Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
Monitoring the security of audio biomedical signals communications in wearable IoT healthcare

Saeid Yazdanpanah A, Saman Shojae Chaeikar B,*, Alireza Jolfaci C

A Department of Computer Engineering, Khorraramabad Branch, Islamic Azad University, Khorramabad, Iran
B Australian Institute of Higher Education, Sydney, Australia
C College of Science and Engineering, Flinders University, Adelaide, Australia

ARTICLE INFO

Keywords:
Audio security
Audio signal processing
Data hiding
Healthcare data
IoT security

ABSTRACT

The COVID-19 pandemic has imposed new challenges on the healthcare industry as hospital staff are exposed to a massive coronavirus load when registering new patients, taking temperatures, and providing care. The Ebola epidemic of 2014 is another example of a pandemic which a hospital in New York decided to use an audio-based communication system to protect nurses. This idea quickly turned into an Internet of Things (IoT) healthcare solution to help to communicate with patients remotely. However, it has grabbed the attention of criminals who use this medium as a cover for secret communication. The merging of signal processing and machine-learning techniques has led to the development of steganalyzers with very higher efficiencies, but since the statistical properties of normal audio files differ from those of purely speech audio files, the current steganalysis practices are not efficient enough for this type of content. This research considers the Percent of Equal Adjacent Samples (PEAS) feature for speech steganalysis. This feature efficiently discriminates the least significant bit stego speech samples from clean ones with a single analysis dimension. A sensitivity of 99.82% was achieved for the steganalysis of 50% embedded stego instances using a classifier based on the Gaussian membership function.

1. Introduction

Cryptography is the science of making data unintelligible to unauthorized people without concealing the existence and other critical information about the encrypted data such as duration and frequency of communications, message size, and sender and recipient’s identities to the adversaries [1,2]. Considering the need for concealed communications, steganography was designed as another security method that can conceal secrets within the body of digital media, enhance privacy, prevent traffic analysis, and allow the transfer of secrets in an imperceptible manner. However, considering the applications of steganography, digital forensic scientists have designed a countermeasure that allows them to supervise potential secret data exchanges that may be performed by hackers, terrorists, and lawbreakers [3,4].

Image, video, and audio are the digital media formats that have been widely used by steganographers as cover media [4,5]. Among these formats, audio files occupy an outstanding proportion of today’s media transfer, and therefore, create magnificent opportunities for illegal data transfer without grabbing attention. Notwithstanding, digital forensic scientists have paid less attention to the audio format, as compared to images, for developing state-of-the-art audio steganalysis algorithms, and this domain requires modern steganalyzers that exploit the statistical characteristics of audio formats [6].

As shown in Fig. 1, steganalysis methods can be classified according to the signal characteristics. Multiplicative, phase-encoding, and echo-embedding are the three targeted steganalysis subclasses. The process of multiplicative subclass may be described as $s[n] = c[n](1 + m[n])$ where $c[n]$, $m[n]$ and $s[n]$ are cover, secret message, and stego, respectively [7]. Normal steganalyzers cannot efficiently detect multiplicative steganography, and hence, this class adds absolute value logarithms of audio samples to the steganalysis discrimination factor [1]. Phase coding, the second targeted class, is based on the fact that relative phases between blocks are preserved, while that amongst consecutive blocks are changed [8].

The third class, namely, echo embedding, holds a bank of kernels in which one selected kernel convolves with segmented parts of the cover audio file [9]. Positive-Negative (PN) and Forward-Backward (FB) are two sample echo embedding kernels.

The universal class can be classified into calibrated and non-calibrated branches. In the calibrated methods, the challenge is to find features that have been formed mostly based on the embedded message instead of the original signals [1]. The first subclass in calibrated methods

* Corresponding author.
E-mail address: s.chaeikar@aih.nsw.edu.au (S. Shojae Chaeikar).

https://doi.org/10.1016/j.dcan.2022.11.002
Received 22 December 2021; Received in revised form 27 October 2022; Accepted 1 November 2022
Available online 14 November 2022
2352-8648/© 2022 Chongqing University of Posts and Telecommunications. Publishing Services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
is self-generated cover estimation. Hausdorff Distance (HD), Audio Quality Metrics (AQM), generalized Gaussian distribution, and Gaussian Mixture Model (GMM) are the main methods in this subclass [1]. The second subclass is a constant referencing in which $M(x \cdot y)$ measures the discrepancies between $x$ and $y$ signals and also is extendable to whole signals [10]. Linear basis, which is the third subclass, deals with estimating the original cover space model [10] and works efficiently if the high-dimensional model of cover space is known. Re-embedding, the final class, utilizes both given and re-embedded samples for steganalysis [10]. The intersection between the aforementioned calibrating class methods is that they all need the cover signals for calibration.

