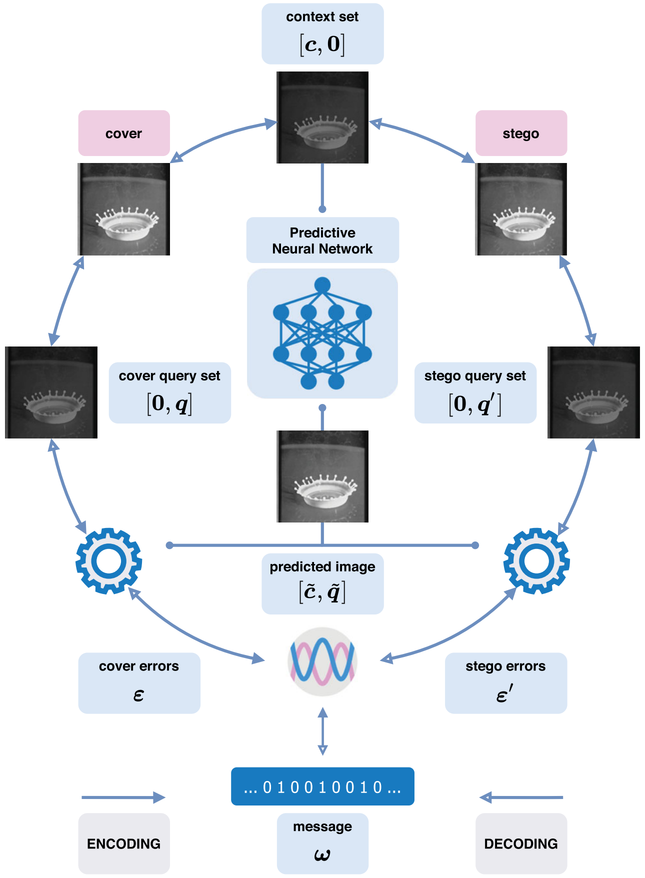

context set
$[\boldsymbol{c}, \mathbf{0}]$
cover
stego
Predictive
Neural Network
cover query set
$[\mathbf{0}, \boldsymbol{q}]$
stego query set
$[\mathbf{0}, \boldsymbol{q}']$
predicted image
$[\tilde{\boldsymbol{c}}, \tilde{\boldsymbol{q}}]$
cover errors
$\boldsymbol{\varepsilon}$
stego errors
$\boldsymbol{\varepsilon}'$
... 0 1 0 0 1 0 0 1 0 ...
ENCODING
message
$\boldsymbol{\omega}$
DECODING

Figure 1. Workflow of reversible steganography with prediction error modulation.

3. Nonlinear discrete optimisation

The essence of reversible steganographic coding is designating one or multiple error values as the carrier and determining how these values change to represent different message digits. A rule of thumb for reversible steganographic coding is to choose the prediction errors of the peak frequency as the carrier. While the peak frequency implies the highest capacity, this capacity-greedy strategy is not necessarily optimal in terms of minimising distortion.

3.1. *Problem definition*

According to the typical law of error, the frequency of an error can be expressed as an exponential function of its numerical magnitude, disregarding sign (Wilson, 1923). In other words, the frequency distribution of prediction errors is expected to centre around zero. In general, a smaller absolute error tends to have a higher occurrence. A special exception is that the occurrence of zero might be lower than the occurrence of a certain small absolute error considering that the latter is the sum of both positive and negative error occurrences. Consider an absolute error histogram as shown in Figure 2. The problem of reversible steganographic coding is to establish a mapping between the values in $[0, n]$ and

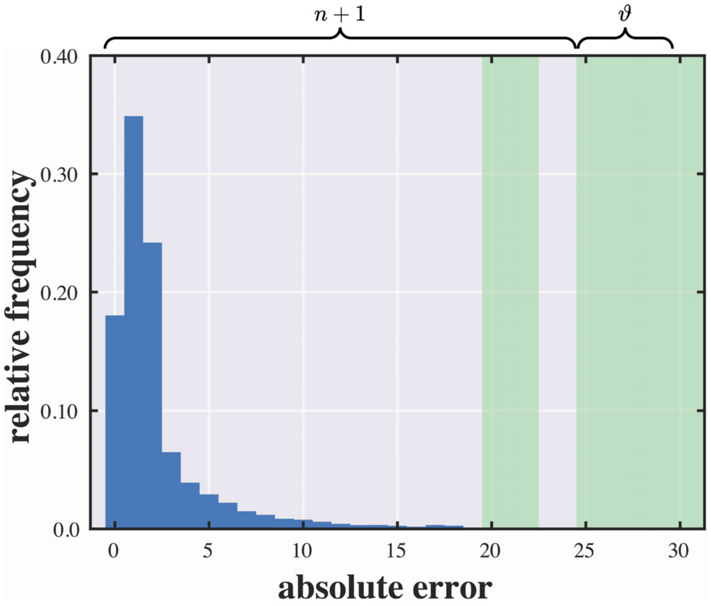


Figure 2. Example of absolute error distribution with highlighted zero occurrences.

the values in $[0, n + \vartheta]$, where ϑ denotes the extra quota and is typically defined as less than or equal to the number of successive empty bins in the absolute error histogram. Encoding is a *one-to-many* mapping that links a cover value to one or more stego values. A message digit can only be represented if the connections are greater than one. Different cover values can never yield the same stego value. This is done in order to avoid an overlap between values (i.e. an ambiguity in decoding). Therefore, a cover value may be changed to a different stego value even if it does not represent any message digit. We impose a constraint that each cover value can only be mapped to the nearest available stego values since a *non-cross* mapping drastically reduces the problem dimension. An example of a cover/stego mapping is illustrated in Figure 3.

