Predictive Noise Detection in Gramophone Records

Christoph F. Stallmann and Andries P. Engelbrecht
Department of Computer Science, University of Pretoria, Pretoria, South Africa
Email: {cstallmann, engel}@cs.up.ac.za

***Abstract*—Gramophones have regained widespread popularity over the past few years. Being an analogue storage medium, gramophone records are subject to distortions which are mainly caused by scratches. This paper empirically analyses various outlier detection algorithms and proposes a novel predictive approach for noise detection. Twelve different forecasting models were utilized for the predictive deviation method. Once outliers are identified, they can be reconstructed using interpolation algorithms or time series approximation. Experiments were conducted on 800 songs from eight genres, both with artificial and real gramophone noise. The algorithms were compared according to their detection rate, computational speed and the tradeoff between accuracy and speed. It was found that the novel absolute predictive deviation using the autoregressive integrate moving average model performed best overall. The experiments also indicated that it was easier to detect noise in stable signals from genres, compared to noise in volatile signals.**

***Index Terms*—gramophone records, noise detection, audio prediction, signal modelling, time series, outliers**

I. Introduction

The first commercial gramophone record was produced by Berliner in 1889, following decades of research and experimentation by Scott, Cros, Edison and Bell. Although gramophones were discontinued as mainstream music medium in the late 1980s, they continued to be popular amongst audiophiles with a steady sales growth over the past few years. The US alone recorded more than six million record sales in 2013, a 33% increase from the previous year [1]. Besides the interest in modern records, most music and other audio recordings prior to the 1960s were only produced and released for the gramophone. Many of these archived recordings are now being digitized by museums, music labels and collectors.

Gramophone records are an analogue storage medium and are, therefore, subject to noise caused by scratches and improper handling. This paper discusses and empirically analysis various outlier detection algorithms that can be utilized to detect disruptions in audio signals that were caused by physical scratches on a record. Identified outliers can then be mathematically reconstructed using an interpolation algorithm in order to remove the noise and improve the audio quality of the recording [2]. Various polynomials and time series models are discussed which form part of a novel predictive outlier detection approach. The methodology and performance measurement used during the experiments are described, followed by the empirical results. Finally, the algorithms are compared according to their detection accuracy and execution speed.

Manuscript received August 15, 2015; revised May 3, 2016.

II. State of the Art

Over the years various methods were proposed for the identification of outliers in gramophone audio. One approach uses mono records which are played back with a stereo turntable, generating two identical signals from one groove which are then correlated to identified outliers [3]. In more recent years, bidirectional processing in the time domain was used, which relies on unidirectional detection techniques to eliminate impulse disturbances of gramophone recordings [4]. Another approach utilizes a model-based predictor that matches highly repetitive click patterns to a set of previously generated noise templates [5]. Sprechmann proposed a technique where multiple copies of the same record were used to refurbish the audio signal, with the hope that the scratches and damages do not occur in the same place on the different copies [6]. Czyzewski formulated a frequency-based click detector which subtracts the frequency spectrum of the impulse-related part of the signal from the whole spectral representation of the signal, therefore, highlighting the segments which are affected most by noise [7].

Most of these approaches are limited, since they require special equipment, multiple records, or have restrictive processing techniques. This paper discuses a number of generic outlier detection algorithms and proposes a novel prediction approach which does not have these restrictions imposed. Although specific attention is given to gramophone noise, due to the genericity, the algorithms in this paper can be applied to other types of noise in audio signals, such as corrupted packets in VoIP or noise filtering in radios with a poor reception.

III. Outlier Detection Algorithms

This section discusses five outlier detection methods, namely the Standard Score (SS), Median Absolute Deviation (MAD), Mahalanobis Distance (MHD),

doi: 10.18178/ijsps.4.6.451-458

Nearest Neighbour Deviation (NND), Mean Absolute Spectral Deviation (MASD) and a Novel Absolute Predictive Deviation (APD) approach.

A. Standard Score

The SS is a statistical relationship between an observation and its population mean. A score greater than zero indicates that the point of interest is above the mean, whereas a negative score denotes a value below the mean. The SS for the data point y_t in the series y at time delay t is calculated as:

$$d_{ss}(y_t) = \frac{y_t - \mu}{\sigma} \quad (1)$$

where μ is the mean and σ the standard deviation of y. A rule of thumb is to mark values as outliers if their absolute standard score is 2.5 or greater for a set of up to 80 samples. If the score is calculated with more than 80 samples, the threshold is increased to 3 [8]. It was however shown that the absolute maximum possible score is dependent on the sample count n and is more accurately calculated as $n - 1/\sqrt{n}$ [9]. The SS may be exaggerated when a few or even a single extreme outlier is present, which is especially prominent in small datasets. Moderate outliers can therefore go undetected in the presence of extreme outliers.

