Arquitetura embarcável para detecção de eventos sonoros utilizando inteligência artificial

Luiz Carlos Silva de Araújo Filho[1], **Carlos Maurício Seródio Figueiredo**[2]

[1]Escola Superior de Tecnologia – Universidade Estadual do Amazonas (UEA)
Manaus – AM – Brasil

[2]Samsung Ocean Center – Escola Superior de Tecnologia – Universidade Estadual do Amazonas (UEA) – Manaus – AM – Brasil

{lcsdafl.eng16,cfigueiredo}@uea.edu.br

Abstract. *Machine Learning Techniques have revolutionized monitoring applications in intelligent environments, regardless of what is being monitored. This article proposes an event detection pipeline based on audio signals. This solution starts from the problem that the monitoring of real environments must contain audios of interest and unknown audios, in sequences of very variable sounds, bringing the need to combine different learning techniques in a single model. In particular, we propose the combination of anomaly detection techniques, followed by audio segment classifiers and, later, a final classifier for sequences of events of interest. We also evaluated the performance of such a model on an embedded platform. Results show the feasibility of the model, with a general accuracy of 93.75% over a test dataset, and a prediction time of 0.45s on a popular embedded platform.*

Resumo. *Técnicas de Aprendizado de Máquina têm revolucionado as aplicações de monitoramento em ambientes inteligentes, seja qual for o tipo da grandeza monitorada. Este artigo propõe um pipeline de detecção de eventos baseado em sinais de áudio. Tal solução parte do problema de que o monitoramento de ambientes reais deve conter áudios de interesse e áudios desconhecidos, em sequências de sons muito variáveis, trazendo a necessidade de combinar diferentes técnicas de aprendizado em um só modelo. Particularmente, propomos a combinação técnicas de detecção de anomalia, seguida de classificadores de trechos de áudios e, posteriormente, um classificador final para sequências de eventos de interesse. Ainda, avaliamos o desempenho de tal modelo em uma plataforma embarcada. Resultados mostram a viabilidade do modelo, com acurácia geral de 93,75% sobre um dataset de teste, e tempo de predição de 0,45s em plataforma embarcada popular.*

1. Introdução

Sistemas Pervasivos e Ubíquos de monitoramento têm se tornado mais comuns conforme a popularidade de sistemas embarcados cada vez mais avançados e com a adoção de técnicas de aprendizado de máquina. Exemplos são câmeras inteligentes, assistentes digitais de voz e sensores diversos.

Redes Neurais Artificiais Profundas têm revolucionado aplicações de monitoramento devido a sua capacidade de detectar padrões em dados de entrada, sejam

esses imagens [Xin and Wang 2019], áudio [Purwins et al. 2019] ou outros sensores [Mohammadi et al. 2017]. Muito se vê do impacto de tais métodos em aplicações de visão computacional, onde avanços significativos foram vistos nas tarefas de detecção de objetos em imagens, como na competição *ImageNet* [Russakovsky et al. 2015], então logo começou-se a estudar se tal impacto seria semelhantes na área de Processamento Digital de Sinais, em particular com dados de áudio.

Nos últimos 4 anos, muitos trabalhos têm surgido propondo o uso de aprendizado profundo para a detecção de eventos de áudio [Stowell et al. 2015]. O objetivo principal de tais trabalhos consiste em identificar elementos emissores de sons ou ambientes característicos em áudios captados em diferentes contextos. Assim, de forma similar à popular competição ImageNet, para Visão Computacional, vem surgindo competições com focos nos avanços em processamento de áudio. Em particular, destacamos a competição DCASE (*Detection and Classification of Acoustic Scenes and Events*), onde diferentes problemas são propostos com bases de dados particulares a serem resolvidos da melhor forma pelo competidores. Tal competição tem duas vertentes principais, a de detecção de cenas de áudio (identificação dos ambientes onde se captura o áudio), ou de detecção de eventos (identificação do emissor de um som característico).

