

dr inż. Rafał Korycki

ekspert z zakresu inżynierii dźwięku i obrazu Biura Badań Kryminalistycznych Agencji Bezpieczeństwa Wewnętrznego
r.korycki@abw.gov.pl

Identyfikacja koderów MP3 oraz urządzeń rejestrujących na podstawie badania plików fonicznych poddanych stratnej kompresji¹

Streszczenie

W pracy nakreślono problem identyfikacji koderów MP3 oraz urządzeń rejestrujących na podstawie analizy nagrań fonicznych poddanych stratnej kompresji. Zaproponowana metoda może być wykorzystana jako wsparcie dla innych rozwiązań służących do wykrywania podwójnej kompresji oraz detekcji nieciągłości. Prezentowane podejście polega na statystycznej analizie zmiennych pozyskanych bezpośrednio ze strumienia danych MP3 i stanowiących nieodłączne parametry kompresji. Wyznaczone wektory składające się z 46 cech zostały użyte jako sekwencje treningowe liniowej analizy dyskryminacyjnej (LDA), jednego z najpopularniejszych algorytmów uczenia maszynowego z nadzorem. Skuteczność metody identyfikacji koderów MP3 oraz urządzeń rejestrujących została przetestowana na specjalnie przygotowanej w tym celu bazie nagrań muzycznych składającej się z blisko miliona plików MP3. Wyniki badań zostały omówione w kontekście wpływu parametrów kompresji na możliwość detekcji fałszerstw w cyfrowych nagraniach fonicznych.

Słowa kluczowe badanie autentyczności nagrań cyfrowych, identyfikacja koderów MP3, identyfikacja urządzeń rejestrujących, analiza dowodów cyfrowych, liniowa analiza dyskryminacyjna

Wstęp i cel pracy

Badanie autentyczności nagrań fonicznych polega na ocenie oryginalności analizowanego zapisu oraz na wykrywaniu śladów ingerencji w jego ciągłość. Zgodnie z przesłaniem orzeczenia Sądu Najwyższego w sprawie o sygn. akt III K 49/61 z 10 marca 1961 [16] zadaniem biegłych wykonujących opinie na zlecenie organów wymiaru sprawiedliwości i organów ścigania jest ustalenie, czy badane nagranie jest oryginalne (swoiste, identyczne), jak również ujawnienie ewentualnych śladów ingerencji w jego ciągłość. Niezwykle ważna jest też definicja nagrania autentycznego według standardu AES27-1996 opracowanego przez międzynarodową organizację AES (Audio Engineering Society) [1]. Wskazano w niej na konieczność badania oryginalności zapisów oraz wykrywania śladów wykonanego montażu, zaznaczając przy tym potrzebę analizy zgodności technik rejestracji z zastosowanymi podczas wytwarzania nagrania. Ponadto zwrócono

uwagę na integralność zapisu z całokształtem zdarzeń akustycznych towarzyszących nagraniu.

W publikacjach [9] i [10], w których opisano wyniki badań realizowanych w ramach projektu badawczego, autor przedstawił opracowany zestaw algorytmów umożliwiających m.in. odróżnianie nagrań poddanych pojedynczej kompresji (np. plików utrwalonych za pomocą dyktafonów cyfrowych) od nagrań kodowanych ponownie, które mogły zostać zmodyfikowane przy użyciu oprogramowania do edycji dźwięku. Ponadto autor zaproponował metodę identyfikacji nieciągłości w nagraniach poddanych stratnej kompresji MP3, która umożliwia wykrywanie procesów kasowania i wstawiania fragmentów zapisu. Do badań wykorzystano wektor cech wyznaczany na podstawie immanentnych parametrów plików MP3 oraz wybrane metody uczenia maszynowego z nadzorem. Istotnym wnioskiem z powyższych badań sformułowanym w ww. publikacjach było ustalenie, że do prawidłowej realizacji procesów predykcji pierwotnej przepływności bitowej,

¹ Badania prowadzone w ramach projektu pt.: „Projektowanie badań empirycznych i analizy materiałów dotyczących specyfiki metod kryminalistyki w pracy służb specjalnych służb porządku publicznego” finansowanego przez Narodowe Centrum Badań i Rozwoju w formie grantu nr 0023/R/ID3/2012/02.

Name	Encoder	Bitrate	Length	Size	Bad Last Frame	Qu...	Frames	Profile
COLDPLAY_10_07.mp3	Blade	128	00:00:09.8	160,074	yes		377	--
COLDPLAY_10_08.mp3	Xing (old)	128	00:00:09.8	160,070	yes		377	--
COLDPLAY_10_09.mp3	Xing (old)	128	00:00:09.8	160,077	yes		377	--
COLDPLAY_10_10.mp3	Lame (old) or m3e	128	00:00:09.8	160,071	yes		377	--
CRANBERRIES_01_01.mp3	Blade	128	00:00:09.8	160,070	yes		377	--
CRANBERRIES_01_02.mp3	Blade	128	00:00:09.8	160,071	yes		377	--
CRANBERRIES_01_03.mp3	Lame (old) or m3e	128	00:00:09.8	160,077	yes		377	--
CRANBERRIES_01_04.mp3	Xing (old)	128	00:00:09.8	160,076	yes		377	--
CRANBERRIES_01_05.mp3	Blade	128	00:00:09.8	160,071	yes		377	--
CRANBERRIES_01_06.mp3	Blade	128	00:00:09.8	160,071	yes		377	--
CRANBERRIES_01_07.mp3	Blade	128	00:00:09.8	160,069	yes		377	--
CRANBERRIES_01_08.mp3	Blade	128	00:00:09.8	160,073	yes		377	--
CRANBERRIES_01_09.mp3	Blade	128	00:00:09.8	160,077	yes		377	--
CRANBERRIES_01_10.mp3	Blade	128	00:00:09.8	160,070	yes		377	--
CRANBERRIES_02_01.mp3	Xing (old)	128	00:00:09.8	160,078	yes		377	--
CRANBERRIES_02_02.mp3	Blade	128	00:00:09.8	160,075	yes		377	--
CRANBERRIES_02_03.mp3	Blade	128	00:00:09.8	160,064	yes		377	--
CRANBERRIES_02_04.mp3	Blade	128	00:00:09.8	160,074	yes		377	--
CRANBERRIES_02_05.mp3	Xing (old)	128	00:00:09.8	160,065	yes		377	--
CRANBERRIES_02_06.mp3	Xing (old)	128	00:00:09.8	160,073	yes		377	--
CRANBERRIES_02_07.mp3	Xing (old)	128	00:00:09.8	160,068	yes		377	--
CRANBERRIES_02_08.mp3	Blade	128	00:00:09.8	160,077	yes		377	--
CRANBERRIES_02_09.mp3	Blade	128	00:00:09.8	160,070	yes		377	--
CRANBERRIES_02_10.mp3	Blade	128	00:00:09.8	160,075	yes		377	--
CRANBERRIES_05_01.mp3	Blade	128	00:00:09.8	160,068	yes		377	--
CRANBERRIES_05_02.mp3	Blade	128	00:00:09.8	160,077	yes		377	--
CRANBERRIES_05_03.mp3	Blade	128	00:00:09.8	160,072	yes		377	--

Ryc. 1. Interfejs użytkownika programu EncSpot z raportem z klasyfikacji plików MP3.

detekcji wielokrotnej kompresji oraz wykrywania montażu w nagraniach poddanych wielokrotnym procesom kodowania i dekodowania konieczna jest wiedza na temat kodera, który został użyty do stworzenia badanego pliku.