The methods in the noncalibrated branch extract the required steganalysis features from time or frequency domain signals [1]. The statistical models of histogram, linear prediction residue, Markov-process-based features, and chaotic-based features are the main time-domain features [11, 12]. The main source of features for the frequency domain are the reversed psychoacoustic model of human hearing, feature fusion, Mel-Frequency Cepstrum Coefficients (MFCC), and the MFCC derivative of audio signals [1].

Despite the development of various audio steganalysis algorithms, the number of fundamental researches in the field of speech steganalysis is very limited. More importantly, the current practices are not efficient enough to preserve the security of the biomedical audio signal communications. This paper proposes a novel speech steganalysis algorithm within the class of noncalibrated universal algorithms. Compared to the state-of-the-art, the proposed algorithm facilitates a fast detection of stego signals due to its single analysis dimension which requires lighter computation.

The rest of this paper is organized as follows: Section 2 reviews the background and related works for WAV, MP3, VoIP, and speech formats. The designed feature and structure of the algorithm are elaborated in Section 3. Section 4 describes the conducted experiments and compares the achievements of the work with the related researches. Finally, Section 5 presents the conclusion of the paper.

### 2. Background and literature

Applications of audio communications on the Internet have been increasing constantly, and the majority of Internet of Things (IoT) based health monitoring systems have added this feature to their services [13, 14]. The popularization of this communication channel has turned it into a favorable channel for cyber adversaries. For instance, a wide range of algorithms and tools such as S-Tool, Hide4PGP, and MP3Stego exist that can embed secret messages in the audio files [15]. As a consequence, audio steganalysis algorithms have become a critical tool for preserving the security of the cyberworld.

Liu et al. [16] introduced a method based on the derivative-based Fourier spectrum and Mel-cepstrum for the steganalysis of WAV files. They reported the method as reliable if the ratio of data hiding is high. Afterwards, to enhance the sensitivity of their method, Liu et al. added Mel-cepstrum coefficients and second-order derivatives of the Markov transition features of audio signals to their algorithm. In another work, Fig. 1. Classification of audio-steganalysis algorithms according to signal characteristics.

Tint and Mya [17] utilized MFCC, Zero Crossing Rate (ZCR), short-time energy features, and spectral flux in their designed steganalyzer. Evaluation results proved that the algorithm works efficiently in the detection of stego files made by Hide4PGP, S-tools4, and Stegowav. The virtual ear [18] is sensitive to variations in the high-frequency regions and is structured based on the Reversed-Mel scale.

In the field of MP3 steganalysis [19], employed a recompression calibration method for detecting the stego files generated by MP3Stego. The algorithm performs well in high-ratio embedding. In another work, Yan et al. [20] designed a steganalysis algorithm for attacking files generated by MP3Stego. The algorithm calculates the Standard Deviation (SD) of the second-order differential sequence in the quantization phase.

Because of the massive application of VoIP technology in messengers, social networks, and various platforms, it is crucial to have specifically designed VoIP steganalyzers. To this end, Ren et al. [21] proposed an Adaptive Multi-Rate (AMR) steganalyzer to detect secret messages hidden based on [22,23] steganography algorithms. Their steganalyzer works based on the probability of the same pulse position and achieved more than 99% accuracy for an embedding ratio of 60%.

In [24], Ren et al. used the Markov model of the adjacent codebook for attacking Huffman codebook stego samples that were created based on Advanced Audio Coding (AAC). Under certain criteria, the algorithm reached 100% accuracy.

One of the earliest Machine-Learning (ML) steganalyzers was designed by Ozer et al. [25]. The model was a mix of SVM and Audio Quality Metrics (AQM) and evaluated against six data hiding techniques – four watermarking and two steganography techniques. Based on the chosen data-hiding technique, the steganalyzer accuracy fluctuates between 87% and 100%. Later, Kraetzner and Dittmann [26] utilized MFCCs as a classification feature and fed the extracted coefficients into a Radial Basis Function (RBF) SVM for detecting stego samples. In a more advanced work [27], presented a Deep Belief Network (DBN) algorithm for detecting stego samples produced by StegHide, Hide4PGP, and FreqSteg. Moreover, the authors claimed that the algorithm could identify which one of the tools produced the stego sample.

