Diagnóstico de CSM en estegoanálisis

Daniel Lerch-Hostalot y David Megías,
Internet Interdisciplinary Institute (IN3),
Universitat Oberta de Catalunya (UOC),
CYBERCAT-Center for Cybersecurity Research of Catalonia
Avgda. Carl Friedrich Gauss, 5, 08860, Castelldefels
E-mail: {dlerch,dmegias}@uoc.edu

***Resumen*—En este artículo se presenta una metodología para detectar el problema de *Cover Source Mismatch* (CSM) en estegoanálisis en imágenes. El método propuesto determina si un clasificador ha sido entrenado con un conjunto de datos incompleto y si no es apropiado para clasificar una imagen concreta. En este caso, la técnica desarrollada detecta que estamos intentando clasificar una muestra no alineada y elige no clasificarla. En el artículo se muestra que esta metodología permite incrementar considerablemente la precisión del clasificador a cambio de no clasificar ciertas muestras. Este método permite aplicar estegoanálisis en escenarios reales donde aparece el problema del CSM. Además, se presenta un procedimiento simple para completar el conjunto de entrenamiento proporcionando nuevas imágenes para completar las regiones del espacio no cubiertas por los datos de entrenamiento iniciales.**

***Index Terms*—Estegoanálisis, *Cover-Source Mismatch*, *Machine Learning*.**

I. Introducción

La esteganografía es una colección de técnicas usadas para ocultar información dentro de objetos aparentemente inocentes. Hoy en día, estos objetos suelen ser medios digitales debido a su amplio uso. Por otra parte, se denomina estegoanálisis a las diferentes técnicas usadas para detectar la existencia de mensajes ocultos mediante esteganografía.

La mayoría de los métodos de estegoanálisis en el estado del arte usan *Machine Learning* [20], [6], [21], [23], [22]. En los sistemas de estegoanálisis basados en *Machine Learning* se utiliza un conjunto de imágenes esteganográficas –también llamadas imágenes *stego*– y de imágenes sin alterar –conocidas como imágenes *cover*– para entrenar un clasificador. Posteriormente, ese clasificador ya entrenado se utiliza para identificar qué imágenes, de un conjunto de test, son *cover* y cuáles son *stego*. Este enfoque funciona muy bien en condiciones de laboratorio, es decir, cuando el conjunto de entrenamiento es similar al conjunto de test. Sin embargo, en el mundo real, el conjunto de imágenes usado por el esteganógrafo para ocultar información puede ser muy diferente al usado por el estegoanalista para entrenar el clasificador [11], incluso aunque el estegoanalista intente disponer de un conjunto de imágenes lo más completo posible. Esto ocurre cuando las imágenes de test no están bien representadas en el conjunto de entrenamiento. Las imágenes de test pueden haber sido obtenidas usando otros modelos de cámaras o pueden haberse tomado en condiciones muy diferentes. En estegoanálisis, este problema se conoce como *cover source mismatch* (CSM), descrito por primera vez en [4].

Se han propuesto diferentes enfoques para abordar el problema del CSM. Durante la competición BOSS [2], algunos participantes intentaron una solución conocida como "entrenamiento con bases de datos contaminadas" que consiste en eliminar el ruido de imágenes del conjunto de test e incluirlas en el conjunto de entrenamiento [7]. Otro enfoque diferente consiste en crear un conjunto de entrenamiento lo más completo posible. En [17], los autores entrenan un clasificador usando grandes cantidades de imágenes. Debido a los altos tiempos de procesamiento y a la gran cantidad de memoria requeridas, los autores optan por usar clasificación *online* y clasificadores ligeros. En [14], se presentan tres estrategias diferentes: (1) entrenamiento con una mezcla de bases de datos de imágenes; (2) uso de diferentes clasificadores para cada una de las bases de datos y test mediante la base de datos más cercana; y (3) utilizar el segundo enfoque pero añadiendo las imágenes de una en una. Otro método similar, conocido como *islet* [19], introduce una paso de preprocesamiento que consiste en organizar las imágenes en grupos, asignando un clasificador a cada grupo. En [24], se presenta un esquema para construir una base de datos completa. Finalmente, en [16] se presenta un método no supervisado que no requiere de ningún conjunto de entrenamiento. En esta técnica, el problema del CSM se evita generando un conjunto de datos de entrenamiento basado en los datos del conjunto de test, transformando el problema en supervisado.

Tal y como se propone en [12], puede ser interesante que los métodos de estegoanálisis produzcan como resultado una muestra *no clasificada* cuando la muestra que se pretende clasificar no está bien representada en el conjunto de entrenamiento. En este artículo, presentamos una metodología a partir de este principio. La idea propuesta está basada en [16], extendiéndola para detectar este tipo de muestras. Al no clasificar las muestras que no están bien representadas en el conjunto de entrenamiento, lo que produciría una cantidad significativa de errores, la precisión final de la clasificación se ve incrementada significativamente. Adicionalmente, se propone una metodología para completar el conjunto de entrenamiento teniendo en cuenta el número de muestras no clasificadas.

Este trabajo se ha financiado parcialmente con los fondos del proyecto TIN2014-57364-C2-2-R "SMARTGLACIS".

