AITIA: Embedded AI Techniques for Embedded Industrial Applications

Marcelo Brandalero[1], Muhammad Ali[2], Laurens Le Jeune[3], Hector Gerardo Muñoz Hernandez[1], Mitko Veleski[1], Bruno da Silva[45], Jan Lemeire[45], Kristof Van Beeck[3], Abdellah Touhafi[45], Toon Goedemé[3], Nele Mentens[3], Diana Göhringer[2], Michael Hübner[1]

[1]*Chair of Computer Engineering, Brandenburg University of Technology Cottbus-Senftenberg, Germany*
[2]*Chair of Adaptive Dynamic Systems, Technical University Dresden, Germany*
[3]*Dept. of Electrical Engineering (ESAT) at KU Leuven, Belgium*
[4]*Dept. of Industrial Sciences (INDI), VUB, Brussels, Belgium*
[5]*Dept. of Electronics and Informatics (ETRO) at VUB – imec, Belgium*

***Abstract*—New achievements in Artificial Intelligence (AI) and Machine Learning (ML) are reported almost daily by the big companies. While those achievements are accomplished by fast and massive data processing techniques, the potential of embedded machine learning, where intelligent algorithms run in resource-constrained devices rather than in the cloud, is still not understood well by the majority of the industrial players and Small and Medium Entereprises (SMEs). Nevertheless, the potential embedded machine learning for processing high-performance algorithms without relying on expensive cloud solutions is perceived as very high. This potential has led to a broad demand by industry and SMEs for a practical and application-oriented feasibility study, which helps them to understand the potential benefits, but also the limitations of embedded AI. To address these needs, this paper presents the approach of the AITIA project, a consortium of four Universities which aims at developing and demonstrating best practices for embedded AI by means of four industrial case studies of high-relevance to the European industry and SMEs: sensors, security, automotive and industry 4.0.**

***Index Terms*—artificial intelligence, machine learning, embedded hardware, sensors, network intrusion detection, driver assistance, industry 4.0**

I. Introduction

AI and ML techniques are pervading all devices and technologies, with intelligent data processing being brought closer to the embedded systems to sustain latency and security requirements. Even in the automotive industry, machine learning is used to equip camera systems with intelligent features. In industrial automation, machine learning is used e.g. in model predictive maintenance or also in model predictive closed control loops. Recently a highly relevant project showed the importance of machine learning in the domain of high throughput bottle filling machines. Another fully new domain for machine learning is mass spectrometry or near infrared spectrometry where trained networks support the fast detections of chemical ingredients.

This work is supported by COllective Research NETworking (CORNET). The Belgian partners are funded by VLAIO under grant number HBC.2018.0491, while the German partners are funded by the BMWi (Federal Ministry for Economic Affairs and Energy) under IGF-Project Number 249 EBG.

In Flanders and Belgium we see an enormous growth of startups focusing on the creation of smart products who might benefit from the inclusion of machine learning techniques into their designs. Belgium counts about 2300 startups active in the creation of software or hardware for smart Information and Communication Technology (ICT) products. Also well established companies within different sectors see the need for stepping up to the innovation flow of smart objects with integrated machine learning. In Germany, a growing number of startups, especially in the domain of security, embedded systems, internet of things and industrial automation, need support for the development of machine learning algorithms and integration on embedded systems. Recently, the high demand from SMEs in the domain of machine learning using reconfigurable architecture was underlined. There is currently a high demand for new end-to-end solutions that practically address all these requirements. We envision one such a platform in this project, which is described in Fig. 1.

Currently there is a large gap between the achievements made in research and what is used in real industrial applications. The SMEs in the user group and in the broader target group do not have the R&D capacity to invest the necessary effort to adopt AI technology in their products. Nevertheless, the companies are convinced that the technology will improve their products and lead to a significant increase in their turnover. In this work, we present the AITIA project, a consortium of four European universities that aims to bridge the gap between the academic knowledge of solutions for embedded AI and ML and industry products.