In [28], the authors introduced an SVM-based VoIP speech steganalyzer that exploits position distribution characteristics even in low bit-rate VoIP speeches. Their discrimination factors are a set of Markov transition probabilities of pulse position based on joint probability matrices for identifying pulse-to-pulse correlation, probability distribution of pulse positions for long-time distribution of features, and short-time

### Table 1

Notations used.

| Notation  | Description                                      |
|-----------|--------------------------------------------------|
| $D$       | The imported data array of signals from wav file |
| $D_i$     | The $i$th sample in the imported data array from wav file |
| $S_p$     | Speech samples array                             |
| $S_i$     | Silence samples array                            |
| $S_p^i$   | The $i$th sample in the speech samples array     |
| $S_i^i$   | The $i$th sample in the silence samples array    |
| $P_{FEAS}_p$ | Percent of Equal Adjacent Samples (FEAS) in the speech samples array |
| $P_{FEAS}_s$ | Percent of Equal Adjacent Samples (FEAS) in the silence samples array |
| $\tau$    | The threshold for speech and silence separation within signal power domain |
| $P$       | The array of power of signals                    |
| $P_i$     | Power value of the $i$th sample in the power array |
| $P_{max}$ | Maximum power value in the power array           |
| $P_{min}$ | Minimum power value in the power array           |
| $P$       | Normalized power array                           |
| $P_i$     | The $i$th sample within the normalized power array |
| $G(x)$    | Gaussian Membership Function (GMF)              |
| $G_{avg}$ | The average GMF values for speech and silence FEAS |
| $m$       | Mean                                             |
| $m_{sp}$  | Mean of speech-band training data                |
| $m_{sil}$ | Mean of silence-band training data               |
| $\sigma$  | Standard Deviation (SD)                          |
| $\sigma_{sp}$ | SD of speech-band                             |
| $\sigma_{sil}$ | SD of silence-band                          |
| $TP$      | True Positive                                    |
| $FN$      | False Negative                                   |

---

S. Yazdanpanah et al. Digital Communications and Networks 9 (2023) 393–399

---

394
invariance characteristics of speech signals. The algorithm was evaluated against a large set of G.729a-encoded speech samples.

3. Proposed steganalysis method

The designed steganalysis algorithm is built based upon a novel feature named 'Percent of Equal Adjacent Samples', and thus, hereinafter, the algorithm is referred to as PEAS. To formulate the PEAS process and workflow, and also to keep the language clear and consistent throughout the paper, Table 1 summarizes the used notations and abbreviations.

3.1. PEAS workflow

PEAS steganalysis has three main components: signal classification, feature extraction, and instance classification. The first component, i.e., signal classification, separates speech and silence signals using four subcomponents: (1) calculating signal power, (2) normalizing power, (3) applying threshold, and (4) signal classification. The second component, i.e., feature extraction, calculates the PEAS values of speech and silence signals. The final component, i.e., instance classification, is in charge of defining the steganalysis results: clean or stego. Fig. 2 shows the overall PEAS steganalysis workflow.

3.2. Signal classification

The signal classification component aims at classifying the input

Fig. 2. Workflow of Percent of Equal Adjacent Samples (PEAS) steganalysis components.

Fig. 3. Signal classification process.
signals into speech and silence classes. To this end, it calculates and then normalizes the signal power using the first two subcomponents. The third subcomponent applies the threshold value \( \tau \) to separate speech and silence areas. Based on the location of the signals in the normalized power array and the detected speech and silence areas, the fourth subcomponent generates two separate arrays containing speech and silence signals. These steps are illustrated and labeled as A to E in Fig. 3.

The power of a signal is calculated based on Equation (1). However, since the power of a signal depends on the recording quality, two separate recordings of one voice may differ considerably. Normalizing power is a process by which this issue can be resolved as it converts the given signals into the same scale. As the signal powers of a given instance are organized within an array, a min-max feature scaling, as shown in Equation (2), was used to normalize the power array.

\[
P_i = |D_i|^2, \quad \text{for} \quad i = 1, 2, \ldots, \text{len}(P)
\]

\[
\hat{P}_i = \left( \frac{P_i - P_{\text{min}}}{P_{\text{max}} - P_{\text{min}}} \right)
\]

Once the power array is normalized, the signals can be classified into speech and silence bands according to the value of \( \tau \). In Fig. 3, this process is illustrated and labeled as D, E, and F. The efficient value of \( \tau \) is defined in Section 3.6 based on the results of several experiments. In the next step, a mapping between the detected speech and silence bands and input samples constructs the content of the speech and silence arrays. Algorithm 1 shows the steps of this process.