Muitos desses trabalhos focam na avaliação da acurácia ou métricas como F1 *Score*, que visam identificar a quantidade de detecções certas ou erradas dos modelos propostos, sobre bases de dados particulares. Esses são compostos de sons de um universo bem delimitado de detecção, muito diferentes da diversidade que podemos encontrar em situações reais. Assim, a detecção de eventos em uma situação real pode se deparar com muitos casos não previstos na base de dados de treinamento, afetando uma sequência de detecção. Para situações de instância de eventos desconhecidos, trabalhos da literatura usam algoritmos de detecção de anomalia, tais como [Koizumi et al. 2019]. No entanto, tais trabalhos não se preocupam com a tarefa de detecção do evento. Assim, o objetivo deste trabalho é propor uma arquitetura de detecção que combine as fases de detecção de anomalia e classificação de eventos de áudio e realizar um estudo da aplicação dos mesmos mensurando desempenho combinado de classificação e desempenho computacional em uma plataforma embarcada.

A Seção 2 apresenta os principais conceitos e referências bibliográficas relacionadas a este trabalho. A Seção 3 descreve os modelo utilizados e a Seção 4 apresenta os cenários de testes e os resultados obtidos. Por fim, a Seção 5 apresenta as considerações finais e trabalhos futuros.

2. Trabalhos Relacionados

Pesquisas no tema fazem parte da grande área de estudo de Processamento Digital de Sinais, particularmente, quando o sinal em questão é uma fonte sonora. Dois domínios iniciais de estudo envolveram Reconhecimento de Fala e Recuperação de Informação Musical, apresentando avanços significativos e com aplicações práticas. Esses avanços mostraram a possibilidade de aplicações em deteções mais gerais, sendo inclusive alvo de Desafios Científicos para estimular seu progresso, como na competição anual DCASE, promovido pelo Comitê Técnico do *IEEE Audio and Acoustic Signal Processing*,

A grande maioria dos resultados mais recentes apresentados na competição previamente mencionada aplicam técnicas de *Deep Learning*, como visto em [Chen et al. 2019],

o qual utilizou uma rede neural convolucional para vencer a tarefa de *Acoustic Scenes Classification* (ASC) da edição de 2019. A arquitetura da rede, no entanto, é muito profunda, sendo baseada na rede canônica VGG [Simonyan and Zisserman 2014a], originalmente projetada como uma rede para classificação de imagens, mas é uma arquitetura que costuma ser recondicionada para diversas finalidades, nesse caso para processamento de sinais sonoros. Essa abordagem costuma ser utilizada consistentemente ao longo dessa competição, podendo ser observada também em [Sakashita and Aono 2018] (vencedor da tarefa de ASC de 2018), em [Lin and Wang 2019] (vencedor da tarefa de *Sound Event Detection* (SED) em ambiente doméstico de 2019).

Ainda observando esses resultados, é possível perceber tendências direcionando pesquisas para abordagens que envolvam *ensembles*, ou seja, utiliza comitês de modelos inteligentes para, no fim, receber um resultado com taxa de acerto maior a partir de uma decisão entre os resultados individuais obtidos. Tanto [Chen et al. 2019] quanto [Sakashita and Aono 2018] e ainda [Kapka and Lewandowski 2019] (vencedor da tarefa de detecção e identificação de eventos sonoros) utilizam essa metodologia, com diversas combinações de abordagens.

Outro aspecto comum entre essas abordagens é a forma que os sinais foram pré-processados. Todos utilizaram alguma variação de representação do sinal no domínio da frequência, seja utilizando uma transformada de Fourier de tempo curto (STFT), ou analisando um cepstrum de frequências-mel.

Abordagens mais simples podem ser encontradas em edições anteriores da competição, como pode ser visto em [Eghbal-Zadeh et al. 2016], vencedor da tarefa de ASC, o qual utilizou uma combinação de extração de características com a utilização de uma rede baseada no VGG [Simonyan and Zisserman 2014b] e utilizando o sinal de entrada após a aplicação da STFT, bem como são analisadas diferentes abordagens em [Stowell et al. 2015], todas envolvendo Coeficientes de Cepstral de Frequência Mel(MFCC), ou ainda apresentam caminhos que acabam se aliando a outras metodologias em submissões mais recentes, como em [Adavanne et al. 2016] que utilizam Redes Neurais Recorrentes para classificar eventos sonoros.