B. Median Absolute Deviation

The mean and standard deviation are greatly influenced by a few extreme outliers in the population. The MAD utilizes the median of the population, which reduces the risk of a single extreme value affecting the outcome of the score, since the median is statistically more robust than the mean [10]. The median is computationally expensive to calculate and can be accelerated with more efficient approaches, such as quickselect, which is based on quicksort [11], or successive binning [12]. A MAD score at time delay t is defined as:

$$d_{mad}(y_t) = \frac{c\left(y_t - \tilde{y}\right)}{mad_y} \qquad mad_y = \text{median}(|y_i - \tilde{y}|)_{i=1}^{n} \quad (2)$$

where $\tilde{y}$ is the median of the subset y with n samples and c a constant greater than zero. Based on simulations, Iglewicz and Hoaglin suggested setting c to 0.6745 and flagging samples as outliers if the MAD score exceeds 3.5 [13]. However, the threshold of 3.5 depends on the dynamic range of the series y and should therefore be adjusted according to the input data.

C. Mahalanobis Distance

Mahalanobis introduced a relative measure to determine the distance from a data point to a common position. The MHD accounts for the covariance between variables and accommodates variances in different directions. It differs from the Euclidean distance in that it is scale-invariant and therefore does not change when the scales of length are multiplied by a common factor. A function $f(x)$ is said to be scale-invariant for a scale factor λ if $f(\lambda x) = \lambda^{\Delta} f(x)$ for some exponent Δ. Given a vector $\mathbf{y}$ of n multivariate independent random data points and a vector μ holding the means of the independent variables, the MHD is defined as:

$$d_{mhd}(\mathbf{y}) = \sqrt{(\mathbf{y} - \boldsymbol{\mu})^{\mathrm{T}} \mathbf{C}^{-1} (\mathbf{y} - \boldsymbol{\mu})} \quad (3)$$

where $\boldsymbol{C}^{-1}$ is the inverse of the covariance matrix. The MHD is called a normalized Euclidean distance if the covariance matrix is diagonal [14] and reduces to the Euclidean distance if the covariance matrix is equal to the identify matrix [15].

D. Nearest Neighbour Deviation

Outliers can be detected by calculating the deviation of a subset of k samples from a larger dataset, which is commonly referred to as kNN outlier detection. The deviation for continuous attributes is typically calculated using the Euclidean distance between vectors of attributes [16]-[18]. However, other means for determining the deviation exists, such as the Mahalanobis, Kullback-Leibler and Hamming distances [19]. If the data is multivariate, the distance is calculated for each individual attribute and then combined to represent the distance for all multivariate attributes [20].

NN outlier detectors are broadly categorised into global and local methods. The former approach determines a kNN global anomaly by calculating the distance to the k^{th} neighbour [21]. Using the mean distance instead of the distance to the k^{th} neighbour is more robust with regards to statistical fluctuation and often the preferred method [22], [23]. The global kNN score of point y_t at time delay t using $k/2$ samples on both sides of y is calculated as follows:

$$d_{nnd}(y_t) = \frac{1}{k}\left(\sum_{i=1}^{\frac{k}{2}} |y_t - y_i| + \sum_{j=\frac{k}{2}+2}^{k+1} |y_t - y_j|\right) \quad (4)$$

To ensure that both sides of y contribute equally, the window size k should be an even number.

The second category of NN anomaly detectors employs the Local Outlier Factor (LOF). LOF flags outliers by calculating the local deviation of a point with respect to its k nearest neighbours [24]. Various extensions and improvements to LOF were proposed, such as the local outlier probability [25], the connectivity based outlier factor [26], influenced outlierness [27], and the local correlation integral [28]. Benchmarking with optimal parameters between global kNN, LOF and the mentioned LOF extensions showed that the kNN global score on average performed best over a number of datasets, with LOF and its extensions only achieving a slightly better detection rate on individual datasets [22].

E. Mean Absolute Spectral Deviation

Noise can be detected in the time domain by identifying points that substantially deviate from the surrounding samples. Transforming the signal into the frequency domain moves the problem from detecting which samples are distorted to determining which frequencies are affected by the disruptions. Applying spectral methods to identify anomalies in the frequency domain is suitable, since outliers often cause a phase and amplitude shift in the Fourier series. An algorithm was

proposed by Shittu and Shangodoyin that makes use of Maximum Likelihood Estimation (MLE) to approximate the parameters of a Fourier model in order to determine the variance between the approximation and the actual values [29]. However, it was found that the algorithm performed well for three datasets, but very poorly for two others. Another proposition utilizes warped linear prediction on the frequency domain of audio data by using bilinear conformal mapping to emphasize outliers in higher frequencies [30].

Outliers can be detected by moving a window over the signal, generating frequency spectra and comparing them to the surrounding frequencies [7]. The mean absolute deviation is computed using the Euclidean distance between the frequency spectrum of a window at a certain time delay and the amplitudes of neighbouring frequencies. The Discrete Fourier Transform (DFT) calculates a set of discrete frequencies f from a sample window *y*, where the resolution of the DFT is depended on the window size and the windowing function. Given a window of *n* samples, the MASD is calculated using:

$$d_{masd}(\mathfrak{f}) = \frac{1}{n-1}\sum_{i=un}^{vn} |\mathfrak{f}_{i-1} - \mathfrak{f}_i| \quad (5)$$

where *u* and *v* are additional parameters in [0, 1] which control the range of frequencies considered to be affected by noise.

