PROGRAM

Euromicro: Machine Learning Driven Technologies and Architectures for Intelligent Internet of Things (ML-IoT)

August 29 –31, 2018

Prague | Czech Republic

Message from the Chairs

It is with great pleasure to welcome you to the Euromicro ML-IoT 2018, in Prague, Czech Republic. In this increasingly compute centric world, where the Internet of Things (IoT) and Artificial Intelligence (AI) tsunami are affecting every aspect of our daily lives, there is increased demand for researchers from different areas such as Machine Learning (ML), distributed computing, embedded systems, and big data to synergize their efforts to better understand untapped opportunities and to produce highly efficient, deployable, intelligent ML-driven IoT systems. In this context, in close collaboration with Digital System Design (DSD) and Software Engineering and Advanced Applications (SEAA), Euromicro organizes the first International event on Machine Learning (ML) Driven Technologies and Architectures for Intelligent Internet of Things (ML-IoT) to promote research and technology transfer in this important cutting-edge field. ML-IoT 2019 consists of one session, with three papers. This session intends to address all aspects of intelligent IoT from Device, to Edge/Fog and Cloud, covering the design of circuits, architecture, network, cloud, cross-layer intelligence, big data, applications, as well as human-machine interaction. ML-IoT also discusses the associated challenges that need to be overcome for achieving the goal of accuracy, privacy, reliability and security.

Once again thanks to Euromicro, their chairpersons and officers, in particular Prof. Erwin Schoitsch and Prof. Hana Kubatova, who continue to manage and keep running the Euromicro conference series, for their support. We want to thank also the organizing organizations including Czech Technical University in Prague hosting our event. We do hope you will be able to join us next year for ML-IoT 2019. Welcome to ML-IoT 2018 in Prague and ENJOY!

Dr.-Ing. Farshad Firouzi, mVISE, Germany

Dr. Bahar Farahani, Shahid Beheshti University, Iran

Dr. Kunal Mankodiya, University of Rhode Island, USA

ML-IoT 2018 Committees

General Chair

- Dr.-Ing. Farshad Firouzi, mVISE, Germany

Program Chair

- Dr. Bahar Farahani, Shahid Beheshti University, Iran
- Dr. Kunal Mankodiya, University of Rhode Island, USA

Technical Program Committee

- Prof. Lech Jozwiak, Eindhoven University of Technology, Netherlands
- Prof. Walter Stechele, TU München, Germany
- Prof. Nicolas Sklavos, University of Patras, Greece
- Prof. Arda Yurdakul, Bogazici University, Turkey
- Prof. Henk Corporaal, Eindhoven University of Technology, Netherlands
- Prof. Fereidoon Shams Aliee, Shahid Beheshti University, Iran
- Prof. Rolf Drechsler, University of Bremen, Germany
- Prof. Radu Grosu, Vienna University of Technology, Austria
- Prof. Emad Samuel Malki Ebeid, University of Southern Denmark, Denmark
- Prof. Puneet Goyal, IIT Ropar, India
- Prof. Assad Abbas, COMSATS Institute of Information Technology, Pakistan
- Dr. Prakash Kumar Ray, Nanyang Technological University, Singapore
- Dr. Ankesh Jain, University of Ulm, Germany
- Dr. Arpan Pal, TCS Research and Innovation
- Dr. Ilias Gerostathopoulos, Technische Universität München, Germany
- Dr. Ritwik Giri, Starkey Hearing Technologies
- Dr. C. P. Ravikumar, Texas Instrument, India
- Maurice Peemen, Eindhoven University of Technology, Netherlands

A Machine Learning Driven IoT Solution for Noise Classification in Smart Cities

Yasser Alsouda
Dep. of Physics and Electrical Eng.
Linnaeus University
351 95 Växjö, Sweden
Email: ya222ci@student.lnu.se

Sabri Pllana
Dep. of Computer Science and Media Tech.
Linnaeus University
351 95 Växjö, Sweden
Email: sabri.pllana@lnu.se

