VIQID: A NO-REFERENCE BIT STREAM-BASED VISUAL QUALITY IMPAIRMENT DETECTOR

Nicolas Staelens[a] *Nick Vercammen*[a] *Yves Dhondt*[b]
Brecht Vermeulen[a] *Peter Lambert*[b] *Rik Van de Walle*[b] *Piet Demeester*[a]

[a]Ghent University - IBBT, Department of Information Technology, Ghent, Belgium
[b]Ghent University - IBBT, Department of Electronics and Information Systems, Ghent, Belgium

ABSTRACT

In order to ensure adequate quality towards the end users at all time, video service providers are getting more interested in monitoring their video streams. Objective video quality metrics provide a means of measuring (audio)visual quality in an automated manner. Unfortunately, most of the current existing metrics cannot be used for real-time monitoring due to their dependencies on the original video sequence. In this paper we present a new objective video quality metric which classifies packet loss as visible or invisible based on information extracted solely from the captured encoded H.264/AVC video bit stream. Our results show that the visibility of packet loss can be predicted with a high accuracy, without the need for deep packet inspection. This enables service providers to monitor quality in real-time.

Index Terms— Visual Quality, Impairment Detection, No-Reference, H.264/AVC

1. INTRODUCTION

When video is streamed over IP-based networks, such as the Internet, packet loss can severely degrade the visual quality of the received video signal. In order to deliver, ensure and maintain optimal quality towards the end-users at all time, video service providers are getting more interested in objective video quality metrics for continuous real-time monitoring of their video streams.

Objective video quality metrics can be classified into three categories depending on the availability of the original video sequence:

- *Full-Reference* (FR) quality metrics require access to the complete original video sequence since their quality evaluation is usually based on a frame-by-frame comparison between the original and the impaired sequence. Two of the most well known FR video quality metrics are Peak Signal-to-Noise Ratio and Structural SIMilarity (SSIM) [1].
- *Reduced-Reference* (RR) metrics measure quality by comparing certain features which are extracted from the original and the impaired video sequence. At the sender side, these features are extracted and signaled to the receiver using an ancillary error-free channel.
- *No-Reference* (NR) metrics perform their quality evaluation using only the received (erroneous) video stream. As these metrics do not depend on the original sequence nor need an ancillary channel, they are most appropriate for monitoring purposes.

Furthermore, quality metrics can be pixel-based, bit stream-based or a combination of both. Pixel-based metrics require a complete decoding of the received video sequence, whereas bit stream metrics only perform a parsing of the encoded video stream in order to estimate visual quality. Hybrid quality metrics use a combination of pixel information and bit stream processing. From a real-time monitoring point of view, it is clear that NR bit stream-based video quality metrics are the most interesting ones since they do not require access to the original video sequence nor the decoding of the received video stream.

Traditionally, quality metrics provide an average quality rating for the entire video sequence or over a certain time window. For example, SSIM outputs a score between -1 and 1 where the latter stands for perfect quality. However, when monitoring video streams it could be preferable to receive instantaneous feedback when visual artifacts occur due to network impairments. This is also interesting in the case of monitoring long video sequences such as television programs or movies. By counting the number of visual impairments or tracking the mean time between visual artifacts, service providers can gain more insight into the quality of the delivery network and their video streams.

In this paper we focus on predicting the visibility of network artifacts[1], introduced by packet loss in the network, in H.264/AVC encoded video sequences and introduce *ViQID*, a novel bit stream based visual quality impairment detector. For modeling the visibility of network impairments, we use a decision tree which classifies packet losses as visible or invisible based on parameters extracted solely from the captured

[1]no compression artifacts are taken into account.

encoded video bit stream.

The remainder of this paper is structured as follows. We start in section 2 by providing an overview of already conducted research concerning the detection of visual quality degradations in video sequences. Next, section 3 describes the subjective test conducted in order to obtain the ground truth for constructing our model. In section 4 we propose and evaluate different decision trees for predicting packet loss visibility. Finally, we conclude the article and present future work.

2. BIT STREAM BASED QUALITY IMPAIRMENT DETECTION

In this section, we provide a general overview of already conducted research related to the visiblity of visual impairments.