F. Absolute Predictive Deviation

Prediction-based outlier detection employs a forecasting model to determine the next values in a time series and if the predicted values deviate from the observed values with a certain degree, they are marked as outliers. Various predictive outlier detectors were proposed, using models such as multilayer perceptrons [31], autoregressive models [32], [33] and nearest cluster prediction [31]. Given a forecasting model *m* with a lag of *n* points that predicts the next value in the series *y* at time delay *t*+1, outliers are calculated with the APD as follows:

$$d_{apd}(y) = |y_{t+1} - m(y_{t-n+1}, \dots, y_t)| \quad (6)$$

An alternative approach makes use of the Mahalanobis distance to determine the deviation from the original signal [34]. If the absolute deviation in (6) exceeds a given threshold, the sample is flagged as an outlier. This approach is sound for univariate outliers, but can skew the model estimation for multivariate outliers, depending on the characteristics of the input data and the forecasting model. If y_{t+1} was flagged as an outlier, the observed value at *t*+1 should not be used for future model estimations, that is for estimations at y_{t+2}, ..., y_{t+r}, where *r* is the number of sequential points that contain noise. The problem is mitigated by utilising one of two alternative approaches. The first approach makes use of recurrent prediction where a single sample is forecasted at a time. Outliers at *t*+1 are replaced with their predicted value before estimating the next sample at *t*+2. The second approach utilizes batch prediction to estimate a model once for the given samples and then predict all *r* sequential outliers at once. Although batch prediction is computationally less expensive than recurrent forecasting, since the model has to be estimated

only once for an entire batch of sequential outliers, it relies on the model's ability to accurately predict up to *r* points. If the model is able to accurately predict enough samples into the future, batch prediction is advised, otherwise recurrent prediction should be used. The notation APD-*m* will be used in this paper, where *m* represents the forecasting model.

IV. Prediction Models

This section briefly introduces various polynomials and time series models that are utilized in the novel APD outlier detection. Interested readers are referred to [2] for a more detailed discussion on the given models.

A. Standard Polynomials

Standard Polynomials (STP) is a mathematical expression of a set of terms, where each term consists of a variable and a coefficient, defined as:

$$m_{stp}(x) = \alpha_d x^d + \alpha_{d-1}x^{d-1} + \cdots + \alpha_0 = \sum_{i=0}^{d} \alpha_i x^i \quad (7)$$

where *x* represent the variables' time delay, α_i the coefficients, and *d* the degree of the polynomial. The coefficients are approximated using Linear Least Squares (LLS) regression.

B. Fourier Polynomials

Fourier introduced a series to model a complex partial differentiable equation as a superposition of simpler oscillating sine and cosine functions. The discrete Fourier Polynomial (FOP) with a finite sum of sine and cosine functions is given as:

$$m_{fop}(x) = \frac{\alpha_0}{2} + \sum_{i=1}^{d} [\alpha_i \cos(i\pi x) + \beta_i \sin(i\pi x)] \quad (8)$$

where α_i and β_i are the polynomial's coefficients of order *d* that are estimated with a LLS fit.

C. Newton Polynomials

Newton formulated a polynomial of least degree that coincides at all points of a finite dataset. Given *n* + 1 data points (x_i,y_i), the Newton Polynomial (NEP) is defined as:

$$m_{nep}(x) = \sum_{i=0}^{n} \alpha_i h_i(x) \qquad h_i(x) = \prod_{j=0}^{i-1}(x - x_i) \quad (9)$$

where $h_i(x)$ is the i^{th} Newton basis polynomial. The coefficients α_i are typically computed with Newton's divided differences, but can also be approximated using LLS regression.

D. Hermite Polynomial

Hermite introduced a polynomial closely related to the Newton and Lagrange polynomials. Besides calculating a polynomial for *n*+1 points, Hermite also considered the derivatives at these points. The Hermite Polynomial (HEP) using the first derivative is defined as:

$$m_{hep}(x) = \sum_{i=0}^{n} h_i(x) f(x_i) + \sum_{i=0}^{n} \overline{h}_i(x) f'(x_i) \quad (10)$$

where h_i and $\overline{h}_i$ are the first and second fundamental Hermite polynomials. Although Hermite originally used Lagrange polynomials, Hermite's concept of osculation can be applied to any polynomials as long as the derivatives are known. This paper examines Osculating Standard Polynomials (OSP) and Osculating Fourier Polynomials (OFP).

E. Autoregressive Model

The Autoregressive (AR) model is an infinite impulse response filter that models a random process where the generated output is linearly depended on the previous values in the process. The model generates internal dynamics, since it retains memory by keeping track of the feedback. Given a sequential series y with $n+1$ data points, the AR model of order p predicts the value of a point at time delay t with the previous values of the series as follows:

$$y_t = c + \varepsilon_t + \sum_{i=1}^{p} \alpha_i y_{t-i} \quad (11)$$

where c is a constant, typically considered to be zero, ε_t the white noise error term, almost always considered to be Gaussian white noise, and α_i the coefficients for the model. The AR coefficients are estimated using a LLS fit.