Arianit Kurti
RISE Interactive
Research Institutes of Sweden
602 33 Norrköping, Sweden
Email: arianit.kurti@ri.se

***Abstract*—We present a machine learning based method for noise classification using a low-power and inexpensive IoT unit. We use Mel-frequency cepstral coefficients for audio feature extraction and supervised classification algorithms (that is, support vector machine and k-nearest neighbors) for noise classification. We evaluate our approach experimentally with a dataset of about 3000 sound samples grouped in eight sound classes (such as, car horn, jackhammer, or street music). We explore the parameter space of support vector machine and k-nearest neighbors algorithms to estimate the optimal parameter values for classification of sound samples in the dataset under study. We achieve a noise classification accuracy in the range 85% – 100%. Training and testing of our k-nearest neighbors ($k = 1$) implementation on Raspberry Pi Zero W is less than a second for a dataset with features of more than 3000 sound samples.**

***Index Terms*—urban noise, smart cities, support vector machine (SVM), k-nearest neighbors (KNN), mel-frequency cepstral coefficients (MFCC), internet of things (IoT).**

I. Introduction

About 85% of Swedes live in urban areas and the quality of life and the health of citizens is affected by noise. Noise is any undesired environmental sound. The world health organization (WHO) recommends [1] for good sleeping less than 30dB noise level in the bedroom and for teaching less than 35dB noise level in classroom. Recent studies [2] have found that exposure to noise pollution may increase the risk for health issues, such as, heart attack, obesity, impaired sleep, or depression.

Following the Environmental Noise Directive (END) 2002/49/EC, each EU member state has to assess environmental noise and develop noise maps every five years. As sources of noise (such as, volume of traffic, construction sites, music and sport events) may change over time, there is a need for continuous monitoring of noise. Health damaging noise often occurs for only few minutes or hours, and it is not enough to measure the noise level every five years. Furthermore, *the sound at the same dB level may be percepted as annoying noise or as a pleasant music*. Therefore, it is necessary to go beyond the state-of-the-art approaches that measure only the dB level [3]–[5] and in future we also *identify the type of the noise*. For instance, it is important that the environment protection unit and law enforcement unit of a city know whether the noise is generated by a jackhammer at construction site or by a gun shot. The Internet of Things (IoT) is a promising technology for improving many domains, such as eHealth [6], [7], and it may be also used to address the issue of noise pollution in smart cities [8].

In this paper, we present an approach for noise classification in smart cities using machine learning on a low-power and inexpensive IoT unit. Mel-frequency cepstral coefficients (MFCC) are extracted as audio features and applied to two classifiers: support vector machine (SVM) and k-nearest neighbors (KNN). The evaluation of SVM and KNN with respect to accuracy and time is carried out on a Raspberry Pi Zero W. For evaluation we prepared a dataset of 3042 samples of environmental sounds from UrbanSound8K [9] and Sound Events [10] in eight different classes (including gun shot, jackhammer, or street music). SVM classification performance is affected by parameters γ and C, whereas parameter k and minimum distance type (that is, Euclidean, Manhattan, or Chebyshev distance) influence the KNN performance. We explore the parameter space of SVM and KNN algorithms to estimate the optimal parameter values for classification of sound samples. The achieved noise classification accuracy is in the range 85% – 100% and the time needed for training and testing of KNN model for $k = 1$ on Raspberry Pi Zero W is below one second.

Major contributions of this paper include,

- a machine learning approach for noise classification;
- implementation of our approach for noise classification on Raspberry Pi Zero W;
- experimental evaluation of our approach using a dataset of 3042 samples of environmental sounds;
- exploration of parameter space of KNN and SVM to estimate the best parameter values with respect to our sound samples dataset.

The rest of this paper is organized as follows. Section II gives an overview of machine learning and the Raspberry Pi platform. The proposed method for noise classification is described in Section III. Section IV presents experimental evaluation of our approach, and Section V discusses the related work. The paper is concluded in Section VI.