Kanumuri *et al.* used two different modeling approaches in [2] to determine whether packet loss, occurring in MPEG-2 encoded sequences, results in visible impairments. First, a RR classifier is constructed which extracts information from the complete video bit stream, the received bit stream and the decoded video. The decision tree classifier was trained on data obtained through a subjective test where users were asked to indicate when they saw an artifact. Based on these data, a packet loss was classified to be visible when 75% or more of the subjects perceived the error. If 25% or less of the subjects perceived the artifact, the packet loss was classified as invisible. The remaining errors were classified as indeterminable and not used as training data.

In a second approach, the authors used a General Linear Model (GLM) to predict the probability that a packet loss is visible and used these probabilities to decide whether a packet loss is visible or not. Using a GLM, thresholds for identifying visible losses can be set dynamically without the need for reconstructing the model. This is not the case when a decision tree is used.

In [3], saliency-based factors were included as additional parameters during the construction of a GLM for predicting packet loss visibility. Results indicated that the prediction performance improved when visual attention was taken into account.

A GLM was also used in [4] to model the probability that individual and multiple packet losses result in visible impairments for H.264/AVC encoded video sequences. The proposed model is also a RR model which needs access to the decoded (lossy) video and to features extracted from the original encoded video. Results show that the amount of motion is not a significant feature for predicting packet loss visibility when motion-compensated error concealment is used.

Reibman *et al.* [5] showed that the overall accuracy of estimating packet loss visibility can be increased when scene characteristics such as camera motion and proximity to scene changes are taken into account. Concerning the influence of camera motion, results indicate that impairments are less visible in still scenes compared to panning and zooming scenes.

Suresh *et al.* introduced the Mean Time Between Failures (MTBF) [6] as a new means for subjective measurement of quality instead of using the more traditional Mean Opinion Score (MOS) grading scales [7, 8]. During a subjective test, users are asked to indicate the occurrence of visual impairments during playback by, for example, pressing a buzzer or a space bar. The MTBF measure is based on failure statistics and represents how often an average viewer perceives any kind of visual artifact. It is argued that, amongst other, MTBF simplifies the subjective test procedure and that it is more robust in the context of heterogeneous stimuli.

In [9], the Automatic Video Quality (AVQ) metric is described which is capable of detecting and quantifying visible compression and network artifacts. The former are estimated as a function of the quantization step size and scene activity whereas the latter are estimated during the decoding process. Since the AVQ metric uses information from both the encoded bit stream and the decoded pixel values, it is a hybrid no-reference video quality metric. Results in [10] indicate a high correlation between the objective AVQ scores and the subjective MTBF measurements.

Still, all research is currently mainly focused on RR and hybrid video metrics for predicting visual quality. However, we developed a new model which only needs information that can be extracted from the encoded bit stream, without the necessity of decoding.

3. SUBJECTIVE TEST SETUP

In order to obtain the ground truth for constructing and validating our model, a subjective test was conducted using the Degradation Category Rating (DCR) methodology [8]. This methodology implies that the test sequences are displayed pairwise. During the experiment, both the original video sequence and the impaired version of it were shown simultaneously, one next to the other. Immediately after watching each pair of sequences, the user was required to rate the visual degradation between the impaired and the original sequence using a five-level scale ranging from 'imperceptible' to 'very annoying'. At the beginning of the subjective test, three training sequences were shown to the subjects to indicate the level of impairments they could expect.

We used four different standard video-only sequences of CIF resolution (352x288 pixels) which represent different content types: *akiyo*, *foreman*, *mobile* and *stefan*. Since we are focusing on predicting the visibility of network impairments only, we encoded the sequences at maximal image quality. Each sequence was encoded at 30 frames per second (fps) using 0, 2 and 3 consecutive B-pictures between two reference pictures. B-pictures were not used as reference. Two different GOP sizes were used: 17 in the case of 3 B-pictures and 16 in the case of 0 and 2 B-pictures. Every picture was encoded using only one slice.

The sequences were impaired using *xStreamer* [11], our

in-house developed modular multimedia streamer. Figure 1 depicts the configuration we used for impairing the sequences.