3.2. Model formulation

Let us denote by a_i the frequency of the value i and by x_i the number of extra cover-to-stego links for the value i. The total number of links for the value i equals $x_i + 1$. The number of bits that can be represented by modifying the value i is $\log_2(x_i + 1)$ and thus the capacity is computed by

$$\mathfrak{C} = \sum_{i=0}^{n} a_i \log_2(x_i + 1). \tag{1}$$

In fact, the number of bits that can be represented by modifying 0 equals $\log_2(2x_0 + 1)$ because 0 can be mapped to both positive and negative values. For example, there are three different states 0 and ±1 when $x_0 = 1$, and five different states 0, ±1 and ±2 when $x_0 = 2$. To be concise, we simplify the case by mapping 0 randomly to a positive or negative value so that the capacity computation for 0 is identical to that for other values at the cost of slightly underestimating the capacity offered by the former. The probability of changing

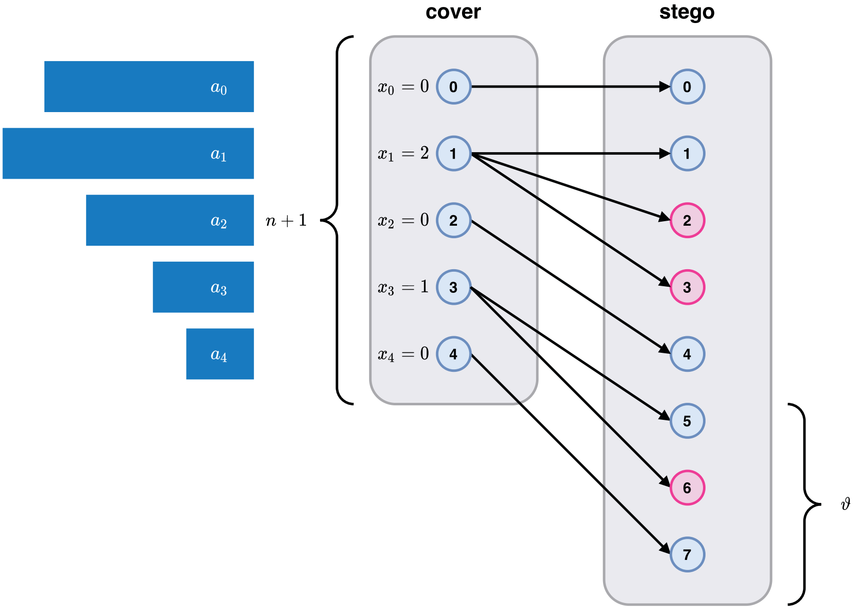


Figure 3. Example of reversible steganographic coding.

a cover value to each stego value is $1/(x_i+1)$. The deviations of the first to the last stego value are $0+y_i$ to x_i+y_i respectively, where y_i denotes the sum of all the previous extra links (i.e. the cumulative deviation). Hence, the expected distortion in terms of the squared deviations is computed by

$$\mathfrak{D} = \sum_{i=0}^{n} a_i \left(\frac{(0+y_i)^2 + (1+y_i)^2 + \cdots + (x_i+y_i)^2}{x_i+1} \right), \tag{2}$$

where

$$y_i = \sum_{j=0}^{i-1} x_j. \tag{3}$$

We can simplify the algebraic expression by

$$\begin{aligned} &\frac{(0+y_i)^2 + (1+y_i)^2 + \cdots + (x_i+y_i)^2}{x_i+1} \\ &\quad = \frac{(0^2 + 2y_i \cdot 0 + y_i^2) + (1^2 + 2y_i \cdot 1 + y_i^2) + \cdots + (x_i^2 + 2y_i \cdot x_i + y_i^2)}{x_i+1} \\ &\quad = \frac{(0^2 + 1^2 + \cdots + x_i^2) + 2y_i(0 + 1 + \cdots + x_i) + y_i^2(x_i+1)}{x_i+1} \\ &\quad = \frac{x_i(x_i+1)(2x_i+1)}{6(x_i+1)} + \frac{2y_i x_i (x_i+1)}{2(x_i+1)} + \frac{y_i^2(x_i+1)}{x_i+1} \\ &\quad = \frac{1}{3}x_i^2 + \frac{1}{6}x_i + x_i y_i + y_i^2. \end{aligned} \tag{4}$$

The reason for computing squared deviations rather than absolute deviations is that image quality is often measured by the peak signal-to-noise ratio (PSNR), which is defined via the mean squared error (MSE). Our goal is to solve for the decision variables $x_i \in \{0, \ldots \vartheta\}$ which minimise the distortion objective subject to the capacity constraint. The sum of all the extra cover-to-stego links is not allowed to exceed the quota ϑ. To summarise, the mathematical optimisation problem for reversible steganographic coding is

$$\begin{aligned} \min \quad & \mathfrak{D} = \sum_{i=0}^{n} a_i \left(\frac{1}{3} x_i^2 + \frac{1}{6} x_i + x_i y_i + y_i^2 \right), \\ \text{s.t.} \quad & \mathfrak{C} = \sum_{i=0}^{n} a_i \log_2(x_i + 1) \geq \text{payload}, \\ & \sum_{i=0}^{n} x_i \leq \vartheta, \\ \text{var.} \quad & x_i \in \{0, \ldots, \vartheta\}, \quad \forall i = 0, \ldots, n. \end{aligned}$$

3.3. Brute-fORCE search

Brute-force search is a baseline method for benchmarking optimisation algorithms. The solution space that exhausts all possible combinations of the decision variables is equal to $(\vartheta + 1)^{n+1} \in \mathcal{O}(c^n)$. By taking account of the quota constraint, we can reduce the solution space from the number of possible combinations to the number of feasible combinations. In number theory and combinatorics, the partition function $\text{part}(t)$ computes the number of ways of writing t as a sum of the positive integers in $[1, t]$. Let $\boldsymbol{\Lambda}_t$ denote a matrix of $\text{part}(t)$ rows and t columns which enumerates all possible partitions:

$$\boldsymbol{\Lambda}_t = \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_{\text{part}(t)} \end{bmatrix} = \begin{bmatrix} \lambda_{1,1} & \cdots & \lambda_{1,t} \\ \vdots & \ddots & \vdots \\ \lambda_{\text{part}(t),1} & \cdots & \lambda_{\text{part}(t),t} \end{bmatrix}. \tag{5}$$