El resto del artículo está organizado de la manera siguiente. La Sección II introduce algunos conceptos relevantes que se usan en el método presentado. La Sección III presenta el método propuesto. Los resultados experimentales obtenidos usando cinco bases de datos de imágenes diferentes en los que aparece el problema del CSM se presentan en la Sección IV. Finalmente, en la Sección V se presentan las conclusiones y algunas sugerencias para realizar investigación futura.

II. Preliminares

Consideremos un escenario en el que el algoritmo y la ratio de inserción –pero no la clave– son conocidos (al menos aproximadamente) por el estegoanalista. Dado un conjunto de imágenes *cover* podemos construir un conjunto de entrenamiento usando una mitad de imágenes sin alterar y ocultando un mensaje en la otra mitad. Para ello, se usará el algoritmo y la ratio de inserción conocidos y un valor aleatorio de la clave secreta. A continuación, si usamos una técnica de *Machine Learning* basada en la extracción de características (no todas las técnicas de *Machine Learning* requieren extracción de características [24]), necesitaremos extraerlas. Llamaremos a este conjunto de entrenamiento A^{train}. En el estegoanálisis basado en *Machine Learning* la metodología habitual consiste en entrenar un clasificador usando A^{train}. A continuación, dicho clasificador se utiliza para clasificar A^{test}, es decir, un conjunto de imágenes para el cual, *a priori*, no tenemos información acerca de qué imágenes son *cover* y cuáles son *stego*.

La metodología que proponemos usa algunos conjuntos de datos adicionales. Primero, definimos los conjuntos B^{train} y C^{train} tal y como se describe en [16]: el conjunto B^{train} es el resultado de ocultar información aleatoria en todas las imágenes de A^{train} usando el mismo algoritmo, una ratio de inserción similar y una clave aleatoria; y el conjunto C^{train} es el resultado de ocultar información aleatoria en todas las imágenes de B^{train}, siguiendo el mismo procedimiento. Por lo tanto dispondremos de tres conjuntos: el conjunto A^{train} que contiene imágenes *cover* y *stego*, el conjunto B^{train} que contendrá imágenes *stego* y "doble *stego*", y el conjunto C^{train} que contendrá imágenes"doble *stego*" y "triple *stego*".

Introducimos la siguiente notación: α_i es una muestra del conjunto A^{train}, β_i es la muestra correspondiente del conjunto B^{train}, donde $\beta_i = \mathrm{Embed}(\alpha_i, Bitrate)$, y γ_i es la muestra correspondiente en C^{train}: $\gamma_i = \mathrm{Embed}(\beta_i, Bitrate)$; donde "Embed" significa que incrustamos un mensaje usando información aleatoria, una clave aleatoria, una ratio de inserción ($Bitrate$) similar y el mismo algoritmo.

Siguiendo el mismo procedimiento, usando las imágenes de A^{test} preparamos los conjuntos B^{test} y C^{test}. En este caso a_i, b_i y c_i hacen referencia a las muestras de los conjuntos A^{test}, B^{test} y C^{test}, respectivamente, donde $b_i = \mathrm{Embed}(a_i, Bitrate)$ y $c_i = \mathrm{Embed}(b_i, Bitrate)$.

La única diferencia con los conjuntos de entrenamiento es que en los conjuntos de test no se conocen las clases (etiquetas), es decir, no sabemos si las imágenes son *cover* o *stego* en A^{test}, si son *stego* o "doble *stego*" en B^{test} o si son "doble *stego*" o "triple *stego*" en C^{test}.

Ahora, supongamos que disponemos de un clasificador C_A (con su correspondiente extractor de características, si es necesario) capaz de clasificar imágenes en *cover* ($\mathcal{C}_A$) y *stego* ($\mathcal{S}_A$) con una tasa de error aceptable. De la misma manera, supongamos que disponemos de los clasificadores C_B y C_C, capaces de clasificar imágenes en *stego* ($\mathcal{S}_B$) y "doble *stego*" ($\mathcal{D}_B$), y en "doble *stego*" ($\mathcal{D}_C$) y "triple *stego*" ($\mathcal{T}_C$) respectivamente.