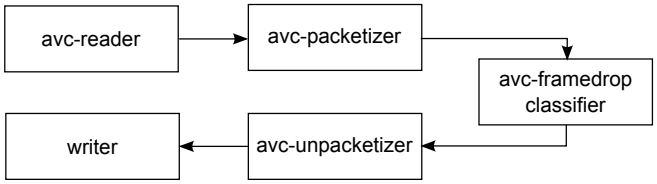

Fig. 1. RTP packets, which carry data from particular slices, are dropped using the *avc-framedrop-classifier* component. The resulting impaired sequence is saved to a new file after unpacketizing.

First, the raw H.264/AVC Annex B encoded bit stream is packetized according to RFC 3984 [12] into RTP packets. No aggregation is performed during packetization. As such, the loss of a RTP packet will never affect more than one slice. The *avc-framedrop-classifier* drops all RTP packets which carry data from a certain slice. As a slice is only decoded when it is received completely error-free, dropping only one RTP packet or all packets carrying data from the same slice always results in an undecodable slice. In our case, dropping a slice results in the loss of an entire picture. Finally, after unpacketizing the RTP packets, the resulting impaired raw H.264/AVC Annex B bit stream is saved to a new file.

An adjusted version of the JM reference software version 16.1 was used to decode the impaired sequences as the original version fails to process corrupted bit streams in all but the simplest cases. In the modified version, the decoder skips all frames before the first IDR frame it encounters. As a result, if the very first frame of a bitstream is lost, the entire first GOP will be skipped. Once the first IDR is processed, the decoder detects missing frames by means of gaps in the picture order count (POC). To do so, two variables are used: the difference between the POCs of any two successive frames and the distance between the POCs of two successive reference frames. The first value is used to detect when a non reference frame is lost. When a non reference frame is lost, it is replaced by the previous frame in the display order. The second value is used to detect when a reference frame is lost. In such a case, the reference frame is replaced by the nearest reference frame in time. Missing IDR frames are detected by checking the POC of the current frame with the POC with the last decoded or concealed reference frame. If this POC is lower, then an IDR frame is missing and needs concealing. In such a case, the last reference frame of the previous GOP is used.

Each encoded sequence was impaired using 10 times a single slice drop, 10 times two consecutive slice drops and 10 times three consecutive slice drops. Of these 10 streaming scenarios, the first dropped slice was randomly selected to be 4 times a B-slice, 4 times a P-slice and 2 times an I-slice. Slices were dropped in decoding order. This resulted in a total number of 312 impaired video sequences which were divided into 4 datasets of 78 sequences. The datasets were created to contain an equal number of impaired sequences for each of the four different content types. 35 subjects participated with the subjective test of which some of them evaluated more than one dataset. Each dataset was evaluated by exactly 20 subjects.

4. CONSTRUCTING AND EVALUATING DECISION TREES FOR VISUAL IMPAIRMENT DETECTION

In order to model the visibility of packet loss, we propose the use of a decision tree for its ease of understanding, interpretation and implementation. Furthermore, a decision tree can be regarded as a white box which enables us to completely understand the internal structure of the model. As we are targeting a real-time no-reference bit stream video quality metric, we only consider a minimal number of parameters which can easily be extracted from the encoded bit stream for building our classification tree. These parameters are listed and explained in Table 1. Four different content classes (A through D) were defined, based on the amount of motion in the sequence. Content class A corresponds with the sequence containing the lowest amount of motion, *akiyo* in our case. The *stefan* sequence was assigned class D as it contains the highest amount of motion. *Mobile* and *foreman* were divided into content classes B and C, respectively. In this article, we do not consider the problem of content classification. As such, this classification was performed as a pre-processing step and signaled inside the bit stream itself. Using methods described in [13] and [14] it is however possible to derive motion characteristics from the compressed video bit stream.

class	Content classification based on the amount of motion in the sequence
slice_type	Slice type of the corresponding dropped slice
impaired_pics	Number of impaired pictures due to the loss of a particular slice
cons_losses	Number of consecutive slice losses
cons_b_losses	Number of consecutive B-slice losses

Table 1. Parameters considered for building our decision tree. Only parameters which can be easily extracted from the encoded bit stream are taken into account.