F. Moving Average Model

The Moving Average (MA) is a statistical calculation where a series of averages are generated from subsets of the full dataset. A study by Slutsky on applying the MA on random events lead to the formulation of a finite impulse response filter where univariate time series are modelled with white noise terms with some additional interpretation added to the model [35]. Slutzky [35] and Yule [36] independently discovered that the moving summation of random data series oscillates when no such fluctuation exists in the original observation. The MA model is defined as:

$$y_t = \mu + \varepsilon_t + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i} \quad (12)$$

where μ is the mean of the series, typically assumed to be zero, β_i the model coefficients of order q and ε_t, ..., ε_{t-q} the white noise error terms. The error terms are assumed to be independent and identically distributed random variables, meaning that all random variables are mutually independent and are subject to the same probability distribution. The MA coefficients are approximated using MLE which in turn is maximized through a gradient-based method such as the Broyden-Fletcher-Goldfarb-Shanno (BFGS) [37] or the Berndt-Hall-Hall-Hausman (BHHH) [38] algorithms.

G. Autoregressive Moving Average Model

The Autoregressive Moving Average (ARMA) model is a combination of the AR and MA models. Proposed by Whittle [39], Box and Jenkins later popularized the model by describing a method for determining the model orders and an iterative method for estimating the model coefficients [40]. The ARMA model is defined as:

$$y_t = c + \varepsilon_t + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i} \quad (13)$$

where p and q are the AR and MA model orders respectively. The model coefficients are typically approximated with MLE using BFGS or BHHH.

H. Autoregressive Integrated Moving Average Model

The Autoregressive Integrated Moving Average (ARIMA) model is a generalization of the ARMA model which is applied if the observed data shows some characteristics of non-stationarity, such as seasonality, trends and cycles [40]. A differencing operation is added as an initial step to the ARMA model to remove possible non-stationarity. The ARMA model in (13) can also be expressed in terms of the lag operator as $\alpha(L)y_t = \beta(L)\varepsilon_t$, where $\alpha(L)$ and $\beta(L)$ are the lag polynomials of the AR and MA processes respectively. The ARIMA model incorporates the difference operator, $y_t - y_{t-1} = (1-L)y_t$, as follows:

$$(1 - \sum_{i=1}^{p} \alpha_i L^i)(1-L)^d y_t = (1 + \sum_{i=1}^{q} \beta_i L^i)\varepsilon_t \quad (14)$$

where p is the AR order, q the MA order and d the order of integration. ARIMA coefficients approximation follows the same technique as the ARMA model.

I. Autoregressive Conditional Heteroskedasticity

The Autoregressive Conditional Heteroskedasticity (ARCH) model was developed by Engel for financial markets that show periods of low volatility followed by periods of high volatility and vice versa [41]. ARCH achieves non-constant conditional variance by calculating the variance of the current error term ε_t as a function of the error terms ε_{t-i} in the previous i time periods. Therefore the forecasting is done on the error variance at time t, compared to the AR model which does its prediction directly on the previously observed time series values. The ARCH process for a zero mean series is defined as:

$$y_t = \sigma_t \varepsilon_t \qquad \sigma_t = \sqrt{\alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2} \quad (15)$$

where ε_t is Gaussian white noise and σ_t is the conditional variance, modelled by an AR process. Since ARCH makes use of an AR process, the coefficients can be estimated through LLS fitting using Yule-Walker equations. Since the distribution of ε_{t-i}^2 is naturally not normal, the Yule-Walker approach does not provide an accurate estimation. The initial coefficients can be set with the Yule-Walker approach and then iteratively refined using MLE.

J. Generalized Autoregressive Conditional Heteroskedasticity

The generalized autoregressive conditional heteroskedasticity (GARCH) model is a generalization of the ARCH model proposed by Bollerslev which also uses the weighted average of past squared residuals without the declining weights ever reaching zero [42]. GARCH uses an ARMA model for the error variance as follows:

$$\sigma_t = \sqrt{\alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2} \quad (16)$$

where α_i and β_i are the model coefficients and p and q the GARCH and ARCH orders respectively. The model coefficients are solved the same way as ARMA coefficients.

V. Methodology

This section explains the methodology and procedures followed to obtain the empirical results. The test data, noise generation and masking, performance measurement, computational speed and tradeoff is discussed.

A. Test Data and Noise

Benchmarking was performed on a set of 800 songs in eight genres, namely classical, country, electronic, jazz, metal, pop, reggae and rock music. The tracks were encoded in stereo using the Free Lossless Audio Codec (FLAC) with a sample rate of 44.1kHz. In order to evaluate the algorithm's performance in a controlled environment, the songs were subjected to artificially generated noise. Another set of 83 songs recorded from real gramophones was used as a validation set to verify the performance of the artificially disrupted tracks. Fig. 1 shows typical audio disruptions in gramophone recordings.