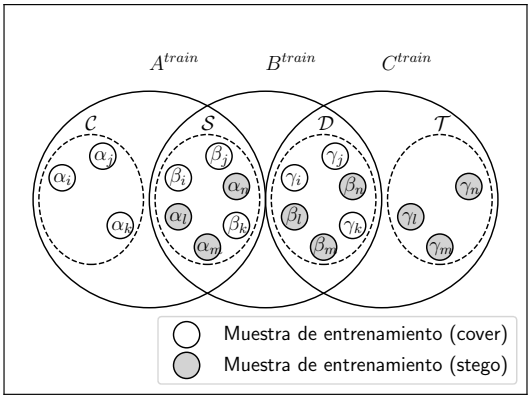

Figura 1. Conjuntos A^{train}, B^{train} y C^{train}

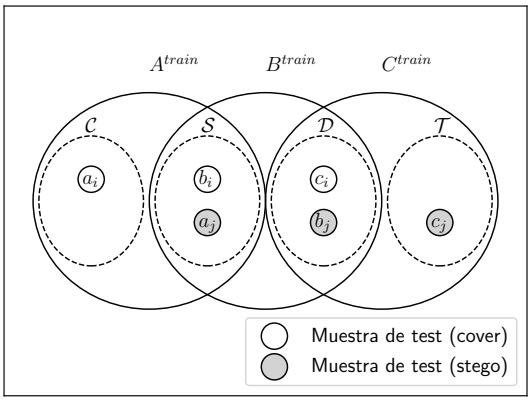

Figura 2. Muestras alineadas

III. Método propuesto

Si disponemos de un conjunto de test que queremos clasificar y de un clasificador entrenado con un conjunto de entrenamiento sin CSM, podremos clasificar el conjunto de test con una probabilidad de error determinada [6]. Si extendemos esta idea a los conjuntos de datos introducidos en la sección anterior, podemos obtener una herramienta para detectar muestras no alineadas entre el conjunto de entrenamiento y el conjunto de test. Dado que existe una biyección entre los elementos de A^{test}, B^{test} y C^{test}, si una imagen de A^{test} se clasifica como *cover* por C_A, esperaremos que la imagen correspondiente de B^{test} se clasifique como *stego* (y no "doble

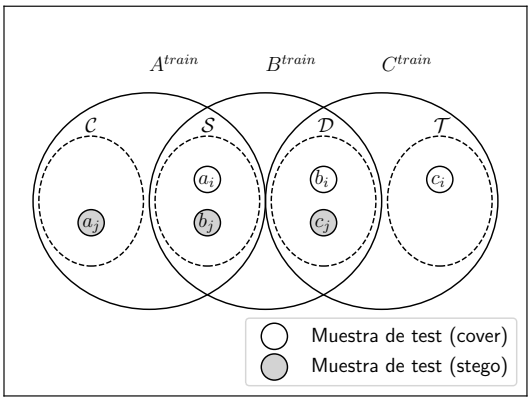

Figura 3. Muestras no alineadas indetectables

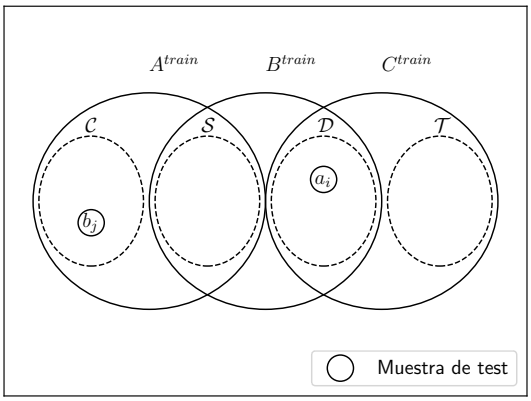

Figura 5. Muestras detectables por F_2

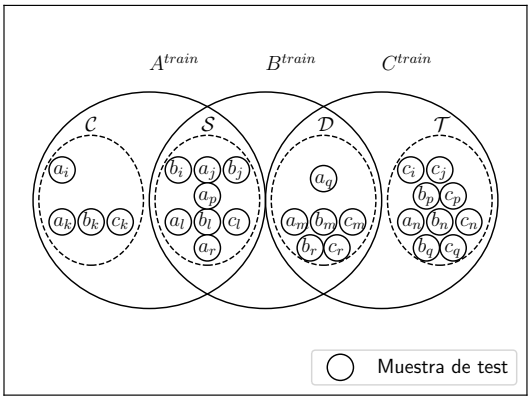

Figura 4. Muestras no alineadas detectables por F_1

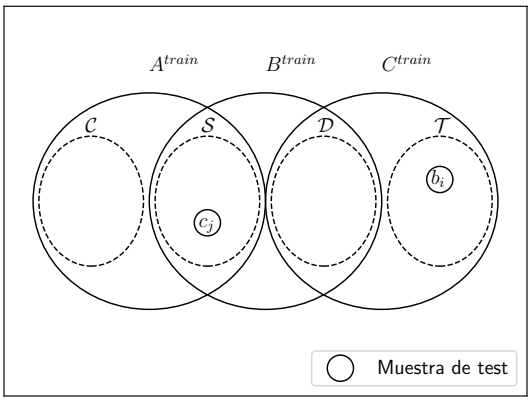

Figura 6. Muestras detectables por F_3

stego") por C_B. Adicionalmente, esperamos que la imagen correspondiente en C^{test} se clasigique como "doble *stego*" y no como "triple *stego*". De forma similar, si una imagen en A^{test} se clasifica como como *stego* por C_A, esperaremos que la imagen correspondiente en B^{test} se clasifique como "doble *stego*" por C_B, y que la imagen correspondiente en C^{test} se clasifique como "triple *stego*" por C_C.

Llamaremos muestra no alineada a la imagen que no cumpla las restricciones esperadas. Si una imagen no es clasificada de forma consistente usando los tres clasificadores, será mejor no clasificarla, pues nuestros clasificadores podrían no ser apropiados para trabajar con dicha imagen. Por lo tanto, el método propuesto puede retornar tres posibles valores para cada imagen clasificada: *cover*, *stego* o no clasificada.