Tratando-se de detecção de anomalias, há um *survey*, [Chandola et al. 2009], que apresenta algumas abordagens que já foram utilizadas para esses fins, divididas em possíveis aplicações. Levanta-se uma atenção maior para as seções que tratam das técnicas de detecção baseadas em classificação e baseadas em K-Vizinhos Mais Próximos (KNN). A primeira busca agrupar os dados em classes e utilizar técnicas comuns de classificação para cumprir a sua tarefa, enquanto que a última consiste em medir, em um espaço hipotético criado pelo modelo para posicionar os dados para os quais ele foi treinado, a distância entre as novas amostras que estão sendo testadas e as amostras para o qual ele foi treinado.

Analisando pesquisas mais recentes, é possível perceber que houve avanços em termos de detecção de anomalias, como descrito em [Munir et al. 2018], que mostra uma abordagem para detecção de anomalias em séries temporais, desse modo pertencendo ao mesmo domínio de dados que áudios, baseada em Redes Neurais Convolucionais (CNNs), apesar da literatura [Hochreiter and Schmidhuber 1997] normalmente apontar para Redes Neurais Recorrentes (RNNs), mais especificamente *Long Short-Term Memory* (LSTMs),

para tratar de dados que possuem relações temporais entre si.

3. Materiais e Métodos

Nesta seção estarão descritos a problemática particular que foi utilizada para fins de validação da solução, o dispositivo em que esta implementação foi embarcada, a disposição, organização, natureza e origem dos dados utilizados durante os testes e a própria solução em forma de arquitetura, bem como cada um dos processos presentes nela, descrevendo ainda a forma que estes foram implementados para realizar os testes de prova de conceito.

3.1. Problemática

Considerando o que foi levantado na Seção Trabalhos Relacionados, este trabalho tratará de propor uma arquitetura para um sistema capaz de identificar sons característicos de um evento considerado como invasão em uma situação específica, de tal forma que funcionasse como um sistema de segurança ou alarme para situações como de uma furadeira tentando perfurar um cofre, ou de alguém tentando arrombar um cadeado, dependendo apenas de quais situações serão consideradas.

Foi levado em conta que este sistema seria embarcado em algum dispositivo, como um microcomputador, portanto houve preocupação com o desempenho que o sistema apresentará durante sua utilização.

3.2. Dispositivo

Por ser de fácil acesso e baseado em um sistema operacional de simples utlização e compatível com boa parte das bibliotecas que serão utilizadas, será utilizado um Raspberry PI 3B+, o qual possui o qual possui um processador *quad-core* com 1.4GHz, 1GB de memória RAM e um *micro-SD* de 16GB, sendo este o modelo mais recente na época dos testes.

Essa escolha foi dada, também, em função de seu custo reduzido em relação ao seu poder computacional, sendo esse capaz de comportar o sistema que será descrito posteriormente neste artigo.

3.3. Dados Utilizados

Para fins de demonstração da solução, a base de dados será composta de três classes que podem ser consideradas um ataque: Furadeira, Lixadeira e Maçarico; além de uma classe que engloba sons esperados e conhecidos que aconteçam normalmente.

Essa base de dados foi obtida através de sessões de gravações que representem uma simulação de uma situação real de invasão de forma relevante para o treinamento da inteligência artificial, isto é, simulando uma invasão externa, foram gravadas sequências para cada classe, cada uma contendo de 20 a 100 amostras. Como trata-se de uma gravação em tempo real, é inevitável que, em alguns momentos, houvesse áudios que não fossem relevantes para o objetivo, sendo assim foi feita uma seleção manual que consistiu em ouvir cada áudio individualmente e remover qualquer sinal de silêncio, ruído ou áudio que não representasse a classe em questão. Ao fim desta seleção, a base de dados ficou com um total de 1579 amostras, as quais foram divididas através de um *Holdout* em proporção de 70% para treino, 20% para teste e 10% para validação.

Para o treinamento do modelo de anomalias, bastou utilizar os áudios mencionados anteriormente, ou seja, dos ataques e do comportamento normal esperado e, ao medir o *score* obtido por um áudio anômalo, seria possível perceber uma diferença muito grande quando comparado com um dos áudios para o qual foi treinado. Para se obter estas métricas, foi utilizado um *subset* (ESC-10) da base de dados ESC-50 [Piczak]. Este subset engloba um grupo de 400 amostras, sendo composto por 10 classes com 40 amostras por classe. Para fins de validação da detecção de anomalias, o total foi reduzido para 15 áudios de 5 segundos cada, escolhidos aleatoriamente entre as 10 classes, e, como a classificação é feita com amostras de um segundo, cada áudio foi considerado uma sequência e foi repartido em 5 novas amostras de 1 segundo.