II. Background

A. Machine Learning

Machine Learning is described by Mitchell [11] as follows, a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Commonly the supervised machine learning techniques are used for classification of data into different categories. Supervised learning means building a model based on known set of data (input and output) to predict the outputs of new data in the future. In the midst of the diversity of classification algorithms, selecting the proper algorithm is not straightforward, since there is no perfect one that fits with all applications and there is always a trade-off between different model characteristics, such as: complexity, accuracy, memory usage, and speed of training.

B. Raspberry Pi and Mic-Hat

Figure 1 depicts our hardware experimentation platform that comprises a Raspberry Pi Zero W and a ReSpeaker 2-Mic Pi HAT.

The Raspberry Pi [12] is a low-power and low-cost single-board computer with a credit card size. It may be used as an affordable computer to learn programming or to build smart devices. A *Raspberry Pi Zero W* with a Wi-Fi capability is used for our experiments. The Raspberry Pi Zero W (see Table I) comes with a single-core CPU running at 1GHz, 512MB of RAM, and costs only about $10.

We use for sound sensing a dual-mic array expansion board for Raspberry Pi called *ReSpeaker 2-Mic Pi HAT* [13]. This board is developed based on WM8960 and has two microphones for collecting data and is designed to build flexible and powerful sound applications.

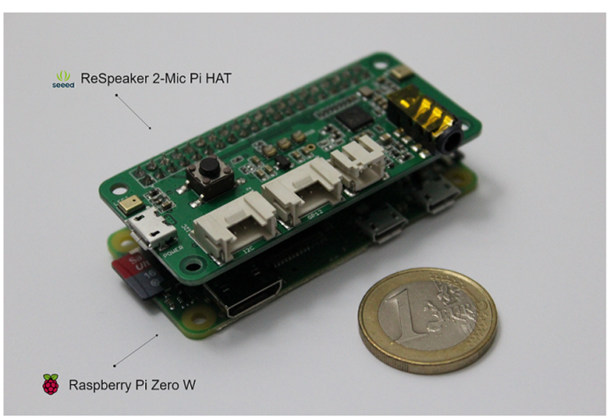


Fig. 1. Noise classification hardware platform consists of a Raspberry Pi Zero W and a ReSpeaker 2-Mic Pi Hat.

III. A Machine Learning Based Method for Noise Classification

In this section we describe our method for classification of noise using machine learning on Raspberry Pi. The proposed noise classification system is illustrated in Figure 2. MFCCs are extracted from a training dataset of sound samples to train SVM and KNN models that are used to predict the type of sensed environmental sounds.

TABLE I
MAJOR PROPERTIES OF THE RASPBERRY PI ZERO W

Property	Raspberry Pi Zero W
SOC	Broadcom BCM2835
core	1 x ARM1176JZF-S, 1GHz
RAM	512MB
storage	micro SD
USB	1 x micro USB port
wireless LAN	802.11 b/g/n
bluetooth	4.1
HDMI	mini
GPIO	40 pins
power (idle)	80mA (0.4W)

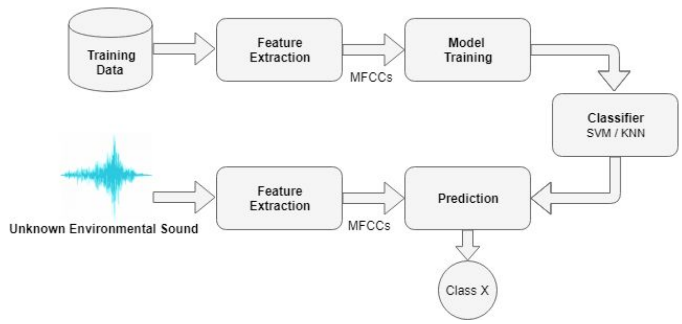


Fig. 2. Our machine learning based approach for noise classification.