Hay más restricciones que podemos introducir para detectar muestras no alineadas. Llamaremos a estas restricciones *filtros*, y al filtro descrito en el párrafo anterior lo llamaremos F_1. Es decir, F_1 es el resultado de clasificar a_i con C_A, b_i con C_B y c_i con C_C, y verificar que los resultados son consistentes:

$$F_1(i) \equiv \begin{cases} \text{Si } C_A(a_i) = \mathcal{S}_A, & \text{Si } (C_B(b_i) \neq \mathcal{D}_B) \vee (C_C(c_i) \neq \mathcal{T}_C), \\ & \text{etiquetar como "no clasificada",} \\ \text{Si no,} & \text{Si } (C_B(b_i) \neq \mathcal{S}_B) \vee (C_C(c_i) \neq \mathcal{D}_C), \\ & \text{etiquetar como "no clasificada".} \end{cases}$$

en la Figura 1 podemos ver una representación gráfica de las clases. La alineación perfecta de las muestras de entrenamiento y de test se muestra en la Figura 2. en la Figura 4, podemos ver muestras no alineadas que pueden ser detectadas con el filtro F_1.

Ahora podemos definir otros filtros. Si clasificamos $a_i \in A^{test}$ usando C_B, esperamos que se clasifique como *stego* y no "doble *stego*". Si a_i se clasifica "doble *stego*" por C_B es que hay algún tipo de desalineación. La misma idea se puede externder a imágenes *cover*: si usamos C_A para clasificar una muestra $b_i \in B^{test}$ esperamos que b_i se clasifique como *stego* y no como *cover*. Llamaremos a este tipo de filtro F_2. en la Figura 5 podemos ver el tipo muestras que detecta.

$$F_2(i) \equiv \begin{cases} \text{Si } C_B(a_i) \neq \mathcal{S}_B, & \text{etiquetar como "no clasificada",} \\ \text{Si } C_A(b_i) \neq \mathcal{S}_A, & \text{etiquetar como "no clasificada".} \end{cases}$$

Una idea similar se puede aplicar usando los conjuntos B^{test} y C^{test} en lugar de A^{test} y B^{test}, respectivamente. Si clasificamos una muestra $b_i \in B^{test}$ con C_C esperamos obtener "doble *stego*" y no "triple *stego*". De la misma manera, si usamos C_B para clasificar una muestra $c_i \in C^{test}$ esperaremos obtener "doble *stego*" y no *stego*. Llamaremos a este filtro F_3

y podemos ver, en la Figura 6, qué tipo de muestras detecta:

$$F_3(i) \equiv \begin{cases} \text{Si } C_C(b_i) \neq \mathcal{D}_C, & \text{etiquetar como “no clasificada”,} \\ \\ \text{Si } C_B(c_i) \neq \mathcal{D}_B, & \text{etiquetar como “no clasificada”.} \end{cases}$$

Nótese que existe un caso que no puede ser detectado por estos filtros: cuando una muestra se clasifica de forma incorrecta por todos los clasificadores. en la Figura 3 podemos ver una representación de este tipo de casos.

Existen varias formas de aplicar estos filtros. Hemos elegido las siguientes combinaciones (1) filtrado *soft*: únicamente aplicamos F_1; (2) filtrado *medium*: aplicamos F_1 y F_2; y (3) filtrado *hard*: aplicamos los filtros F_1, F_2 y F_3. Los diferentes tipos de filtrado se han seleccionado de esta manera debido a que es más difícil cumplir los requisitos en C^{test} comparado con B^{test} o A^{test}, pues cuantas más operaciones de inserción se realizan, más difícil es para el clasificador separar las dos clases. Los experimentos realizados muestran que C_C es menos preciso al clasificar imágenes en $\mathcal{D}_C$ y $\mathcal{T}_C$ que C_B al clasificar las mismas imágenes en $\mathcal{S}_B$ y $\mathcal{D}_B$. Así, el filtrado *soft* no tiene que cumplir ninguna restricción en C^{test}, el filtrado *medium* introduce algunas restricciones en C^{test} y, finalmente, el filtrado *hard* usa todos los filtros, ampliando así la dificultad para cumplir todas las restricciones.

Tal y como se muestra en la Sección IV, el filtrado incrementa la precisión de el clasificador de forma considerable, lo que nos permite lidiar con el problema del CSM, a cambio de obtener muestras sin clasificar. Podemos aprovechar esta situación para mejorar el clasificador. Si el número de muestras sin clasificar es elevado, podemos intentar mejorarlo añadiéndole nuevas imágenes. Después de reentrenar el clasificador, podemos clasificar de nuevo el conjunto de test para comprobar si el número de muestras no clasificadas se reduce. Si es así, sabremos que las nuevas imágenes completan nuestro conjunto de entrenamiento.

IV. Resultados experimentales

Los experimentos se han realizado usando imágenes de las bases de datos siguientes:

- BOSS, que contiene imágenes de la competición *Break Our Steganographic System!* [2]. Está formada por 10.000 imágenes de 512×512 píxeles, obtenidas de siete cámaras diferentes.
- ESO del European Southern Observatory [5], con imágnes de tamaño variable alrededor de 1200×1200 píxeles.
- Interactions (INTE), que contiene imágenes de Interactions.org [10] de tamaño variable alrededor de 600×400 píxeles.
- NOAA, que contiene imágenes de la National Oceanic and Atmospheric Administration [18] con tamaños variables alrededor de 2000×1500 píxeles.
- Albion (ALBI), que contiene imágenes de la Plant Image Database del Albion College [1] de tanaño 1024×685 píxeles.
- Calphotos (CALP), que contiene imágenes del Regents of the University of California [3] de tamaño variable alrededor de 700×500 píxeles.