**Algorithm 1** Pseudocode for allocating the imported signals into speech and silence arrays based on the optimized threshold

1. procedure SIGNAL-CLASSIFIER(P, S p, S i)
2. for \( i = 1 \) to \( \text{len}(P) \) do
3. if \( \hat{P}_i \geq \tau \) then
4. append(\( P_i, S p \));
5. else
6. append(\( P_i, S i \));
7. end if
8. end for
9. Return (\( S p, S i \));
10. end procedure

3.3. Feature extraction

Feature extraction, i.e., the second component, receives two arrays containing speech and silence samples and calculates the PEAS values of the arrays.

3.3.1. PEAS feature

The PEAS feature is regarded as a discrimination factor between clean and stego samples. To calculate PEAS, all adjacent samples, from the first signal until the last one, are compared. If the values of the compared adjacent samples are equal, the counter of equal adjacent samples increases by 1. PEAS is the proportion of the value of the counter to the total number of the samples in the file. The formulas for calculating the PEAS values of speech and silence bands are presented in Equations (3) and (4), respectively. The concept of PEAS is illustrated in Fig. 4.

\[
\text{PEAS}_{Sp} = \frac{\sum_{i=0}^{\text{len}(Sp)-2}[Sp_i = Sp_{i+1}] \times 100}{\text{len}(Sp)}
\]

\[
\text{PEAS}_{Si} = \frac{\sum_{i=0}^{\text{len}(Si)-2}[Si_i = Si_{i+1}] \times 100}{\text{len}(Si)}
\]

In addition, the pseudocode given in Algorithm 2 shows the steps for calculating PEAS for speech and silence signals.

**Algorithm 2** Pseudocode for calculating PEAS in speech and silence bands

1. procedure PEAS(S p, S i)
2. \( S p\text{Count} = 0; \)
3. \( S i\text{Count} = 0; \)
4. for \( i = 0 \) to \( \text{len}(S p) - 2 \) do
5. if \( S p_i = S p_{i+1} \) then
6. \( S p\text{Count} += 1; \)
7. end if
8. end for
9. \( \text{PEAS}_{Sp} = (S p\text{Count} \times 100) / \text{len}(S p); \)
10. for \( i = 0 \) to \( \text{len}(S i) - 2 \) do
11. if \( S i_i = S i_{i+1} \) then
12. \( S i\text{Count} += 1; \)
13. end if
14. end for
15. \( \text{PEAS}_{Si} = (S i\text{Count} \times 100) / \text{len}(S i); \)
16. Return (\( \text{PEAS}_{Sp}, \text{PEAS}_{Si} \));
17. end procedure

3.3.2. Signal correlativity regression

Data embedding distorts the harmony among the signals of a recorded speech. Consequently, the PEAS values of stego signals differs from those of clean ones. In other words, the more the data embedded, the more is the distortion in the natural harmony between adjacent samples. At low embedding ratios, the signal distortions are hardly sensible; in contrast, at higher ratios, the distance increases, and stego signals are more accurately detectable. In PEAS speech steganalysis, the levels of distortions between adjacent samples are utilized as discrimination factors.

3.4. Instance classification

The third component, i.e., instance classification, classifies the analyzed sample as clean or stego. It receives the PEAS values from the feature extraction component and calculates their Gaussian membership degrees compared to the ranges in the extracted clean and stego reference profiles. The outputs of the first subcomponent are five sets of Gaussian membership degrees obtained by comparing PEAS values with five reference profiles. The second subcomponent calculates the average value for all Gaussian membership degree pairs and passes them into the decision-making subcomponent. The final subcomponent chooses the embedding ratio of the corresponding reference profile with the
Table 2: The extracted PEAS values from a given instance are compared to this function, each element is mapped to a value in the range of 0–1. This value, i.e., the so-called membership degree or membership value, quantifies the membership grade of each element in form of a fuzzy set (0–1).

To calculate the Gaussian membership degree for a given value, two parameters m and σ are required. In this work, the two parameters are taken from the corresponding training dataset. Since the given voice instances were examined in terms of speech and silence bands and also with five embedding ratios, 10 sets of m, σ are required to construct a profile that covers the steganalysis of voice instances in all embedding ratios. The basic structure of the PEAS reference profile is given in Table 2. The extracted PEAS values from a given instance are compared with the profiles to determine whether it is a stego.