A. Dataset

To investigate the performance of the system, we conduct experiments with eight different classes of environmental sounds: quietness, silence, car horn, children playing, gun shot, jackhammer, siren, and street music. For the purpose of this study we chose noise-relevant environmental sound clips from popular sound datasets, such as UrbanSound8K [9] and Sound Events [10]. The total dataset contains 3042 sound excerpts with length up to four seconds. Table II provides the information about environmental sound samples that we use for experimentation.

TABLE II
CLASSES OF SOUND SAMPLES IN THE DATASET

Class	Samples	Duration
Quietness	40	02 min 00 sec
Silence	40	02 min 00 sec
Car horn	312	14 min 38 sec
Children playing	560	36 min 47 sec
Gun shot	235	06 min 39 sec
Jackhammer	557	32 min 34 sec
Siren	662	43 min 17 sec
Street music	636	42 min 24 sec
Total	3042	2 hrs 0 min 19 sec

B. Feature Extraction

Features extraction is the first step in an automatic sound classification system. MFCCs [14] are a well-known feature set and are widely used in the area of sound classification because they are well-correlated to what the human can hear. MFCCs are obtained using the procedure depicted in Figure 3.

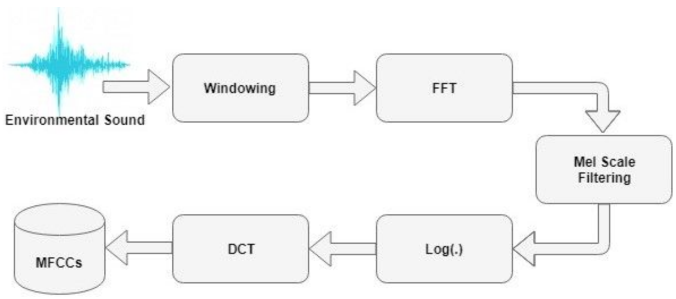


Fig. 3. The procedure for generating MFCCs of environmental sounds.

Foote [15] proposes the use of the first 12 MFCCs plus an energy term as sound features. In this paper, we computed the first 12 MFCCs of all frames of the entire signal and appended the frame energy to each feature vector, thus each audio signal is transformed into a sequence of 13-dimensional feature vector.

C. Classification

In this section we examine two supervised classification methods: support vector machine and k-nearest neighbors.

1) Support Vector Machines (SVM): SVM [16] is a popular supervised algorithm mostly used for solving classification problems. The main goal of the SVM algorithm is to design a model that finds the optimal hyperplane that can separate all training data into two classes. There may be many hyperplanes that separate all the training data correctly, but the best choice will be the hyperplane that leaves the maximum margin, which is defined as the distance between the hyperplane and the closest samples. Those closest samples are called the support vectors.

Considering the example of two linearly separable classes (circles and squares) shown in Figure 4, both hyperplanes (one and two) can classify all the training instances correctly, but the best hyperplane is one since it has a greater margin ($m_1 > m_2$).

When the data is nonlinearly separable, the nonlinear classifier can by created by applying the kernel trick [17]. Using the kernel trick, the non-separable problem can be converted to a separable problem using kernel functions that transform low dimensional input space to high dimensional space. Selecting the appropriate kernel and its parameters has a significant impact on the SVM classifier. Another important parameter for the SVM classifier is the soft margin parameter C, which controls the trade-off between the simplicity of the decision boundary and the misclassification penalty of the training points. A low value of C makes the classifier

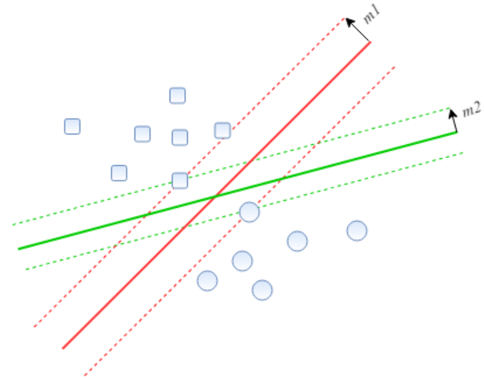


Fig. 4. An illustration of SVM for a 2-class classification problem.

tolerant with misclassified data points (that is, *smooth decision boundary*), while a high value of C makes it aiming to a perfect classification of the training points (that is, *complex boundary decision*).