II. Related Work

Current tools for developing algorithms using machine learning are mainly from academic groups. Therefore almost all solutions are not really capable to be used in the development of industrial products. Example for such tools are Tensor flow and Caffee. Certainly there are some solutions available but they target mainly one specific target hardware and mostly only one domain. It is fully unclear for SMEs how to start with a machine learning approach for a given

978-1-7281-6371-0/20/$31.00 ©2020 IEEE

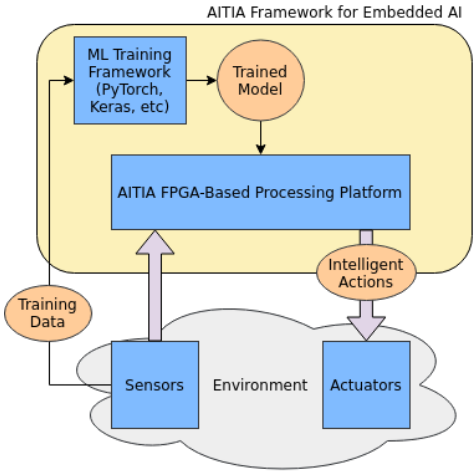

Fig. 1. Overview of the AITIA Framework for Embedded AI.

problem. This begins with the problem analysis, definition of input data, definition and selection of the network, training method and size of training data sets and, finally, the selection of the right reconfigurable hardware and its implementations. This knowledge, present in academics, should be synchronized with the SMEs. The benefit is that SMEs can grow with this knowledge and provide highly attractive positions for graduates of the universities.

Many works have targeted the execution of AI/ML algorithms in Field-Programmable Gate Arrays (FPGAs). Different frameworks for mapping arbitrary ML algorithms (mostly neural networks) have been proposed to that end [1], [2]; recent work discusses those and others comprehensively [3]. The execution of both Recurrent Neural Networkss (RNNs) [4]–[7] and Convolutional Neural Networkss (CNNs) [3], [8]–[13] has been covered in recent works. Optimization techniques to improve the performance and/or energy efficiency of these FPGA-based architecture include quantization [13] and partial reconfiguration [14].

The continuous report of new achievements suggest that a great deal of open challenges still need to be addressed in order to bring the applications of ML from academia to the SMEs.

III. The Four AITIA Use-Cases

This section describes the four use-cases that will be addressed within the scope of AITIA: *smart sensing*, *network intrusion detection*, *driver assistance* and *industry 4.0 management*. In smart sensing, ML is leveraged to augment sensor capabilities. For Network Intrusion Detection, ML offers a stronger capability of detecting intrusions when compared to traditional methods. AI can also provide better assistance to drivers in different automotive aspects. Finally, when used in an Industry 4.0 setting, ML offers improved management of sensors and actuators.

A. Smart Sensing

The use of near-sensor ML has the potential to enable new capabilities for industry and consumers alike [15]. The needs for low latency and efficient bandwidth consumption of many applications can be alleviated by bringing intelligence to the network's edge. Furthermore, such smart devices are capable of locally process the sensor's data, reducing the overall power consumption, optimizing the bandwidth usage and preserving privacy.

1) Motivation: Heterogeneous sensor arrays, composed of different type of sensors (infra-red, visual cameras, acoustics, ...) provide valuable multi-sensory information. The intelligent processing of this multi-sensory information, by using ML techniques, enable new smart sensing capabilities for industrial applications [16], [17].

Here, we consider the use of ML techniques for smart sensing in order to improve the sensors' quality-of-service (QoS), to enable sensors' anomaly detection or to perform acoustic event classification. The target heterogeneous sensor array is a multi-mode acoustic camera [18] which combines visual information with acoustic signals acquired by a microphone array. A SoC FPGA processes the information from both type of sensors which combination is displayed in a heatmap format, where the visual and the acoustic information are overlayed. The following functionalities are targeted:

- **QoS improvement**: While even low-quality visual cameras provide relative large resolutions, large acoustic image resolutions lead to high computational cost, which decreases the overall number of frames-per-second. As an alternative, the quality of acoustic images will be improved in terms of resolution by exploiting deep learning algorithms.
- **Anomaly detection**: The microphone arrays offer redundant information which can be used for anomaly detection [19]. Different ML techniques will be evaluated to exploit this multi-sensory information.
- **Embedded classification**: The multi-sensory information that heterogeneous sensor arrays provide enable the capability to recognize events which otherwise could not be detected by using single sensors. Existing ML techniques will be evaluated to process this multi-sensory information for detection and recognition of acoustic events in urban environments.