The Gaussian membership function is the chosen classifier in PEAS. The membership degree is calculated using Equation (5), where x is a variable holding the feature value, and G(x) is its membership degree. As the PEAS values for speech and silence bands are the selected steganalysis features, by replacing the variable x in Equation (5) with PEAS_sp and PEAS_si, we can obtain Equations (6) and (7), respectively.

\[ G(x) = \exp \left( -\frac{(x - m)^2}{2\sigma^2} \right) \]  

(5)

\[ G(\text{PEAS}_p) = \exp \left( -\frac{(\text{PEAS}_p - m_p)^2}{2\sigma_p^2} \right) \]  

(6)

\[ G(\text{PEAS}_s) = \exp \left( -\frac{(\text{PEAS}_s - m_s)^2}{2\sigma_s^2} \right) \]  

(7)

Using Equations (6) and (7), two Gaussian membership degrees are produced. To determine a single value for decision making, the average of these membership values is computed by using Equation (8).

\[ G_{av}(\text{PEAS}_p, \text{PEAS}_s) = \frac{G(\text{PEAS}_p) - G(\text{PEAS}_s)}{2} \]  

(8)

To achieve a final and unified equation, in Equation (8), G(x) is replaced with its equivalent in Equation (5). Equation (9) is the final equation, and it is a combination of the average Gaussian membership function and \( \text{PEAS}_p \) and \( \text{PEAS}_s \) formulas. To obtain the final steganalysis result, five calculations are performed using Equation (9) with m and σ of \( \text{PEAS}_p \) and \( \text{PEAS}_s \) in the five embedding ratios. The embedding ratio returning the highest average Gaussian membership degree defines whether the given sample is identified as clean or stego. Algorithm 3 presents the code for defining the steganalysis result using Equation (9).

\[ G_{av}(\text{PEAS}_p, \text{PEAS}_s) = \exp \left[ -\frac{\sum_{i=0}^{\text{len}(\text{sp})-2} [\text{sp}_i - \text{sp}_{i+1}]^2}{\text{len}(\text{sp}) \cdot 2\sigma_p^2} \right] + \exp \left[ -\frac{\sum_{i=0}^{\text{len}(\text{si})-2} [\text{si}_i - \text{si}_{i+1}]^2}{\text{len}(\text{si}) \cdot 2\sigma_s^2} \right] \]  

(9)

3.5. Database

In this research, the speech part of the Austeg dataset was used to train and evaluate the PEAS steganalysis algorithm. The speech part of this dataset consists of 18,000 noisy and noise-free speech instances in English, Farsi, and Chinese languages and three chunk lengths of 3, 6, and 10 s. The instances are in 44100 Hz, 16-bit mono WAV format. To produce the required stego instances, the Wavsteg tool in Python was used to embed the Austeg dataset instances by the LSB replacement algorithm in ratios of 12.5%, 25%, 37.5%, and 50%. The final dataset, including the stego copies, consisted of 90,000 instances in 0% (clean), 12.5%, 25%, 37.5%, and 50% embedding ratios.

3.6. Training

As mentioned, 70% of the utterance instances in the database were used for training the classifier. Since the Gaussian membership function

Table 3: Extracted values for m and σ for speech and silence bands of sample voices in the training database; for five embedding ratios and five thresholds.

| Threshold | Channel | Clean (0%) | 12.5% stego | 25% stego | 37.5% stego | 50% stego |
|-----------|---------|------------|------------|------------|------------|------------|
|           | Clean (0%) | 12.5% stego | 25% stego | 37.5% stego | 50% stego |
| 0.10      | Speech | 0.0059 | 0.0007 | 0.0035 | 0.0043 | 0.0214 | 0.0263 | 0.0066 | 0.0086 | 0.5183 | 0.6908 |
| 0.15      | Speech | 0.0031 | 0.0003 | 0.0019 | 0.0021 | 0.0111 | 0.0129 | 0.0037 | 0.0045 | 0.2955 | 0.3821 |
| 0.20      | Speech | 0.0083 | 0.0018 | 0.0012 | 0.0011 | 0.0066 | 0.0071 | 0.0024 | 0.0027 | 0.1841 | 0.2014 |
| 0.25      | Speech | 0.0032 | 0.0011 | 0.0008 | 0.0007 | 0.0042 | 0.0044 | 0.0015 | 0.0017 | 0.1231 | 0.1274 |
| 0.30      | Silent | 0.0008 | 0.0007 | 0.0005 | 0.0004 | 0.0028 | 0.0028 | 0.0011 | 0.0011 | 0.0858 | 0.0809 |
| 0.35      | Silent | 0.0024 | 0.0012 | 0.0011 | 0.0008 | 0.0042 | 0.0044 | 0.0015 | 0.0017 | 0.1231 | 0.1274 |
Table 4
Specificity and Sensitivity values of Support Vector Machine (SVM) and Gaussian Membership Function (GMF) implementation of PEAS speech steganalysis for five thresholds.

