Explainability and Interpretability for Media Forensic Methods: Illustrated on the Example of the Steganalysis Tool Stegdetect

Christian Kraetzer and Mario Hildebrandt
Dept. of Computer Science, Otto-von-Guericke University Magdeburg, Germany

Keywords: Media Forensics, Explainability and Interpretability, Machine Learning, Artificial Intelligence, Quality Assurance and Proficiency Testing in Forensics.

Abstract: For the explainability and interpretability of the outcomes of all forensic investigations, including those in media forensics, the quality assurance and proficiency testing work performed needs to ensure not only the necessary technical competencies of the individual practitioners involved in an examination. It also needs to enable the investigators to have sufficient understanding of machine learning (ML) or 'artificial intelligence' (AI) systems used and are able to ascertain and demonstrate the validity and integrity of evidence in the context of criminal investigations.

In this paper, it is illustrated on the example of applying the multi-class steganalysis tool Stegdetect to find steganographic messages hidden in digital images, why the explainability and interpretability of the outcomes of media forensic investigations are a challenge to researchers and forensic practitioners.

1 INTRODUCTION

Forensic procedures in many domains, but especially in media forensics, see an increase in machine learning (ML) or 'artificial intelligence' (AI) based analysis and investigation tools. Such methods have to undergo rigorous quality assurance evaluations and proficiency testing like all other methods to be used in trustworthy forensic processes. The requirements for such evaluations are very illustratively summarised for example in the corresponding European Network of Forensic Science Institutes (ENFSI) Best Practice Manuals (BPM), see e.g., (European Network of Forensic Science Institutes (ENFSI), 2021) for the guidelines on digital image authentication. In contrast to validation methodologies for well established human based analysis methods, which, to a large extent, rely on internationally acknowledged proficiency tests, the evaluation of ML or AI based methods has not jet reached the same degree of maturity. This is the reason why the (UNICRI and INTERPOL, 2023) state that using such systems "*for high-stakes decisions such as those taken in criminal justice and law enforcement contexts is controversial.*" In quality assurance of such methods, besides their accuracy, robustness and efficiency, also factors focusing on human control and oversight have to be addressed. Part of this complex is the question how good an human expert (here, a forensic practitioner) can interpret the method and its output for the 'customer' in a forensic process (the beneficiary of the forensic report, usually a police officer, prosecutor, judge or jury). In this context, it has to be differentiated between explainability and interpretability of machine learning (ML) / 'artificial intelligence' (AI) methods. (UNICRI and INTERPOL, 2023) defines the explainability of such methods as follows: "*Explainability (in a narrow sense) refers to the ability of developers and users of an AI system to understand its functioning, meaning how the system makes decisions or generates outputs. It focuses on the inner workings of the AI system, its internal logic or underlying processes.*" In this paper this is addressed by process modelling and validation work, including opening up the black box of an ML-based analysis tool, replacing the core part of the tool (the used classification algorithm) by an interchangeable module and then benchmarking the detection performance changes occurring when different, well established, classification algorithms are used (with corresponding trained models) instead of the original one while still using the same feature space.

Interpretability, as the second of these two aspects, is defined in (UNICRI and INTERPOL, 2023) as: "*the ability to provide reasoning for a specific outcome the system has produced – in other words, to*

Kraetzer, C. and Hildebrandt, M.
Explainability and Interpretability for Media Forensic Methods: Illustrated on the Example of the Steganalysis Tool Stegdetect.
DOI: 10.5220/0012423800003660

In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 4: VISAPP*, pages 585-592
ISBN: 978-989-758-679-8; ISSN: 2184-4321

understand why a certain result has been generated." Not all AI methods are in their workings and decisions entirely explainable. (UNICRI and INTERPOL, 2023) points out that with the recently developed research field of 'Explainable AI' an entire scientific domain has formed, focusing on work that intends to ensure the interpretability of non-explainable models. In this paper, the issue of interpretability is addressed by creating reproducible data sets with fixed parameters, including different JPEG-compression algorithms - although this does not ensure interpretability for real world use cases, this intermediate step is necessary in order to assess the output of the ML classification models.