Each vector $\boldsymbol{\lambda}_\ell$ represents a possible partition in which each element is the quantity of a candidate integer (i.e. the summand). For example, $\boldsymbol{\Lambda}_2$, $\boldsymbol{\Lambda}_3$ and $\boldsymbol{\Lambda}_4$ are

$$\begin{array}{ccc} 1 & 2 & \\ 2 & 0 & \boldsymbol{\lambda}_1 \\ 0 & 1 & \boldsymbol{\lambda}_2 \end{array}, \quad \begin{array}{cccc} 1 & 2 & 3 & \\ 3 & 0 & 0 & \boldsymbol{\lambda}_1 \\ 1 & 1 & 0 & \boldsymbol{\lambda}_2 \\ 0 & 0 & 1 & \boldsymbol{\lambda}_3 \end{array}, \quad \begin{array}{ccccc} 1 & 2 & 3 & 4 & \\ 4 & 0 & 0 & 0 & \boldsymbol{\lambda}_1 \\ 2 & 1 & 0 & 0 & \boldsymbol{\lambda}_2 \\ 0 & 2 & 0 & 0 & \boldsymbol{\lambda}_3 \\ 1 & 0 & 1 & 0 & \boldsymbol{\lambda}_4 \\ 0 & 0 & 0 & 1 & \boldsymbol{\lambda}_5 \end{array}.$$

The total number of feasible solutions can be calculated by adding up the number of feasible solutions given by each individual partition matrix from $\boldsymbol{\Lambda}_1$ to $\boldsymbol{\Lambda}_\vartheta$; that is,

$$\sum_{t=1}^{\vartheta} \text{feasible}(\boldsymbol{\Lambda}_t, n^*), \tag{6}$$

where $n^* = n + 1$ denotes the number of integers in $[0, n]$. For each matrix $\boldsymbol{\Lambda}_t$, the number of feasible solutions is computed by summing the number of possible combinations given by each partition vector $\boldsymbol{\lambda}_\ell$, denoted by

$$\text{feasible}(\boldsymbol{\Lambda}_t, n^*) = \sum_{\ell=1}^{\text{part}(t)} \text{comb}(\boldsymbol{\lambda}_\ell, n^*). \tag{7}$$

A combination is a selection of values from a set of n^* values based on a given partition vector and hence the number of combinations is computed by

$$\text{comb}(\boldsymbol{\lambda}_\ell, n^*) = \prod_{i=1}^{t} \binom{n^* - \sum_{j=1}^{i-1} \lambda_j^*}{\lambda_i^*}, \tag{8}$$

where $\lambda_i^* = \lambda_{\ell,i}$ represents a convenient notation without explicitly writing out the index of the partition vector (for reducing the verbosity). The number of combinations is a product of t binomial coefficients and each term is meant to choose (and remove) an unordered subset of λ_i^* values from the remaining values in the set of n^* values. Let us take $\boldsymbol{\Lambda}_3$ for example. The number of combinations for partition vectors $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$ and $\boldsymbol{\lambda}_3$ are computed as follows:

$$\begin{aligned}
\text{comb}(\boldsymbol{\lambda}_1, n^*) &= \binom{n^*}{3}\binom{n^* - 3}{0}\binom{n^* - 3 - 0}{0}, \\
\text{comb}(\boldsymbol{\lambda}_2, n^*) &= \binom{n^*}{1}\binom{n^* - 1}{1}\binom{n^* - 1 - 1}{0}, \\
\text{comb}(\boldsymbol{\lambda}_3, n^*) &= \binom{n^*}{0}\binom{n^* - 0}{0}\binom{n^* - 0 - 0}{1}.
\end{aligned}$$

The number of combinations can be approximated by

$$\begin{aligned}
&\prod_{i=1}^{t} \binom{n^* - \sum_{j=1}^{i-1} \lambda_j^*}{\lambda_i^*} \\
&= \binom{n^*}{\lambda_1^*}\binom{n^* - \lambda_1^*}{\lambda_2^*} \cdots \binom{n^* - \lambda_1^* - \lambda_2^* - \cdots - \lambda_{t-1}^*}{\lambda_t^*} \\
&= \frac{n^*!}{\lambda_1^*!(n^* - \lambda_1^*)!} \times \frac{(n^* - \lambda_1^*)!}{\lambda_2^*!(n^* - \lambda_1^* - \lambda_2^*)!} \times \cdots \times \frac{(n - \sum_{j=1}^{t-1} \lambda_j^*)!}{\lambda_t^*!(n - \sum_{j=1}^{t} \lambda_j^*)!} \\
&= \frac{n^*!}{\lambda_1^*!\lambda_2^*! \ldots \lambda_t^*!(n^* - \sum_{j=1}^{t} \lambda_j^*)!} \\
&= \frac{n^*(n^* - 1)(n^* - 2) \ldots (n^* - (\sum_{j=1}^{t} \lambda_j^* - 1))}{\lambda_1^*!\lambda_2^*! \ldots \lambda_t^*!} \\
&\leq \frac{n^*(n^* - 1)(n^* - 2) \ldots (n^* - (t - 1))}{\lambda_1^*!\lambda_2^*! \ldots \lambda_t^*!} \approx n^t.
\end{aligned} \tag{9}$$

Hence, the complexity of this brute-force algorithm is approximately equal to

$$\sum_{t=1}^{\vartheta}\sum_{\ell=1}^{\text{part}(t)} \text{comb}(\boldsymbol{\lambda}_{\ell}, n^{*}) \approx \sum_{t=1}^{\vartheta} \text{part}(t) \cdot n^{t} \in \mathcal{O}(n^{c}). \tag{10}$$

4. Linearisation

The difficulty of our optimisation problem lies in the nonlinear nature of the capacity constraint and the distortion objective. To apply off-the-shelf optimisation tools, we have to tackle these nonlinearities.