3.4. Arquitetura

O problema de SED envolve alguns entraves que devem ser levados em conta, principalmente quando aplicados a situações de detecção de invasão em tempo real. Para tanto, propõe-se uma arquitetura baseada em um *pipeline*, representado pelo diagrama mostrado na Figura 1, o qual executa as seguintes funções: recebe uma amostra de um segundo do ambiente, pré-processa essa amostra, envia esse dado como entrada para o detector de anomalias e, caso esse dado seja identificado como anomalia, é enviado diretamente para uma lista de 10 posições, do contrário essa mesma entrada será alimentada para o classificador de 1 segundo e a saída deste será inserida nessa lista e, caso esta ainda não esteja completa, repete os mesmos processos, desde a captura da amostra. Caso já esteja cheia, é usada como entrada para um classificador de 10 segundos que apresentará a saída final do *pipeline*.

Para fins de demonstração da viabilidade, uma implementação básica foi utilizada, sendo esta treinada e validada utilizando a base de dados descrita na Seção 3.3. A título de comparação, técnicas de *Machine Learning* mais complexas também foram empregadas para testar tanto a sua acurácia quanto seu desempenho computacional.

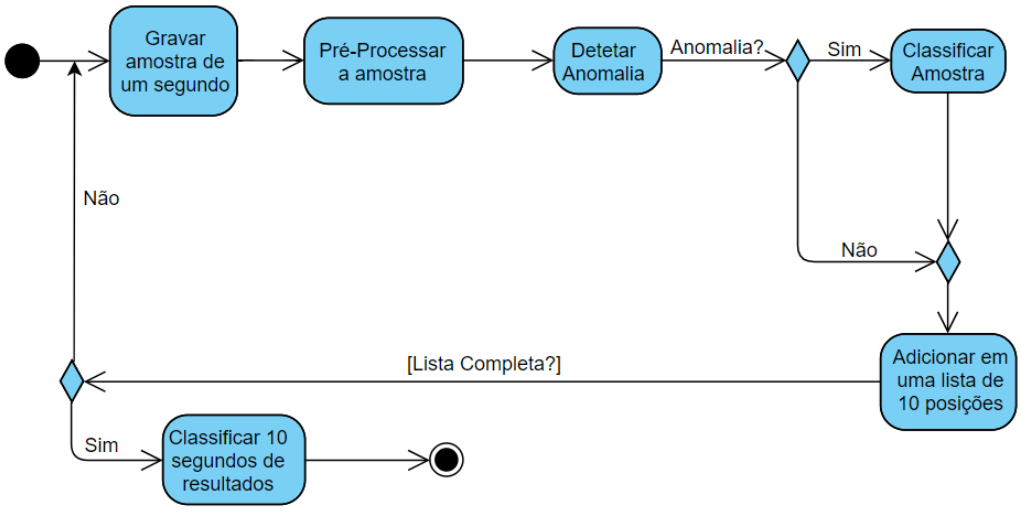


Figura 1. *Pipeline* proposto para o sistema de detecção eventos sonoros em tempo real

3.4.1. Pré Processamento

Considerando o que normalmente é utilizado para este tipo de problema e as particularidades deste, foi utilizada extração de características nesta etapa do processo, mais especificamente foi realizada a extração de 64 *Mel-Frequency Cesptral Coefficients* (MFCCs), utilizando taxa de amostragem de 16000hz. Todos os dados foram normalizados. Este pré processamento foi empregado tanto para o classificador MLP quanto para o detector de anomalias KNN e autoencoder.

Para o classificador CNN foram extraídos espectrogramas utilizando STFT, seguidos de extração de bancos de filtro log-Mel, os quais formaram um espectrograma que foi utilizado como entrada para a rede. Apenas a parte real dos espectrogramas foi considerada.

3.4.2. Classificação de 1 segundo

Apesar da literatura apresentada neste artigo apontar para o estado da arte em uma direção envolvendo *Deep Learning* e redes neurais convolucionais, a quantidade de dados normalmente utilizada é consideravelmente maior do que foi possível obter, ainda mais considerando a natureza da captura, torna-se difícil conseguir uma quantidade relevante para que uma solução profunda seja capaz de generalizar os padrões. Sendo assim, ainda foi utilizada uma rede neural, mas dessa vez uma Rede Perceptron de Multicamadas (MLP).