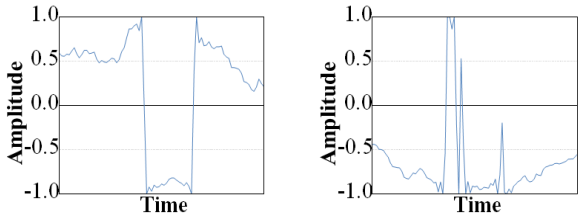

Figure 1. Typical disruptions caused by scratches on records.

A common approach in audio processing is to generate disruptions in clean audio data with Gaussian white noise [43]-[45]. It was suggested that positive pulses with a constant magnitude and a mixture of white noise and impulses should be used [46]. The test data in this paper was distorted with four different types of artificial generated noise which resemble the disruptions caused by scratches in Fig. 1. The noise was generated using positive and negative pulses with varying magnitudes and then subjected to a Gaussian white noise process. Most scratches do not affect more than 30 sequential samples. Benchmarking was conducted with noise of up to 50 samples in order to accommodate longer disruptions. The algorithms are, however, able to detect noise of any duration.

B. Noise Masking

The detection algorithms generate a per-sample noise map with values in [0, z] where z is determined by each individual algorithm. A binary mask is generated for each sample i in the signal, indicating whether or not the sample is an outlier. Given a threshold θ and a noise map η, the mask is generated as follows:

$$\tilde{\eta}_i = \begin{cases} 1 & for\ \eta_i \geq \theta \\ 0 & otherwise \end{cases} \tag{17}$$

C. Performance Measurement

The algorithms' performance was evaluated using the Sensitivity (SEN), Specificity (SPE) and the Matthews Correlation Coefficient (MCC). The True Positives (TP) and True Negatives (TN) are the number of correctly identified outliers and inliers respectively, whereas the False Positives (FP) and False Negatives (FN) are the number of incorrectly flagged inliers and outliers respectively. The SEN is the ability of an algorithm to correctly identify outliers, whereas the SPE is the capacity to which inliers are correctly recognized. The SEN and SPE is calculated as:

$$\text{SEN} = \frac{\text{TP}}{\text{TP+FN}} \qquad \text{SPE} = \frac{\text{TN}}{\text{TN+FP}} \tag{18}$$

The MCC is used as a combined measurement to evaluate how well outliers are correctly identified and penalizing mislabeled inliers. The MCC is computed using:

$$\text{MCC} = \frac{\text{TP}\times\text{TN}-\text{FP}\times\text{FN}}{\sqrt{(\text{TP+FP})(\text{TP+FN})(\text{TN+FP})(\text{TN+FN})}} \tag{19}$$

D. Computational Speed and Tradeoff

The execution time of the algorithms are measured as a speed, that is the duration in seconds it takes to process a second of audio data using a single processor thread and is denoted as s\s. In order to evaluate the tradeoff between the detection accuracy and the execution speed, a tradeoff measurement is needed. Based on the scoring metric in [47], the Speed-Accuracy Tradeoff (SAT) is calculated using:

$$\text{SAT} = \left(\frac{1-\kappa}{\hat{\kappa}-\check{\kappa}} + \frac{\tau}{\hat{\tau}-\check{\tau}}\right)^{-1} \tag{20}$$

where κ is the detection MCC and τ the computational speed measured in s\s. $\kappa^{\wedge}$ and $\kappa^{\vee}$ are the best and worst MCCs, with $\tau^{\wedge}$ and $\tau^{\vee}$ the fastest and slowest execution times respectively. A higher SAT score indicates a more efficient tradeoff between the accuracy and the speed. An Intel Core i7 2600 at 3.4GHz machine with 16GB memory was used for the experiments.

VI. Empirical Results

Table I shows the threshold from (17), the sensitivity, specificity, overall detection accuracy, computational time and tradeoff of the outlier detection algorithms. All algorithms' parameters were optimized using fractional factorial design.

The NND had the highest sensitivity, but due to the lowest specificity amongst all algorithms had an overall low MCC. The MHD achieved the best specificity, slightly higher than that of the SS and MAD. The overall best detection accuracy was achieved by the APD using the ARIMA model, with a MCC of 0.837. The APD-HEP had the fastest execution time, and besides the APD-NEP and MASD is the only algorithm that can be executed in real time using a single thread. The APD-AR was the most efficient algorithm by achieving a good detection rate within a limited timespan. The proximity-based algorithms, namely SS, MAD, MHD and the NND, had a

good detection accuracy, but still remained inferior to most predictive algorithms.