2) Challenges: Deep learning algorithms, like the one applied in [20] for infra-red and visual cameras, are good candidates for image upscaling. Nonetheless, the lack of datasets with acoustic images demands additional effort, since labelled data is needed for the ML training. Although many solutions has been proposed for anomaly detection [21], there is a lack of datasets for the training stage. We will exploit our sensor array [18] to generate our own datasets through real recordings. This will allow us to detect defective behaviour such as complete microphone failures, non-linearly corrupted microphones and other anomalies like studied in [22]. The audio signal acquired by the microphone array will be used

for sound classification. The selection of the most suitable ML techniques for specific smart sensing applications is not trivial. For instance, different ML techniques can be used for sound recognition achieving similar accuracy. Existing datasets of urban sounds can be used for training ML techniques such as k-nearest neighbors algorithm (k-NN), support vector machines (SVM), etc. Moreover, the time needed for the recognition is usually not reported [23]. Our target solution will not only offer a high accuracy in recognizing events but it will also present the shortest execution time.

3) Evaluation: Metrics such as accuracy, precision, recall of F1-score will be used to evaluate each use case. Parameters such as performance (e.g. in frames-per-second), resources and power consumption will also determine the most interesting solutions. Additionally will be considered:

- **QoS Improvement**: ML techniques are expected to outperform non-AI techniques such as Bilinear interpolation by using the root mean square error (RMSE) or the peak signal-to-noise ratio (PSNR) for measuring the interpolation error [24].
- **Anomaly Detection**: AI-based solutions are expected to allow the detection of anomalies, such as sensor malfunctioning, which would not be possible with non-AI based techniques.
- **Embedded classification**: A higher performance (in terms of execution time and/or frames-per-second) is expected thanks to running ML techniques on reconfigurable architectures, outperforming alternative embedded solutions while offering similar or even higher accuracy than the state-of-the art.

B. Network Intrusion Detection

1) Motivation: For the use-case about security, we chose to examine the application of machine learning in network intrusion detection. Such intrusion detection sytems (IDS) aim to detect intrusions or attacks against a computer system [25]. Two different types exist: network-based intrusion detection systems (NIDS) which detect intrusion in a computer network, and host-based intrusion detection systems (HIDS) that detect intrusions on a specific host. For such systems, we aim to investigate if the use of machine learning could leverage their performance. Currently, NIDS implementations are rule-based. Examples of this are Zeek [26] (previously known as Bro [27]), Snort [28] or Suricata [29]. These rule-based approaches only protect against attacks that are explicitely described in the rules, which leaves the network susceptible to unknown attacks. Here, machine learning might be able to provide protection by automatically learning new attacks, instead of only relying on specific rules.

2) Challenges: Network intrusion detection is no trivial task, for a number of reasons. Firstly, two different methods are used to perform detection, each with their own specific advantages and disadvantages. On the one hand, misuse-based detection methods use knowledge of existing attacks to detect intrusions. On the other hand, anomaly-based intrusion detection systems identify attacks by their deviation from normal network behaviour [25]. Misuse-based systems are good at accurately detecting the attacks they know, but are unable to detect unknown or day-zero attacks. On the contrary, an anomaly-based IDS has the ability to detect attacks without needing to actually know how a specific attack behaves. However, as not all anomalous behaviour in network traffic corresponds to an attack, anomaly-based systems have a high false postive rate.

Secondly, it is hard to obtain realistic and representative datasets [30]. The datasets that are used most often (KDDCup1999 [31] or NSL-KDD [32]) are based on a dataset generated in 1998 (DARPA1998 [33]). Not only does this imply that the attacks featured in those datasets are outdated, it also means that newer attacks (such as distributed denial-of-service) are not present. More recent datasets are available [34], [35], but are not widely adopted yet. If machine learning implementations of NIDS are designed, they need to be trained and evaluated on relevant datasets.

Finally, while sophisticated implementations of NIDS exist, achieving high throughputs on a single software-based device remains very difficult [36]. Hardware-based approaches are necessary to obtain a higher throughput. Therefore, the goal of this research is to create a machine learning based hardware implementation.