The ML-based forensic analysis tool considered in this paper is the steganalysis tool Stegdetect. It implements steganalysis (as practice of detecting steganographic[1] communication) for multiple different steganographic tools for JPEG images. In short, Stegdetect tries to detect the presence of steganographically hidden information in those images and if corresponding traces are found it performs an embedding method attributation and reports a decision confidence. Stegdetect was introduced in 2002 in the seminal work of Provos and Honeyman (Provos and Honeyman, 2002). It is described in more detail in Section 2.2 of this paper.

The contributions in this paper are:

- Turning Stegdetect from a black-box detection engine into a gray-box setup, to make its workings more explainable and allow for a validation of the feature space and original classification algorithm used in this tool
- The replacement of the original classification algorithm in Stegdetect and a benchmarking of a set of alternative methods
- Empirical experiments addressing the explainability and interpretability of Stegdetect decisions

The empirical evaluations are performed using the ALASKA2 image steganography reference database (Kaggle, 2020). On one hand, it allows for easy performance of obtained steganalysis performances with other publications (since ALASKA2 is widely established in this field). On the other hand, the image characteristics (especially the resolution of 512x512) of the 75000 images in the ALASKA2 set are assumedly very close to images that might have been used by Provos and Honeyman in 2002. Thereby, a certain degree of comparability to the results in that publication could be assumed.

The paper is structured as follows: In Section 2 some details on the state of the art are summarised. These include a brief introduction to explainability and interpretability in ML/AI for law enforcement and forensics as well as a brief summary on multi-class steganalysis with Stegdetect, which is the basis for the empirical evaluations performed within this paper. In Section 3 the experimental setup is introduced, while Section 4 discusses the evaluation results. The Section 5 closes the paper with a summary and conclusions.

2 BACKGROUND

In Section 2.1 a perspective on the issues of explainability and interpretability in ML/AI for law enforcement and forensics is presented. This perspective is strongly based on the United Nations Interregional Crime and Justice Research Institute (UNICRI) and International Criminal Police Organization (INTERPOL) joined efford on a 'Toolkit for Responsible AI Innovation in Law Enforcement' ('AI Toolkit'). In Section 2.2 the ML-based tool Stegdetect, which acts as the illustration example for explainability and interpretability issues in this paper, is introduced in some detail.

2.1 Explainability and Interpretability in ML/AI for Law Enforcement and Forensics

Due to the huge amount of benefits that machine learning (ML) or 'artificial intelligence' (AI) based methods can offer in law enforcement and forensics there is a strong pull from policy makers to see them integrated into an increasing number of procedures and forensic processes. A good example for this pull is the following quote from the 60 page document (Vaughan et al., 2020) issued by the National Police Chiefs' Council (NPCC) of the United Kingdom of Great Britain and Northern Ireland (GB) in 2020: "*The insights DF* [=digital forensics] *science can bring to an investigation are unique in forensic science disciplines; society's pervasive use of technology gives new power to DF science, allowing phones, computers and even smart speakers, watches or doorbells to act as 'digital witnesses' to what happens in daily life. We can get rapid insights from DF*

[1] Steganography is considered to be the art and science of hidden communication. In contrast to cryptography, where only the content of a communication is hidden but the communication itself is visible, in steganography also the existence of the communication channel is hidden, by embedding/hiding the message into innocent looking cover objects, e.g., digital images.

analysis which in the past could have taken months of physical surveillance. It also gives unprecedented access to someone's innermost thoughts from the content of conversations, or search histories. If policing is to use this ability, it is vital it does so responsibly and sensitive to the ethical issues that arise. As well as new investigative opportunities, advances in technology offer opportunities to expand DF services. Rapid growth in cloud services will allow us to simplify and rationalise DF data storage. These same cloud services allow investigations access to more processing power, to harness the power of automation and explore the potential of new and evolving technologies such as machine learning."

Until now, forensic practitioners are very hesitant to rely to much on ML/AI in trustworthy forensic processes. The reasons lie on one hand in issues of accuracy and proficiency and on the other hand in concerns regarding explainability and interpretability.

