МАТЕМАТИЧЕСКИЕ МЕТОДЫ СТЕГАНОГРАФИИ

УДК 004.056.5

ЭФФЕКТИВНОЕ ОБНАРУЖЕНИЕ СТЕГАНОГРАФИЧЕСКИ СКРЫТОЙ ИНФОРМАЦИИ ПОСРЕДСТВОМ ИНТЕГРАЛЬНОГО КЛАССИФИКАТОРА НА ОСНОВЕ СЖАТИЯ ДАННЫХ

В. А. Монарев*, А. И. Пестунов**

*Институт вычислительных технологий СО РАН, г. Новосибирск, Россия

**Новосибирский государственный университет экономики и управления, г. Новосибирск, Россия

Предлагается концепция интегрального классификатора, предназначенного для повышения точности методов стегоанализа, которые базируются на машинном обучении. Вместо одиночного классификатора, принимающего решение о пустоте или заполненности контейнера, предлагается обучать набор классификаторов, каждый из которых предназначен для обработки контейнеров с определёнными свойствами. В качестве реализации данной концепции представлен интегральный классификатор, основанный на сжатии данных, что подразумевает выбор отдельного классификатора из набора на основе коэффициентов сжатия контейнеров. Эффективность предлагаемого классификатора для решения задачи обнаружения скрытой информации экспериментально продемонстрирована для современных методов адаптивного внедрения HUGO, WOW и S-UNIWARD на изображениях-контейнерах из известной базы BOSSbase 1.01. Показано, что в зависимости от метода внедрения и количества скрываемой информации ошибку обнаружения можно снизить на 0,05–0,16 по сравнению с лучшими из известных результатов.

Ключевые слова: *стегоанализ, ошибка обнаружения, HUGO, WOW, UNIWARD, метод опорных векторов, PSRM-признаки, SRM-признаки, ансамблевый классификатор, интегральный классификатор.*

DOI 10.17223/20710410/40/5

EFFICIENT STEGANOGRAPHY DETECTION BY MEANS OF COMPRESSION-BASED INTEGRAL CLASSIFIER

V. A. Monarev*, A. I. Pestunov**

*Institute of Computational Technologies of SB RAS, Novosibirsk, Russia

**Novosibirsk State University of Economics and Management, Novosibirsk, Russia

E-mail: viktor.monarev@gmail.com, pestunov@gmail.com

We propose a model of an integral classifier in order to solve the problem of binary steganalysis by means of machine-learning tools more efficiently. The problem of binary steganalysis consists in recognizing whether a given container is empty or contains a certain payload embedded via a certain steganographic algorithm. In steganalysis, such problem is often solved using such machine-learning techniques as the support

vector machine and the ensemble classifier. Instead of using a single classifier (as it is done now) which is intended to make an ultimate decision about whether the container is empty or not, the proposed in this paper integral classifier consists of several classifiers and works in such a way that each of them processes only those containers which satisfy a certain condition. Within the proposed model, we develop a compression-based integral classifier which works as follows. The training set of classifiers is splitted into several subsets according to the containers compression rate; then a corresponding number of classifiers are trained, but each classifier is injected only with an ascribed subset. The testing containers are distributed between the classifiers (also according to their compression rate) and the decision about the certain container is made by the chosen classifier. In order to demonstrate the power of the integral classifier, we performed some experiments using the famous de-facto standard images database BOSSbase 1.01 as a source of the containers along with contemporary content-adaptive embedding algorithms HUGO, WOW and S-UNIWARD. Comparison with state-of-the-art results (obtained for the single support vector machine and the ensemble classifier) demonstrated that, depending on the case, the integral classifier allows to decrease the detection error by 0.05–0.16.

Keywords: *steganalysis, detection error, HUGO, WOW, UNIWARD, support vector machine, ensemble classifier, integral classifier, projected spatial rich model, spatial rich model, compression.*

Введение

Классическая задача бинарного стегоанализа [1 – 3] состоит в том, чтобы определить, присутствует в заданном контейнере информация или нет. При этом, в отличие от так называемого «слепого стегоанализа» [4 – 6], метод внедрения и размер внедрения предполагаются известными. Ошибку обнаружения стегоанализа принято вычислять эмпирически для некоторого контрольного множества как отношение числа ошибок первого и второго рода или, другими словами, ложных срабатываний и пропущенных обнаружений [1 – 3, 7, 8]. В целом, такой подход позволяет сравнивать точность различных методов стегоанализа, однако он тем не менее использует усреднение и не позволяет учесть особенности изображений, влияющие на достоверность обнаружения. Другими словами, в контрольном множестве часто оказывается возможным выбрать подмножества, которые характеризуются тем, что вычисленная только по ним ошибка обнаружения может быть как больше, так и меньше ошибки, вычисленной по всему множеству.

В работе [9] авторами предложен подход, названный «предварительной фильтрацией» и позволяющий учесть эту особенность. Идея подхода заключается в том, что из всего контрольного множества некоторым образом выбирается подмножество, ошибка обнаружения по которому меньше, чем по всему множеству. В рамках данного подхода предложено три метода разной эффективности для решения этой задачи. Однако несмотря на то, что выбранное подмножество может быть достаточно велико (в [9] его размер варьировался приблизительно от 5 до 40 % от всего контрольного множества), все-таки значительная часть контейнеров не используется при подсчёте ошибки обнаружения.