| Classifier       | Threshold | Specificity (%) | Sensitivity (%) |
|------------------|-----------|-----------------|-----------------|
|                  | 0         | 12.5            | 25              | 37.5          | 50            |
| SVM - linear     | 0.10      | 35.59           | 86.13           | 81.13         | 100           | 100           |
|                  | 0.15      | 40              | 80.87           | 78.1          | 100           | 100           |
|                  | 0.20      | 35.71           | 77.47           | 73.36         | 100           | 100           |
|                  | 0.25      | 28.07           | 81.64           | 80.78         | 100           | 99.545        |
|                  | 0.30      | 32.69           | 81.27           | 73.87         | 100           | 99.547        |
| SVM – polynomial (Scale) | 0.10 | 53.62           | 93.18           | 16.45         | 100           | 100           |
|                  | 0.15      | 61.21           | 90.41           | 11.73         | 100           | 100           |
|                  | 0.20      | 8.4             | 95.28           | 95.83         | 100           | 100           |
|                  | 0.25      | 57.52           | 90.25           | 13.8          | 100           | 100           |
|                  | 0.30      | 11.01           | 93.8            | 96.31         | 100           | 100           |
| SVM – Gaussian (Auto) | 0.10 | 30.61           | 89.96           | 84.33         | 100           | 100           |
|                  | 0.15      | 26.79           | 89.4            | 84.03         | 100           | 100           |
|                  | 0.20      | 28              | 84.82           | 82.85         | 100           | 100           |
|                  | 0.25      | 29.7            | 90.78           | 85.31         | 100           | 100           |
|                  | 0.30      | 29.95           | 86.91           | 84.87         | 100           | 100           |
| SVM – Gaussian (Scale) | 0.10 | 30.48           | 85.48           | 84.72         | 100           | 100           |
|                  | 0.15      | 27.83           | 86.8            | 85.32         | 100           | 100           |
|                  | 0.20      | 26.38           | 88.49           | 85.34         | 100           | 99.54         |
|                  | 0.25      | 29.2            | 90.23           | 85.91         | 100           | 100           |
|                  | 0.30      | 26.4            | 92.29           | 87.09         | 100           | 100           |
| GMF              | 0.10      | 71.32           | 62.96           | 67.92         | 91.03         | 99.46         |
|                  | 0.15      | 81.2            | 78.36           | 81.4          | 93.74         | 99.82         |
|                  | 0.20      | 66.3            | 65.94           | 61.11         | 89.96         | 100           |
|                  | 0.25      | 66.3            | 65.94           | 61.11         | 89.96         | 100           |
|                  | 0.30      | 62              | 67.2            | 61.46         | 90.86         | 100           |

Table 5
Comparison table.

| Name                | Type   | Classifier | No. of analysis dimensions | Detection rate |
|---------------------|--------|------------|----------------------------|----------------|
|                     |        |            |                            | Specificity (%) | Sensitivity (%) |
|                     |        |            |                            | 0              | 12.5        | 25     | 37.5   | 50    | 100   |
| Kocal et al. [12]   | Speech | SVM        | 22                         | –              | –           | –      | –      | 84    | 91    |
| Miao et al. [29]    | Speech | SVM        | 2                          | –              | –           | 85     | 95     | –     | –     |
| PEAS steganalysis   | Speech | GMF        | 1                          | 81.2           | 78.36       | 81.4   | 93.74  | 99.82 | –     |
| Paulin et al. [27]  | Audio  | DBN        | 25 coefficients + 256 Gaussians | –          | –           | –      | –      | 82.25 | –     |
| Jayasree and Amritha [30] | VoIP | SVM        | 2                          | –              | 55          | 73     | –      | –     | –     |
| Tian et al. [38]    | VoIP   | SVM        | 3                          | –              | –           | –      | –      | 92    | –     |
Considering this range, even the highest SVM-based Specificity value is not good enough for developing a reliable and applicable steganalyzer. When the Gaussian membership function is used as a classifier, it does not outperform SVM-based implementations in terms of Sensitivity; however, in terms of the Specificity, there exists a huge gap between them. For a threshold of 0.15, the Gaussian membership function has a very reliable Sensitivity; meanwhile it has a Specificity of 81.2%. Therefore, this classifier is chosen as the PEAS steganalysis detection engine, and the most efficient threshold is defined as 0.15.