We used the Waikato Environment for Knowledge Analysis (WEKA) [15], an open source data mining software package, for constructing our decision tree. Subjective results from three out of the four available datasets were used for training the decision tree and one dataset was used for validation. In order to evaluate the performance of the classifier,

the overall accuracy and the true positive (TP) rate are measured. The latter represents the percentage of visible packet loss that has been correctly classified as being visible and the percentage of invisible packet loss correctly classified as not perceivable.

In order to classify packet loss as visible or invisible, we used the same thresholds set by Kanumuri *et al.* [2]. As such, when 75% or more of the subjects gave a quality score of 5 to a sequence, the impairment was classified as invisible. Likewise, packet loss was classified as visible when 25% or less of the subjects provided a quality rating of 5 to a sequence. The remaining impairments, in our case about 20% of the entire training data, were also classified as indeterminable and not taken into account when building the classifier. It is important to mention that by using these two different thresholds, not every packet loss can be classified as visible or invisible. The resulting classifier is depicted in Figure 2.

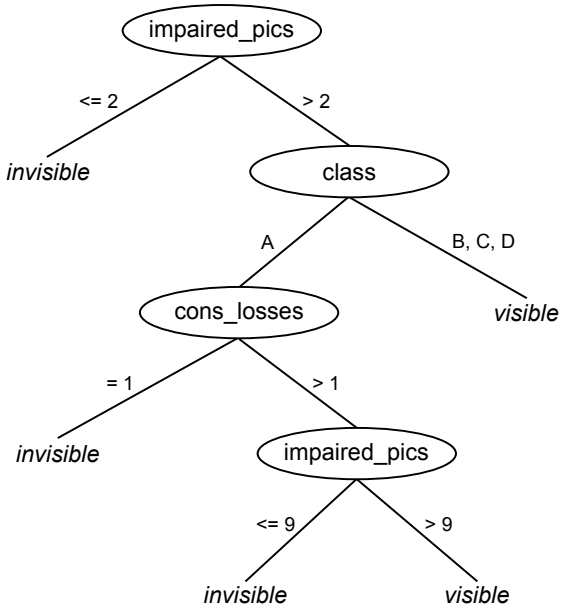

Fig. 2. Classification tree based on the same detection thresholds used in [2].

First, a split is made on the number of impaired pictures. This parameter corresponds with the visual impairment drift and can easily be predicted using the slice type of the dropped slice and the location of the loss within the GOP. In our case, *impaired_pics* less than or equal to 2 corresponds with 1 or 2 consecutive B-slice drops. As P-slices and I-slices are used as reference, losing such a slice results in an *impaired_pics* count larger than 2. Video content does not play a role when losing up to 2 consecutive B-slices. In the case of losing a P-slice or an I-slice, a split is further made based on content type. Packet loss is visible in sequences with medium and high amounts of motion. Only in the case of very low motion sequences (such as the akiyo sequence), packet loss visibility depends on the number of consecutive slice losses and the slice type. Losing only 1 slice does not result in a visible artifact, even if this dropped slice is a P-slice or an I-slice. When multiple consecutive slices are lost, impairment visibility depends on the location of the drops within the GOP. Multiple slice drops in the beginning of the GOP will be detected more rapidly as the temporal drift of the impairment is longer. Furthermore, when multiple slices are dropped, more error concealment must be performed due to the loss of additional reference pictures.

Our classifier has a 10 fold cross-validation accuracy of 93.16%, with TP rates for visible and invisible predictions of respectively 97.0% and 84.5%. When evaluating our constructed tree against the validation set, an overall accuracy of 85.92% is obtained. In this case, TP rates for visible and invisible classifications are 90.6% and 72.2% respectively. These results are also listed in Table 2 and show that the classifier is able to reliably predict the occurence of visual impairments. Furthermore, the overall tree size remains small and requires only 3 parameters for the classification.