В настоящей работе предлагается подход, позволяющий расширить идею предварительной фильтрации и добиться снижения ошибки обнаружения, которая будет вычислена уже по всему контрольному множеству, а не только по его части. Для этой цели вводится понятие интегрального классификатора, состоящего из набора отдельных

классификаторов, каждый из которых обрабатывает только те контейнеры, которые отфильтрованы для него. Теоретически интегральный классификатор может быть реализован разными способами. В работе предложен интегральный классификатор на основе сжатия данных. Обучающее множество разбивается на несколько частей в соответствии с коэффициентом их сжатия и после этого обучается такое же количество классификаторов, причём каждый из них обучается на своём подмножестве. Во время фазы тестирования контрольного множества очередной контейнер отправляется на классификатор, обученный на контейнерах, коэффициент сжатия которых наиболее близок к коэффициенту сжатия данного контейнера.

Идея использования коэффициента сжатия как критерия выбора классификатора возникла на основе известного факта о том, что проводить стегоанализ «шумных» изображений (noisy images) сложнее, чем изображений с большими областями приблизительно одного цвета (plain images). Как известно, изображения первого типа сжимаются хуже, чем второго. Соответственно наша догадка состоит в том, что классификаторы для плохо сжимаемых контейнеров должны обучаться на плохо сжимаемых контейнерах, и то же самое — для хорошо сжимаемых. Заметим, что сжатие данных уже использовалось в стегоанализе [10, 11], но эти работы посвящены либо обнаружению LSB-стеганографии, либо созданию количественных методов обнаружения, когда размер внедрения является неизвестной величиной, подлежащей определению. Второе отличие заключается в том, что в настоящей работе сжатие используется на предварительном этапе выбора классификатора, но не для построения самого метода стегоанализа непосредственно.

Эффективность предлагаемого подхода к обнаружению скрытой информации экспериментально продемонстрирована на изображениях из стандартизованной базы BOSSbase 1.01 [12], часто используемой специалистами по стеганографии и стегоанализу. Подлежащая обнаружению информация внедрялась при помощи современных методов адаптивной стеганографии HUGO [13], S-UNIWARD [14] и WOW [15]. В качестве эталонных значений ошибки обнаружения взяты лучшие на данный момент результаты В. Холуба и Дж. Фридрич [2]. Эксперименты подтвердили гипотезу о повышении эффективности стегоанализа при использовании нескольких классификаторов и добавлении фазы предварительного распределения контейнеров по ним. В зависимости от метода внедрения и его размера ошибку обнаружения удалось снизить на 0,05–0,16.

1. Описание предлагаемого интегрального классификатора

1.1. Общая схема интегрального классификатора

Процессы обучения интегрального классификатора и обнаружения скрытой информации с его помощью схематично изображены на рис. 1 и 2. Использование интегрального классификатора подразумевает выбор параметра L, определяющего количество составляющих его одиночных классификаторов; в вырожденном случае при $L = 1$ интегральный классификатор фактически становится одиночным (рис. 3).

На стадии обучения интегрального классификатора все обучающее множество разбивается на L непересекающихся подмножеств (возможно, различного размера) в соответствии с коэффициентами сжатия входящих в них контейнеров. Для определённости будем считать, что подмножество 1 содержит контейнеры с наибольшим коэффициентом сжатия, а подмножество L — с наименьшим.