W wielu publikacjach autorzy zwracali już uwagę na możliwość pozyskiwania informacji o zastosowanym algorytmie kompresji na podstawie analizy pliku poddanego kompresji. Wybrane prace dotyczyły m.in. określania kilku wybranych parametrów kodowania, np. współczynników skalowania i kwantyzacji [5], [7], oraz identyfikacji różnych algorytmów kompresji (np. MP3, AAC, OGG, WMA, AMR) [2], [6]. Na rynku dostępne są również aplikacje komercyjne (np. EncSpot [4]) służące do rozpoznawania rodzaju użytego kodera MP3 na podstawie analizy plików poddanych kompresji. Jednakowoż warto nadmienić, że programy te bazują w dużej mierze na informacjach znajdujących się w nagłówkach oraz metadanych analizowanych plików (takich jak np. tagi ID3 [14]). Jakkolwiek pozyskane na tej podstawie informacje są bardzo szczegółowe, mogą być jednak dowolnie modyfikowane bez wpływu na parametry utraconych nagrań. Ponadto w przypadku niektórych implementacji algorytmu MP3 identyfikacja kodera jest obarczona dużym błędem. Przykładowo, podczas próby rozpoznania kodera SoloH aplikacja EncSpot wskazywała losowo jeden z trzech innych koderów: Blade, Xing oraz Lame. Na rycinie 1 przedstawiono interfejs użytkownika programu EncSpot z wynikiem przykładowej identyfikacji. Dopiero Böhme i Westfeld zaproponowali rozwiązanie służące do rozróżniania poszczególnych implementacji tego samego algorytmu kompresji [3]. Zastosowanie opisanej metody było jednak ograniczone do wyspecyfikowanej listy koderów przedstawionej

przez autorów ze względu na użycie do klasyfikacji jedynie dziesięciu cech symbolicznych przyjmujących wartości dyskretne.

Celem badań opisywanych w bieżącej publikacji było opracowanie metod służących do wykrywania rodzajów i wersji różnych implementacji algorytmu MP3 użytych do tworzenia badanych plików. Bazując na rozwiązaniach opisanych w wymienionych publikacjach, autor zaproponował własny zestaw cech, umożliwiający zarówno prawidłową detekcję kodera użytego do kompresji badanego nagrania, jak i szybką adaptację klasyfikatorów do nowych lub nieuwzględnionych w niniejszym opracowaniu implementacji algorytmu MP3. Aby sprawdzić skuteczność opracowanych metod, stworzono własną bazę danych nagrań fonicznych składającą się z blisko miliona plików MP3. W opinii autora, takie podejście sprzyja obiektywizacji prowadzonych badań oraz umożliwia łatwe porównywanie badanych metod. Ponadto wykorzystano wciąż popularne metody uczenia maszynowego, które mogą być przydatne w rozpoznawaniu i klasyfikacji cech różniących poszczególne kodery. Badania pokazały, że dzięki odpowiednio przygotowanym sekwencjom uczącym jest możliwe stworzenie algorytmu służącego do identyfikacji koderów MP3 oraz urządzeń rejestrujących na podstawie badania plików fonicznych poddanych stratnej kompresji.

Mając na uwadze coraz bardziej powszechne wykorzystanie przenośnych rejestratorów dźwięku, w których do zapisu plików stosuje się stratną kompresję, podczas prowadzonych badań dokonano analizy różnych algorytmów kodowania. Autor skoncentrował się na analizie nagrań poddanych stratnej kompresji z użyciem algorytmu MPEG-1 Layer 3 (MP3) [12]. Wybór ten został podyktowany bogatą literaturą

przedmiotu, dostępnością dokumentacji oraz powszechnym użyciem koderów MP3. Warto nadmienić, że zaprezentowane w niniejszej pracy metody można stosować również w odniesieniu do innych algorytmów kompresji (np. AAC – Advanced Audio Coding, Ogg Vorbis), w przypadku których standard nie definiuje w sposób precyzyjny i jednoznaczny każdej operacji wykonywanej przez algorytm.

Na potrzeby opisu przeprowadzonych badań autor postanowił wprowadzić następujące określenia: „nagrania źródłowe”, odnoszące się do fragmentów utworów muzycznych pochodzących z oryginalnych płyt CD, oraz „nagrania oryginalne” w odniesieniu do tychże fragmentów poddanych pojedynczej kompresji.

Badania były prowadzone w ramach projektu badawczego pt.: „Projektowanie badań empirycznych i analizy materiałów dotyczących specyfiki metod kryminalistyki w pracy służb specjalnych służb porządku publicznego” finansowanego przez Narodowe Centrum Badań i Rozwoju w formie grantu nr 0023/R/ID3/2012/02.

Wybrane zagadnienia związane z kompresją MP3

Algorytm MP3 opisany w dokumentacji do standardu ISO 11172-3 [12] określa metody wyznaczania dużej liczby zmiennych istotnych z punktu widzenia procesu kompresji i definiuje sposób zapisu sygnału fonicznego. Daje to gwarancję, że strumień binarny wytworzony przez koder będzie prawidłowo odczytywany oraz interpretowany. Jakkolwiek wspomniana dokumentacja jest niezwykle szczegółowa, to nie precyzuje każdego kroku wykonywanego przez program, pozostawiając twórcom aplikacji swobodę w implementacji poszczególnych bloków kodera. W efekcie algorytm MP3 może być realizowany w różny sposób, a mimo to pliki foniczne poddane kompresji i pochodzące z różnych źródeł będą odtwarzane przez dekoder [12], [15].

Strumień danych pliku MP3 składa się z ramek o stałej długości. W ramce znajdują się dwa granule (*granules*), z których każdy zawiera dane potrzebne do zdekodowania wartości współczynników MDCT dla jednego (nagrania monofoniczne) lub dwóch kanałów (nagrania stereofoniczne). Ramki są złożone z nagłówków (*header*), kontroli integralności pliku CRC (*cyclic redundancy code*), informacji pobocznych (*side information*) oraz danych głównych (*main data*). Nagłówek ma 32 bity długości i rozpoczyna go 12 bitów synchronizacji, dzięki czemu każda ramka jest indywidualnie lokalizowana przez dekoder. Takie rozwiązanie umożliwia strumieniową transmisję danych, a odbiorca może rozpocząć odsłuchiwanie ciągłego przekazu w dowolnym momencie jego trwania. Ponadto w nagłówku pliku zapisywane są m.in. informacje o wersji i warstwie standardu MPEG-1, przepływności (*bit rate*), częstotliwości próbkowania, trybie kodowania

sygnałów dwukanałowych, stosowaniu emfazy² czy ochronie integralności pliku (*protection bit*)³.

Informacje poboczne umieszczone w kolejnej sekcji ramki są niezbędne do prawidłowego odczytania danych głównych zawierających skwantowane i zakodowane wartości prążków widma. Sekcję rozpoczyna zmienna *main_data_end*, będąca wskaźnikiem do początku nagłówka kolejnej ramki (mierząc od końca sekcji zajmowanej przez dane główne ramki bieżącej). W ten sposób definiowany jest rozmiar rezerwuaru, którego zadaniem jest wirtualne zwiększenie pojemności ramki, wówczas gdy wytworzenie strumienia o mniejszej objętości wiązałoby się z przekroczeniem wartości progowej szumu kwantyzacji estymowanej na podstawie wartości entropii psychoakustycznej widma fragmentu sygnału, wyznaczanej w bloku modelu psychoakustycznego.