Para los siguientes experimentos, hemos preparado una base de datos de entrenamiento usando 5000 imágenes escogidas aleatoriamente de la base de datos BOSS. Cada una de las 5000 imágenes aparece dos veces en el conjunto de entrenamiento, como *stego* (con mensaje) y como *cover* (sin mensaje). Para los conjuntos de test A^{test} hemos obtenido 250 imágenes de las bases de datos de manera que sean diferentes a las imágenes de entrenamiento. De las 250 imágenes, 125 se usan como *cover* y a las 125 restantes se les incrusta un mensaje aleatorio con una clave aleatoria. Estos conjuntos de datos han sido seleccionados con CSM, por lo que siempre usaremos el mismo conjunto de entrenamiento (que proviene de la base de datos BOSS) y diferentes conjuntos de test. Uno de los conjuntos de entrenamiento proviene también de la base de datos BOSS para poder analizar el método propuesto sin CSM.

Para los experimentos se han usado tres algoritmos de esteganografía diferentes en el dominio espacial: *Highly Undetectable steGO* (HUGO) [7], *Wavelet Obtained Weights* (WOW) [8] y *UNIversal WAvelet Relative Distortion* (UNIWARD). [9]. Además, para cada imagen, se ha generado una clave aleatoria diferente. Cada uno de los experimentos se ha realizado usando el mismo algoritmo y ratio de inserción en el conjunto de entrenamiento y en el de test. Con los algoritmos de esteganografía seleccionados, la inserción requerida para construir los conjuntos A^{test}, B^{test} y C^{test} no siempre es posible. En algunos casos (unas 2 ó 3 imágenes por conjunto de test), el proceso de inserción falla. Hemos decidido conservar dichas imágenes y marcarlas como no clasificadas, dado que en una situación real no podríamos descartarlas.

Además, hemos usado el conocido *framework Rich Models* (RM) [6] con *Ensemble Classifiers* (EC) [13]. Este *framework* cumple los requisistos introducidos en la Sección III, es decir, permite clasificar los conjuntos A^{test} (en $\mathcal{C}_A$ y $\mathcal{S}_A$), B^{test} (en $\mathcal{S}_B$ y $\mathcal{D}_B$) y C^{test} (en $\mathcal{D}_C$ y $\mathcal{T}_C$) para HUGO, WOW, UNIWARD y algoritmos similares. No pretendemos que el método propuesto funcione para algoritmos diseñados para eludir el *framework* RM+EC, como el presentado en [15], dado que el método propuesto depende del clasificador subyacente.

IV-A. Detección de muestras no alineadas

En esta sección se presentan los resultados obtenidos con el método *Non-Aligned Sample Detection* (NASD). En la Tabla I, podemos ver diferentes bloques, cada uno con su correspondiente base de datos de entrenamiento, el algoritmo y la ratio de incrustación usados. En cada fila, la base de datos de test va variando, mostrando a continuación los resultados de clasificación. La primera fila de cada bloque corresponde al caso en el que no hay CSM. Las otras filas muestran situaciones en que la base de datos de entrenamiento y la de test son diferentes y, por lo tanto, se produce CSM. Los resultados se expresan en ratio total de errores “Err_T” (error sobre todas las imágenes), ratio de error “Err_C” (error sobre las imágenes clasificadas). También se proporcionan los positivos