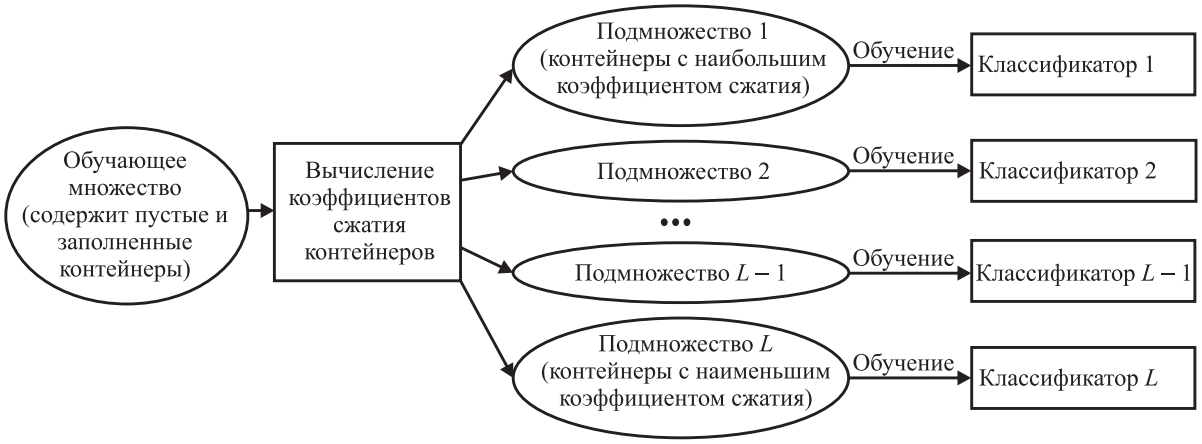

Рис. 1. Обучение интегрального классификатора, основанного на сжатии данных

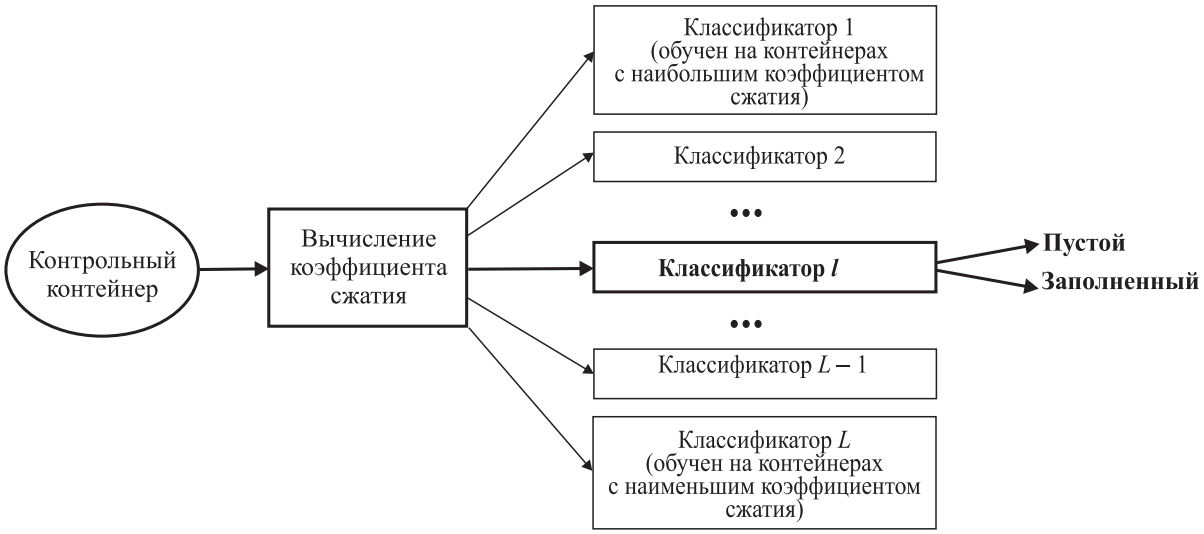

Рис. 2. Определение наличия/отсутствия скрытой в контейнере информации при помощи интегрального классификатора, основанного на сжатии данных (у контрольного контейнера вычисляется коэффициент сжатия, согласно которому выбирается классификатор, дающий ответ; этот классификатор обучен на контейнерах, имеющих коэффициенты сжатия, близкие к коэффициенту сжатия данного контрольного контейнера)

Рис. 3. Обнаружение скрытой информации при помощи одиночного классификатора

После разбиения необходимо взять L классификаторов и обучить каждый из них на контейнерах из соответствующего ему подмножества. На стадии тестирования контрольное множество также разбивается на L подмножеств в соответствии с коэффициентами сжатия, после чего контейнеры из этих подмножеств обрабатываются классификатором, обученным на контейнерах с близким коэффициентом сжатия. Если

контрольные контейнеры поступают по одному, то их можно обрабатывать «на лету», выбирая классификатор согласно заданным порогам для коэффициента сжатия.

В принципе, в описанной схеме разбиение обучающего множества и выбор классификатора может производиться не только на основе коэффициента сжатия, но и на основе любой характеристики, которую можно вычислить как функцию от контейнера в качестве аргумента.

1.2. Алгоритм обнаружения скрытой информации при помощи интегрального классификатора

Алгоритм 1 демонстрирует пошаговую работу интегрального классификатора. В качестве входных параметров алгоритм принимает обучающее множество $\mathcal{X}$ и контрольное множество $\mathcal{Y}$. На первых шагах выбираются: метод сжатия $\texttt{Compress}(\cdot)$; L — количество классификаторов, составляющих интегральный классификатор; размеры подмножеств, на которые разбивается обучающее множество $size_1, \ldots, size_L$ ($size_1 + \ldots + size_L = |\mathcal{X}|$); размеры подмножеств, на которые разбивается контрольное множество $size'_1, \ldots, size'_L$ ($size'_1 + \ldots + size'_L = |\mathcal{Y}|$).

Алгоритм 1. Алгоритм работы интегрального классификатора

1: **Функция** Интегральный-классификатор($\mathcal{X}$, $\mathcal{Y}$)
   $\mathcal{X}$ — обучающее множество, $\mathcal{Y}$ — контрольное множество
2: Выбрать функцию сжатия $\texttt{Compress}(\cdot)$.
3: Выбрать количество классификаторов L.
4: Выбрать размеры $size_l$, $l = 1, \ldots, L$, подмножеств, на которые разбивается обучающее множество ($size_1 + \ldots + size_L = |\mathcal{X}|$).
5: Выбрать размеры $size'_l$, $l = 1, \ldots, L$, подмножеств, на которые разбивается контрольное множество ($size'_1 + \ldots + size'_L = |\mathcal{Y}|$).
6: $(Subset_1, \ldots, Subset_L)$:=Разбить-множество($\mathcal{X}$,$\texttt{Compress}(\cdot)$,$L, size_1, \ldots, size_L$).
7: $(Detector_1, \ldots, Detector_L)$:=Обучить-классификаторы($L, Subset_1, \ldots, Subset_L$).
8: $(Subset'_1, \ldots, Subset'_L)$:=Разбить-множество($\mathcal{Y}$, $\texttt{Compress}(\cdot)$,$L, size'_1, \ldots, size'_L$).
9: $\mathcal{Y}_{\text{Пустые}} = \varnothing$; $\mathcal{Y}_{\text{Заполненные}} = \varnothing$.
10: **Для всех** $y \in \mathcal{Y}$
11: $detectorNumber$:=Номер-классификатора($y, L, Subset'_1, \ldots, Subset'_L$);
12: $detectionResult := Detector_{detectorNumber}(y)$.
13: **Если** $detectionResult = \text{Empty}$, **то**
14: $\mathcal{Y}_{\text{Пустые}} := \mathcal{Y}_{\text{Пустые}} \cup \{y\}$;
15: **иначе**
16: $\mathcal{Y}_{\text{Заполненные}} := \mathcal{Y}_{\text{Заполненные}} \cup \{y\}$.
17: **Вернуть** $\mathcal{Y}_{\text{Пустые}}$ — пустые контейнеры (согласно классификатору);
$\mathcal{Y}_{\text{Заполненные}}$ — заполненные контейнеры.