Informacje poboczne zawierają również bity prywatne (*private bits*) oraz zmienną *scfsi* (*scalefactor selection information*). Określa ona, czy ten sam zestaw parametrów definiowanych jako *scalefactors* i służących do skalowania wartości prążków widma w danym paśmie krytycznym jest wykorzystywany w obydwu granulach, czy też dla każdego z nich musi być przesyłany indywidualnie. Kolejny zestaw danych składających się na informacje poboczne jest przekazywany osobno dla każdego z granuli. Należą do nich m.in. zmienna *big_values*, która stanowi liczbę prążków widma, do kodowania których stosuje się więcej niż trzy poziomy kwantyzacji, bity wskazujące na użycie jednego z czterech predefiniowanych okien czasowych w drugim zespole filtrów, czy zmienne zapisane jako *table_select* będące numerami tablic Huffmana, których należy użyć do zdekodowania danych głównych zakodowanych w pliku MP3 z wykorzystaniem algorytmu kompresji Huffmana.

Ostatnia sekcja ramki składa się z wartości współczynników skalowania (*scalefactors*), danych binarnych zakodowanych z wykorzystaniem algorytmu kompresji Huffmana oraz informacji dodatkowych (*ancillary data*). Zadaniem współczynników skalowania jest wzmacnianie lub tłumienie prążków widma MDCT znajdujących się w zakresach definiowanych jako *scalefactor bands* w celu skutecznego

- 2 Tryb emfazy oznacza, że plik został poddany kompresji z wykorzystaniem jednego z dwóch trybów redukcji zakłóceń, polegających na wzmacnianiu dużych częstotliwości podczas kodowania i ich redukcji w czasie dekodowania. Podobne rozwiązania (np. Dolby-B™) są powszechnie stosowane w systemach analogowych.
- 3 Jeśli bit ochrony zostanie ustawiony, do każdej ramki dodawany jest 16 bitów CRC umożliwiających sprawdzenie poprawności 16 bitów nagłówka oraz 16 bitów informacji pobocznych. Są to dane istotne do prawidłowego odtworzenia zawartości ramki. W przypadku wystąpienia błędów ramka może zostać wyciszona lub zastąpiona ramką poprzednią.

maskowania szumu kwantyzacji na podstawie danych przekazanych przez model psychoakustyczny. Zakresy *scalefactor bands* są z kolei dobierane w taki sposób, aby możliwie najdokładniej odwzorować pasma krytyczne [12], [15].

Ekstrakcja wektora cech

Na podstawie analizy dokumentacji do standardu ISO/IEC 11172-3 [12] oraz wstępnych badań doświadczalnych został skonstruowany wektor 46 cech umożliwiający rozpoznanie wybranej implementacji algorytmu MP3, która posłużyła do stworzenia badanego pliku. Do budowy wektora użyto wartości wyodrębnionych z każdej z ramek i wyselekcjonowanych w taki sposób, aby osoba dokonująca edycji nagrania nie miała możliwości ich modyfikacji bez wprowadzenia dających się usłyszeć zniekształceń (np. podczas edycji strumienia danych binarnych pliku MP3 w edytorze). Za punkt odniesienia posłużyły wyniki badań uzyskane na podstawie zestawu dziesięciu cech symbolicznych przyjmujących wartości dyskretne [3]. Powyższe rozwiązanie, jakkolwiek umożliwia skuteczną klasyfikację w ramach zaproponowanej przez autorów listy kodeków, znacznie utrudnia poszerzenie istniejącego zestawu o nowe implementacje algorytmu MP3. W związku z tym, tworząc nowy wektor cech o długości 46 elementów, autor użył również zmiennych ciągłych oraz rozszerzył katalog parametrów wyodrębnianych z plików MP3. Dzięki temu dodanie nowego kodeka do listy rozpoznawanych wymaga jedynie wygenerowania odpowiedniej sekwencji treningowej dla algorytmów uczenia maszynowego.

Ze względu na różnice pomiędzy implementacjami algorytmu kompresji MP3 rzeczywiste wartości przepływności bitowej różnią się od ich nominalnych odpowiedników⁴ zapisanych w nagłówkach każdej z ramek. Ma to związek ze sposobem wyznaczania parametrów kontrolnych definiowanych dla zagnieżdżonych pętli kwantyzacji i kompresji, w ramach których algorytm znajduje kompromis między rozmiarem pliku a poziomem wprowadzanych zniekształceń. Powyższą zależność wykorzystano do stworzenia pierwszej cechy, którą wyznaczono według zależności:

$$F1 = \frac{(f_{size} - h - m) \cdot f_s}{1152 \cdot \sum_{i=1}^p \beta_{nom}^{(i)}}, \quad (1)$$

gdzie f_{size} oznacza rozmiar pliku, h jest wielkością nagłówka, m oznacza metadane zapisane w pliku, f_s jest częstotliwością próbkowania, p stanowi liczbę ramek w pliku, natomiast $\beta_{nom}^{(i)}$ jest wartością przepływności nominalnej dla i -tej ramki.

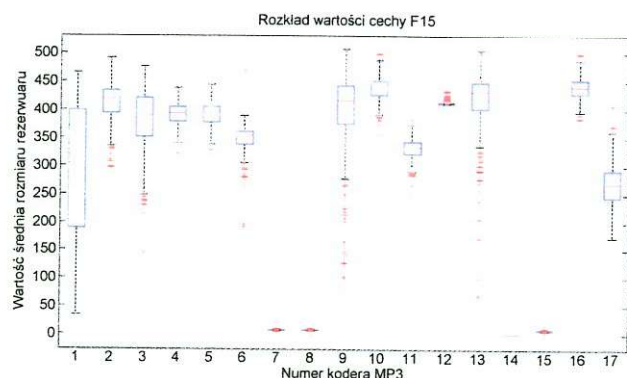
4 Standaryzowane wartości przepływności bitowej dla plików MP3 wynoszą odpowiednio: 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256, 320 kbps.

W trakcie wykonywania algorytmu kompresji rozmiar granuli jest dobierany indywidualnie w taki sposób, aby spełnić kryterium dopasowania do długości ramki przy najniższym możliwym poziomie zniekształceń. Zależnie od implementacji pierwszy granul w każdej ramce może być większy lub mniejszy od drugiego, w związku z tym kolejne trzy cechy F2:F4 określają liczbę ramek, w których granul pierwszy jest większy od drugiego, wyznaczoną dla obydwu kanałów oraz oddzielnie dla kanału lewego i prawego. Kolejnych osiem elementów wektora cech F5:F12 przyjmuje wartości dyskretne determinowane wyborem jednej z technik stosowanych podczas zapisu nagrań dwukanałowych. Sposób przetwarzania danych pochodzących z poszczególnych kanałów jest zapisany w nagłówku każdej ramki. Warto nadmienić, że w niektórych kodekach MP3, np. 8Hz, Blade, Shine, nie stosuje się trybu *joint stereo*⁵.

Kolejna grupa cech jest wyznaczana na podstawie zmiennej *main_data_end* określającej rozmiar rezerwuaru. Sterowanie powyższym mechanizmem jest zależne w dużej mierze od programisty tworzącego kodek. Zatem szybkość „zapełniania” rezerwuaru w przypadku pojawienia się sygnału fonicznego może być zależna od rodzaju oraz wersji implementacji algorytmu MP3. Przykładowo, kodeki 8Hz, Blade oraz Shine, dzielą między ramkami dużo mniej danych niż kodeki z rodziny Lame. Mając na uwadze powyższe rozważania, zdefiniowano cechy F13:F14 określające liczbę bajtów dzielonych między pierwszymi dwiema parami ramek oraz cechy F15:F18 stanowiące odpowiednio wartość średnią i odchylenie standardowe rozmiaru rezerwuaru oraz jego maksymalną i minimalną pojemność. Na rycinie 2 przedstawiono rozkłady wartości cechy F15 (wartość średnia rozmiaru rezerwuaru) dla pierwszych siedemnastu kodeków wskazanych w publikacjach [9] i [10] wygenerowane za pomocą funkcji *boxplot* środowiska obliczeniowego MATLAB [11]. Powyższa funkcja tworzy zestaw wykresów, w których wskaźnik centralny (pozioma czerwona linia) jest medianą, krawędzie pola są 25. oraz 75. centylem, wypustki określają wartości maksymalne, natomiast wartości o znacznych odstępach (*outliers*) zostały oznaczone czerwonymi krzyżami.