Tabla I
RESULTADOS DE CLASIFICACIÓN PARA NASD

		Train	Test	RM+EC	H-NASD						M-NASD						S-NASD					
ALGO	BR	DB	DB	Err	Err_C	Err_T	TP	TN	FP	FN	NC	Err_C	Err_T	TP	TN	FP	FN	NC	Err_C	Err_T	TP	TN	FP	FN	NC
HUGO	0.4	BOSS	BOSS	0.11	0.04	0.03	74	87	3	4	82	0.04	0.03	78	95	3	5	69	0.05	0.04	93	105	5	6	41
HUGO	0.4	BOSS	ALBI	0.07	0.05	0.02	41	43	4	0	162	0.04	0.02	57	45	4	0	144	0.06	0.04	57	110	5	5	73
HUGO	0.4	BOSS	CALP	0.25	0.16	0.02	14	12	4	1	219	0.17	0.03	26	13	7	1	203	0.16	0.06	32	55	14	2	147
HUGO	0.4	BOSS	ESO	0.50	0.00	0.00	2	1	0	0	247	0.00	0.00	2	1	0	0	247	0.40	0.02	2	4	0	4	240
HUGO	0.4	BOSS	INTE	0.50	0.00	0.00	0	0	0	0	250	0.00	0.00	0	0	0	0	250	0.00	0.00	0	0	0	0	250
HUGO	0.4	BOSS	NOAA	0.50	0.00	0.00	0	0	0	0	250	0.00	0.00	0	0	0	0	250	0.00	0.00	0	0	0	0	250
HUGO	0.2	BOSS	BOSS	0.22	0.10	0.03	37	38	2	6	167	0.00	0.06	43	54	3	13	137	0.13	0.09	60	84	6	16	84
HUGO	0.2	BOSS	ALBI	0.35	0.43	0.12	7	34	11	20	178	0.45	0.16	8	40	16	24	162	0.41	0.21	31	46	21	32	120
HUGO	0.2	BOSS	CALP	0.39	0.50	0.01	0	2	2	0	246	0.75	0.02	0	2	3	3	242	0.34	0.08	11	29	10	11	189
HUGO	0.2	BOSS	ESO	0.47	0.00	0.00	0	2	0	0	248	0.00	0.00	1	2	0	0	247	0.44	0.15	1	47	0	38	164
HUGO	0.2	BOSS	INTE	0.47	0.00	0.00	0	0	0	0	250	0.00	0.00	0	0	0	0	250	0.48	0.34	0	93	1	85	71
HUGO	0.2	BOSS	NOAA	0.49	0.00	0.00	0	0	0	0	250	0.00	0.00	0	0	0	0	250	0.49	0.30	0	77	0	75	98
WOW	0.4	BOSS	BOSS	0.16	0.05	0.02	55	65	2	4	124	0.07	0.04	62	75	2	8	103	0.10	0.08	77	98	7	13	55
WOW	0.4	BOSS	ALBI	0.50	0.40	0.01	3	0	2	0	245	0.76	0.05	4	0	13	0	233	0.72	0.05	4	1	13	0	232
WOW	0.4	BOSS	CALP	0.50	1.00	0.00	0	0	1	0	249	1.00	0.00	0	0	1	0	249	0.50	0.00	0	1	1	0	248
WOW	0.4	BOSS	ESO	0.43	0.00	0.00	0	1	0	0	249	0.00	0.00	0	1	0	0	249	0.46	0.21	0	62	0	52	136
WOW	0.4	BOSS	INTE	0.50	0.00	0.00	0	0	0	0	250	0.00	0.00	0	0	0	0	250	0.59	0.08	0	13	0	19	218
WOW	0.4	BOSS	NOAA	0.50	0.00	0.00	0	0	0	0	250	0.00	0.00	0	0	0	0	250	0.58	0.30	0	54	0	74	122
WOW	0.2	BOSS	BOSS	0.33	0.12	0.01	6	9	1	1	233	0.31	0.06	14	19	8	7	202	0.28	0.14	30	57	16	18	129
WOW	0.2	BOSS	ALBI	0.48	0.00	0.00	0	0	0	0	250	0.00	0.00	0	0	0	0	250	0.48	0.10	0	28	0	26	196
WOW	0.2	BOSS	CALP	0.51	0.00	0.00	0	0	0	0	250	0.00	0.00	0	0	0	0	250	0.50	0.02	2	4	1	5	238
WOW	0.2	BOSS	ESO	0.50	0.00	0.00	0	0	0	0	250	0.77	0.04	3	0	10	0	237	0.55	0.09	8	11	16	7	208
WOW	0.2	BOSS	INTE	0.50	0.00	0.00	0	0	0	0	250	0.00	0.00	0	0	0	0	250	0.63	0.04	5	1	7	3	234
WOW	0.2	BOSS	NOAA	0.48	0.00	0.00	0	0	0	0	250	0.68	0.06	7	0	15	0	228	0.54	0.22	36	12	41	15	146
UNIW	0.4	BOSS	BOSS	0.13	0.06	0.04	59	72	2	7	110	0.07	0.04	68	78	2	9	93	0.09	0.08	88	99	6	13	44
UNIW	0.4	BOSS	ALBI	0.28	0.42	0.04	7	8	7	4	224	0.63	0.11	8	8	23	4	207	0.36	0.16	8	63	23	17	139
UNIW	0.4	BOSS	CALP	0.46	0.56	0.02	3	1	4	1	241	0.61	0.09	10	4	21	1	214	0.54	0.12	16	9	28	1	196
UNIW	0.4	BOSS	ESO	0.44	0.50	0.00	0	1	1	0	248	0.67	0.02	0	2	4	0	244	0.48	0.12	1	32	4	27	186
UNIW	0.4	BOSS	INTE	0.47	0.00	0.00	0	0	0	0	250	0.00	0.00	0	0	0	0	250	0.62	0.10	0	15	0	24	211
UNIW	0.4	BOSS	NOAA	0.38	0.00	0.00	0	0	0	0	250	1.00	0.00	0	0	1	0	249	0.44	0.14	0	46	1	35	168
UNIW	0.2	BOSS	BOSS	0.24	0.12	0.03	21	29	1	6	193	0.15	0.05	33	37	2	10	168	0.13	0.07	54	71	3	15	107
UNIW	0.2	BOSS	ALBI	0.44	0.36	0.02	5	2	2	2	239	0.48	0.10	10	17	7	18	198	0.47	0.15	25	18	19	19	169
UNIW	0.2	BOSS	CALP	0.48	0.00	0.00	0	1	0	0	249	0.60	0.01	1	1	3	0	245	0.44	0.03	6	4	8	0	232
UNIW	0.2	BOSS	ESO	0.47	0.00	0.00	0	0	0	0	250	0.00	0.00	0	0	0	0	250	0.44	0.08	1	24	1	19	205
UNIW	0.2	BOSS	INTE	0.51	0.00	0.00	0	0	0	0	250	0.00	0.00	0	0	0	0	250	0.78	0.03	0	2	1	6	241
UNIW	0.2	BOSS	NOAA	0.51	0.00	0.00	0	0	0	0	250	0.00	0.00	0	0	0	0	250	0.49	0.08	0	21	0	20	209
TOTAL				**0.40**	**0.12**		**334**	**408**	**49**	**56**		**0.21**		**435**	**494**	**148**	**103**		**0.32**		**648**	**1396**	**258**	**702**	