In contrast with the results from [4], the amount of motion does influence the visibility of packet loss in our sequences. However, as the tree indicates, packet losses are only masked in very low motion sequences which then corresponds more with the results in [5]. The classifiers from [2], [3] and [4] all use parameters which need to be extracted from the video exactly at the location of the loss. In our case, packet loss always results in one or more pictures being dropped which, in turn, simplifies our classifier. As a result, only information is used which can be extracted from the captured bit stream without the need for deep packet processing.

	Test set	*Cross-validation*
Accuracy	85.92%	93.16%
TP visible	90.6%	97.0%
TP invisible	72.2%	84.5%

Table 2. Performance statistics of the decision tree when the subjective data is processed according to the visiblity thresholds from [2].

In this paper, we are also interested in constructing a decision tree capable of classifying each occurring packet loss as visible or invisible and thus avoiding classifying packet loss as indeterminable. Therefore, we classified packet loss as visible when 65% or more of the subjects rated the corresponding sequence less than 5, otherwise it was classified as invisible. This threshold is lower compared to the one used in [2] because we want to build a more stringent decision tree and avoid classifying packet loss as indeterminable. Furthermore, as we showed the original and the impaired video sequence pairwise during the subjective test, the influence of packet loss on visual quality is more rapidly perceived. Using this new threshold, the classifier depicted in Figure 3 was constructed. Performance statistics of this decision tree are

summarized in Table 3.

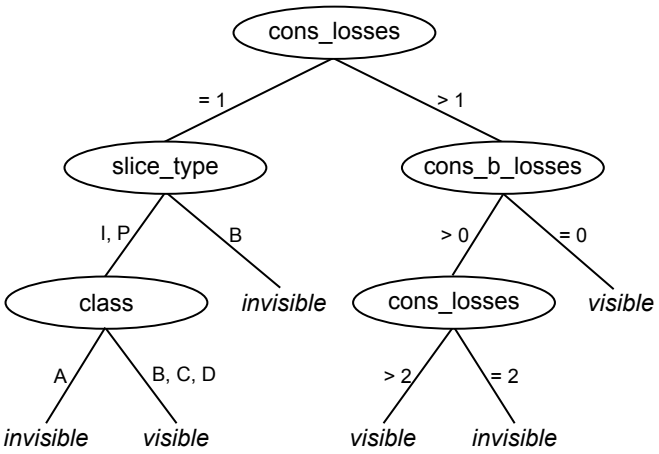

Fig. 3. Decision tree for predicting the visibility of slice drops due to packet loss.

This classifier makes a first decision based on the number of consecutive slice losses. When only one slice is lost, the type of that missing slice in combination with sequence content determines the visibility of the packet loss. As such, only the loss of a single P-slice or I-slice will result in a visible artifact in sequences with medium and high amounts of motion; losing a single B-slice never results in a visible impairment. In the case of multiple consecutive slice losses, a split is further made using the number of consecutive B-slice losses. Packet loss will be visible when multiple consecutive slices of reference pictures are lost. Visual artifacts will also be perceived when more than 2 consecutive B-slices are lost. Video content does not influence impairment visibility when losing multiple consecutive slices.

In contrast with the tree from Figure 2, the number of consecutive impaired pictures is not used during the classification process. Hence, only parameters which can be extracted during bit stream monitoring are taken into account.

	Test set	*Cross-validation*
Accuracy	85.90%	85.04%
TP visible	91.5%	94.6%
TP invisible	68.4%	61.8%

Table 3. Accuracy and TP rates for our classifier, based on a detection threshold of 65%. Performance was evaluated using a test set and 10 fold cross-validation.

An overall accuracy of 85% is obtained when validating the resulting tree using both the test set and 10 fold cross-validation. Based on the TP rates, the tree performs slightly better in correctly classifying packet loss. During real-time monitoring of visual quality, it is important not to misclassify packet loss as being invisible to often. In that sense, it is better to signal visible impairments optimistically rather than not detecting visual artifacts at all. Concerning the tree size, 11 nodes and 4 parameters are used to classify packet loss visibility.

Our two constructed decision trees classify packet loss visibility with a high accuracy. All parameters used in the decision process can be extracted while monitoring the video streams without the need for decoding or deep packet parsing. Furthermore, due to the reduced number of parameters and overall tree size, the classifiers can be used for detecting visual impairments in real-time.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a novel bit stream-based NR video quality metric which can be used for real-time video quality impairment detection. This enables service providers to monitor their video streams. Using data obtained from a subjective test, different decision tree classifiers were constructed and evaluated for classifying packet loss as visible or invisible.