4.1. Logarithmic capacity constraint

The capacity constraint involves the calculation of logarithm of variables $\log_2(x_i + 1)$. The logarithmic function is nonlinear. A useful linearisation trick is to re-model the problem with binary-integer variables. We binarise each decision variable x_i with the domain $[0, \vartheta]$ into a 0/1 vector (or a one-hot vector) of length $\vartheta + 1$, as illustrated in Figure 4. The vector consists of 0s with the exception of a single 1 whose position indicates the value of x_i; that is,

$$\mathbf{x}_i = [\mathrm{x}_i^0, \ldots, \mathrm{x}_i^{\vartheta}] \in \{0, 1\}^{\vartheta+1}, \tag{11}$$

such that

$$\mathbb{1} \cdot \mathbf{x}_i^{\mathsf{T}} = 1, \quad \forall i = 0, \ldots, n. \tag{12}$$

We can retrieve x_i using the dot product of vectors

$$x_i = [0, \ldots, \vartheta] \cdot \mathbf{x}_i^{\mathsf{T}} = \mathbf{v}\mathbf{x}_i^{\mathsf{T}}. \tag{13}$$

Accordingly, the quota constraint becomes

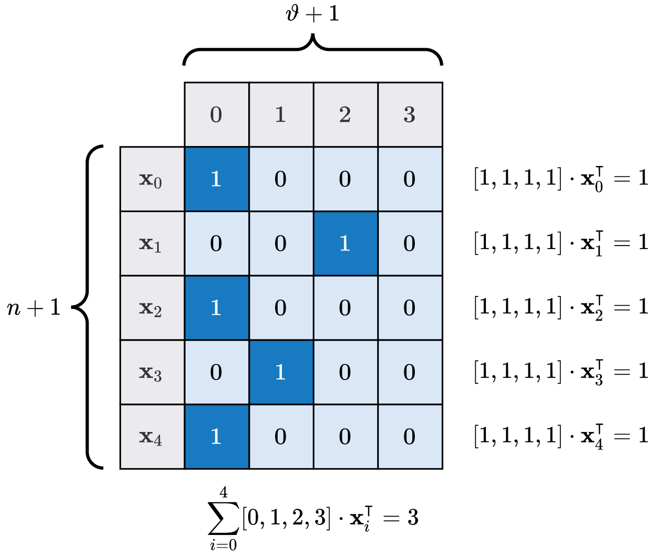
$$\sum_{i=0}^{n} \mathbf{v}\mathbf{x}_i^{\mathsf{T}} \leq \vartheta. \tag{14}$$

In a similar manner, the logarithm can be derived using the dot product of vectors

$$\begin{aligned} \log_2(x_i + 1) &= \left[\log_2(0+1), \ldots, \log_2(\vartheta + 1)\right] \cdot \mathbf{x}_i^{\mathsf{T}} \\ &= \mathbf{v}_{\log}\mathbf{x}_i^{\mathsf{T}}. \end{aligned} \tag{15}$$

Hence, we rewrite the capacity constraint as

$$\mathfrak{C} = \sum_{i=0}^{n} a_i \mathbf{v}_{\log}\mathbf{x}_i^{\mathsf{T}}. \tag{16}$$



	0	1	2	3
$\mathbf{x}_0$	1	0	0	0
$\mathbf{x}_1$	0	0	1	0
$\mathbf{x}_2$	1	0	0	0
$\mathbf{x}_3$	0	1	0	0
$\mathbf{x}_4$	1	0	0	0

$[1,1,1,1] \cdot \mathbf{x}_0^\mathsf{T} = 1$
$[1,1,1,1] \cdot \mathbf{x}_1^\mathsf{T} = 1$
$[1,1,1,1] \cdot \mathbf{x}_2^\mathsf{T} = 1$
$[1,1,1,1] \cdot \mathbf{x}_3^\mathsf{T} = 1$
$[1,1,1,1] \cdot \mathbf{x}_4^\mathsf{T} = 1$

$$\sum_{i=0}^{4} [0,1,2,3] \cdot \mathbf{x}_i^\mathsf{T} = 3$$

Figure 4. Example of binary-integer decision variable.

4.2. *Quadratic distortion objective*

The distortion objective involves three nonlinear terms x_i^2, y_i^2 and $x_i y_i$. These terms are quadratic functions of variables. The first term can be approached using the dot product as before; that is

$$x_i^2 = [0^2, \ldots, \vartheta^2] \cdot \mathbf{x}_i^\mathsf{T} = \mathbf{v}_{\text{sq}} \mathbf{x}_i^\mathsf{T}. \tag{17}$$

The remaining two terms contain the partial sum of variables y_i, which is computed by

$$y_i = \sum_{j=0}^{i-1} \mathbf{v} \mathbf{x}_j^\mathsf{T}. \tag{18}$$

To linearise the univariate quadratic term y_i^2 and the bivariate quadratic term $x_i y_i$, we introduce two non-negative continuous *slack* variables $z_{y_i^2} \geq 0$ and $z_{x_i y_i} \geq 0$. Replacing the quadratic terms with the dot product and the slack variables results in a linear distortion objective

$$\mathfrak{D} = \sum_{i=0}^{n} a_i \left(\frac{1}{3} \mathbf{v}_{\text{sq}} \mathbf{x}_i^\mathsf{T} + \frac{1}{6} \mathbf{v} \mathbf{x}_i^\mathsf{T} + z_{x_i y_i} + z_{y_i^2} \right). \tag{19}$$

We begin by solving this mixed-integer linear programming problem, which does not yet reflect the quadratic terms regarding cumulative distortion, and obtain an initial solution comprising $\tilde{\mathbf{x}}_i$, $\tilde{z}_{y_i^2}$, and $\tilde{z}_{x_i y_i}$. The initial slack variables would be zeros because the objective is to minimise distortion. To make the slack variables reflect the quadratic terms properly, we add the following constraints

$$\begin{aligned} z_{y_i^2} &\geq y_i^2, \\ z_{x_i y_i} &\geq x_i y_i. \end{aligned} \tag{20}$$