Функция Разбить-множество (алгоритм 2) возвращает L непересекающихся подмножеств, образующих разбиение множества Z. В интегральном классификаторе (алгоритм 1) эта функция вызывается дважды: в шаге 6 — с целью разбиения обучающего множества $\mathcal{X}$ и в шаге 8 — с целью разбиения контрольного множества $\mathcal{Y}$. На первом шаге данной функции каждый контейнер $z \in \mathcal{Z}$ сжимается, чтобы определить размер его архива $|\texttt{Compress}(z)|$. Далее все контейнеры сортируются согласно размерам архивов в порядке возрастания, отсортированный список обозначим $(z_{(1)}, z_{(2)}, \ldots, z_{(|\mathcal{Z}|)})$. Наконец, на последнем шаге формируются подмножества; первые

$size_1$ контейнеров помещаются в первое подмножество, следующие $size_2$ — во второе и т. д.; последние $size_L$ контейнеров образуют последнее подмножество.

Алгоритм 2. Разбиение множества на подмножества согласно коэффициенту сжатия

1: **Функция** Разбить-множество($\mathcal{Z}$, `Compress(·)`, L, $size_1$, …, $size_L$)
   $\mathcal{Z}$ — обучающее или контрольное множество, подлежащее разбиению;
   `Compress(·)` — функция сжатия;
   L — количество подмножеств (определяется количеством классификаторов);
   $size_l$, $l = 1, \ldots, L$ — размеры подмножеств ($size_1 + \ldots + size_L = |\mathcal{Z}|$).
2: **Для всех** $z \in \mathcal{Z}$
   вычислить $|\texttt{Compress}(z)|$.
3: Отсортировать контейнеры по неубыванию значения $|\texttt{Compress}(z)|$.
4: Сформировать L подмножеств множества $\mathcal{Z}$ следующим образом:
   $Subset_1 = \{z_{(1)}, \ldots, z_{(size_1)}\}$; $Subset_2 := \{z_{(size_1+1)}, \ldots, z_{(size_1+size_2)}\}$;
   $Subset_3 := \{z_{(size_1+size_2+1)}, \ldots, z_{(size_1+size_2+size_3)}\}$; …;
   $Subset_L := \{z_{(size_1+\ldots+size_{L-1}+1)}, \ldots, z_{(|\mathcal{Z}|)}\}$;
5: **Вернуть** $Subset_1$, $Subset_2$, …, $Subset_L$.

Функция Обучить-классификаторы (алгоритм 3) принимает L подмножеств обучающего множества и обучает на них L одиночных классификаторов различать пустые и заполненные контейнеры; каждое подмножество используется для обучения одного классификатора. Классификаторы могут быть различными, но в стегоанализе, как правило, используются классификаторы на основе метода опорных векторов (support vector machine) или ансамблевые (ensemble classifier). Эти классификаторы работают со специфичными для стегоанализа признаками, в частности, для изображений-контейнеров применяются SRM [1], PSRM [2], SPAM [16] и др.

Алгоритм 3. Обучение классификаторов, образующих интегральный классификатор

1: **Функция** Обучить-классификаторы($L, Subset_1, \ldots, Subset_L$)
   L — количество классификаторов;
   $\{Subset_l : l = 1, \ldots, L\}$ — разбиение обучающего множества.
2: Взять L необученных классификаторов $Detector_1$, $Detector_2$, …, $Detector_L$.
3: **Для всех** $l = 1, \ldots, L$
   обучить $Detector_l$ на множестве $Subset_l$.
4: **Вернуть** $Detector_1$, $Detector_2$, …, $Detector_L$.

Фаза обработки контрольного множества состоит из двух этапов: 1) выбор классификатора и 2) его применение. Выбор классификатора осуществляется функцией Номер-классификатора (алгоритм 4), которая по заданному контейнеру из контрольного множества $\mathcal{Y}$ возвращает номер классификатора, решение которого будет принято за решение интегрального классификатора. Номер определяется согласно разбиению контрольного множества.

Алгоритм 4. Выбор классификатора для обработки контрольного контейнера

Функция НОМЕР-КЛАССИФИКАТОРА(y, L, $Subset'_1$, $Subset'_2$, …, $Subset'_L$)
y — контейнер из контрольного множества;
L — количество классификаторов;
$\{Subset'_l : l = 1, \ldots, L\}$ — разбиение контрольного множества.
Для $l = 1, \ldots, L$
  Если $y \in Subset'_l$, **то**
    $detectorNumber := l$.
Вернуть $detectorNumber$.