3) Potential approaches: There are several potential approaches to create a NIDS. One intuitive way would be to design a new NIDS model from the ground up. By taking previous results and other successful approaches into account, this new NIDS could potentially combine state-of-the-art techniques with new contributions. While this could lead to good results, it would be a very time-intensive undertaking. With the development of a hardware implementation in mind, spending too much time devising a completely new NIDS might obstruct this goal.

Consequently, the main other potential approach is to search for existing suitable NIDS designs and implement those in the context of this project. However, as this approach would no longer require coming up with a completely new design, more time would become available for a hardware implementation.

Defining the suitability of a design would depend on the requirements of the desired NIDS. If for example real-time classification of network traffic is desired, an approach aggegrating traffic data over longer time intervals would not be suitable. Depending on whether the NIDS should be misuse-based or anomaly-based, the range of suitable algorithms changes too. Finally, for this project, the algorithm should be adaptable to a hardware implementation without losing too much functionality.

If this second approach is chosen, a method for comparing various algorithms in various situations should be used. Figure 2 illustrates the general flow of a NIDS approach: from the raw traffic data, first abstract features are derived. A machine learning classifier is used in a second stage to decide if a network intrusion is detected. Several state-of-the-art approaches use a CNN as classifier, and one or another way to convert the input flow to a feature image that is fed in this CNN

[37]–[39]. For comparison on a given dataset, we can change both the preprocessing method and the classification algorithm. Changing the preprocessing method provides different input features for the machine learning algorithm, while changing the algorithm itself introduces other ways to learn from the input features. Mixing and matching preprocessing methods and machine learning algorithms for different datasets might then provide a workflow that is robust in many situations while achieving good intrusion detection.

4) Evaluation: Of course, there needs to be a way to define performance for a machine learning-based NIDS. Firstly, the detection performance itself should be evaluated. This is most often done using metrics such as accucary, false positive rate, precision, recall or F-measure. Sometimes more specific measures such as Matthews Correlation Coefficient can also be used [40], if the dataset requires it.

Other evaluation criteria are necessary for the evaluation of the hardware implementation. More concretely, these citeria include bandwidth (in Gb/s), energy consumption (in W), latency (in s), used resources (number of flip-flops,...) and packet loss rate (due to buffer overflows). The latency in this case is the time it takes for the system to decide, upon receiving a network packet, if the packet is malicious. The packet loss rate then is the fraction of a flow of data that is lost, for example by dropping packets when a buffer is full.

C. Driver assistance

AI is vastly used in different automotive applications and is a key part in many Advanced Driver-Assistance Systems (ADAS) used in cars. The key domain of automotive industry that utilizes AI is object detection and image segmentation.

1) Motivation: AI can be used in different domains in automotive. From safety applications to adding luxury for the passengers, it has a lot of applications. Some of the key motivation points for using AI in automotive use case are given below:

- **Driving Assistance**: Intelligent devices in automotive not only help in driver assistance but are also used in many safety operations such as emergency braking, blind spot monitoring and car distance detection. By monitoring different sensors, AI in automotive can identify dangerous situations which can alert the driver or take control of the vehicle to avoid accidents. A very detailed survey is presented in [41] about driving assistance systems and their future trends.
- **Cloud Services**: Conventional cars show drivers check-engine lights, low battery and other alerts for maintenance. Intelligent devices in automotive can detect issues in cars before they start to effect vehicle performance by monitoring different sensor data. In [42], fault detection and classification technique is presented for automotive internal combustion engines. The vibration data from crank angle is used for fault detection. Using this method commonly known defects of engines were identified with a success rate of 97%.
- **Risk Assessment**: AI can access driver's recent history data and can do a risk assessment on driver's ability. There are many factors which can effect driver's ability such as health issues or less sleep.
- **Driver Monitoring**: In another technique, AI detects human eye dilation and predicts if the driver is under stress. This can be used as a safety measure. For instance, the authors in [43] propose a stress detection approach by monitoring driver's electrocardiogram (ECG) while driving. This monitoring is used to alert driver, its family or the road users to avoid accidents in case of high stress level. Also driver gestures can be used for infotainment control.