Tabla II
RESULTADOS PARA NASD-R

		Train	Test	RM+EC	H-NASD						
ALGO	BR	DB	DB	Err	Err_C	Err_T	TP	TN	FP	FN	NC
HUGO	0.40	BOSS	CALP	0.25	0.16	0.02	14	12	4	1	219
HUGO	0.40	BOSS+ALBI	CALP	0.23	0.12	0.03	29	29	3	5	184
HUGO	0.40	BOSS+ESO	CALP	0.23	0.06	0.01	29	31	4	0	186
HUGO	0.40	BOSS+NOAA	CALP	0.34	0.66	0.06	5	3	16	0	226
HUGO	0.40	BOSS+INTE	CALP	0.25	0.36	0.03	8	6	7	1	228
HUGO	0.40	BOSS+ALBI+ESO	CALP	0.21	0.07	0.02	40	39	3	3	165
HUGO	0.40	BOSS+ALBI+ESO+NOAA	CALP	0.18	0.07	0.02	40	41	2	5	160
HUGO	0.40	BOSS+ALBI+ESO+NOAA+INTE	CALP	0.20	0.09	0.03	36	39	2	6	167

verdaderos (TP), los falsos positivos (FP), los negativos verdaderos (TN) y los falsos negativos (FN), así como el número de muestras no clasificadas. Como referencia, también indicamos los resultados obtenidos usando el *framework* EC+RM. Sin embargo, una comparación directa no sería justa, puesto que EC+RM clasifica todas las muestras, mientras que el método propuesto puede elegir no clasificar. Por otra parte, el método propuesto está también basado en RM+EC, pero con una fase adicional de filtrado.

En la Tabla I, se observa que NASD con filtrado *hard* ("H-NASD") detecta el CSM de forma precisa. Cuando hay CSM los falsos positivos y falsos negativos son casi cero, dejando una gran cantidad de imágenes sin clasificar, tal y como se espera. A medida que se va usando un filtrado menos estricto, el número de no clasificados decrece, aunque incrementa el error de clasificación. Al usar H-NASD se produce un considerable número de muestras no clasificadas, incluso en casos donde no hay un CSM obvio. En contraste, obtenemos unos errores de clasificación muy reducidos. En la última fila de la Tabla I mostramos el número total de TP, TN, FP y FN. El método propuesto, con todas las opciones de filtrado, nos llevan a un error de clasificación mejorado (Err_C) respecto a RM+EC. Como se puede observar, el método propuesto permite identificar las imágenes *cover* y *stego* con una alta fiabilidad a cambio de dejar las muestras con CSM sin clasificar.

IV-B. NASD reforzando la base de datos de entrenamiento

En esta sección presentamos los resultados para NASD con refuerzo de la base de datos (NASD-R). La idea tras este experimento consiste en mostrar que el conjunto de entrenamiento puede ser completado sin añadir imágenes innecesarias. En este caso hemos usado algunas de las bases de datos para reforzar el entrenamiento.

La primera fila de la Tabla II muestra un ejemplo con CSM: se usan imágenes de BOSS para entrenar e imágenes

de Calphotos para el test. Bajo estas condiciones, y usando el filtrado H-NASD, se producen 219 no clasificados. En la segunda fila se muestra el resultado después de reforzar el conjunto de entrenamiento con imágenes de la base de datos Albion. El número de no clasificados disminye a 184. La tercera fila muestra el resultado después de reforzar el conjunto de entrenamiento con imágenes de ESO. En este caso, el número de no clasificados es de 186. En la cuarta y la quinta filas la base de datos se refuerza con imágenes de NOAA e Interactions, respectivamente. El número de no clasificados no mejora, de hecho, empeora. A la vista de los resultados, vemos que Albion y ESO pueden ser convenientes para mejorar el conjunto de entrenamiento. La sexta fila muestra el resultado obtenido después de reforzar el conjunto de entrenamiento con ambas bases de datos, Albion y ESO. En este caso, el número de no clasificados disminuye hasta 165. El error de clasificación es de $0{,}02$ ($\mathrm{Err_T}$), o $0{,}07$ si consideramos únicamente las imágenes clasificadas. En la séptima y octava filas se muestra el resultado después de reforzar el conjunto de entrenamiento también con NOAA e Interactions. El resultado no mejora significativamente, lo que tiene sentido si vemos los resultados anteriores. Este procedimiento puede ser repetido con otras imágenes hasta que la mayoría de las imágenes queden clasificadas, lo cual nos indicaría que el conjunto de entrenamiento es apto para clasificar el conjunto de test. De esta manera, el problema de CSM quedaría eliminado.