TABLE I. THE THRESHOLD, SENSITIVITY, SPECIFICITY, DETECTION ACCURACY, SPEED AND TRADEOFF OF THE OUTLIER DETECTION ALGORITHMS

Algorithm	THLD	SEN	SPE	MCC	Speed	SAT
SS	3.289	0.681	0.999	0.792	5.467	1.211
MAD	3.822	0.637	0.999	0.741	19.16	0.824
MHD	3.275	0.685	0.999	0.799	15.47	1.053
NND	0.680	0.825	0.993	0.754	1.691	1.100
MASD	8.095	0.384	0.998	0.558	0.506	0.627
APD-STP	0.179	0.785	0.997	0.802	13.81	1.093
APD-OSP	0.176	0.783	0.998	0.804	30.86	0.863
APD-FOP	0.196	0.802	0.997	0.818	24.69	0.985
APD-OFP	0.197	0.801	0.997	0.820	68.23	0.606
APD-NEP	0.174	0.736	0.999	0.800	0.771	1.373
APD-HEP	0.546	0.536	0.998	0.647	0.497	0.783
APD-AR	0.181	0.811	0.998	0.836	4.269	1.530
APD-MA	0.190	0.753	0.996	0.747	24.77	0.785
APD-ARMA	0.181	0.823	0.998	0.835	26.64	1.014
APD-ARIMA	0.182	0.803	0.998	0.837	11.50	1.323
APD-ARCH	0.202	0.801	0.998	0.83	14.59	1.211
APD-GARCH	0.202	0.801	0.998	0.83	14.59	1.211

Fig. 2 illustrates the change in the sensitivity with an increasing duration in the sequential distorted samples. The APD detection represents the best predictive algorithm, that is the ARIMA model. All algorithms struggled to detect univariate noise, that is a single outlying sample. The outlier detectors, except MASD, had a quick sensitivity increase with noise longer than two samples and stayed relatively stable for noise durations of up to 50 samples. MASD performed considerably poorer compared to the other algorithms. Although not shown in the graph, the MASD was tested with noise of up to 200 samples, which resulted in a steady increase in the sensitivity. MASD was therefore able to accurately detect long multivariate outliers, but had difficulty with noise durations shorter than 50 samples, which is more common in gramophone distortions.

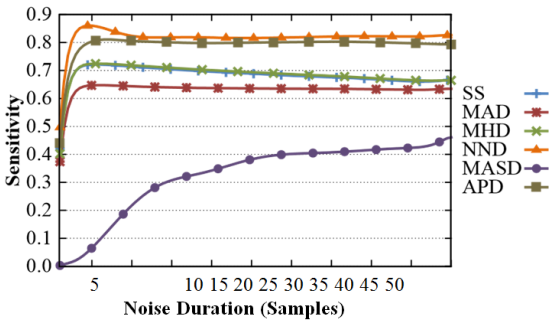

Figure 2. The detection sensitivity for an increasing duration.

Fig. 3 illustrates the detection MCC of the outlier identifiers for different genres, with APD employing the ARIMA model. The MAD performed best for classical music, but was inferior to the SS, MHD and APD for all other genres. The APD performed well on average and showed a significant superiority with more volatile signals, especially the electronic, metal, and pop genres.

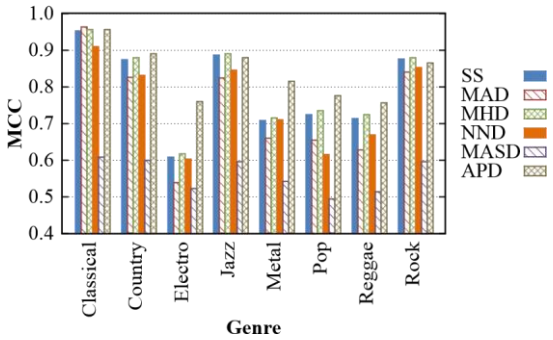

Figure 3. The detection accuracy for different genres.

Table II compares the algorithms' detection accuracy on artificially generated and real gramophone noise with the MCC's standard deviation range given in the second column. As expected, all algorithms performed slightly worse when tested on real gramophone recordings. However, the difference between the two noise groups is statistically insignificant. The gramophone's MCC falls within the artificial MCC's standard deviation range and the artificial noise generation is therefore considered a sound representation of the gramophone noise.

Once the noisy samples are flagged, they are reconstructed using one of the algorithms in [2] or [48]. A perceptual evaluation was conducted with 15 participants. The songs had a pleasant listening experience with most participants unable to distinguish between the original and reconstructed signals. Most of the noise that went undetected had a short duration and was difficult to identify by the human ear.

TABLE II. THE DETECTION ACCURACY (MCC) FOR ARTIFICIALLY GENERATED AND REAL GRAMOPHONE NOISE

Algorithm	Artificial Noise	Gramophone Noise
SS	0.7924 ±0.172	0.7442
MAD	0.7407 ±0.195	0.7101
MHD	0.7989 ±0.168	0.7541
NND	0.7539 ±0.174	0.7100
MASD	0.5582 ±0.084	0.5001
APD-ARIMA	0.8367 ±0.108	0.7975

VII. CONCLUSION

This paper analysed and benchmarked six different algorithms that are able to detect disruptions in audio signals caused by scratches on gramophone records. A novel predictive deviation outlier detection was proposed, utilizing one of twelve different forecasting models. The algorithms were benchmarked against each other by comparing the SEN, SPE and MCC of the detection process and measuring the execution time. It was found that the predictive outlier detection using the ARIMA model performed best on average. Predictive identification using the AR model had the most efficient tradeoff by detecting most outlier for a limited execution time.