The accuracy of such methods is discussed in (UNICRI and INTERPOL, 2023) as: "*Accuracy corresponds to the degree to which an AI system can make correct predictions, recommendations or decisions. It is important that agencies verify that any system they are developing and/or intend to use is highly accurate, as using inaccurate AI systems can result in various types of harm.*" [...] *The accuracy of an AI system is dependent on the way the system was developed, and in particular the data that was used to train it. In fact, training the system with sufficient and good quality data is paramount to building a good AI model.* [...] *In most cases, it is preferable that the training data relates to the same or a similar context as the one where the AI system will be used.*"

The definitions for explainability and interpretability have already been discussed in Section 1 above. As response to the issue of explainability requirements, (UNICRI and INTERPOL, 2023) points toward the research field of 'explainable AI', which "[...] *aims to ensure that even when humans cannot understand 'how' an AI system has reached an output, they can at least understand 'why' it has produced that specific output. This field distinguishes explainability in a narrow sense, as different from interpretability.* [...] *In the context of criminal investigations, the explainability of AI systems used to obtain or analyze evidence is particularly important. In fact, in some jurisdictions, criminal evidence obtained with the support of AI systems has been challenged in courts on the basis of a lack of understanding of the way the systems function. While the requirements for evidence admissibility are different in each jurisdiction, a sufficient degree of explainability needs to be ensured for any AI system used to obtain and examine criminal evidence. This helps guaranteeing, alongside the necessary technical competencies, that law enforcement officers involved in investigations and forensic examinations have sufficient understanding of the AI systems used to be able to ascertain and demonstrate the validity and integrity of criminal evidence in the context of criminal proceedings.*"

2.2 Multi-Class Steganalysis with Stegdetect

In their seminal paper, (Provos and Honeyman, 2002) criticise the current state-of-the-art in steganalysis approaches at the point of time of their publication in 2002 as being practically irrelevant, due to faulty basic assumptions (modelling as a two-class problem and statistical over-fitting to the training sets). In contrast to these publications Provos and Honeyman construct a multi-class pattern recognition based image steganalysis detector called Stegdetect: Each input image for Stegdetect is considered to be member of one of four classes, either it is an unmodified cover or it is the result of the application of one out of three different steganographic tools (JSteg, JPHide and Out-Guess 0.13b) which have been amongst the state-of-the-art at this point of time. Stegdetect is then applied blindly (without knowledge about the true class) to two million images downloaded from eBay auctions and one million images obtained from USENET archives. As a result, Stegdetect classifies over 1% of all images seem to have been steganographically altered (mostly by JPHide) and therefore contain hidden messages. Based on these findings, the authors describe in (Provos and Honeyman, 2002) also a second tool called Stegbreak for plausibility considerations, i.e., for verifying the existence of messages hidden by JPHide in the images identified by Stegdetect. Their verification approach is based on the assumption that at least some of the passwords used as embedding key for the steganographic embedding are weak passwords. Based on this assumption, they implement for Stegbreak a dictionary attack using JPHide's retrieval function and large (about 1,800,000 words) multi-language dictionaries. This attack is applied to all images that have been flagged as stego objects by the statistical analyses in Stegdetect. To verify the correctness of their tools, Provos and Honeyman insert tracer images into every Stegbreak job. As expected the dictionary attack finds the correct passwords for these tracer images. However, they do not find any single genuine hidden message. Even though the result of this large scale investigation is negative, the method-

ology and concepts for addressing the interpretability behind the work in (Provos and Honeyman, 2002) are remarkable[2] and are widely considered to be amongst the first (modern) works on forensic steganalysis.
This forensic steganalysis process, as an conceptual model, is considered in (Fridrich, 2009) to consist of the following six steps: 1. selection of investigation targets, 2. reliable 2-class detection that distinguishes stego images from cover images, 3. identification of the embedding method, 4. identification of the steganographic software, 5. searching for the stego key and extracting the embedded data, and 6. decoding/deciphering the extracted data and obtaining the secret message (cryptanalysis). In this process model, this paper addresses the steps 2 and 3.

3 EXPERIMENTAL SETUP

Stegdetect was seeing functional updates and bugfixes for some years. The final version released in 2004 by the original authors is Stegdetect 0.6. This last version added detection support for the steganography algorithm F5 to the capabilities and is used as basis for the experiments conducted here.
Three sets of evaluations are performed within this paper with the following test goals:

- First (test goal T1), baseline tests are performed with an unmodified Stegdetect (using its original pre-trained models) on steganography algorithms supposedly supported by Stegdetect and with image data that should be similar to the material considered in (Provos and Honeyman, 2002).