In this way, we reformulate a problem with a nonlinear objective into a problem with a linear objective and nonlinear constraints. We make use of the solution obtained previously to linearise these nonlinear constraints and solve the mixed-integer linear programming problem iteratively. To begin with, we express the variables in terms of the previous solution:

$$\begin{aligned} x_i &= \tilde{x}_i + \delta_{x_i}, \\ y_i &= \tilde{y}_i + \delta_{y_i}, \end{aligned} \tag{21}$$

where $\tilde{x}_i$ and $\tilde{y}_i$ are treated as constants. Then, we apply the first-order Taylor series to approximate the univariate quadratic term as

$$\begin{aligned} f(y_i) &= f(\tilde{y}_i + \delta_{y_i}) \\ &= f(\tilde{y}_i) + f'(\tilde{y}_i)\delta_{y_i} + \cdots \\ &= \tilde{y}_i^2 + 2\tilde{y}_i\delta_{y_i} + \cdots \\ &\approx \tilde{y}_i^2 + 2\tilde{y}_i(y_i - \tilde{y}_i) \\ &= 2\tilde{y}_i y_i - \tilde{y}_i^2, \end{aligned} \tag{22}$$

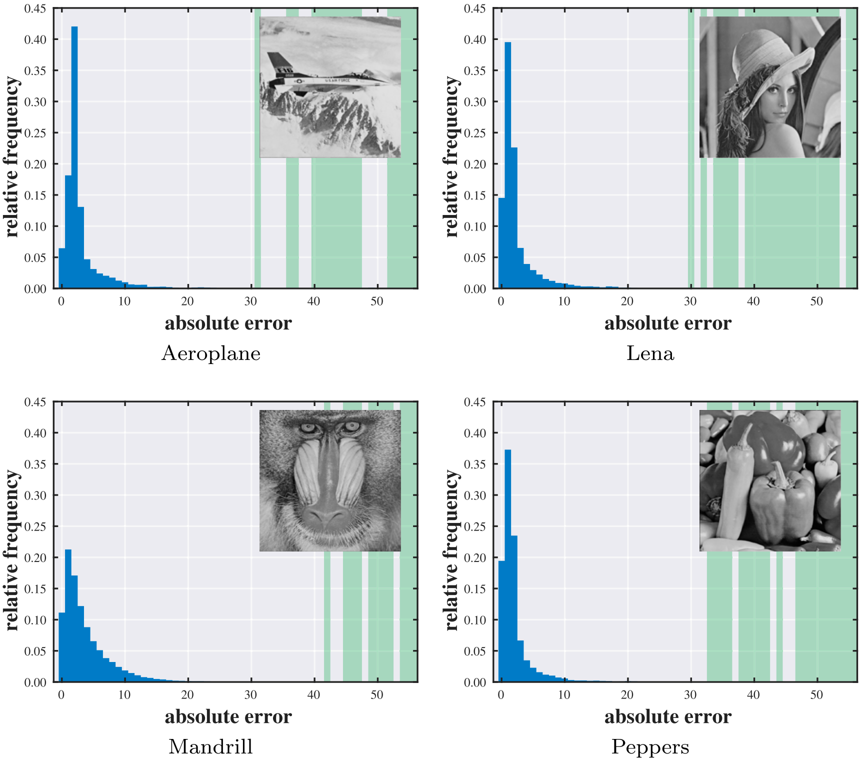


Figure 5. Absolute error histograms with highlighted empty bins. (a) Aeroplane. (b) Lena. (c) Mandrill and (d) Peppers.

and similarly the bivariate quadratic term as

$$
\begin{aligned}
f(x_i, y_i) &= f(\tilde{x}_i + \delta_{x_i}, \tilde{y}_i + \delta_{y_i}) \\
&= f(\tilde{x}_i, \tilde{y}_i) + \frac{\partial f(\tilde{x}_i, \tilde{y}_i)}{\partial x_i}\delta_{x_i} + \frac{\partial f(\tilde{x}_i, \tilde{y}_i)}{\partial y_i}\delta_{y_i} + \cdots \\
&= \tilde{x}_i\tilde{y}_i + \tilde{y}_i\delta_{x_i} + \tilde{x}_i\delta_{y_i} + \cdots \\
&\approx \tilde{x}_i\tilde{y}_i + \tilde{y}_i(x_i - \tilde{x}_i) + \tilde{x}_i(y_i - \tilde{y}_i) \\
&= \tilde{x}_i y_i + \tilde{y}_i x_i - \tilde{x}_i\tilde{y}_i.
\end{aligned} \tag{23}
$$

As a result, the nonlinear constraints are transformed into linear constraints

$$
\begin{aligned}
2\tilde{y}_i y_i - z_{y_i^2} &\le \tilde{y}_i^2, \\
\tilde{x}_i y_i + \tilde{y}_i x_i - z_{x_i y_i} &\le \tilde{x}_i\tilde{y}_i.
\end{aligned} \tag{24}
$$

To recapitulate, the nonlinear discrete optimisation problem is approached by means of an iterative method that solves a mixed-integer linear programming problem with binary-integer variables and non-negative continuous slack variables:

$$
\begin{aligned}
\text{min} \quad & \mathfrak{D} = \sum_{i=0}^{n} a_i \left(\frac{1}{3}\mathbf{v}_{\text{sq}}\mathbf{x}_i^{\mathsf{T}} + \frac{1}{6}\mathbf{v}\mathbf{x}_i^{\mathsf{T}} + z_{x_i y_i} + z_{y_i^2} \right), \\
\text{s.t.} \quad & \mathfrak{C} = \sum_{i=0}^{n} a_i \mathbf{v}_{\log}\mathbf{x}_i^{\mathsf{T}} \ge \text{payload}, \\
& \sum_{i=0}^{n} \mathbf{v}\mathbf{x}_i^{\mathsf{T}} \le \vartheta, \\
& \mathbb{1} \cdot \mathbf{x}_i^{\mathsf{T}} = 1, \quad \forall i = 0, \ldots, n, \\
& 2\tilde{y}_i y_i - z_{y_i^2} \le \tilde{y}_i^2, \quad \forall i = 0, \ldots, n, \\
& \tilde{x}_i y_i + \tilde{y}_i x_i - z_{x_i y_i} \le \tilde{x}_i\tilde{y}_i, \quad \forall i = 0, \ldots, n, \\
\text{var.} \quad & \mathbf{x}_i \in \{0,1\}^{\vartheta+1}, \quad \forall i = 0, \ldots, n, \\
& z_{y_i^2} \ge 0, \quad \forall i = 0, \ldots, n, \\
& z_{x_i y_i} \ge 0, \quad \forall i = 0, \ldots, n, \\
\text{where} \quad & x_i = \mathbf{v}\mathbf{x}_i^{\mathsf{T}} \quad \& \quad y_i = \sum_{j=0}^{i-1} \mathbf{v}\mathbf{x}_j^{\mathsf{T}}.
\end{aligned}
$$

5. Simulation

We carry out experimental analysis on the optimality of the proposed method benchmarked against the brute-force method. The experimental setup is described as follows. For the predictive model, we use the residual dense network (RDN), which has its origins in low-level computer vision tasks such as super-resolution imaging (Y. Zhang et al., 2018)

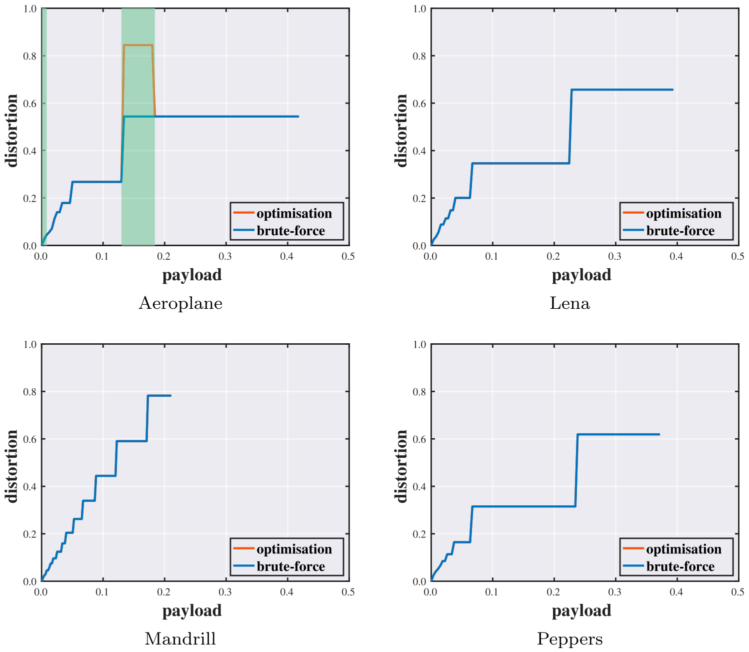


Figure 6. Payload–distortion curves for optimality analysis against brute-force search with highlighted discrepancies ($\vartheta = 1$).

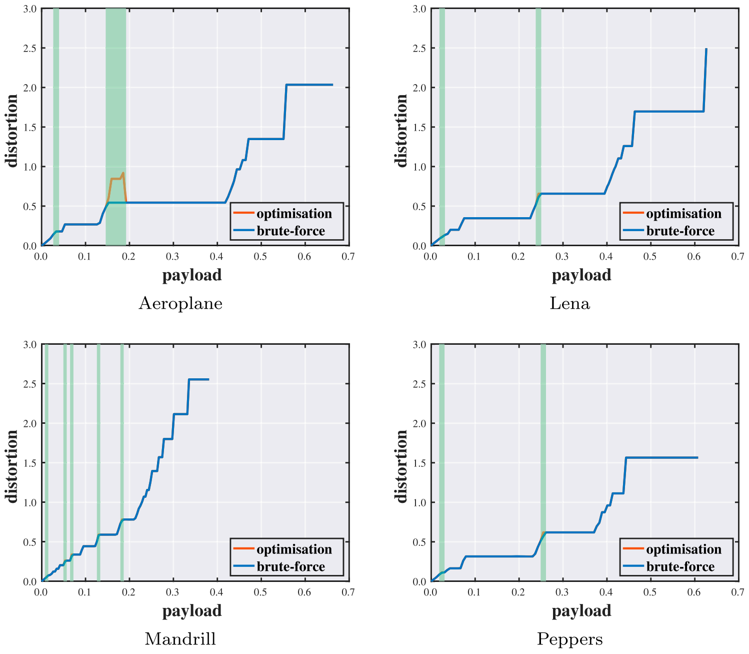
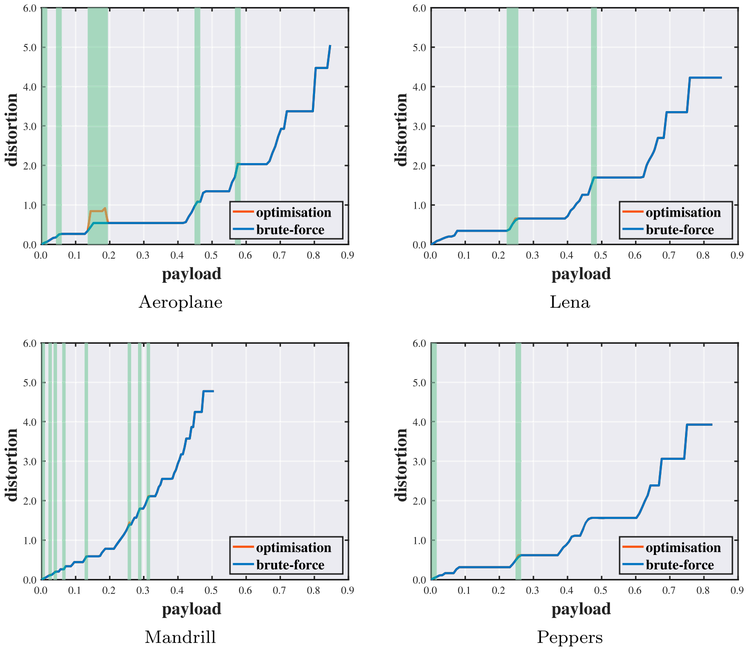


Figure 7. Payload–distortion curves for optimality analysis against brute-force search with highlighted discrepancies ($\vartheta = 2$).