6. Discussion

To measure in which aspects PEAS steganalysis outperforms the related works, these algorithms are compared in terms of specificity, sensitivity, and the number of analysis dimensions in Table 5. This is notable that some related works have not been evaluated exactly in terms of identical embedding ratios, and thus, for some embedding ratios, they either have no sensitivity value or the real ratio is slightly different.

When the algorithms are compared according to the processing load, PEAS yields the best performance as it has only one analysis dimension. The works in Refs. [12,27] involve a large number of dimensions, and the dimensions reported by Tian et al. [28], Miao et al. [29], and Jayasree and Amritha [30] are at least twice the dimensions of PEAS. Note that none of the compared works reported a specificity value. Referring to Table 4, on ignoring this criterion, the implementation of the PEAS algorithm based on an SVM polynomial kernel is enhanced to an average of 97.702% with a starting value of 95.28% for an embedding ratio of 12.5%. However, there is a tradeoff with Specificity, which decreases to 8.4%; in other words, a large number of false alarms are produced.

According to the Sensitivity results reported in Table 5, PEAS outperforms [30] by a distance of more than 23% for the lowest embedding ratio of 12.5%. No other work has reported a sensitivity for this ratio.

For an embedding ratio of 25% [29], reported a higher sensitivity, and for a ratio of 37.5%, none of the other researches listed in Table 5 has given any value. Similar to the lowest ratio, for 50% embedding, PEAS outstandingly outperforms all the related works with a sensitivity value of 99.82%. This value is even higher than the sensitivities reported in Refs. [12,27] and [28] for full-capacity embedding. In conclusion, the comparisons show that, PEAS outperforms the related works for most of the embedding ratios.