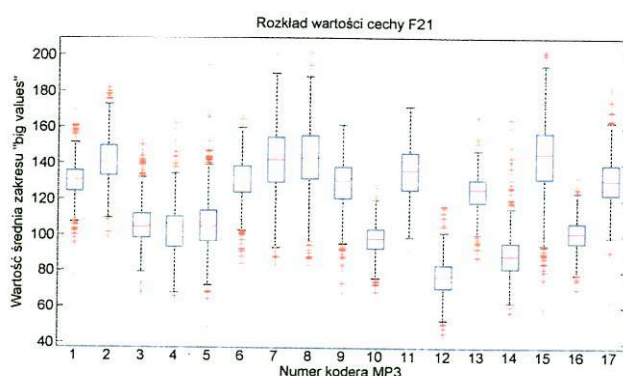
Wartości prążków widma dla poszczególnych granuli są grupowane w pasma, a następnie kodowane z wykorzystaniem różnych tablic Huffmana. Podział ten jest realizowany ze względu na maksymalne wartości MDCT, zakładając przy tym, że w zakresie dużych częstotliwości występują skwantowane współczynniki widma o małych amplitudach, z których część nie musi być w ogóle zapisywana. Najbardziej znaczący zakres częstotliwości jest określany jako *big_values*

5 Tryb *joint stereo* polega na oddzielnym kodowaniu części wspólnej wyznaczonej dla obydwu kanałów oraz różnicy między nimi. Występuje w dwóch wersjach: *intensity* oraz *mid/side*.



Ryc. 2. Rozkłady wartości cechy F15 dla pierwszych 17 koderów MP3 wskazanych w [9] i [10].

i odnosi się do prążków reprezentowanych przez więcej niż trzy poziomy kwantyzacji. Ich najmniejsze oraz największe wartości, jak również wartość średnią i odchylenie standardowe opisują cechy F19:F22. Na rycinie 3 przedstawiono rozkłady wartości cechy F21 (wartość średnia szerokości obszaru *big_values*) dla pierwszych siedemnastu koderów wskazanych w publikacjach [9] i [10] wygenerowane za pomocą funkcji *boxplot*.



Ryc. 3. Rozkłady wartości cechy F21 dla pierwszych 17 koderów MP3 wskazanych w [9] i [10].

Z kolei zmienne F23:F24 są determinowane entropią histogramu wyznaczanego dla każdego z kanałów na podstawie liczby wystąpień pierwszych 60 współczynników MDCT [3]:

$$\begin{aligned} F23 &= -\sum_{j=1}^{60} d_j^{(L)} \log_2 d_j^{(L)} \\ F24 &= -\sum_{j=1}^{60} d_j^{(R)} \log_2 d_j^{(R)} \end{aligned} \quad (2)$$

gdzie $d_j^{(L)}$ oraz $d_j^{(R)}$ oznaczają częstotliwość występowania j -tego prążka widma MDCT odpowiednio dla lewego („L”) i prawego („R”) kanału.

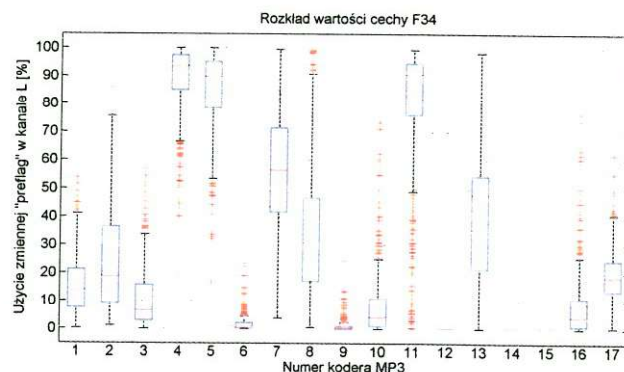
Zakres określany jako *big_values* jest dzielony na trzy obszary za pomocą dwóch zmiennych *region_address1* oraz *region_address0* określających liczbę zajmowanych przez nie pasm krytycznych (*scalefactor bands*). Ich wartości minimalne oraz maksymalne

znalazły odzwierciedlenie w cechach F25:F28. Ponadto w zależności od rozkładu energii widma sygnału fonicznego do kodowania wykorzystywane są różne tablice Huffmana. Jakkolwiek dopuszczalne jest stosowanie 32 tablic, to część z nich pozostaje nieużywana w niektórych koderach. W związku z tym kolejnym pięciu cechom F29:F33 przypisano odpowiednio użycie tablicy 1 oraz tablic 9:12 w pierwszym obszarze *big_values*, tablicy 0 we wszystkich trzech obszarach oraz tablicy 23 w regionie drugim i trzecim.

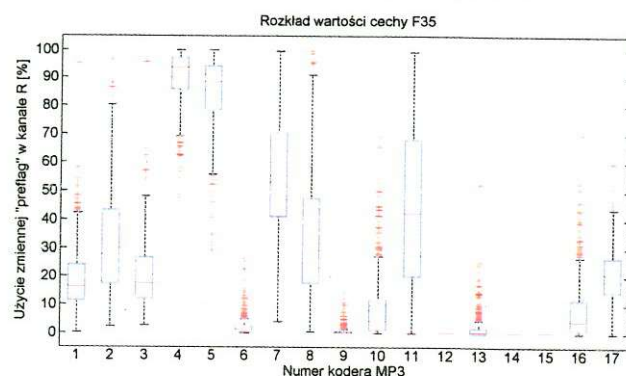
W specyfikacji ISO 11172-3 [12] zapisano, że stosowanie emfazy polegającej na wzmacnianiu dużych częstotliwości podczas kodowania i zmniejszaniu ich poziomu podczas dekodowania w celu redukcji zakłóceń jest zależne od woli programisty tworzącego koder. W związku z tym kolejne dwie cechy zostały wyznaczone na podstawie zmiennej *preflag* opisującej użycie tej funkcji w każdej z ramek:

$$\begin{aligned} F34 &= \frac{1}{p} \sum_{i=1}^p \text{preflag}(b_i^{(L)}) \\ F35 &= \frac{1}{p} \sum_{i=1}^p \text{preflag}(b_i^{(R)}) \end{aligned} \quad (3)$$

gdzie p oznacza liczbę ramek w pliku, natomiast $b_i^{(L)}$ oraz $b_i^{(R)}$ stanowią odpowiednio bloki danych dla kanału lewego i prawego. Na rycinach 4 i 5 przedstawiono procentowe rozkłady wartości cech F34 oraz F35 dla pierwszych 17 koderów wskazanych w publikacjach [9] i [10] wygenerowane za pomocą funkcji *boxplot*.



Ryc. 4. Rozkład wartości cechy F34 dla pierwszych 17 koderów MP3 wskazanych w [9] i [10].