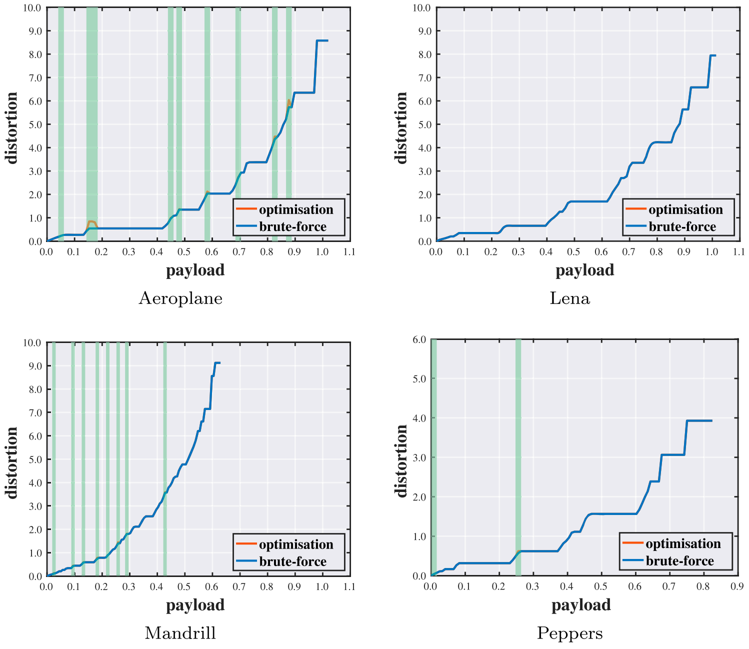
Aeroplane

Lena

Mandrill

Peppers

Figure 8. Payload–distortion curves for optimality analysis against brute-force search with highlighted discrepancies ($\vartheta = 3$).



Aeroplane

Lena

Mandrill

Peppers

Figure 9. Payload–distortion curves for optimality analysis against brute-force search with highlighted discrepancies ($\vartheta = 4$).

and image restoration (Y. Zhang et al., 2021). This neural network model is characterised by a tangled labyrinth of residual and dense connections. It is trained on the BOSS-base dataset (Bas et al., 2011), which originated from an academic competition for digital steganography, and comprises a large collection of greyscale photographs covering a wide variety of subjects and scenes. The algorithms are tested on selected images from the USC-SIPI dataset (Weber, 2006). All the images are resized to a resolution of 256×256 pixels via Lanczos resampling (Duchon, 1979). The border pixels along with half of the rest of the pixels are designated as the context. Accordingly, the number of query pixels equals $(254 \times 254)/2$. We display both distortion and capacity as divided by the number of query pixels.

Figure 5 shows the absolute error distribution for each test image. It is observed that most of the error values are below around 30 to 50, depending on the image. We set $n = 55$ conservatively in the sense that nearly every value of non-zero occurrence is included. We implement the algorithms with respect to different quota settings ($\vartheta = 1, 2, 3, 4$). Figures 6–9 show performance evaluations of the proposed optimisation algorithm. Each point of the curve indicates the minimum distortion of a solution under a specific capacity constraint. In the vast majority of cases, the solutions found by the proposed method are identical to those given by the brute-force method. When failing to find the optimal solutions, the objective values reached are within a small distance from the optimal ones. Hence, even though optimal solutions cannot always be guaranteed, the results suggest that the proposed method can attain near-optimal performance.

6. Conclusion

This paper studies a mathematical optimisation problem applied to reversible steganography. We formulate automatic coding in prediction error modulation as a nonlinear discrete optimisation problem. The objective is to minimise distortion under a constraint on capacity. We discuss the complexity of a brute-force search algorithm and the linearisation techniques for the logarithmic capacity constraint and the quadratic distortion objective. The problem is transformed into an iterative mixed-integer linear programming problem with binary-integer variables and slack variables. Our simulation results validate the near-optimality of the proposed algorithm.

Disclosure statement

No potential conflict of interest is reported by the author.

References

Alattar, A. M. (2004). Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Transactions on Image Processing*, *13*(8), 1147–1156.

Anderson, R., & Petitcolas, F. (1998). On the limits of steganography. *IEEE Journal on Selected Areas in Communications*, *16*(4), 474–481.

Barni, M., Bartolini, F., Cappellini, V., & Piva, A. (1998). A DCT-domain system for robust image watermarking. *Signal Processing*, *66*(3), 357–372.

Bas, P., Filler, T., & Pevný, T. (2011). Break our steganographic system: The ins and outs of organizing BOSS. In *Proceedings of the international workshop on information hiding (IH)* (pp. 59–70).

Boden, M., Bryson, J., Caldwell, D., Dautenhahn, K., Edwards, L., Kember, S., Newman, P., Parry, V., Pegman, G., Rodden, T., Sorrell, T., Wallis, M., Whitby, B., & Winfield, A. (2017). Principles of robotics: Regulating robots in the real world. *Connection Science*, *29*(2), 124–129.

Castelvecchi, D. (2016). Can we open the black box of AI?. *Nature*, *538*(7623), 20–23.

Celik, M. U., Sharma, G., Tekalp, A. M., & Saber, E. (2005). Lossless generalized-LSB data embedding. *IEEE Transactions on Image Processing*, *14*(2), 253–266.

Chang, C. C. (2020). Adversarial learning for invertible steganography. *IEEE Access*, *8*, 198425–198435.

Chang, C. C. (2021). Neural reversible steganography with long short-term memory. *Security and Communication Networks*, *2021*, Article 5580272.