2) Challenges: The main challenges in this use case are accuracy and performance. Since most of the ADAS are real time and an error can be very catastrophic. So the algorithms used, needs to be in safety critical accuracy. Also the algorithm execution latency should be low enough so it does not miss hard deadlines in safety critical situations. The algorithms used (especially deep learning algorithms) have high memory and computational requirements. If convolutional neural networks are considered as application test case, then 90% of the operations are convolution operations [44] which are computationally intensive. Also the parameters of these algorithms have huge memory requirements. To fit these algorithms on embedded systems is a challenge in itself. Another concern is, the algorithm developers do not consider hardware requirements when developing and focus is purely on the achieving higher accuracy. This creates a gap between software and hardware designs, creating algorithms which have high accuracy but are impossible to fit on embedded platforms. This is because algorithms are very big with huge parameters to be processed.

3) Potential approaches: In order to use an embedded platform for this use case, light weight deep learning algorithms should be used. There is also some work already available to reduce parameter size which is helpful for embedded platforms. Ristretto [45] is a framework which allows parameter approximation of convolutional neural networks depending on the hardware. This tool reduces the bit-width of a given network by simulating hardware arithmetic of a hardware accelerator. This reduces the hardware size by giving low bit-width parameters for the network. For embedded platforms, the most common approach for deploying AI algorithms is by building hardware accelerators [46]. Although it is performance efficient and a reliable solution but it lacks flexibility and robustness. A combination of a processor and accelerator will add more programmability and flexibility to the architecture. With the introduction of RISC-V [47] as an open source instruction set architecture, a lot of open source system-on-chip and MPSoCs are available for research [48]. RISC-V also allows to add custom instructions to the architecture thus it can be extended to an Application Specific Instruction Set Processor (ASIP). A combination of multiple ASIPs, network-on-chip and different hardware accelerator topologies, can be a feasible solution to overcome AI constraints.

Fig. 2. Different steps in the approach of comparing and implementing existing systems.

4) Evaluation: The evaluation criteria for this use case should be in three domains: area, power and performance. Since these are three main constraints for embedded platforms. Also keeping in check that AI algorithms used do not lose accuracy significantly when different optimization techniques are used such as parameter optimization and pruning, etc.

D. Industry 4.0

1) Motivation: The vision is to use Machine Learning as a driver to extend the functionality of next generation sensors and actuators for the Industry 4.0. This new class of intelligent devices should support (1) self-calibration, (2) predictive maintenance, (3) self-organization and (4) autonomous control.

Below we list the potential applications to the Industry 4.0 domain.

- **Self-calibration**: intelligent devices can use multi-dimensional data from neighboring sensors to perform self-calibration on the sensors and actuators.
- **Predictive maintenance**: intelligent devices can automatically detect anomalies in the industrial processes and predict malfunctions in sensors and actuators. This information is processed collaboratively from multi-dimensional data gathered by the industrial sensor network and the malfunctioning equipment can be identified;
- **Self-organization**: intelligent devices are connected in a mesh network and can organize themselves in a meaningful way to process information on the task at hand and are able to react in case of failure of individual nodes (i.e., even when predictive maintenance fails). As an example, a network of sensors is used to measure temperature in different stages of an industrial process; in case one of the nodes fails, the network reorganizes itself and uses learned data to predict the temperature for the failed node.
- **Autonomous control**: the intelligent devices can automatically control the industrial processes to optimize the throughput while maintaining the required quality standards and requiring minimum amount of human intervention.

2) Challenges: The two main challenges of this task are achieving **high performance**, **security** (in close connection to use-case 3.3 above) and **high dependability**. The devised solutions should have low latency in processing sensor data and evaluating commands for the actuators to have minimal impact on the production throughput. For that reason, processing should be distributed (lack of a centralized controller) and carried at the edge in order to avoid the high-latency data communication costs. An additional justification for the edge processing requirement is the protection of the data and continuity of operation: edge processing enables devices to potentially operate disconnected from the network, keeping the data local and ensuring a higher uptime.