A rede é constituída de quatro camadas totalmente conectadas, com uma camada de *dropout* entre cada uma delas. As camadas possuem, respectivamente, 128, 64, 32 e 4 nós. A função de ativação utilizada em cada camada foi a ReLu [Agarap 2018].

Foi treinada durante 100 épocas, com *batches de tamanho 8*. Ao fim do treino, utilizando o *split* de validação, ou seja, com dados totalmente novos, foi possível se obter uma acurácia de 99% para 171 amostras das 4 classes.

Como a acurácia alcançada já é muito perto de 100%, ao testar uma rede mais complexa, já é esperado que a acurácia seja maior, mas por uma margem mínima. Ainda assim, foi implementada a arquitetura baseada em CNN encontrada em [Sakashita and Aono 2018], a fim de comparar a eficiência entre as duas soluções.

3.4.3. Detecção de Anomalias

Um problema encontrado nas soluções baseadas em classificação é a situação na qual o áudio capturado trata-se de um caso totalmente alheio ao treinamento da rede, ou seja, é uma classe nova, que não havia sido planejada ou prevista durante o treinamento e, ainda que houvesse um esforço dedicado à isso, para alguns problemas é simplesmente impossível agregar todas as diferentes possibilidades, como é o caso do problema proposto neste artigo: tratando-se de classificação baseada em áudio, com certeza haverá algum momento que um som diferente será apresentado ao sistema: um cachorro latindo, por exemplo, causaria uma classificação certamente errada, mas com comportamento imprevisível por parte da rede.

Pensando nisso que é notável a necessidade de tratamento para estas anomalias de

forma genérica. Uma das formas de fazê-lo é utilizando uma biblioteca em python chamada *PyOD*, uma biblioteca que disponibiliza diversos métodos de detecção de *outliers*, que, no caso deste trabalho, serão considerados anomalias.

Utilizar apenas um modelo de K-vizinhos mais próximos (KNN) para determinar anomalias pode ser muito suscetível à *overfitting*, sendo assim para este teste serão treinadas 20 versões de KNNs, com a quantidade de vizinhos variando de 10 a 200, incrementando em 10 para cada nova versão e, tendo todos os resultados, será utilizada a média para agregar os resultados de cada um dos modelos e, por fim, decidir se a amostra trata-se ou não de uma anomalia.

Novamente, a fim de comparação, ainda utilizando a mesma ferramenta foi criado um *autoencoder* para este mesmo fim, com a seguinte configuração de camadas: 25, 10, 2, 10 e 25 nós totalmente conectados.

3.4.4. Classificação de 10 segundos

Após 10 segundos de gravação e classificação, esses dados serão agrupados na forma de uma lista com cada um dos dez resultados individuais ao classificador de 10 segundos, que decidirá a saída final do *pipeline*, informando a natureza desse período de gravação. A natureza da saída dessa classificação depende exclusivamente da especificidade do problema a ser tratado.

Para este protótipo, foi utilizado um critério de maioria simples, identificando entre as quatro classes para as quais o modelo foi treinado e uma outra classe que engloba qualquer anomalia detectada.

4. Resultados Obtidos

Como já era esperado, quando alimentada para predizer a classificação dos áudios anômalos, a rede não apresenta uma resposta suficiente para a natureza do áudio, sendo assim ela basicamente tenta adivinhar o que o áudio pode ser dentre as opções para a qual ela foi treinada. Algo que vale ser ressaltado é que, dentre os 99 áudios anômalos que estavam sendo utilizados, 17 foram classificados como furadeira, 42 como lixadeira e 40 como Maçarico, mas nenhum como comportamento normal.

É possível verificar todos os resultados obtidos, para todos os modelos testados, na Tabela 1. Conforme o disposto nos Trabalhos Relacionados, geralmente os melhores resultados não são obtidos ao se utilizar uma MLP para classificar eventos sonoros e KNN para detecção de anomalias, no entanto, talvez por conta do escopo menor, ou da quantidade limitada de dados, as diferenças ou foram mínimas, no caso dos classificadores, ou até, de certo modo melhores, no caso da detecção de anomalias, onde foi possível observar que, apesar de se obter uma acurácia menor, seu F1-score foi consideravelmente melhor.