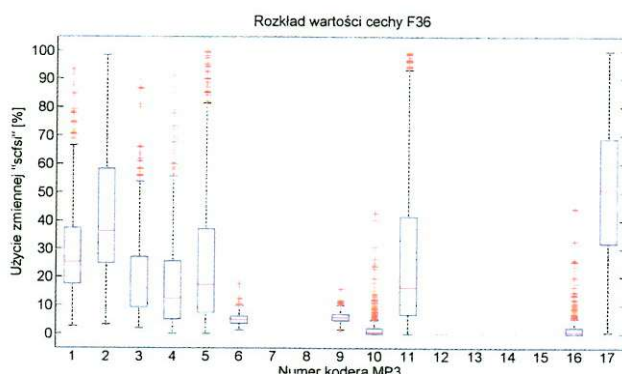


Ryc. 5. Rozkład wartości cechy F35 dla pierwszych 17 koderów MP3 wskazanych w [9] i [10].

Następna badana zmienna (*scfsi*) jest również opcjonalna i pozwala określić, w których ramach ten sam zestaw parametrów definiowanych jako *scalefactors* i służących do skalowania wartości prążków widma w danym paśmie krytycznym jest wykorzystywany w obydwu granulach, a w których dla każdego z nich musi być przesyłany indywidualnie. Zatem kolejny element składowy wektora cech jest wyznaczany w sposób następujący:

$$F36 = \frac{1}{p} \sum_{i=1}^p scfsi(b_i), \quad (4)$$

gdzie p oznacza liczbę ramek w pliku, natomiast b_i stanowią bloki danych. Na rycinie 6 przedstawiono procentowe rozkłady wartości cechy F36 dla pierwszych 17 koderów wskazanych w publikacjach [9] i [10] wygenerowane za pomocą funkcji *boxplot*.



Ryc. 6. Rozkład wartości cechy F36 dla pierwszych 17 koderów MP3 wskazanych w [9] i [10].

Długość ramek oraz ich liczba przypadająca na jedną sekundę nie zawsze są tożsame z odpowiadającą im liczbą próbek wejściowego sygnału fonicznego. Konieczne jest zatem dodanie, najczęściej co kilka ramek, stosownego okna (*slot*), które ma na celu jej wydłużenie. Taka operacja, określana jako wypełnianie (*padding*), jest zależna od implementacji, a szczegóły jej użycia mogą posłużyć do identyfikacji rodzaju oraz wersji implementacji algorytmu MP3. Przykładowo, dla częstotliwości próbkowania 44100 Hz i przepływności 128 kbps na początku plików poddanych kompresji za pomocą koderów Blade oraz iTunes występują kolejno 24 ramki i 25 ramek wypełnionych przedzielonych jedną ramką bez tzw. *slotu*. Z kolei w koderze Helix powyższa sekwencja jest poprzedzona pojedynczą ramką bez wypełnienia. Występują również kodery (np. Adobe CS6), w których wszystkie ramki mają tę samą długość. W związku z tym w kolejnych sześciu cechach F37:F42 zawarto odpowiednio: minimalną i maksymalną długość ramek, rozmiar pierwszej ramki oraz liczbę wypełnionych ramek w pierwszych trzech grupach wspomnianej sekwencji.

Algorytm MP3 przewiduje stosowanie długich oraz krótkich okien czasowych w drugim zespole filtrów

przed wyznaczeniem transformacji MDCT, czego efektem jest wyznaczenie odpowiednio jednej realizacji liczącej 576 prążków widma lub trzech realizacji po 192 prążki widma. Dzięki temu rozwiązaniu dla lepszej reprezentacji sygnałów tonalnych stosuje się bloki o większej liczbie próbek, natomiast okna krótkie służą do zapewnienia dobrego odwzorowania sygnałów szybko zmieniających się w czasie. Aby przełączanie między wymienionymi rodzajami analizy następowało płynnie i nie powodowało zniekształceń, stosuje się okna przejściowe (start i stop). Jednakże w wielu wersjach koderów (np. Xing) krótkie bloki w ogóle nie są stosowane. Z kolei w implementacjach takich jak np. Plugger mogą się zdarzać pewne niedozwolone przejścia, np. bezpośrednio między blokami długimi i krótkimi. Powyższe obserwacje znalazły zastosowanie podczas tworzenia kolejnej cechy [3]:

$$F43 = \begin{cases} 1 & \text{dla } typ(b_i) = 0 \text{ dla wszystkich } i \\ 2 & \text{dla } typ(b_i) = 0 \wedge typ(b_{i+1}) = 2 \\ 3 & \text{dla } typ(b_i) = 2 \wedge typ(b_{i+1}) = 3 \\ 4 & \text{dla } |\{b_i | typ(b_i) = 2\}| = \\ & \quad = |\{b_i | typ(b_i) = 3\}| = 1 \\ 5 & \text{w pozostałych przypadkach} \end{cases}, \quad (5)$$

gdzie 0 i 2 oznaczają odpowiednio okna długie oraz krótkie, natomiast 1 i 3 odnoszą się do okien przejściowych.

Niezależnie od sposobu wykorzystania rezerwuaru podczas wykonywania algorytmu kompresji MP3 mogą pojawić się w ramce miejsca niezapełnione danymi. Specyfikacja ISO 11172-3 [12] nie precyzuje wartości bajtów, które powinny służyć do wypełniania tych pustych przestrzeni. Programiści mają zatem swobodę wyboru powyższych wartości (tzw. *stuffing bits*). Przykładowo, w plikach poddanych kompresji za pomocą koderów Gogo można odnaleźć sekwencję 47 i 4F zapisaną w kodzie heksadecymalnym, co tłumacząc na kod ASCII oznacza „GO”. Wobec tego kolejna zmienna wektora cech została wykorzystana do opisu powyższych danych:

$$F44 = \begin{cases} 1 & \text{dla wypełnienia zerami} \\ 2 & \text{dla wypełnienia jedynekami} \\ 3 & \text{dla wypełnienia sekwencją} \\ & \quad 0x47 \text{ i } 0x4F \\ 4 & \text{dla wypełnienia sekwencją} \\ & \quad 0xAA \text{ lub } 0x55 \\ 5 & \text{dla braku wypełnienia} \\ 6 & \text{w pozostałych przypadkach} \end{cases}. \quad (6)$$

Ostatnie dwie cechy F45:F46 przyjmują wartości binarne i wskazują odpowiednio wadliwą ostatnią ramkę oraz błąd synchronizacji, które najczęściej wynikają z niewłaściwego wyznaczania końca pliku oraz początku kolejnego nagłówka.

Klasyfikacja koderów MP3

Na podstawie analizy rozkładu wartości pięciu cech (ryc. 2–6) można zaobserwować istotne różnice w przyjmowanych przez nie wartościach w zależności od zastosowanego koderu. Jednak aby dokonać precyzyjnej identyfikacji użytej wersji algorytmu MP3, konieczne jest wykorzystanie rozwiązania umożliwiającego automatyczną klasyfikację wielowymiarowych wektorów danych. Stosuje się w tym celu metody maszynowego uczenia z nadzorem, które polegają na wnioskowaniu o nieznanej funkcji na podstawie sekwencji danych treningowych. Dane treningowe składają się z zestawu przykładów, które zawierają wektor danych oraz pożądany stan wyjściowy (zwany sygnałem kontrolnym). Algorytm uczący analizuje sekwencje treningowe i wnioskuję o nieznanej funkcji (modelu), którą określa się mianem klasyfikatora (w przypadku danych dyskretnych) lub funkcji regresji (dla danych ciągłych). Właściwie dobrany klasyfikator

powinien poprawnie przewidywać stan wyjściowy dla każdego poprawnego zestawu danych [13].