7. Conclusion

This paper proposes a novel speech steganalysis algorithm based on PEAS for detecting hidden secrets in speech signals of healthcare IoT communications. The algorithm can be expressed in four general steps: normalizing the power of the given signal, allocating the signals to speech or silence classes, extracting the PEAS feature from both speech and silence signals, and finally classifying the given speech instance as clean or stego by using a Gaussian Membership Function (GMF) classifier. The algorithm was evaluated in terms of Specificity and Sensitivity. The results reported 81.2% for Specificity and 78.36%, 81.4%, 93.74%, and 99.82% Sensitivity for embedding ratios of 12.5%, 25%, 37.5%, and 50%, respectively. In terms of Specificity, PEAS steganalysis outperforms the compared related works at three embedding ratios of 12.5%, 37.5%, and 50% with a single analysis dimension. In contrast, in the compared works, number of the analysis dimensions varies between 2 and 281.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] H. Ghaseimzadeh, M.H. Kayzanrad, Comprehensive review of audio steganalysis methods, IET Signal Process. 12 (6) (2018) 673–687.
[2] J. Shayan, S.M. Abdullah, S. Karimizadeh, An overview of objectionable image detection, in: 2015 International Symposium on Technology Management and Emerging Technologies (ISTMET), IEEE, 2015, pp. 396–400.
[3] S.S. Chaeikar, A. Ahmad, Ensemble wavelet image steganalysis: a low dimension method for lsb detection, Signal Process. Image Commun. 70 (2019) 233–245.
[4] S.S. Chaeikar, A. Ahmad, Sw: a blind lsb image steganalysis technique, in: Proceedings of the 10th International Conference on Computer Modeling and Simulation, 2018, pp. 14–18.
[5] W. Duan, R. Nasiri, S. Karimizadeh, Smart city concepts and dimensions, in: Proceedings of the 2019 7th International Conference on Information Technology: IoT and Smart City, 2019, pp. 488–492.
[6] L. Yang, Z. Wang, S. Gao, Pipeline magnetic flux leakage image detection algorithm based on multiscale ssd network, IEEE Trans. Ind. Inf. 16 (1) (2019) 501–509.
[7] L. Yang, Z. Wang, S. Gao, Pipeline magnetic flux leakage image detection algorithm based on multiscale ssd network, IEEE Trans. Ind. Inf. 16 (1) (2019) 501–509.
[8] S.S. Chaeikar, M. Zaman, A.B.A. Manaf, A.M. Zeki, Psw statistical lsb image steganalysis, Multimed. Tool. Appl. 77 (11) (2018) 805–835.
[9] S. Yazdanpanah, M. Khayyamish, M. Mostof, Lshr speech steganalysis based on percent of equal adjacent samples, J. Circ. Syst. Comput. 31 (6) (2022) 2250118.
[10] M. Hansaballah, M. A. Hameed, A. I. Awad, K. Muhammad, A novel image steganography method for industrial internet of things security, IEEE Trans. Ind. Eng. 99 (2021).
[11] X.-M. Hu, H.-J. Zhang, X. Huang, Steganalysis of audio: attacking the steghide, in: 2005 International Conference on Machine Learning and Cybernetics, 7, IEEE, 2005, pp. 3937–3942.
[12] O.H. Kocal, E. Yürüklü, E. Dilarıveroğlu, Speech steganalysis based on the delay vector variance method, Turk. J. Electr. Eng. Comput. Sci. 24 (5) (2016) 4129–4141.
[13] C. Han, R. Xue, R. Zhang, X. Wang, A new audio steganalysis method based on linear prediction, Multimed. Tool. Appl. 77 (12) (2018) 15431–15455.
[14] A.B.A. Manaf, M. Zaman, R.B. Ahmad, F. Jaryani, H. Taherdoost, S. Shojae Chaeikar, H.R. Zeidanloo, Genetic audio steganography, Int. J. Recent Trends Eng. 1 (1) (2009)1-3.
[15] M. Zaman, A.B.A. Manaf, S.M. Abdullah, S. Shojae Chaeikar, Correlation between Poor and Bit Per Sample Rate in Audio Steganography,International conference on telecommunication and information;International conference on signal processing, Janpan, 2021,pp.163-168.
[16] Q. Liu, A.H. Sung, M. Qiao, Temporal derivative-based spectrum and mel-cepstrum audio steganalysis, IEEE Trans. Inf. Forensics Secur. 4 (3) (2009) 359–368.
[17] Y. Tint, K.T. Myn, Audio steganalysis using features extraction and classification, international journal of research & reviews in computer science, Science, 3 (2)(2012)192–195.
[18] H. Ghaseimzadeh, M.K. Ajrami, Reversed-mel cepstrum based audio steganosis, in: 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE), IEEE, 2014, pp. 679–684.
[19] X. Yu, R. Wang, D. Yan, J. Zhu, Mp3 audio steganalysis using calibrated side information feature, J. Comput. Inf. Syst. 8 (10) (2012) 4241–4248.
[20] D. Yan, R. Wang, X. Yu, J. Zhu, Steganalysis for mp3/stego using differential statistics of quantization step, Digit. Signal Process. 23 (4) (2013) 1181–1185.
[21] Y. Ren, T. Cai, M. Tang, L. Wang, Amr steganalysis based on the probability of same pulse position, IEEE Trans. Inf. Forensics Secur. 10 (9) (2015) 1801–1811.
[22] H. Miao, L. Huang, Z. Chen, W. Yang, A. Al-Hawbani, A new scheme for covert communication via 3g encoded speech, Comput. Electr. Eng. 38 (6) (2012) 1490–1501.
[23] B. Geiser, P. Vary, High rate data hiding in acelp speech codecs, in: 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2008, pp. 4085–4088.
[24] Y. Ren, Q. Xiong, L. Wang, Steganalysis of aac using calibrated markov model of adjacent codebook, in: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2016, pp. 679–684.
[25] H. Ozer, I. Avcioba, B. Sankur, N.D. Memon, Steganalysis of audio based on audio quality metrics, in: Security and Watermarking of Multimedia Contents V., 5020, International Society for Optics and Photonics, 2003, pp. 55–66.
[26] C. Kraetzner, J. Dittmann, Mel-cepstrum based steganalysis for voip steganography, in: Security, Steganography, and Watermarking of Multimedia Contents IX, 6505, International Society for Optics and Photonics, 2007, 650505.
[27] C. Paulin, S.A. Selouani, E. Hervet, Audio steganalysis using deep belief networks, Int. J. Speech Technol. 19 (3) (2016) 585–591.
[28] H. Tian, Y. Wu, Y. Huang, J. Liu, Y. Chen, T. Wang, Y. Cai, Steganalysis of low bitrate speech based on statistic characteristics of pulse positions, in: 2015 10th International Conference on Availability, Reliability and Security, IEEE, 2015, pp. 455–460.
[29] H. Tian, L. Huang, Y. Shen, X. Lu, Z. Chen, Steganalysis of compressed speech based on markov and entropy, in: International Workshop on Digital Watermarking, Springer, 2013, pp. 65–76.
[30] N. Jayasree, A. Amritha, A model for the effective steganalysis of voip, in: Artificial Intelligence and Evolutionary Algorithms in Engineering Systems, Springer, 2015, pp. 379–387.