3) Evaluation: For some of the proposed functional requirements, the evaluation criteria are the ability to replace a human operator and perform the task "just as good" (e.g.: when it comes to self-calibration and self-organization). A quantitative evaluation criterion, in this case, could be the required time for calibration or organization, or the impact of this time on the production throughput. The ability for successful predictive maintenance should be evaluated by an increase in the mean work/time to failure (MWTF or MTTTF) driven by decreased idle time from failed equipment. Autonomous control can be evaluated by the ability of proposed ML-based solutions to achieve the same production throughput while maintaining the same quality standards, or by an improvement in the quality (indicated, for instance, by a reduced average deviation from the norm). Finally, processing should be carried locally (in the industrial plant) and data communication should not be centralized in order to sustain the non-functional requirement of security.

IV. Conclusions

This paper covers the goals of the AITIA project and its use-cases. The project aims to bridge the gap between the academic knowledge on AI/ML and bring it to ready-to-use solutions that SMEs can use in their products.

References

[1] R. DiCecco, G. Lacey, J. Vasiljevic, P. Chow, G. Taylor, and S. Areibi, "Caffeinated FPGAs: FPGA framework For Convolutional Neural Networks," in *2016 International Conference on Field-Programmable Technology (FPT)*, Dec. 2016, pp. 265–268.

[2] Y. Wang, J. Xu, Y. Han, H. Li, and X. Li, "DeepBurning: Automatic Generation of FPGA-based Learning Accelerators for the Neural Network Family," in *DAC'16*, ser. DAC '16. New York, NY, USA: ACM, 2016, pp. 110:1–110:6, event-place: Austin, Texas. [Online]. Available: http://doi.acm.org/10.1145/2897937.2898003

[3] S. I. Venieris, A. Kouris, and C.-S. Bouganis, "Toolflows for Mapping Convolutional Neural Networks on FPGAs: A Survey and Future Directions," *ACM Computing Surveys*, vol. 51, no. 3, pp. 56:1–56:39, Jun. 2018. [Online]. Available: http://doi.acm.org/10.1145/3186332

[4] Y. Sun, A. Ben Ahmed, and H. Amano, "Acceleration of Deep Recurrent Neural Networks with an FPGA Cluster," in *Proceedings of the 10th International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies*, ser. HEART 2019. New York, NY, USA: ACM, 2019, pp. 18:1–18:4, event-place: Nagasaki, Japan. [Online]. Available: http://doi.acm.org/10.1145/3337801.3337804

[5] E. Nurvitadhi, D. Kwon, A. Jafari, A. Boutros, J. Sim, P. Tomson, H. Sumbul, G. Chen, P. Knag, R. Kumar, R. Krishnamurthy, S. Gribok, B. Pasca, M. Langhammer, D. Marr, and A. Dasu, "Why Compete When You Can Work Together: FPGA-ASIC Integration for Persistent RNNs," in *FCCM'19*, Apr. 2019, pp. 199–207.

[6] Z. Li, S. Wang, C. Ding, Q. Qiu, Y. Wang, and Y. Liang, "Efficient Recurrent Neural Networks using Structured Matrices in FPGAs," *arXiv:1803.07661 [cs, stat]*, Mar. 2018, arXiv: 1803.07661. [Online]. Available: http://arxiv.org/abs/1803.07661

[7] M. Rizakis, S. I. Venieris, A. Kouris, and C.-S. Bouganis, "Approximate FPGA-Based LSTMs Under Computation Time Constraints," in *ARC'18*, ser. Lecture Notes in Computer Science, N. Voros, M. Huebner, G. Keramidas, D. Goehringer, C. Antonopoulos, and P. C. Diniz, Eds. Springer International Publishing, 2018, pp. 3–15.

[8] C. Zhang, D. Wu, J. Sun, G. Sun, G. Luo, and J. Cong, "Energy-Efficient CNN Implementation on a Deeply Pipelined FPGA Cluster," in *ISLPED'16*, ser. ISLPED '16. New York, NY, USA: ACM, 2016, pp. 326–331, event-place: San Francisco Airport, CA, USA. [Online]. Available: http://doi.acm.org/10.1145/2934583.2934644

[9] J. H. Kim, B. Grady, R. Lian, J. Brothers, and J. H. Anderson, "FPGA-based CNN inference accelerator synthesized from multi-threaded C software," in *SOCC'17*, Sep. 2017, pp. 268–273.