- Second (test goal T2), Stegdetect is used only as a feature extractor allowing for re-training of detector models using different classifiers and thereby for an estimation of the actual performance of the Stegdetect feature space. In those tests steganography algorithms supposedly supported by Stegdetect are used as well as algorithms for which Stegdetect was not used before.

- Third (test goal T3), a sequence of smaller tests is used to determine what the re-trained classifiers (and as a consequence the Stegdetect feature space) are really representing, either image steganalysis (as originally assumed) or a JPEG encoder artefact classification. I.e., it is evaluated whether the Stegdetects feature space actually detects the slight variations within the image caused by steganographic embedding or rather the artefacts caused by different JPEG compression algorithms used to create the training and testing images.

3.1 T1: Evaluation of Stegdetect for Detecting Image Steganography

As one of the few existing multi-class image steganalysis tools existing, Stegdetect is used for the empirical evaluations in this paper. The work performed is done using the latest official version released by Niels Provos as Stegdetect 0.6 ((Provos and Honeyman, 2002), (Provos, 2004)).

Since this version of Stegdetect (and the pre-trained detection models contained therein) was published in 2002, it was decided for our paper to utilise the cover data of the ALASKA2 dataset (Kaggle, 2020) well established in the image steganography and steganalysis community, since it consists of digital images similarly sized to those widely used in the early 2000s. By using ALASKA2 data (instead of much newer image steganography reference databases like StegoAppDB) it can be assumed that Stegdetects pre-trained models should work for the supported steganography algorithms at those image sizes.

In the evaluation setup for those baseline tests for T1, all 75000 cover images from the ALASKA2 set are used for embedding of a single stego message. Afterwards, Stegdetect is run on the stego images as well as on the cover images in order to determine the detection performance.

For the exemplary stego algorithms, we use F5 (Feng, 2012) JPHide (Church, 2017), both of which are supposedly supported by Stegdetect.

3.2 T2: Using the Stegdetect Feature Spaces in Conjunction with Various Classifiers

Motivated from the differences in detection performances reported in (Provos and Honeyman, 2002) and the results for our own baseline tests (T1), the second objective in this paper is to perform an evaluation of the feature space of Stegdetect. Besides the actual classification in a black-box mode, the tool allows in a gray-box (debug) setup also the exporting the feature vectors that are extracted. It separates them into the following four feature sub-spaces: 1. Differential of Squares Error features, 2. Gradient features, 3. Roughness features, 4. Spline interpolation features.

[2] Unfortunately, no similar effort regarding the explainability of Stegdetect as a tool that might be used in forensics was performed in (Provos and Honeyman, 2002).

In the experiments within this paper, this gray-box setup is used to enable a feature-level fusion by concatenating the four feature vectors for those subspaces in order to form a combined feature vector. Turning Stegdetect from a black-box into a gray-box feature extractor allows for re-training detector models using different classifiers from the Weka data mining suite (Frank et al., 2016) and thereby for a better estimation of the usability of the Stegdetect feature space. In addition to the stego algorithms natively supported by Stegdetect (F5 and JPHide), for those tests also Jsteg (Champine, 2011), Steghide (Hetzl, 2003) and the LSB stego tool Stegano (Bonhomme, 2023) with an additional JPEG compression are utilized in the training and test procedures.

3.3 T3: Steganalysis vs. JPEG Encoder Artefact Classification

The third experimental setup is designed to investigate the initial assumption of correlation instead of causality towards the detection results using the Stegdetect feature space. In order to perform the experiments, the feature extraction mode of Stegdetect is used exactly like in T2, again with using the Weka data mining suite to perform the evaluation.

For the evaluation a subset of 9370 randomly drawn genuine samples from the ALASKA2 dataset (Kaggle, 2020) are used and first converted into the PNG format. Afterwards, the pixel domain Least Significant Bit (LSB) replacement tool Stegano (Bonhomme, 2023) is used to embed a stego message into those images. After this preparation phase, the images are converted using ImageMagick's convert tool (with command line options `-quality 100 -sampling-factor 4:4:4`) and the Python Image Library (PIL). We chose the latter, because it is also used as the JPEG encoder within the Python implementation of F5 that is used in the test performed here. With that setup it is obvious that the stego message embedded by Stegano is destroyed by the JPEG compression performed. The artefacts caused in the image by the LSB replacement are overwritten by the modifications and corresponding artefacts cased by the new last step in the image editing history, the JPEG compression.