Os testes para validar esta proposta foram feitos no dispositivo descrito na Seção 3.2. Foram feitos testes de tempo de carregamento individual dos modelos testados, a quantidade de memória consumida, tempo para pré processar uma amostra de áudio e tempo de uma predição. Estas medidas não foram feitas para o classificador de 10 segundos pois trata-se apenas da escolha do resultado com maior quantidade de

Tabela 1. Acurácia, Recall, Precisão e F1-score obtidos para cada um dos modelos propostos

	MLP - Classificação 1 segundo	KNN - Detecção de Anomalias	CNN - Classificação 1 segundo	AutoEncoder - Detecção de Anomalias
Acurácia	98.83	81.71	99.40	87.97
Recall	98.50	98.46	99.25	93.75
Precisão	99.00	74.85	99.50	37.97
F1-Score	98.75	85.04	99.50	54.04

ocorrências, não dependendo de qualquer recurso de inteligência artificial. Estes dados estão dispostos na Tabela 2.

Tabela 2. Tabela com dados de desempenho utilizando um Raspberry PI 3B+

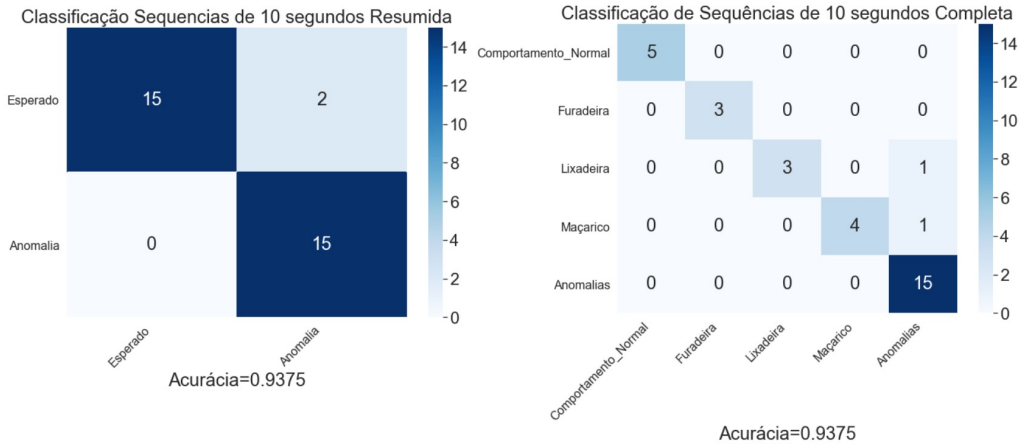
	MLP - Classificação 1 segundo	KNN - Detecção de Anomalias	CNN - Classificação 1 segundo	AutoEncoder - Detecção de Anomalias
Tempo de Carregamento (s)	4.03	0.21	8.12	2.12
Tempo de Pré-Processamento (s)	0.09		0.02	0.09
Tempo para uma predição (s)	0.27	0.09	1.95	0.05
Memória utilizada pelo modelo (MB)	637.3	24.0	2726.3	849.2

Certamente é notável que, quanto mais complexa a abordagem para solucionar o problema, mais computacionalmente intensiva ela se torna. Considerando o exposto nas Tabelas 1 e 2, o maior consumo de recursos não é justificado pelo aumento na taxa de acertos do modelo, sendo assim para o teste do pipeline inteiro serão utilizados como classificador e detector de anomalias o modelo MLP e KNN, respectivamente.

Tendo em vista todo o fluxo do *pipeline* (conforme na Figura 1) e os modelos selecionados, o teste completo foi subseguido da seguinte forma: os dados foram alimentados em sequências de 5 a 10 segundos. O objetivo inicial era, utilizar sequências de 10 segundos, no entanto os dados anômalos encontravam-se todos em sequências de 5 segundos. Tratando-se de um cenário real, sempre seria possível conseguir coletar os 10 segundos de áudio, mas para todos os fins, os resultados seriam similares. Como as sequências de ataque não foram gravadas como múltiplos de 10 segundos, haviam sobras que não completavam os 10 segundos e, para manter um padrão, caso restasse menos de 5 segundos de áudio ao fim da amostra de ataque, estes seriam descartados.