[10] F. U. D. Farrukh, T. Xie, C. Zhang, and Z. Wang, "Optimization for Efficient Hardware Implementation of CNN on FPGA," in *2018 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA)*, Nov. 2018, pp. 88–89.

[11] K. Guo, L. Sui, J. Qiu, J. Yu, J. Wang, S. Yao, S. Han, Y. Wang, and H. Yang, "Angel-Eye: A Complete Design Flow for Mapping CNN Onto Embedded FPGA," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 35–47, Jan. 2018.

[12] L. Xie, X. Fan, W. Cao, and L. Wang, "High Throughput CNN Accelerator Design Based on FPGA," in *2018 International Conference on Field-Programmable Technology (FPT)*, Dec. 2018, pp. 274–277.

[13] J. Zhang and J. Li, "PQ-CNN: Accelerating Product Quantized Convolutional Neural Network on FPGA," in *FCCM'18*, Apr. 2018, pp. 207–207.

[14] G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, D. Giardino, M. Matta, M. Re, F. Silvestri, and S. Spanò, "Efficient Ensemble Machine Learning Implementation on FPGA Using Partial Reconfiguration," in *Applications in Electronics Pervading Industry, Environment and Society*, ser. Lecture Notes in Electrical Engineering, S. Saponara and A. De Gloria, Eds. Springer International Publishing, 2019, pp. 253–259.

[15] G. Plastiras, M. Terzi, C. Kyrkou, and T. Theocharidcs, "Edge intelligence: Challenges and opportunities of near-sensor machine learning applications," in *2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*. IEEE, 2018, pp. 1–7.

[16] A. Hackner, H. Oberpriller, A. Ohnesorge, V. Hechtenberg, and G. Müller, "Heterogeneous sensor arrays: Merging cameras and gas sensors into innovative fire detection systems," *Sensors and Actuators B: Chemical*, vol. 231, pp. 497–505, 2016.

[17] S. Luna, T. F. Stahovich, H. C. Su, and N. V. Myung, "A method for optimizing the design of heterogeneous nano gas chemiresistor arrays," *Electroanalysis*, vol. 31, no. 6, pp. 1009–1018, 2019.

[18] B. da Silva, L. Segers, Y. Rasschaert, Q. Quevy, A. Braeken, and A. Touhafi, "A multimode soc fpga-based acoustic camera for wireless sensor networks," in *2018 13th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*. IEEE, 2018, pp. 1–8.

[19] L. Del Val, A. Izquierdo, J. J. Villacorta, and L. Suárez, "A failure detection methodology using new features of acoustic images of a fan matrix," in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, vol. 259, no. 3. Institute of Noise Control Engineering, 2019, pp. 6563–6571.

[20] F. Almasri and O. Debeir, "Rgb guided thermal super-resolution enhancement," in *2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech)*. IEEE, 2018, pp. 1–5.

[21] K. Noto, C. Brodley, S. Majidi, D. W. Bianchi, and D. K. Slonim, "Csax: Characterizing systematic anomalies in expression data," in *International Conference on Research in Computational Molecular Biology*. Springer, 2014, pp. 222–236.

[22] N. Madhu and R. Martin, "Low-complexity, robust algorithm for sensor anomaly detection and self-calibration of microphone arrays," *IET signal processing*, vol. 5, no. 1, pp. 97–103, 2011.

[23] B. da Silva, A. W Happi, A. Braeken, and A. Touhafi, "Evaluation of classical machine learning techniques towards urban sound recognition on embedded systems," *Applied Sciences*, vol. 9, no. 18, p. 3885, 2019.

[24] A. Amanatiadis and I. Andreadis, "A survey on evaluation methods for image interpolation," *Measurement Science and Technology*, vol. 20, no. 10, p. 104015, 2009.

[25] H. Debar, "An introduction to intrusion-detection systems," in *Proceedings of Connect'2000*, 2002.