The evaluation setup for T3 consists of multiple experiments:

1. Single compression with the PIL algorithm for genuine and stego images
2. Single compression with the ImageMagic algorithm for genuine and stego images
3. Double compression of stego images with PIL in the first place and ImageMagick in the second place, single compression of the genuine images with PIL
4. Double compression of stego images with ImageMagick in the first place and PIL in the second place, single compression of the genuine images with ImageMagick
5. Double compression of stego images with PIL in the first place and ImageMagick in the second place, single compression of the genuine images with ImageMagick
6. Double compression of stego images with ImageMagick in the first place and PIL in the second place, single compression of the genuine images with PIL

Each of the experiments is designed in order to determine whether Stegdetects feature space actually detects the slight variations within the image caused by steganographic embedding or rather the artefacts caused by different JPEG compression algorithms.

In order to exclude classification-algorithm-dependent influences, the experiments are run as a two fold stratified cross validation with the following five classification algorithms from the Weka data mining suite: Bagging, J48 (C4.5 decision tree), Logistic Model Tree (LMT), RandomForest and SMO.

4 EVALUATION RESULTS

In this Chapter, the results for the experiments described in Chapter 3 are summarised.

4.1 Evaluation Results for T1

The evaluation results using the integrated detection models of Stegdetect are shown in Table 1. The true negative rate classifying the unmodified cover images is with 90.05% rather low, which in return means a false alarm rate of almost 10 percent. Moreover the detection rate for JPHide is quite rather low at 31.96%. For F5, Stegdetect offers two different operation modes. Even using the slower (and more precise) detection mechanism, a mere 8.19% of the stego samples for the Python implementation of F5 are correctly detected. Those results significantly deviate from those reported in the work of Provos and Honeyman (Provos and Honeyman, 2002). This drop in the detection performances between 2002 and now indicates that the pre-trained models in Stegdetect did not age well and lost performance on newer images. This assumption is evaluated in the tests performed for T2.

Table 1: Detection performance of Stegdetect (original, pre-trained models) based on 75000 samples each for the classes genuine (=cover), JPHide and F5.

True Negative (on covers)	True Positive JPHide	True Positive F5
90.05%	31.96%	8.19%

4.2 Evaluation Results for T2

The evaluation results for the two-fold stratified cross-validation using the concatenated feature sub-spaces of Stegdetect together with a training of new detector models using four arbitrarily selected classification schemes of the Weka data mining software are summarised in Table 2. Overall pretty convincing detection rates are achieved for F5, Jsteg and the JPEG-converted LSB samples from Stegano. The latter is quite surprising, as the LSB embedding (which was done on PNG versions of the cover image files) should be heavily corrupted or completely destroyed by the JPEG conversion and compression using ImageMagick's convert utility even at a quality factor of 100%. On the other hand, for Steghide and JPHide no usable results are achieved. While this might be reasonable for Steghide, since Stegdetect is not designed for that algorithm, at least detecting JPHide should yield a better performance.

Those results for Stegano particularly raise the question what is distinguished between using those feature spaces - is it actually the embedding of a steganographic message within a JPEG compressed image or is it rather the utilised JPEG compression library in conjunction with the compression parameters. The sole purpose of the T3 experiments is to answer that question.

4.3 Evaluation Results for T3

After achieving questionable results for some of the steganography algorithms in T2 (especially for the LSB replacement performed by Stegano) while using newly trained models on the existing Stegdetect feature space, additional experiments with double compression of the images using two different JPEG encoders (ImageMagick and PIL) are performed.

The evaluation results for the six experiments for cover data and Stegano output described in Section 3.3 for T3 are summarised in Table 3.