V. Conclusiones

En este artículo, se presenta un método de estegoanálisis supervisado capaz de diagnosticar CSM. Adicionalmente, se muestra una metodología que permite mejorar el conjunto de entrenamiento añadiendo nuevas imágenes hasta obtener uno más adecuado para clasificar un conjunto de test dado. El método propuesto se ha verificado usando tres algoritmos de esteganografía: HUGO, WOW y UNIWARD, y cinco bases de datos de imágenes diferentes que producen CSM. Los experimentos muestran que el método propuesto permite clasificar imágenes con una alta precisión a cambio de dejar las imágenes con CSM sin clasificar.

Como trabajo futuro, sería interesante analizar si otras condiciones también causan muestras no alineadas, como por ejemplo un algoritmo o una ratio de incrustación incorrectos.

Referencias

[1] "Plant Image DataBase from Albion College," Available online at: http://www4.albion.edu/plants/, n.d, accessed on 30 de junio de 2018. [Online].

[2] P. Bas, T. Filler, and T. Pevný, ""Break Our Steganographic System": The Ins and Outs of Organizing BOSS," in *Proceedings of the 13th International Conference on Information Hiding*, ser. IH'11. Springer-Verlag, 2011, pp. 59–70.

[3] "Regents of the University of California, Berkeley," Available: http://calphotos.berkeley.edu/, n.d, (Accessed on 30 de junio de 2018).

[4] G. Cancelli, G. Doërr, M. Barni, and I. J. Cox, "A Comparative Study of ± 1 Steganalyzers," in *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*, 2008, pp. 791–796.

[5] "European Southern Observatory," Available: http://www.eso.org/public/images/, n.d, accessed on 30 de junio de 2018.

[6] J. Fridrich and J. Kodovský, "Rich Models for Steganalysis of Digital Images," *IEEE Transactions on Information Forensics and Security*, vol. 7, pp. 868–882, 2012.

[7] G. Gul and F. Kurugollu, "A New Methodology in Steganalysis: Breaking Highly Undetectable Steganography (HUGO)," in *Proceedings of the 13th International Conference on Information Hiding*, 2011, pp. 71–84.

[8] V. Holub and J. Fridrich, "Designing Steganographic Distortion Using Directional Filters." in *International Workshop on Information Forensics and Security (WIFS)*, 2012, pp. 234–239.

[9] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, pp. 1–13, 2014.

[10] "Interactions.org Particle Physics News and Resources," Available: http://www.interactions.org/cms/?pid=1900, n.d, accessed on 30 de junio de 2018. [Online].

[11] A. D. Ker, P. Bas, R. Böhme, R. Cogranne, S. Craver, T. Filler, J. Fridrich, and T. Pevný, "Moving Steganography and Steganalysis from the Laboratory into the Real World," in *Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security*, 2013, pp. 45–58.

[12] A. D. Ker and T. Pevný, "A Mishmash of Methods for Mitigating the Model Mismatch Mess," in *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 9028, 2014, pp. 90 280I–90 280I–15.

[13] J. Kodovský, J. J. Fridrich, and V. Holub, "Ensemble Classifiers for Steganalysis of Digital Media." *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2012.

[14] J. Kodovský, V. Sedighi, and J. Fridrich, "Study of Cover Source Mismatch in Steganalysis and Ways to Mitigate its Impact," in *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 9028, 2014.

[15] S. Kouider, M. Chaumont, and W. Puech, "Adaptive steganography by oracle (ASO)," in *International Conference on Multimedia and Expo (ICME)*. IEEE, 2013, pp. 1–6.

[16] D. Lerch-Hostalot and D. Megías, "Unsupervised steganalysis based on artificial training sets," *Engineering Applications of Artificial Intelligence*, vol. 50, pp. 45–59, 2016.

[17] I. Lubenko and A. D. Ker, "Going from Small to Large Data in Steganalysis," in *Media Watermarking, Security, and Forensics 2012*, ser. Proceedings of SPIE - The International Society for Optical Engineering, vol. 8303, 2012, pp. 0M01–0M10.

[18] "National Oceanic and Atmospheric Administration (NOAA)," Available: http://www.photolib.noaa.gov/, n.d, accessed on 30 de junio de 2018. [Online].

[19] J. Pasquet, S. Bringay, and M. Chaumont, "Steganalysis with Cover-Source Mismatch and a Small Learning Database," in *Proceedings of the 22nd European Signal Processing Conference (EUSIPCO)*, 2014, pp. 2425–2429.

[20] T. Pevný, P. Bas, and J. Fridrich, "Steganalysis by Subtractive Pixel Adjacency Matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, pp. 215–224, 2010.

[21] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep Learning for Steganalysis via Convolutional Neural Networks," in *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 9409, 2015, pp. 94 090J–94 090J-10.

[22] S. Wu, S. Zhong, and Y. Liu, "Deep Residual Learning for Image Steganalysis," *Multimedia Tools and Applications*, pp. 1–17, 2017.

[23] G. Xu, H. Z. Wu, and Y. Q. Shi, "Structural Design of Convolutional Neural Networks for Steganalysis," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, 2016.

[24] X. Xu, J. Dong, W. Wang, and T. Tan, "Robust Steganalysis based on Training Set Construction and Ensemble Classifiers Weighting," in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 1498–1502.