[26] R. Sommer and V. Paxson, "The Zeek Network Security Monitor." [Online]. Available: https://www.zeek.org/

[27] V. Paxson, "Bro: a system for detecting network intruders in real-time," *Computer Networks*, vol. 31, no. 23, pp. 2435 – 2463, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128699001127

[28] M. Roesch, "Snort - Network Intrusion Detection & Prevention System." [Online]. Available: https://www.snort.org/

[29] "Suricata — Open Source IDS / IPS / NSM engine." [Online]. Available: https://suricata-ids.org/

[30] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning For Network Intrusion Detection," in *2010 IEEE SYMPOSIUM ON SECURITY AND PRIVACY*, ser. IEEE Symposium on Security and Privacy, IEEE Comp Soc. 10662 LOS VAQUEROS CIRCLE, PO BOX 3014, LOS ALAMITOS, CA 90720-1264 USA: IEEE COMPUTER SOC, 2010, Proceedings Paper, pp. 305–316, Symposium on Security and Privacy, Oakland, CA, MAY 16-19, 2010.

[31] "KDD Cup 1999 Data." [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[32] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, July 2009, pp. 1–6.

[33] "1998 DARPA Intrusion Detection Evaluation Dataset." [Online]. Available: https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset

[34] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Nov 2015, pp. 1–6.

[35] I. Sharafaldin, A. Habibi Lashkari, and A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *4th International Conference on Information Systems Security and Privacy (ICISSP)*, Portugal, 01 2018, pp. 108–116.

[36] M. Ceska, V. Havlena, L. Holík, J. Korenek, O. Lengál, D. Matousek, J. Matousek, J. Semric, and T. Vojnar, "Deep packet inspection in fpgas via approximate nondeterministic automata," *CoRR*, vol. abs/1904.10786, 2019. [Online]. Available: http://arxiv.org/abs/1904.10786

[37] Y. Zhang, X. Chen, D. Guo, M. Song, Y. Teng, and X. Wang, "Pccn: Parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows," *IEEE Access*, vol. 7, pp. 119 904–119 916, 2019.

[38] Z. Li, Z. Qin, K. Huang, X. Yang, and S. Ye, "Intrusion detection using convolutional neural networks for representation learning," in *Neural Information Processing*, D. Liu, S. Xie, Y. Li, D. Zhao, and E.-S. M. El-Alfy, Eds. Cham: Springer International Publishing, 2017, pp. 858–866.

[39] T. Kim, S. C. Suh, H. Kim, J. Kim, and J. Kim, "An encoding technique for cnn-based network anomaly detection," in *2018 IEEE International Conference on Big Data (Big Data)*, Dec 2018, pp. 2960–2965.

[40] R. K. Malaiya, D. Kwon, J. Kim, S. C. Suh, H. Kim, and I. Kim, "An empirical evaluation of deep learning for network anomaly detection," in *2018 International Conference on Computing, Networking and Communications (ICNC)*, March 2018, pp. 893–898.

[41] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 6–22, winter 2014.

[42] R. Ahmed, M. El Sayed, S. A. Gadsden, J. Tjong, and S. Habibi, "Automotive internal-combustion-engine fault detection and classification using artificial neural network techniques," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 21–33, Jan 2015.

[43] S. NITA, S. BITAM, and A. MELLOUK, "A body area network for ubiquitous driver stress monitoring based on ecg signal," in *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2019, pp. 1–6.

[44] Y. Ma, Y. Cao, S. Vrudhula, and J. Seo, "Optimizing the convolution operation to accelerate deep neural networks on fpga," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 7, pp. 1354–1367, July 2018.

[45] P. Gysel, "Ristretto: Hardware-oriented approximation of convolutional neural networks," *CoRR*, vol. abs/1605.06402, 2016. [Online]. Available: http://arxiv.org/abs/1605.06402

[46] K. Abdelouahab, M. Pelcat, J. Sérot, and F. Berry, "Accelerating CNN inference on fpgas: A survey," *CoRR*, vol. abs/1806.01683, 2018. [Online]. Available: http://arxiv.org/abs/1806.01683
[47] "Risc-v." [Online]. Available: https://riscv.org/
[48] "Risc-v cores and soc overview." [Online]. Available: https://riscv.org/risc-v-cores/