2. Экспериментальные результаты

2.1. Инструментарий для проведения экспериментов

Изображения-контейнеры

Для выполнения экспериментов в качестве контейнеров использовались изображения из известного стандартизованного множества BOSSbase 1.01, опубликованного в рамках конкурса Break Our Steganographic System (BOSS) [12]. Данное множество изображений за последние годы стало де-факто стандартом при экспериментальном расчёте ошибок обнаружения методов стегоанализа [7, 8]. Множество BOSSbase 1.01 состоит из 10 000 изображений, полученных с семи различных фотокамер в формате RAW, которые затем были преобразованы в 8-битовый черно-белый режим и уменьшены до размера 512×512.

Методы внедрения информации

В экспериментах стеганографическое внедрение информации производилось тремя адаптивными методами: HUGO, WOW и S-UNIWARD, поскольку именно эти методы считаются наиболее трудно обнаружимыми на данный момент и именно по ним приведены лучшие из известных результатов обнаружения [2]. Идея адаптивного внедрения заключается в том, что позиции для внедрения выбираются не произвольно, а исходя из свойств изображения; при этом с большей вероятностью внедрение осуществляется в те области, где обнаружить информацию должно быть труднее. HUGO (Highly Undetectable Steganography) — это алгоритм внедрения, основанный на синдромном решетчатом декодировании (syndrome-trellis codes) [13]. WOW (Wavelet Obtained Weights) использует дискретное вейвлет-преобразование [15], а алгоритм S-UNIWARD [14] является упрощённым вариантом WOW.

Создание обучающего и контрольного множеств

Множество BOSSbase 1.01 состоит из 10 000 изображений. Создание обучающего и контрольного множеств на его основе осуществлялось следующим образом. Обозначим через $\mathcal{X}^p$ обучающее множество и через $\mathcal{Y}^p$ — контрольное, где индекс p указывает на размер внедрения в битах на пиксель (б/п).

Всё исходное множество разбивалось на две части $\mathcal{X}_0$ и $\mathcal{Y}_0$, где $|\mathcal{X}_0| = 7\,500$ и $|\mathcal{Y}_0| = 2\,500$. Затем во все изображения из $\mathcal{X}_0$ и $\mathcal{Y}_0$ внедрялось p б/п случайной информации с помощью одного из трёх методов (HUGO, S-UNIWARD и WOW). Полученные изображения с внедрённой информацией обозначим $\mathcal{X}_1^p$ и $\mathcal{Y}_1^p$ соответственно.

Далее обучающее и контрольное множества формировались следующим образом: $\mathcal{X}^p := \mathcal{X}_0 \cup \mathcal{X}_1^p$ и $\mathcal{Y}^p := \mathcal{Y}_0 \cup \mathcal{Y}_1^p$. Таким образом, $|\mathcal{X}^p| = 15\,000$ и $|\mathcal{Y}^p| = 5\,000$. Оба

множества содержат поровну пустых и заполненных контейнеров. Далее индекс p будем опускать (это не должно привести к разночтениям), а обучающее и контрольное множества будем обозначать через $\mathcal{X}$ и $\mathcal{Y}$ соответственно.

Используемые методы сжатия

В качестве метода сжатия использован известный свободно распространяемый архиватор PAQ, основанный на контекстном моделировании (context mixing model) и предсказании частичным совпадением (partial match). В экспериментах использована готовая реализация М. Махоуни [17]. Скрипт для запуска архиватора: paq -11.

Реализация классификатора

В качестве классификатора использована реализация метода опорных векторов (support vector machine) на языке программирования Python [18]. В данной реализации параметры, заданные по умолчанию, не изменялись, за исключением следующих: ядро — линейное, сокращение (srinking) — включено и штрафной параметр $C = 20\,000$. Параметры изменены с целью достижения приемлемой скорости работы программы.

Извлекаемые из изображений признаки для классификации

Для классификации из изображений извлекались так называемые SRM-признаки (Spatial Rich Model) [1], являющиеся одним из наилучших инструментов, применяемых в стегоанализе. Их более новый усовершенствованный вариант — PSRM-признаки (Projection Spatial Rich Model) [2] — лишь незначительно снижает ошибку обнаружения, но существенно увеличивает сложность реализации и замедляет работу программы вплоть до неприемлемых показателей. Пространство SRM-признаков имеет размерность 34,671.

Формула для вычисления ошибки обнаружения

Ошибка обнаружения вычислялась стандартным для современных работ способом [1, 3, 16, 19]: $P_E = (P_{FA} + P_{MD})/2$, где P_{FA} — вероятность ложных срабатываний (пустой контейнер принимается за заполненный); P_{MD} — вероятность пропуска обнаружения (заполненный контейнер принимается за пустой).

2.2. Результаты экспериментов

Приведём результаты экспериментов, которые демонстрируют, что использование интегрального классификатора позволяет повысить эффективность стегоанализа, другими словами, снизить ошибку обнаружения по сравнению с одиночным классификатором. Результаты экспериментов сравниваются со state-of-the-art значениями из [2], где реализован ансамблевый классификатор [3], который немного уступает в точности, но работает быстрее. Для убедительности результатов помимо этих данных вычислены ошибки для нашей реализации одиночного классификатора на основе метода опорных векторов и SRM-признаков; эти ошибки оказались близкими к ошибкам ансамблевого классификатора. Однако точность предлагаемого интегрального классификатора оказалась выше точности одиночных во всех случаях.