Performance statistics show that our two classifiers obtain a high accuracy for classifying packet loss as visible or invisible. Furthermore, our proposed trees are small in size and use only a limited number of parameters.

Our classifiers only consider parameters which can easily be extracted from the encoded bit stream without deep packet inspection or decoding.

Using a lower detection threshold, packet loss can always be classified as being visible or invisible. In this case, the proposed intuitive decision tree performs well in detecting visual impairments caused by slice losses. Monitoring the received video bit stream is sufficient for extracting the parameters necessary for classifying packet loss visibility.

Our results also indicate that impairment visibility depends on sequence content. Static, low motion sequences are less sensitive to slice drops compared to medium and high motion sequences.

The work presented in this paper is a first step towards a full NR bit stream-based video quality metric capable of predicting the occurence of visual degradations with a high accuracy. In future work, we will investigate the influence of different content (ranging from CIF resolution up to High Definition) and different encoding settings (e.g. multiple slices per picture) on the visibility of packet loss. We are hereby targeting a NR bit stream-based metric which enables real-time visual quality monitoring of multiple video stream simultaneously.

6. ACKNOWLEDGEMENTS

The research activities that have been described in this paper were funded by Ghent University and the Interdisciplinary In-

stitute for Broadband Technology (IBBT). Nicolas Staelens would like to thank the Special Research Fund of Ghent University (BOF) for financial support through his PhD grant.

7. REFERENCES

[1] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

[2] S. Kanumuri, P.C. Cosman, A.R. Reibman, and V.A. Vaishampayan, "Modeling packet-loss visibility in MPEG-2 video," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 341–355, April 2006.

[3] T. Liu, X. Feng, A. Reibman, and Y. Wang, "Saliency inspired modeling of packet-loss visibility in decoded videos," *Fourth International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM-09)*, January 2009.

[4] S. Kanumuri, S.G. Subramanian, P.C. Cosman, and A.R. Reibman, "Predicting H.264 Packet Loss Visibility using a Generalized Linear Model," in *IEEE International Conference on Image Processing*, October 2006.

[5] A. R. Reibman and D. Poole, "Predicting packet-loss visibility using scene characteristics," in *Packet Video 2007*, nov. 2007, pp. 308 –317.

[6] N. Suresh and N. Jayant, "Mean Time Between Failures: A Functional Quality Metric for Consumer Video," in *First International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, January 2005.

[7] ITU-R Recommendation BT.500-11, "Methodology for subjective assessment of the quality of television pictures," International Telecommunication Union (ITU), 1992.

[8] ITU-T Recommendation P.910, "Subjective video quality assessment methods for multimedia applications," International Telecommunication Union (ITU), 1999.

[9] N. Suresh, N. Jayant, and O. Yang, "AVQ: A Zero-reference Metric for Automatic Measurement of the Quality of Visual Communications," in *Proceedings of the Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, January 2007.

[10] N. Suresh, R. Palaniappan, P. Mane, and N. Jayant, "Testing of a No-Reference VQ Metric: Monitoring Quality and Detecting Visible Artifacts," in *Proceedings of the Fourth International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, January 2009.

[11] A. Rombaut, N. Staelens, N. Vercammen, B. Vermeulen, and P. Demeester, "xStreamer: Modular Multimedia Streaming," in *Proceedings of the seventeenth ACM international conference on Multimedia*, 2009, pp. 929–930.

[12] S. Wenger, M.M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer, "RTP Payload Format for H.264 Video," February 2005.

[13] T. Yap-Peng, D.D. Saur, S.R. Kulkami, and P.J. Ramadge, "Rapid estimation of camera motion from compressed video with application to video annotation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 10, no. 1, pp. 133 –146, feb 2000.

[14] S. Jeannin and A. Divakaran, "Mpeg-7 visual motion descriptors," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 6, pp. 720–724, 2001.

[15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, vol. 11, no. 1, 2009.