Most of those results indicate that the feature space is more sensitive to the JPEG compression algorithm rather than the embedded steganographic messages. As a result, if genuine (=cover) data and stego data is compressed with the same algorithm, the classification performance is low. There is one exception, though. When using a double compression with ImageMagick in the first compression and PIL in the second compression, the classification accuracy remains high - at $>$ 99.75% - when training with PIL compressed genuine images. Since the LSB replacement performed by Stegano is mostly destroyed during the JPEG compression, this result is quite surprising, especially since all other experiments indicate a total loss of discriminatory power when both sets of images are compressed with the same JPEG encoder in the last step. This phenomenon can not be explained with the current evaluations and would require further investigations.

Generalising the results obtained, it can be stated that the high classification accuracies obtained here (and of course in T2) for Stegano are not due to successfully performed steganalysis but rather due to the fact that the machine learning approach has successfully trained the strongest characteristic present which is in this case the influences imposed to the images by the JPEG encoder used.

5 SUMMARY & CONCLUSIONS

The results presented here for image steganalysis show machine learning driven solutions are suffering from ageing effects of the trained models: In the experiments performed for T1 the performance obtained in 2023 is much lower than the original performance reported in 2002 even though the conditions of the tests are closely reconstructed.

Secondly, detectors that come as a black-box (here Stegdetect) should be turned into (more) transparent (i.e., gray- or white-box) mechanisms, which was here achieved by using the raw feature vectors that could be output from Stegdetect for debug reasons. Based on those feature vectors, a wide range of classification algorithms from a well established data mining suite (here Weka (Frank et al., 2016)) are trained and the results are analysed and compared. This work could easily be extended by applying model interpretation or feature selection strategies also offered within Weka.

Third, Machine learning tends to train/learn the most significant difference in the feature space projections of the classes present in the training data. It is shown, that, in case of the Stegdetect feature space, this tends to be the artefacts caused by the JPEG encoder used. In case of steganography tools that are embedding in the JPEG transform domain (e.g., Jsteg which directly modifies the DCT coefficients) these tools are basically implementing an own, non-standard JPEG encoder. Therefore, in their case the

Table 2: Detection performance using the re-trained Stegdetect (with Weka classifiers) in using the concatenated Stegdetect feature space. Results based on 75000 samples for genuine and stego images in a 2-fold stratified cross-validation.

	F5		Jsteg		Steghide		JPHide		Stegano + ImageMagick convert	
Weka Classifier	TN	TP	TN	TP	TN	TP	TN	TP	TN	TP
Bagging	97.9%	96.9%	97.1%	98.1%	67.2%	35.6%	50.2%	52.7%	99.6%	99.8%
J48	95.9%	96.3%	96.9%	96.9%	84.0%	24.7%	56.3%	55.1%	99.5%	99.6%
RandomForest	98.7%	97.8%	97.9%	98.3%	66.9%	25.0%	40.3%	39.5%	99.8%	99.9%
SMO	96.2%	96.5%	95.5%	96.3%	96.4%	10.8%	55.0%	54.9%	99.6%	99.9%

Table 3: T3: Results of the 2-fold stratified cross-validation using the Weka data mining suite with 9370 genuine and 9370 stego samples; In these tests only the steganography tool Stegano performing LSB replacement in pixel domain was used to create the stego files.

			Training PIL compressed Genuine		Training ImageMagick compressed Genuine	
Last Compression	First Compression	Weka Classifier	Genuine	Stego	Genuine	Stego
Python Image Library (PIL)	none	Bagging	28.06%	27.75%	-	
		J48	100.00%	0.00%		
		LMT	80.90%	18.15%		
		RandomForest	25.86%	23.23%		
		SMO	46.82%	45.94%		
	Image-Magick	Bagging	38.64%	40.22%	99.79%	99.68%
		J48	100.00%	0.00%	99.65%	99.74%
		LMT	80.79%	18.32%	99.88%	99.80%
		RandomForest	26.89%	24.53%	99.83%	99.94%
		SMO	47.85%	48.10%	99.88%	99.78%
ImageMagick	none	Bagging	28.26%	28.80%	-	
		J48	100.00%	0.00%		
		LMT	47.56%	48.88%		
		RandomForest	26.52%	22.37%		
		SMO	45.12%	48.29%		
	PIL	Bagging	99.90%	99.80%	67.04%	98.04%
		J48	99.82%	99.75%	98.09%	97.62%
		LMT	99.93%	99.94%	99.46%	98.92%
		RandomForest	99.99%	99.96%	99.36%	98.16%
		SMO	99.94%	99.94%	99.56%	98.47%

attribution of the stego tool as an attribution of the encoder will give reliable results.