В таблице приведено сравнение ошибок обнаружения методов внедрения HUGO, WOW и S-UNIWARD для одиночных классификаторов и интегрального классификатора на основе метода опорных векторов и SRM-признаков. Параметры интегрального классификатора следующие: метод сжатия — PAQ; $L = 5$; $size_1 = size_2 = size_3 = size_4 = size_5 = 3\,000$; $size'_1 = size'_2 = size'_3 = size'_4 = size'_5 = 1\,000$. Выбор параметров обоснован некоторыми предварительными экспериментами, хотя, в целом, увеличение или уменьшение числа подмножеств слабо влияет на ошибку обнаружения.

Сравнение ошибок обнаружения различных методов на множестве изображений BOSSbase 1.01 при различных размерах внедрения (серым цветом выделены наименьшие ошибки для каждого размера внедрения); приведены ошибки, вычисленные отдельно по подмножествам

Вид классификатора	Контрольное множество	Размер внедрения		
		0,1 б/п	0,2 б/п	0,4 б/п
Внедрение методом HUGO [13]				
Интегральный классификатор	$Subset'_1$	0,01	0,01	0,00
	$Subset'_2$	0,17	0,05	0,01
	$Subset'_3$	0,21	0,11	0,03
	$Subset'_4$	0,34	0,18	0,08
	$Subset'_5$	0,46	0,30	0,19
	Всё контр. множ.	**0,24**	**0,13**	**0,06**
Одиночный классификатор (всё контрольное множество)	Наша реализация: SVM+SRM	**0,35**	0,27	0,15
	Holub, Fridrich [2]: Ансамблевый+SRM	0,36	0,25	0,12
	Holub, Fridrich [2]: Ансамблевый+PSRMQ1	**0,35**	**0,23**	**0,11**
Внедрение методом WOW [15]				
Интегральный классификатор	$Subset'_1$	0,02	0,01	0,00
	$Subset'_2$	0,20	0,08	0,01
	$Subset'_3$	0,25	0,13	0,06
	$Subset'_4$	0,30	0,18	0,11
	$Subset'_5$	0,44	0,29	0,20
	Всё контр. множ.	**0,24**	**0,13**	**0,08**
Одиночный классификатор (всё контрольное множество)	Наша реализация: SVM+SRM	**0,38**	**0,29**	0,21
	Holub, Fridrich [2]: Ансамблевый+SRM	0,39	0,31	0,19
	Holub, Fridrich [2]: Ансамблевый+PSRMQ1	**0,38**	**0,29**	**0,17**
Внедрение методом S-UNIWARD [14]				
Интегральный классификатор	$Subset'_1$	0,01	0,00	0,00
	$Subset'_2$	0,21	0,04	0,00
	$Subset'_3$	0,29	0,14	0,02
	$Subset'_4$	0,33	0,20	0,11
	$Subset'_5$	0,41	0,37	0,16
	Всё контр. множ.	**0,25**	**0,15**	**0,06**
Одиночный классификатор (всё контрольное множество)	Наша реализация: SVM+SRM	**0,37**	**0,30**	**0,17**
	Holub, Fridrich [2]: Ансамблевый+SRM	0,41	0,31	0,20
	Holub, Fridrich [2]: Ансамблевый+PSRMQ1	0,39	**0,30**	0,18

В таблице серым цветом выделены сравниваемые значения. Для каждого размера внедрения и каждого метода внедрения сравниваются значения ошибок интегрального классификатора с наименьшими ошибками среди всех реализаций одиночного классификатора, среди которых две реализации из [2] (ансаблевый классификатор с SRM- или PSRM-признаками) и одна наша реализация (метод опорных векторов с SRM-признаками). Например, для HUGO 0,1 б/п сравнивается 0,24 с 0,35, а для WOW 0,4 б/п — 0,08 с 0,17. Если две реализации дают одинаковую наименьшую ошибку, то выделены два значения (например, для HUGO 0,1 б/п две реализации дают ошибку 0,35).

Результаты показали, что интегральный классификатор снижает ошибку обнаружения. Наиболее впечатляющие результаты достигнуты при внедрениях HUGO 0,1 б/п, WOW 0,1 б/п, WOW 0,2 б/п, S-UNIWARD 0,1 б/п, S-UNIWARD 0,2 б/п, S-UNIWARD 0,4 б/п, где ошибка снизилась более чем на 0,1.

Заключение

В работе предложена концепция интегрального классификатора, предназначенного для повышения точности методов стегоанализа, которые основаны на машинном обучении. Данная концепция расширяет опубликованный ранее подход под названием «предварительная фильтрация» [9], позволяющий выбирать из контрольного множества (отфильтровывать) только те контейнеры, ошибка обнаружения для которых меньше, чем для всего контрольного множества. Хотя количество таких отфильтрованных контейнеров может быть достаточно велико, предварительная фильтрация в чистом виде не позволяет снизить ошибку обнаружения для всего контрольного множества. В то же время интегральный классификатор, предложенный в настоящей работе, предназначен для того, чтобы уменьшить ошибку обнаружения для всего контрольного множества.