Do oceny skuteczności klasyfikacji koderów MP3 oraz urządzeń rejestrujących wykorzystano specjalnie w tym celu zaprojektowaną i stworzoną bazę danych. Do badań wykorzystano 2940 fragmentów utworów muzycznych (o długości 10 sekund każdy) pochodzących z 36 albumów z prywatnej kolekcji autora. Wszystkie nagrania źródłowe (2940) utrwalone w liniowym i bezstratnym formacie LPCM (Linear Pulse Code Modulation) i zapisane w postaci plików z rozszerzeniem „wav” zostały poddane kompresji za pomocą 21 różnych realizacji koderów MP3⁶. Do badań wykorzystano środowisko obliczeniowe MATLAB, w szczególności narzędzia z pakietów Signal Processing Toolbox, Parallel Computing Toolbox, Statistics Toolbox oraz Bioinformatics Toolbox [23]. Klasyfikację wektorów cech wykonano za pomocą liniowej analizy dyskryminacyjnej z funkcją odwracania macierzy za pomocą metody Moora–Penrose’a [8], [11]. Wektory

Tabela 1 Macierz klasyfikacji koderów z wykorzystaniem liniowej analizy dyskryminacyjnej. Nagrania poddano kompresji z przepływnością 112 kbps

Przewidywany koder	Koder rzeczywisty (przepływność: 112 kbps)																
	Lame 396	Lame 398	Lame 399	Adobe 2	Adobe CS6	iTunes	8Hz	Blade	GoGo	Helix	Mp3enc	Plugger	SCMPX	Shine	SoloH	Real	Xing
Lame 396	57,82	0,68	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 398	42,18	99,32	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 399	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe 2	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe CS6	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-
iTunes	-	-	-	-	-	100	-	-	0,68	-	-	-	-	-	-	-	-
8Hz	-	-	-	-	-	-	96,94	-	-	-	-	-	-	-	-	-	-
Blade	-	-	-	-	-	-	1,02	100	-	-	-	-	-	-	1,08	-	-
GoGo	-	-	-	-	-	-	-	-	99,32	-	-	-	-	-	-	-	-
Helix	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-
Mp3enc	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-
Plugger	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-
SCMPX	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-
Shine	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-
SoloH	-	-	-	-	-	-	2,04	-	-	-	-	-	-	-	98,98	-	-
Real	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Xing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100

⁶ Wykaz koderów MP3 wykorzystanych w badaniach można znaleźć w publikacjach [9] i [10] autora.

Tabela 2 Macierz klasyfikacji koderów z wykorzystaniem liniowej analizy dyskryminacyjnej. Nagrania poddano kompresji z przepływnością 128 kbps

Przewidywany koder	Koder rzeczywisty (przepływność: 128 kbps)																
	Lame 396	Lame 398	Lame 399	Adobe 2	Adobe CS6	iTunes	8Hz	Blade	GoGo	Helix	Mp3enc	Plugger	SCMPX	Shine	SoloH	Real	Xing
Lame 396	57,48	2,38	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 398	42,52	97,62	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 399	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe 2	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe CS6	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-
iTunes	-	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-
8Hz	-	-	-	-	-	-	92,52	-	-	-	-	-	-	-	-	-	-
Blade	-	-	-	-	-	-	1,36	100	-	-	-	-	-	-	3,06	-	-
GoGo	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-	-
Helix	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-
Mp3enc	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-
Plugger	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-
SCMPX	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-
Shine	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-
SoloH	-	-	-	-	-	-	6,12	-	-	-	-	-	-	-	96,94	-	-
Real	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-
Xing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100

cech reprezentujące poszczególne kodery były dzielone na dwie równe sekwencje: treningową oraz testową. Sekwencja treningowa była wykorzystywana do tworzenia modeli przez algorytm LDA, natomiast skuteczność klasyfikacji, a w związku z tym również dokładność wykrywania koderów, była sprawdzana na sekwencji testowej.

Wykonywana klasyfikacja polegała na predykcji rodzajów oraz wersji różnych implementacji algorytmu MP3, które zostały użyte do stworzenia badanych plików. W tabelach 1–3 zamieszczono wyniki predykcji koderów (podawane w procentach) w zależności od zastosowanej przepływności, odpowiednio 112 kbps (tabela 1), 128 kbps (tabela 2) oraz 192 kbps (tabela 3). Przykładowo, 96,94% wszystkich plików poddanych kompresji za pomocą kodera 8Hz z wykorzystaniem przepływności 112 kbps zostało sklasyfikowanych prawidłowo, 1,02% plików rozpoznano jako wytworzone przy użyciu kodera Blade, natomiast 2,04% jako poddane kompresji za pomocą implementacji SoloH (patrz tabela 1). Ponadto w przypadku użytej wersji kodera Real nie było możliwe wykonanie kompresji

z przepływnością 112 kbps (oprogramowanie nie udostępniło takiej opcji).

Na podstawie analizy powyższych wyników można zauważyć, że predykcja koderów MP3, jakie zostały użyte do kompresji nagrań fonicznych, jest możliwa, a jej skuteczność dla przeważającej liczby implementacji przewyższa 95% prawidłowych wskazań. W przypadku koderów 8Hz oraz SoloH liczba poprawnie sklasyfikowanych plików maleje wraz ze wzrostem przepływności. Może to być związane z dużym podobieństwem pomiędzy wymienionymi implementacjami algorytmu MP3 (koder SoloH powstał na bazie kodu źródłowego kodera 8Hz). Ponadto należy zauważyć, że użyta do klasyfikacji metoda liniowej analizy dyskryminacyjnej nie okazała się w pełni skuteczna podczas wykrywania wybranych koderów. Na rycinach 4 i 5 rozkłady wartości cech F34 i F35 są różne dla koderów 7 (8Hz) i 15 (SoloH), co powinno umożliwić precyzyjne rozróżnienie tych dwóch implementacji algorytmu MP3. Jednakże wyniki zamieszczone w tabelach 1–3 pokazują, że w każdym z badanych przypadków występują fałszywe wskazania. W związku z powyższym

Tabela 3 Macierz klasyfikacji koderów z wykorzystaniem liniowej analizy dyskryminacyjnej. Nagrania poddano kompresji z przepływnością 192 kbps

Przewidywany koder	Koder rzeczywisty (przepływność: 192 kbps)																
	Lame 396	Lame 398	Lame 399	Adobe 2	Adobe CS6	iTunes	8Hz	Blade	GoGo	Helix	Mp3enc	Plugger	SCMPX	Shine	SoloH	Real	Xing
Lame 396	60,20	-	2,38	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 398	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 399	39,80	-	97,62	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe 2	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe CS6	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-
iTunes	-	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-
8Hz	-	-	-	-	-	-	74,83	-	-	-	-	-	-	-	30,27	-	-
Blade	-	-	-	-	-	-	1,70	100	-	-	-	-	-	-	1,70	-	-
GoGo	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-	0,34
Helix	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	3,74	-
Mp3enc	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-
Plugger	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-
SCMPX	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-
Shine	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-
SoloH	-	-	-	-	-	-	23,47	-	-	-	-	-	-	-	68,03	-	-
Real	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	96,26	-
Xing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	99,66

zalecane byłoby poszukiwanie innych metod uczenia maszynowego z nadzorem, które znacznie lepiej wykorzystywałyby relacje zachodzące pomiędzy badanym cechami, np. binarne drzewa decyzyjne (BDT – *Binary Decision Trees*) opisane przez autora w publikacji [9].

Klasyfikacja urządzeń rejestrujących

Celem prowadzonych badań było opracowanie algorytmu umożliwiającego zarówno skuteczną klasyfikację w ramach zdefiniowanej listy koderów, jak i poszerzenie jej o nowe implementacje algorytmu MP3. W związku z tym tworzony wektor 46 cech został tak zaprojektowany, aby jego złożoność pozwalała na ujawnienie nieznanych jeszcze zależności, które mogą być reprezentowane przez nowe lub nieuwzględnione w niniejszym opracowaniu implementacje algorytmu MP3. Dzięki temu dodanie nowego koderu do listy rozpoznawanych wymaga jedynie wygenerowania odpowiedniej sekwencji treningowej dla algorytmów uczenia maszynowego.