Generalising those results, it can be said that machine learning (ML) / ‘artificial intelligence’ (AI) methods learn to distinguish the most apparent differences between the different classes presented in training. This is not necessarily the one that has been intended in the learning setup but could also be an unforeseen effect, as demonstrated above in the discussions on T3.

For that reason, ML-/AI-based forensic methods need to undergo vigorous quality assurance and proficiency testing before they can be included into trustworthy forensic processes and afterwards a cyclic re-evaluation of the models trained and the feature spaces used needs to be performed during the operational life of such a ML-/AI-based forensic method. Especially in media forensics, models trained are assumed to age pretty badly since the assumed source characteristics are significantly changing over times. This can be well illustrated with digital images, where the technical developments since the late 1990s saw a steady but significant increase in image sizes and resolutions as well as leap-breaks with new image formats (e.g., the High Efficiency Image File Format (HEIF) or High dynamic range (HDR) image formats).

Considering the explainability and interpretability of the outcomes of such forensic investigations, as requested, amongst others, in (UNICRI and INTERPOL, 2023), the quality assurance and proficiency testing work performed needs to ensure not only the necessary technical competencies of the individual practitioners involved in an examination. It also needs to enable the investigators to have sufficient understanding of the ML/AI systems used and to be able to ascertain and demonstrate the validity and integrity of evidence in the context of criminal proceedings.

ACKNOWLEDGEMENTS

This work received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 101021687 (project "UNCOVER").

The authors wish to thank Jana Dittmann for her comments on the evaluation procedure.

Author Contributions. Initial idea & methodology: Christian Kraetzer (CK); Conceptualization: CK, Mario Hildebrandt (MH); Discussion on Explainability and Interpretability in ML/AI for Law Enforcement and Forensics: CK, Empirical evaluations - design: CK, MH; Empirical evaluations - realisation: MH; Writing – original draft: CK; Writing – review & editing: MH.

All authors have read and agreed to the published version of the manuscript.

REFERENCES

Bonhomme, C. (2023). Stegano. *https://github.com/cedricbonhomme/Stegano - last accessed: October 26st, 2023*.

Champine, L. (2011). Jsteg - jpeg steganography. *https://github.com/lukechampine/jsteg - last accessed: October 26st, 2023*.

Church, D. (2017). jphide & seek steganography tools. *https://github.com/h3xx/jphs - last accessed: October 26st, 2023*.

European Network of Forensic Science Institutes (ENFSI) (2021). Best Practice Manual for Digital Image Authentication. Technical Report BPM-DI-03-2021, European Network of Forensic Science Institutes (ENFSI), Wiesbaden, Germany.

Feng, J. (2012). f5-steganography, a python implement of f5 steganography. *https://github.com/jackfengji/f5-steganography - last accessed: October 26st, 2023*.

Frank, E., Hall, M. A., and Witten, I. H. (2016). *The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Burlington, MA, 4th edition.

Fridrich, J. (2009). *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, New York, NY, USA, 1st edition.

Hetzl, S. (2003). Steghide. *https://steghide.sourceforge.net/ - last accessed: October 26st, 2023*.

Kaggle (2020). Alaska2 image steganalysis set. *https://www.kaggle.com/competitions/alaska2-image-steganalysis/data - last accessed: August 21st, 2023*.

Provos, N. (2004). Stegdetect. *https://www.provos.org/p/outguess-and-stegdetect-downloads/ - last accessed: October 24th, 2023*.

Provos, N. and Honeyman, P. (2002). Detecting steganographic content on the internet. In *NDSS*. The Internet Society.

UNICRI and INTERPOL (2023). Toolkit for Responsible AI Innovation in Law Enforcement: Principles for Responsible AI Innovation. Guidelines, United Nations Interregional Crime and Justice Research Institute (UNICRI) and International Criminal Police Organization (INTERPOL), Brussels.

Vaughan, J., Baker, N., and Underhill, M. (2020). Digital Forensic Science Strategy. Policy, National Police Chiefs' Council (NPCC), London, UK.