Эффективность обнаружения скрытой информации при помощи интегрального классификатора продемонстрирована экспериментально для методов адаптивной стеганографии HUGO, WOW и S-UNIWARD на известном стандартизованном множестве изображений-контейнеров BOSSbase 1.01. Результаты экспериментов показали, что в зависимости от размера внедрения ошибка обнаружения уменьшилась на 0,05–0,16 по сравнению с лучшими известными авторам результатами.

Идея интегрального классификатора заключается в том, что для распознавания пустых и заполненных контейнеров обучается не один классификатор, а несколько, причём каждый из них предназначен для обработки контейнеров, обладающих определёнными свойствами. Так, в настоящей работе предложен интегральный классификатор, основанный на сжатии данных; контейнеры распределяются по отдельным составляющим его классификаторам согласно их коэффициентам сжатия. Для каждого контейнера интегральный классификатор выбирает наиболее подходящий одиночный классификатор, результат которого принимается за решение о заполненности или пустоте контейнера. В целом, методика распределения контейнеров по классификаторам может быть совершенно разной, и поиск других характеристик, согласно которым будут распределяться контейнеры по классификаторам для снижения ошибки обнаружения, может оказаться продолжением данного исследования.

Во время предварительных экспериментов мы проверили некоторые комбинации параметров, чтобы подобрать приемлемый с точки зрения эффективности вариант. Этого оказалось достаточно, чтобы превзойти точность известных методов стегоанализа. Однако для поиска оптимальных в различных случаях параметров, вероятно, может быть целесообразно провести дополнительное исследование теоретического характера и разработать методику их выбора.

В описанных экспериментах контрольное множество разбивалось на подмножества согласно коэффициентам сжатия и все контейнеры из одного подмножества обрабатывались одним и тем же классификатором. Если подлежащие обработке контейнеры поступают один за другим, то во избежание необходимости их накопления можно зафиксировать определённые пороги для коэффициентов сжатия и выбирать классификаторы согласно этим порогам для каждого отдельно взятого контейнера «на лету».

ЛИТЕРАТУРА

1. *Fridrich J.* Rich models for steganalysis of digital images // IEEE Trans. Inform. Forensics and Security. 2012. V. 7. No. 3. P. 868–882.
2. *Holub V. and Fridrich J.* Random projections of residuals for digital image steganalysis // IEEE Trans. Inform. Forensics and Security. 2013. V. 8. No. 12. P. 1996–2006.
3. *Kodovsky J., Fridrich J., and Holub V.* Ensemble classifiers for steganalysis of digital media // IEEE Trans. Inform. Forensics and Security. 2010. V. 7. No. 2. P. 434–444.
4. *Борисенко Б. Б.* Модификация карты Хотеллинга, нивелирующая влияние тренда, и её применение при обнаружении цифровых водяных знаков // Прикладная дискретная математика. 2010. №. 2. С. 42–58.
5. *Menori M. and Munir R.* Blind steganalysis for digital images using support vector machine method // Proc. IEEE Intern. Symp. Electronics and Smart Devices (ISESD). 2016. Bandung, Indonesia. IEEE. P. 132–136.
6. *Pevný T., Fridrich J., and Ker A.* From blind to quantitative steganalysis // IEEE Trans. Inform. Forensics and Security. 2010. V. 7. No. 2. P. 445–454.
7. *Cogranne R., Denemark T., and Fridrich J.* Theoretical model of the FLD ensemble classifier based on hypothesis testing theory // Proc. 6th IEEE Intern. Workshop on Inform. Forensics and Security (WIFS), 2014. Atlanta, GA, USA. IEEE. P. 167–172.
8. *Schöttle P., Korff S., and Böhme R.* Weighted stego-image steganalysis for naive content-adaptive embedding // Proc. 4th IEEE Intern. Workshop on Inform. Forensics and Security (WIFS). 2012. Tenerife, Spain. IEEE. P. 193–198.
9. *Монарев В. А., Пестунов А. И.* Повышение эффективности методов стегоанализа при помощи предварительной фильтрации контейнеров // Прикладная дискретная математика. 2016. №. 2. С. 87–99.
10. *Boncelet C., Marvel L., and Raqlin A.* Lossless compression-based steganalysis of LSB embedded images // Proc. 41st Ann. Conf. on Inform. Sciences and Systems (CISS). 2007. Baltimore, MD, USA. IEEE. P. 923–929.
11. *Monarev V. and Pestunov A.* A new compression-based method for estimating LSB replacement rate in color and grayscale images // Proc. 7th IEEE Intern. Conf. on Intelligent Inform. Hiding and Multimedia Signal Processing (IIH-MSP). 2011. Dalian, China. IEEE. P. 57–60
12. *Bas P., Filler T., and Pevný T.* Break our steganographic system — the ins and outs of organizing BOSS // LNCS. 2011. V. 6958. P. 59–70.
13. *Pevný T., Filler T., and and Bas P.* Using high-dimensional image models to perform highly undetectable steganography // LNCS. 2010. V. 6387. P. 161–177.
14. *Holub V. and Fridrich J.* Digital image steganography using universal distortion // Proc. 1st ACM Workshop on Inform. Hiding and Multimedia Security (IHMMSec). 2013. Montpellier, France. ACM. P. 59–68.
15. *Holub V. and Fridrich J.* Designing steganographic distortion using directional filters // Proc. 4th IEEE Intern. Workshop on Inform. Forensics and Security (WIFS). 2012. Tenerife, Spain. IEEE. P. 234–239.
16. *Pevný T., Bas P., and Fridrich J.* Steganalysis by subtractive pixel adjacency matrix // IEEE Trans. Inform. Forensics and Security. 2010. V. 5. No. 2. P. 215–224.
17. `mattmahoney.net/dc/text.html` — Large Text Compression Benchmark. 2017.
18. `scikit-learn.org` — scikit-learn: Machine Learning in Python. 2017.
19. *Monarev V. and Pestunov A.* A known-key scenario for steganalysis and a highly accurate detector within it // Proc. 10th IEEE Intern. Conf. on Intelligent Inform. Hiding and Multimedia Signal Processing (IIH-MSP). 2014. Kitakyushu, Japan. IEEE. P. 175–178.