Future research should focus on improving the detection accuracy with a more effective spectral algorithm. Using the Mahalanobis or nearest neighbour distance on the frequency spectrum instead of the Euclidean deviation may prove beneficial. The input signal can also be automatically categorized according to the volatility of the samples and then processed by the most accurate algorithm for the given volatility. The research will be extended in order to determine the ability of an Artificial Neural Network (ANN) to forecast audio signals. Initial research has shown that predictive ANNs have a performance and speed improvement in the APD compared to the other models presented in this paper.

REFERENCES

[1] F. Richter. (January 2014). The LP is Back! [Online]. Availabel: http://www.statista.com/chart/1465/vinyllp-sales-in-the-us

[2] C. F. Stallmann and A. P. Engelbrecht, "Gramophone noise reconstruction: A comparative study of interpolation algorithms for noise reduction," in *Proc. SIGMAP*, Colmar, France, 2015.

[3] M. Niedzwiecki, "Elimination of clicks and background noise from archive gramophone recordings using the two track mono approach," in *Proc. European Signal Processing Conf.*, 1996, pp. 1749-1752.

[4] M. Niedzwiecki and M. Ciołek, "Elimination of impulsive disturbances from archive audio signals using bidirectional processing," *IEEE Transactions on ASLP*, vol. 21, no. 5, pp. 1046-1059, 2013.

[5] M. Niedzwiecki and M. Ciołek, "Localization of impulsive disturbances in archive audio signals using predictive matched filtering," in *Proc. Int. Conf. on Acoustics, Speech, Signal Processing*, 2014, pp. 2888-2892.

[6] P. Sprechmann, A. M. Bronstein, J. M. Morel, and G. Sapiro, "Audio restoration from multiple copies," in *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, 2013, pp. 878-882.

[7] A. Czyzewski, "Some methods for detection and interpolation of impulsive distortions in old audio recordings," in *Proc. IEEE Applications of Signal Processing to Audio and Acoustics*, 1995, pp. 139-142.

[8] S. Vijendra and P. Shivani, "Robust outlier detection technique in data mining: A univariate approach," *Cornell University Repository*, 2014.

[9] R. E. Shiffler, "Maximum Z scores and outliers," *The American Statistician*, vol. 42, no. 1, pp. 79-80, February 1988.

[10] C. Leys, L. Christophe, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *Journal of Experimental Social Psychology*, vol. 49, no. 4, pp. 764-766, 2013.

[11] C. A. R. Hoare, "Algorithm 65: Find," *Communications of the ACM*, vol. 4, no. 7, pp. 321-322, July 1961.

[12] R. J. Tibshirani, "Fast computation of the median by successive binning," *Cornell University Computing Research Repository*, 2008.

[13] B. Iglewicz and D. C. Hoaglin, *How to Detect and Handle Outliers Front Cover*, 1st ed., Milwaukee, US: ASQ Quality Press 1993.

[14] M. Marghany and M. Hashim, "Comparison between Mahalanobis classification and neural network for oil spill detection using RADARSAT-1 SAR data," *International Journal of the Physical Sciences*, vol. 6, no. 3, pp. 566-576, 2011.

[15] L. Bodis, "Quantification of spectral similarity: Towards automatic spectra verification," Ph.D. dissertation, Eidgenossische Technische Hochschule Zurich, Zurich, Switzerland, 2007.

[16] K. Bhaduri and B. L. Matthews, "Algorithms for speeding up distance based outlier detection," in *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2011, pp. 859-867.

[17] P. Yang and B. Huang, "KNN based outlier detection algorithm in large dataset," in *Proc. IEEE International Workshop on Geoscience and Remote Sensing*, vol. 1, 2008, pp. 611-613.

[18] M. Zhao and V. Saligrama, "Anomaly detection with score functions based on nearest neighbor graphs," in *Proc. Neural Information Processing Systems*, Vancouver, Canada, December 2009, pp. 2250-2258.

[19] J. Walters-Williams and Y. Li, "Comparative study of distance functions for nearest neighbors," in *Advanced Techniques in Computing Sciences and Software Engineering*, Springer, 2010, pp. 79-84.

[20] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, 1st ed., Boston, US: Addison-Wesley, 2005.

[21] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *ACM SIGMOD Record*, vol. 29, no. 2, pp. 427-438, May 2000.

[22] M. Amer and M. Goldstein, "Nearest-Neighbor and clustering based anomaly detection algorithms for RapidMiner," in *Proc. Rapid Miner Community Meeting and Conference*, 2012, pp. 1-12.

[23] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces," in *Proc. European Conference on Principles of Data Mining and Discovery*, 2002, pp. 15-26.

[24] M. Breunig, H. Kriegel, R. Ng, and J. Sander, "Identifying density-based local outliers," *ACM SIGMOD*, vol. 29, no. 2, pp. 93-104, 2000.