Aby zademonstrować możliwość adaptacji zaproponowanego rozwiązania do listy zawierającej 17 koderów, dodano cztery realizacje sprzętowe będące przenośnymi urządzeniami rejestrującymi: Zoom, Tascam, Olympus, oraz Sony. Tym razem pliki MP3 stworzono w sposób bezpośredni, rejestrując za pomocą wymienionych dyktafonów około 10-sekundowe fragmenty utworów muzycznych odtwarzane z głośników aktywnych Adam P22. Wykonywana klasyfikacja polegała na predykcji rodzajów oraz wersji różnych implementacji algorytmu MP3, które zostały użyte do stworzenia badanych plików. W tabeli 4 zamieszczono wyniki predykcji koderów (podawane w procentach) dla przepływności 128 kbps. Przykładowo 93,54% wszystkich plików poddanych kompresji za pomocą koderu 8Hz zostało sklasyfikowanych prawidłowo, natomiast 6,46% jako zakodowanych przy użyciu implementacji SoloH.

Na podstawie analizy powyższych wyników można zauważyć, że predykcja koderów MP3, jakie zostały użyte do kompresji nagrań fonicznych, jest możliwa, a jej skuteczność dla przeważającej liczby implementacji

Tabela 4 Macierz klasyfikacji koderów z wykorzystaniem liniowej analizy dyskryminacyjnej. Nagrania poddano kompresji z przepływnością 128 kbps [9]

Przewidywany koder	Koder rzeczywisty (przepływność: 128 kbps)																				
	Lame 396	Lame 398	Lame 399	Adobe 2	Adobe CS6	iTunes	8Hz	Blade	GoGo	Helix	Mp3enc	Plugger	SCMPX	Shine	SoloH	Real	Xing	Zoom	Tascam	Olympus	Sony
Lame 396	53,07	3,74	7,14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 398	35,03	95,58	6,12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Lame 399	11,90	0,68	86,74	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe 2	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Adobe CS6	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
iTunes	-	-	-	-	-	100	-	-	-	-	-	-	0,68	-	-	-	-	-	-	-	-
8Hz	-	-	-	-	-	-	93,54	4,42	-	-	-	-	-	-	-	-	-	-	-	-	-
Blade	-	-	-	-	-	-	-	92,18	-	-	-	-	-	-	-	-	-	-	-	-	-
GoGo	-	-	-	-	-	-	-	-	99,66	-	-	-	-	-	-	-	-	-	-	-	-
Helix	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	0,34	-	-	-	-	-
Mp3enc	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-
Plugger	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	11,54	-	-
SCMPX	-	-	-	-	-	-	-	-	-	-	-	-	99,32	-	-	-	-	-	-	-	-
Shine	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-	-	-	-
SoloH	-	-	-	-	-	-	6,46	3,40	-	-	-	-	-	-	100	-	-	-	-	-	-
Real	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	99,66	-	-	-	-	-
Xing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	100	-	-	-	-
Zoom	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	98,70	-	-	-
Tascam	-	-	-	-	-	-	-	-	0,34	-	-	-	-	-	-	-	-	-	87,16	-	-
Olympus	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1,30	-	100	-
Sony	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1,30	-	100

przewyższa 95% prawidłowych wskazań. Warto nadmienić, że pomimo braku szczegółowych informacji dotyczących czterech dodanych implementacji algorytmu MP3 realizowanych sprzętowo przez dodane do listy urządzenia rejestrujące (Zoom, Tascam, Olympus oraz Sony), klasyfikacja plików stworzonych za ich pomocą okazała się również możliwa, a jej skuteczność była najmniejsza w przypadku koderu zastosowanego w urządzeniu Tascam DR-40 i wyniosła ok. 87%.

Wpływ znajomości koderu na możliwość wykrywania podwójnej kompresji

W ramach badań wykonanych w związku z realizacją projektu badawczego, których wyniki zostały

opisane w publikacjach [9] i [10], autor przedstawił opracowany zestaw algorytmów umożliwiających wykrywanie źródłowej przepływności bitowej w nagraniach poddanych podwójnej kompresji MP3, odróżnianie nagrań poddanych pojedynczej kompresji od nagrań ponownie kodowanych oraz identyfikację nieciągłości w nagraniach poddanych stratnej kompresji MP3. Istotnym wnioskiem z powyższych badań sformułowanym w wymienionych publikacjach było ustalenie, że do prawidłowej pracy zaproponowanych algorytmów konieczna jest wiedza na temat koderu, który został użyty do stworzenia badanego pliku. Największe znaczenie mają w tym przypadku rodzaj oraz wersja implementacji algorytmu MP3, która została użyta jako ostatnia podczas procesu kompresji analizowanego nagrania.

Mając na uwadze powyższe rozważania, autor postanowił zademonstrować wpływ modelu treningowego wygenerowanego dla algorytmu liniowej analizy dyskryminacyjnej na skuteczność procesu klasyfikacji binarnej mającej na celu odróżnienie nagrań oryginalnych (tzn. nagrań źródłowych poddanych pojedynczej kompresji) od zapisów fonicznych poddanych podwójnej kompresji. Do badań wykorzystano nagrania z utworzonej przez autora bazy danych (por. [9], [10]). Fragmenty utworów (nagrania źródłowe) poddano kompresji z użyciem trzech wybranych implementacji algorytmu MP3: Lame 399, iTunes oraz Adobe CS6, zdekodowano je do formatu bezstratnego, a następnie ponownie zakodowano, stosując te same trzy wybrane kodery. Z utworzonych plików MP3 wyodrębniono wektory o długości 228 cech (por. [9], [10]), które poddano klasyfikacji z wykorzystaniem liniowej analizy dyskryminacyjnej (LDA). W przypadku każdej z wykonanych klasyfikacji wektory cech były dzielone na dwie równe sekwencje: treningową oraz testową. Podczas procesu uczenia klasyfikatora tworzone były trzy modele treningowe odpowiadające trzem implementacjom algorytmu MP3: Lame 399, iTunes oraz Adobe CS6. Innymi słowy każdy z modeli treningowych powstawał na skutek porównania nagrań oryginalnych oraz poddanych podwójnej kompresji, do których stworzenia wykorzystano jeden z trzech wymienionych koderów. Dzięki temu określono procentowy udział poprawnie sklasyfikowanych plików MP3 występujących w sekwencji testowej w zależności od zastosowanego modelu treningowego. Badania zostały powtórzone dla trzech wybranych przepływności: 112, 128 i 192 kbps.

W tabelach 5–7 zamieszczono wyniki detekcji podwójnej kompresji (podawane w procentach) w zależności od użytego modelu treningowego wyznaczone dla trzech wybranych przepływności bitowych: 112 kbps (tab. 5), 128 kbps (tab. 6) oraz 192 kbps (tab. 7). Dla przykładu, w trakcie klasyfikacji nagrań poddanych kompresji za pomocą koderu Lame 399 z przepływnością 112 kbps w przypadku użycia podczas testu modelu, do stworzenia którego wykorzystano również nagrania zakodowane za pomocą tego koderu, 98,81% plików zostało prawidłowo przypisanych do właściwej im klasy. Jednakże zmiana modelu na utworzony z wykorzystaniem plików poddanych kompresji z wykorzystaniem koderu Adobe CS6 zmniejszyła odsetek prawidłowo sklasyfikowanych plików do 52,04% (patrz tab. 5).