REFERENCES

1. *Fridrich J.* Rich models for steganalysis of digital images. IEEE Trans. Inform. Forensics and Security, 2012, vol. 7, no. 3, pp. 868–882.
2. *Holub V. and Fridrich J.* Random projections of residuals for digital image steganalysis. IEEE Trans. Inform. Forensics and Security, 2013, vol. 8, no. 12, pp. 1996–2006.
3. *Kodovsky J., Fridrich J., and Holub V.* Ensemble classifiers for steganalysis of digital media. IEEE Trans. Inform. Forensics and Security, 2010, vol. 7, no. 2, pp. 434–444.
4. *Borisenko B. B.* Modifikatsiya karty KHotellinga, niveliruyushchaya vliyaniye trenda, i eye primeneniye pri obnaruzhenii tsifrovykh vodyanykh znakov [Modified Hotelling's chart excluding trend influence and its application for digital watermarks detection]. Prikladnaya Diskretnaya Matematika, 2010, no. 2, pp. 42–58. (in Russian)
5. *Menori M. and Munir R.* Blind steganalysis for digital images using support vector machine method. Proc. IEEE Intern. Symp. on Electronics and Smart Devices (ISESD), 2016, Bandung, Indonesia, IEEE, pp. 132–136.
6. *Pevný T., Fridrich J., and Ker A.* From blind to quantitative steganalysis. IEEE Trans. Inform. Forensics and Security, 2010, vol. 7, no. 2, pp. 445–454.
7. *Cogranne R., Denemark T., and Fridrich J.* Theoretical model of the FLD ensemble classifier based on hypothesis testing theory. Proc. 6th IEEE Intern. Workshop on Inform. Forensics and Security (WIFS), 2014, Atlanta, GA, USA, IEEE, pp. 167–172.
8. *Schöttle P., Korff S., and Böhme R.* Weighted stego-image steganalysis for naive content-adaptive embedding. Proc. 4th IEEE Intern. Workshop on Inform. Forensics and Security (WIFS), 2012, Tenerife, Spain, IEEE, pp. 193–198.
9. *Monarev V. A. and Pestunov A. I.* Povyshenie effektivnosti metodov stegoanaliza pri pomoshchi predvaritel'noy fil'tratsii konteynerov [Enhancing steganalysis accuracy via tentative filtering of stego-containers]. Prikladnaya Diskretnaya Matematika, 2016, no. 2, pp. 87–99. (in Russian)
10. *Boncelet C., Marvel L., and Raqlin A.* Lossless compression-based steganalysis of LSB embedded images. Proc. 41st Ann. Conf. on Inform. Sciences and Systems (CISS), 2007, Baltimore, MD, USA, IEEE, pp. 923–929.
11. *Monarev V. and Pestunov A.* A new compression-based method for estimating LSB replacement rate in color and grayscale images. Proc. 7th IEEE Intern. Conf. on Intelligent Inform. Hiding and Multimedia Signal Processing (IIH-MSP), Dalian, Chaina, IEEE, 2011, pp. 57–60
12. *Bas P., Filler T., and Pevný T.* Break our steganographic system — the ins and outs of organizing BOSS. LNCS, 2011, vol. 6958, pp. 59–70.
13. *Pevný T., Filler T., and Bas P.* Using high-dimensional image models to perform highly undetectable steganography. LNCS, 2010, vol. 6387, pp. 161–177.
14. *Holub V. and Fridrich J.* Digital image steganography using universal distortion. Proc. 1st ACM Workshop on Inform. Hiding and Multimedia Security (IHMMSec), 2013, Montpellier, France, ACM, pp. 59–68.
15. *Holub V. and Fridrich J.* Designing steganographic distortion using directional filters. Proc. 4th IEEE Intern. Workshop on Inform. Forensics and Security (WIFS), 2012, Tenerife, Spain, IEEE, pp. 234–239.
16. *Pevný T., Bas P., and Fridrich J.* Steganalysis by subtractive pixel adjacency matrix. IEEE Trans. Inform. Forensics and Security, 2010, vol. 5, no. 2, pp. 215–224.
17. `mattmahoney.net/dc/text.html` — Large Text Compression Benchmark, 2017.
18. `scikit-learn.org` — scikit-learn: Machine Learning in Python, 2017.

19. *Monarev V. and Pestunov A.* A known-key scenario for steganalysis and a highly accurate detector within it. Proc. 10th IEEE Intern. Conf. on Intelligent Inform. Hiding and Multimedia Signal Processing (IIH-MSP), 2014, Kitakyushu, Japan, IEEE, pp. 175–178.