[25] H. P. Kriegel, P. Kroger, E. Schubert, and A. Zimek, "LoOP: Local outlier probabilities," in *Proc. ACM IKM Conference*, 2009, pp. 1649-1652.

[26] J. Tang, Z. Chen, A. Fu, and D. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Taipei, Taiwan, May 2002.

[27] W. Jin, A. K. H. Tung, J. Han, and W. Wang, "Ranking outliers using symmetric neighborhood relationship," in *Proc. Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, 2006, pp. 577-593.

[28] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral," in *Proc. International Conference on Data Engineering*, March 2003, pp. 315-326.

[29] O. Shittu and D. Shangodoyin, "Detection of outliers in time series data: A frequency domain approach," *Asian Journal of Scientific Research*, vol. 1, no. 2, pp. 130-137, 2008.

[30] P. A. A. Esquef, M. Karjalainen, and V. Valimaki, "Detection of clicks in audio signals using warped linear prediction," in *Proc. International Conference on Digital Signal Processing*, 2002, pp. 1085-1088.

[31] D. J. Hill and B. S. Minsker, "Anomaly detection in streaming environmental sensor data: A data-driven modeling approach," *Environmental Modelling and Software*, vol. 25, no. 9, pp. 1014-1022, 2010.

[32] M. Niedzwiecki and M. Ciołek, "Elimination of clicks from archive speech signals using sparse autoregressive modeling," in *Proc. European Signal Processing Conference*, 2012, pp. 2615-2619.

[33] R. S. Tsay, "Outliers, level shifts, and variance changes in time series," *Journal of Forecasting*, vol. 7, no. 1, pp. 1-20, 1988.

[34] M. C. Hau and H. Tong, "A practical method for outlier detection in autoregressive time series modelling," *Stochastic Hydrology and Hydraulics*, vol. 3, no. 4, pp. 241-260, 1989.

[35] E. Slutzky, "The summation of random causes as the source of cyclic processes," *Econometrica*, vol. 5, no. 2, pp. 105-146, 1927.

[36] G. Yule, "Why do we sometimes get nonsense correlations between time-series," *Royal Statistical Society Journal*, vol. 89, pp. 1-64, 1926.

[37] C. Broyden, "The convergence of a class of double-rank minimization algorithms," *Journal of Applied Mathematics*, vol. 6, pp. 76-90, 1970.

[38] E. K. Berndt, B. H. Hall, R. E. Hall, and J. A. Hausman, "Estimation and inference in nonlinear structural models," *Annals of Economic and Social Measurement*, vol. 3, no. 4, pp. 653-665, 1974.

[39] P. Whittle, "Hypothesis testing in time series analysis," Ph.D. dissertation, Uppsala University, Uppsala, Sweden, 1951.

[40] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, San Francisco, US: Holden-Day, Incorporated, 1970.

[41] R. Engle, "AR conditional heteroscedasticity with estimates of variance of UK inflation," *Econometrica*, vol. 50, no. 4, pp. 987-1007, 1982.
[42] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307-327, 1986.
[43] L. Oudre, "Automatic detection and removal of impulsive noise in audio signals," *Image Processing on Line*, vol. 5, pp. 267-281, 2015.
[44] P. Esquef and G. Welter, "Audio de-thumping using Huang's emperical mode decomposition," in *Proc. International Conference on Digital Audio Effects*, 2011, pp. 401-408.
[45] J. Howarth and P. Wolfe, "Correction of wow and flutter effects in analog tape transfers," *Journal of the Audio Engineering Society*, vol. 117, 2011.
[46] M. Niedzwiecki and K. Cisowski, "Adaptive scheme for elimination of background noise and impulsive disturbances from audio signals," in *Proc. Quatrozieme Colloque*, Juan-les-Pins, France, 1993, pp. 519-522.
[47] S. Sidiroglou-Douskos, S. Misailovic, H. Hoffmann, and M. Rinard, "Managing performance vs accuracy trade-offs with loop perforation," in *Proc. ACM SIGSOFT Symposium*, 2011, pp. 124-134.
[48] C. F. Stallmann and A. P. Engelbrecht, "Gramophone noise detection and reconstruction using time delay artificial neural networks," *IEEE Transactions on Systems, Man, and Cybernetics*, in review, 2015.

Christoph F. Stallmann completed the Masters degree in Computer Science at the University of Pretoria, South Africa. He has worked at the Council for Scientific and Industrial Research (CSIR) and the South African National Space Agency (SANSA) and has been an assistant lecturer in Computer Science at the University of Pretoria since 2011. His research interests include music and audio processing, signal modelling, and neural networks. His current research focuses on the noise detection and reconstruction of gramophones.

Andries P. Engelbrecht received the Masters and PhD degrees in Computer Science from the University of Stellenbosch, South Africa, in 1994 and 1999 respectively. He is a Professor in Computer Science at the University of Pretoria, and serves as Head of the department. He also holds the position of South African Research Chair in Artificial Intelligence, leads the Computational Intelligence Research Group at the University of Pretoria, consisting of 40 Masters and PhD students, and has published over 220 papers.