Jak można zauważyć, do prawidłowej realizacji procesu detekcji podwójnej kompresji konieczna jest wiedza na temat rodzaju oraz wersji implementacji algorytmu MP3, jaka została użyta do stworzenia badanego pliku. Albowiem jedynie model treningowy wygenerowany na bazie nagrań poddanych kompresji za pomocą tego samego koderu, jaki został użyty do kompresji plików znajdujących się w sekwencji testowej, umożliwi wykonanie klasyfikacji, która będzie obciążona najmniejszym błędem.

Tabela 5 Klasyfikacja binarna nagrań poddanych pojedynczej i podwójnej kompresji MP3 z wykorzystaniem algorytmu LDA oraz różnych modeli treningowych. Do badań wykorzystano pliki MP3 poddane kompresji z przepływnością 112 kbps [9]

Koder wykorzystany do trenowania klasyfikatora	Koder wykorzystany do testowania klasyfikatora		
	Lame 399	Adobe CS6	iTunes
Lame 399	98,81	65,99	37,59
Adobe CS6	52,04	91,67	54,93
iTunes	47,62	41,67	97,79

Tabela 6 Klasyfikacja binarna nagrań poddanych pojedynczej i podwójnej kompresji MP3 z wykorzystaniem algorytmu LDA oraz różnych modeli treningowych. Do badań wykorzystano pliki MP3 poddane kompresji z przepływnością 128 kbps [9]

Koder wykorzystany do trenowania klasyfikatora	Koder wykorzystany do testowania klasyfikatora		
	Lame 399	Adobe CS6	iTunes
Lame 399	98,30	61,90	35,88
Adobe CS6	50,34	87,93	42,01
iTunes	49,15	56,12	97,79

Tabela 7 Klasyfikacja binarna nagrań poddanych pojedynczej i podwójnej kompresji MP3 z wykorzystaniem algorytmu LDA oraz różnych modeli treningowych. Do badań wykorzystano pliki MP3 poddane kompresji z przepływnością 192 kbps [9]

Koder wykorzystany do trenowania klasyfikatora	Koder wykorzystany do testowania klasyfikatora		
	Lame 399	Adobe CS6	iTunes
Lame 399	87,93	51,87	49,49
Adobe CS6	59,69	79,93	55,95
iTunes	49,49	62,93	98,98

Podsumowanie

Wyniki badań wykonanych w ramach projektu badawczego pokazują, że istnieje możliwość wykrywania rodzajów i wersji różnych implementacji algorytmów MP3 użytych do stworzenia badanego pliku. Bazując na dostępnych rozwiązaniach, autor zaproponował własny zestaw cech, umożliwiający nie tylko prawidłową detekcję parametrów kompresji, ale również szybką adaptację klasyfikatorów do nowych lub nieuwzględnionych w niniejszym opracowaniu implementacji algorytmu MP3. W celu sprawdzenia skuteczności stworzonych metod opracowano własną bazę danych nagrań fonicznych składającą się

z blisko miliona plików MP3. Przeprowadzone badania pokazały, że dzięki odpowiednio przygotowanym sekwencjom uczącym jest możliwe stworzenie algorytmu służącego do identyfikacji koderów MP3 oraz urządzeń rejestrujących na podstawie badania plików fonicznych poddanych stratnej kompresji. Fakt ten ma istotne znaczenie w powiązaniu z innymi wynikami badań autora, które zostały opisane w publikacji [9]. W związku z powyższym, mając możliwość identyfikacji użytego kodera MP3, opisana w niniejszej publikacji metoda może stać się istotnym wsparciem dla biegłych, którzy podejmują się badania autentyczności cyfrowych nagrań fonicznych.

Omówioną w niniejszej pracy metodę polegającą na wykrywaniu rodzaju oraz wersji implementacji algorytmu MP3 można wykorzystać zarówno do badania plików MP3, jak i innych algorytmów kompresji (np. AAC, Ogg Vorbis, itp.), w przypadku których obowiązujący standard nie definiuje w sposób precyzyjny i jednoznaczny każdej operacji wykonywanej przez dany algorytm. Warto również podkreślić, że w związku z wykorzystaniem do stworzenia wektora cech danych binarnych zapisanych w pliku MP3 skuteczność zaproponowanej metody jest niezależna od kształtu obwiedni charakterystyki widmowej badanych zapisów fonicznych. Czyni to metodę niezależną od treści badanych nagrań, co niewątpliwie ma miejsce w przypadku m.in. metod rozpoznawania mówców na podstawie głosu oraz innych rozwiązań polegających na analizie i klasyfikacji parametrów wyznaczonych na podstawie widma sygnału fonicznego. W związku z powyższym przedstawiona metoda może być wykorzystana do badania plików poddanych kompresji MP3 niezależnie od utwalonego w nich materiału fonicznego.

Źródła rycin i tabel

Rycina 1: autor
Ryciny 2–6: [9, 10]
Tabele 1–3: autor
Tabele 4–7: [9]

Bibliografia

1. Audio Engineering Society: AES27-1996 (r2007) "Recommended Practice for Forensic Purposes – Managing Recorded Audio Materials Intended for Examination", 1996 (2007).
2. Avciabas I., Hicsonmez S., Sencar H.T.: "Audio codec identification from coded and transcoded audios", *Digital Signal Processing*, vol. 23 (5), 2013, 1720–1730.

3. Böhme R., Westfeld A.: "Feature-based encoder classification of compressed audio streams", *Multimedia Systems*, vol. 11 (2), 2005, 108–120.
4. EncSpot – An MP3 Analyzer, 2002, <http://www.guerillasoft.nstep.com/EncSpot2>
5. Geiger R., Herre J. and Moehrs S.: "Analysing decompressed audio with the Inverse Decoder – towards an operative algorithm", in *Proc. 112th AES Convention*, Munich, May 10–13, 2002.
6. Grigoras C.: "Statistical tools for multimedia forensics", in *Proc. 39th AES International Conference of Audio Forensics*, Hillerød, Denmark, June 17–19, 2010, 27–32.
7. Herre J., Schug M.: "Analysis of Decompressed Audio – The Inverse Decoder", in *Proc. 109th AES Convention*, Los Angeles, USA, Sept. 22–25, 2000.
8. Ji S., Ye J.: "Discriminant Analysis for Dimensionality Reduction: An Overview of Recent Developments", in *Biometrics: Theory, Methods, and Applications*, N.V. Boulgouris, K.N. Plataniotis and E. Micheli-Tzanakou, Eds., John Wiley & Sons, Inc., Hoboken, NJ, USA, 2009, chapter 1, 1–17.
9. Korycki R.: "Authenticity examination of compressed audio recordings using detection of multiple compression and encoders' identification", *Forensic Science International*, vol. 238 (1-3), 2014, 33–46.
10. Korycki R.: „Badanie autentyczności cyfrowych nagrań fonicznych utwalonych w plikach MP3”, *Problemy Kryminalistyki*, vol. 283, 2014.
11. Mathworks R2013a Documentation, Statistics Toolbox and Bioinformatics Toolbox, <http://www.mathworks.com/help/documentation-center.html>
12. MPEG, "Coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbit/s, part 3: Audio", International Standard IS 11172-3, ISO/IEC JTC1/SC29 WG11, 1992.
13. Nilsson N.J.: "Introduction to machine learning", Artificial Intelligence Laboratory, Department of Computer Science, Stanford University, 2010.
14. Nilsson M.: "ID3v2 – The Audience is Informed", 1998, <http://www.id3.org>.
15. Sieler M., Sperschneider R.: "MPEG-Layer3 Bitstream Syntax and Decoding", Wyd. 1.2, 1997.
16. Wyrok SN z dnia 10 marca 1961 r., sygn. III K 49/61, OSNKW 1962, nr 1, poz. 8.