Chang, C. C. (2022). Bayesian neural networks for reversible steganography. *IEEE Access*, *10*, 36327–36334.

Chang, C. C., Li, C. T., & Shi, Y. Q. (2018). Privacy-aware reversible watermarking in cloud computing environments. *IEEE Access*, *6*, 70720–70733.

Coatrieux, G., Pan, W., Cuppens-Boulahia, N., Cuppens, F., & Roux, C. (2013). Reversible watermarking based on invariant image classification and dynamic histogram shifting. *IEEE Transactions on Information Forensics and Security*, *8*(1), 111–120.

Cox, I. J., Kilian, J., Leighton, F. T., & Shamoon, T. (1997). Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, *6*(12), 1673–1687.

Depovere, G., Kalker, T., Haitsma, J., Maes, M., de Strycker, L., Termont, P., Vandewege, J., Langell, A., Alm, C., Norman, P., O'Reilly, G., Howes, B., Vaanholt, H., Hintzen, R., Donnelly, P., & Hudson, A. (1999). The VIVA project: Digital watermarking for broadcast monitoring. In *Proceedings of the international conference on image processing (ICIP)* (pp. 202–205).

De Vleeschouwer, C., Delaigle, J. F., & Macq, B. (2003). Circular interpretation of bijective transformations in lossless watermarking for media asset management. *IEEE Transactions on Multimedia*, *5*(1), 97–105.

Dragoi, I., & Coltuc, D. (2014). Local prediction based difference expansion reversible watermarking. *IEEE Transactions on Image Processing*, *23*(4), 1779–1790.

Duan, X., Jia, K., Li, B., Guo, D., Zhang, E., & Qin, C. (2019). Reversible image steganography scheme based on a U-Net structure. *IEEE Access*, *7*, 9314–9323.

Duchon, C. E. (1979). Lanczos filtering in one and two dimensions. *Journal of Applied Meteorology and Climatology*, *18*(8), 1016–1022.

Fallahpour, M. (2008). Reversible image data hiding based on gradient adjusted prediction. *IEICE Electronics Express*, *5*(20), 870–876.

Fridrich, J., Goljan, M., & Du, R. (2001). Invertible authentication. In *Proceedings of the SPIE conference on security and watermarking of multimedia contents (SWMC)* (pp. 197–208).

Fridrich, J., Goljan, M., Lisonek, P., & Soukal, D. (2005). Writing on wet paper. *IEEE Transactions on Signal Processing*, *53*(10), 3923–3935.

Friedman, G. (1993). The trustworthy digital camera: Restoring credibility to the photographic image. *IEEE Transactions on Consumer Electronics*, *39*(4), 905–910.

Goodfellow, I., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *Proceedings of the international conference on learning representations (ICLR)* (pp. 1–11).

He, S., & Wu, M. (2006). Joint coding and embedding techniques for multimedia fingerprinting. *IEEE Transactions on Information Forensics and Security*, *1*(2), 231–247.

Hu, R., & Xiang, S. (2021). CNN prediction based reversible data hiding. *IEEE Signal Processing Letters*, *28*, 464–468.

Hwang, H. J., Kim, S., & Kim, H. J. (2016). Reversible data hiding using least square predictor via the LASSO. *EURASIP Journal on Image and Video Processing*, *2016*(1), Article 42.

LeCun, Y., Bengio, Y., & Hinton, G. E. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lee, S., Yoo, C. D., & Kalker, T. (2007). Reversible image watermarking based on integer-to-Integer wavelet transform. *IEEE Transactions on Information Forensics and Security*, *2*(3), 321–330.

Li, X., Yang, B., & Zeng, T. (2011). Efficient reversible watermarking based on adaptive prediction-Error expansion and pixel selection. *IEEE Transactions on Image Processing*, *20*(12), 3524–3533.

Liu, T., Liu, Z., Liu, Q., Wen, W., Xu, W., & Li, M. (2020). StegoNet: Turn deep neural network into a stegomalware. In *Proceedings of the annual computer security applications conference (ACSAC)* (pp. 928–938).

Lu, S. P., Wang, R., Zhong, T., & Rosin, P. L. (2021). Large-capacity image steganography based on invertible neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 10816–10825).

Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E. C., & Roli, F. (2017). Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the ACM workshop on artificial intelligence and security (AISEC)* (pp. 27–38).

Rissanen, J. (1984). Universal coding, information, prediction, and estimation. *IEEE Transactions on Information Theory*, *30*(4), 629–636.

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-Key cryptosystems. *Communications of the ACM*, *21*(2), 120–126.

Sachnev, V., Kim, H. J., Nam, J., Suresh, S., & Shi, Y. Q. (2009). Reversible watermarking algorithm using sorting and prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, *19*(7), 989–999.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*(3), 379–423.

Thodi, D. M., & Rodriguez, J. J. (2007). Expansion embedding techniques for reversible watermarking. *IEEE Transactions on Image Processing*, *16*(3), 721–730.

Weber, A. G. (2006). *The USC-SIPI image database: Version 5* (Tech. Rep. No. 315). Los Angeles, CA, USA: Signal and Image Processing Institute, Viterbi School of Engineering, University of Southern California.

Weinberger, M., & Seroussi, G. (1997). Sequential prediction and ranking in universal context modeling and data compression. *IEEE Transactions on Information Theory/Professional Technical Group on Information Theory*, *43*(5), 1697–1706.

Wilson, E. B. (1923). First and second laws of error. *Journal of the American Statistical Association*, *18*(143), 841–851.

Wu, H., & Zhang, X. (2020). Reducing invertible embedding distortion using graph matching model. In *Proceedings of the IS&T international symposium on electronic imaging (EI)* (pp. 1–10).

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2018). Residual dense network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (CVPR)* (pp. 2472–2481).

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2021). Residual dense network for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(7), 2480–2495.

Zhang, Z., Fu, G., Di, F., Li, C., & Liu, J. (2019). Generative reversible data hiding by image-to-Image translation via GANs. *Security and Communication Networks*, *2019*, Article 4932782.