De acordo com a Tabela 3, é perceptível que há uma queda na acurácia quando se utiliza um detector de anomalias, justamente porque é sempre melhor treinar a rede para a maior quantidade de classes que seja possível a coleta de dados. Ainda assim, a queda de desempenho não é grande o suficiente para inviabilizar seu uso. Uma questão que deve ser levada em conta, é que há particularidades dependendo da natureza dos problemas de

Tabela 3. Matriz de Confusão para todo o *pipeline*. Os resultados correspondem aos testes com MLP e KNN



detecção de invasão e, dependendo de como o sistema vai tratar esse aviso de ataque, deve haver uma maior preocupação em se tratar os falso positivos ou falso negativos.

5. Considerações Finais

A busca pelo estado da arte é natural e o esperado quando se procura a solução para um problema, no entanto concessões devem ser tomadas quando se leva em conta necessidades requeridas quando o problema envolve um sistema embarcado. Os resultados obtidos, apesar de em um escopo menor, servem para mostrar que é possível utilizar este sistema com múltiplos processos de forma eficaz e eficiente e, além disso, também há casos onde métodos mais simples obtêm melhores resultados. Conforme fosse possível utilizar métodos mais complexos em dispositivos portáteis, e ainda mesmo com a tecnologia atual, seria interessante rever as etapas do processo a fim de conseguir melhorar seu desempenho. Utilizar outras formas de classificar as amostras de um segundo ou de identificar anomalias, além de utilizar algum outro tipo de modelo inteligente para efetuar a classificação dos dez segundos é algo esperado e que deve ser almejado para utilizações práticas desta solução, mas devem sempre ser escolhidos levando em conta todas as particularidades que determinado problema traz consigo.

Agradecimentos

Este trabalho recebeu apoio da FAPEAM e CNPq por meio do Programa PPP 04/2017.

Referências

Adavanne, S., Parascandolo, G., Pertilä, P., Heittola, T., and Virtanen, T. (2016). Sound event detection in multichannel audio using spatial and harmonic features. Technical report, DCASE2016 Challenge.

Agarap, A. F. (2018). Deep learning using rectified linear units (relu).

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, 41.

Chen, H., Liu, Z., Liu, Z., Zhang, P., and Yan, Y. (2019). Integrating the data augmentation scheme with various classifiers for acoustic scene modeling. Technical report, DCASE2019 Challenge.

Eghbal-Zadeh, H., Lehner, B., Dorfer, M., and Widmer, G. (2016). CP-JKU submissions for DCASE-2016: a hybrid approach using binaural i-vectors and deep convolutional neural networks. Technical report, DCASE2016 Challenge.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Kapka, S. and Lewandowski, M. (2019). Sound source detection, localization and classification using consecutive ensemble of crnn models. Technical report, DCASE2019 Challenge.

Koizumi, Y., Saito, S., Uematsu, H., Kawachi, Y., and Harada, N. (2019). Unsupervised detection of anomalous sound based on deep learning and the neyman–pearson lemma. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(1):212–224.

Lin, L. and Wang, X. (2019). Guided learning convolution system for dcase 2019 task 4. Technical report, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China.

Mohammadi, M., Al-Fuqaha, A., Sorour, S., and Guizani, M. (2017). Deep learning for iot big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, PP.

Munir, M., Siddiqui, S., Dengel, A., and Ahmed, S. (2018). Deepant: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access*, PP:1–1.

Piczak, K. J. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press.

Purwins, H., Li, B., Virtanen, T., Schluter, J., Chang, S.-Y., and Sainath, T. (2019). Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):206–219.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Sakashita, Y. and Aono, M. (2018). Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions. Technical report, DCASE2018 Challenge.

Simonyan, K. and Zisserman, A. (2014a). Very deep convolutional networks for large-scale image recognition.

Simonyan, K. and Zisserman, A. (2014b). Very deep convolutional networks for large-scale image recognition.

Stowell, D., Giannoulis, D., Benetos, E., Lagrange, M., and Plumbley, M. D. (2015). Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia*, 17(10):1733–1746.

Xin, M. and Wang, Y. (2019). Research on image classification model based on deep convolution neural network. *EURASIP Journal on Image and Video Processing*, 2019(1):40.