One of the kernel functions that is commonly used in SVM classification is the radial basis function (RBF). The RBF kernel on two feature vectors (x and x') is expressed by Equation 1.

$$K(x,x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right) = \exp\left(-\gamma\|x-x'\|^2\right) \quad (1)$$

The RBF parameter γ determines the influence of the training data points on determining the exact shape of the decision boundary. With a high value of γ the details of the decision boundary are determined only by the closest points, while for a low value of γ even the faraway points are considered in drawing the decision boundary.

In this paper, we explore the effect of parameters γ and C on SVM model with respect to our dataset of sound samples.

2) K-Nearest Neighbors (KNN): KNN is one of the simplest machine learning algorithms used for classification. The KNN works based on the minimum distance (such as, Euclidean distance) between the test point and all training points. The class of the test point is then determined by the most frequent class of the k nearest neighbors to the test point. Commonly used distances include,

- *Euclidean distance*: d(q, p) = $\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$
- *Manhattan distance*: d(q, p) = $\sum_{i=1}^{n}|q_i - p_i|$
- *Chebyshev distance*: d(q, p) = $\max_i(|q_i - p_i|)$

The KNN classifier is illustrated with an example in Figure 5. Two classes are represented with *squares* and *circles* and the aim of the KNN algorithm is to predict the correct class of the *triangle*. Suppose $k = 3$, then the model will find three nearest neighbors of triangle. To predict the correct class of the triangle, the algorithm can achieve its aim by finding three nearest neighbors of the triangle and the most frequent element determines the class of the *triangle*, which is the class of *squares* in this case.

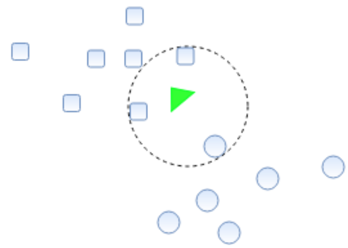


Fig. 5. An illustration of KNN for a 2-class classification problem for $k = 3$.

The KNN algorithm needs a significant amount of memory to run, since it requires all the training data to make a prediction.

IV. Experimental Evaluation

In this section, we investigate the performance of SVM and KNN on eight different classes of environmental sounds: quietness, silence, car horn, children playing, gun shot, jack-hammer, siren, street music. For training the models we use a dataset of 3042 samples of environmental sounds (see Table II). We divide the dataset arbitrary into two sub-sets: 75% are used for training and 25% for testing. All experiments are repeated 20 times with different sub-sets and the obtained results are averaged. We have implemented all algorithms in Python using open source packages for machine learning and audio analysis (that is, scikit-learn [18] and librosa [19]).

A. SVM Parameter Space Exploration

To optimize the performance of SVM, the grid search is used to select the best combination of the parameters γ and C for the RBF kernel. To explore the SVM's cross-validation accuracy, we plot the heat map depicted in Figure 6 as a function of γ and C, where $\gamma \in \{10^{-11} - 10^{1}\}$ and $C \in \{10^{-4} - 10^{8}\}$. Table III shows the SVM model accuracy [%] for various values of γ and C parameters. After evaluating the model, we achieved a 93.87% accuracy for $\gamma = 0.00167$ and $C = 3$, as shown in Figure 7 and Figure 8.

TABLE III
ACCURACY [%] OF SVM

	γ			
C	0.0001	0.00167	0.01	0.1
0.1	64.14	67.07	22.96	21.18
1	79.19	92.31	72.66	29.88
3	82.85	93.87	75.22	31.53
5	84.40	93.86	75.21	31.53
10	85.90	93.83	75.19	31.53
100	89.24	93.70	75.18	31.54

B. KNN Parameter Space Exploration

For KNN classifier we examine the influence of parameter k, the Euclidean distance, Manhattan distance, and the Chebyshev distance (Section III-C2). Figure 9 illustrates the

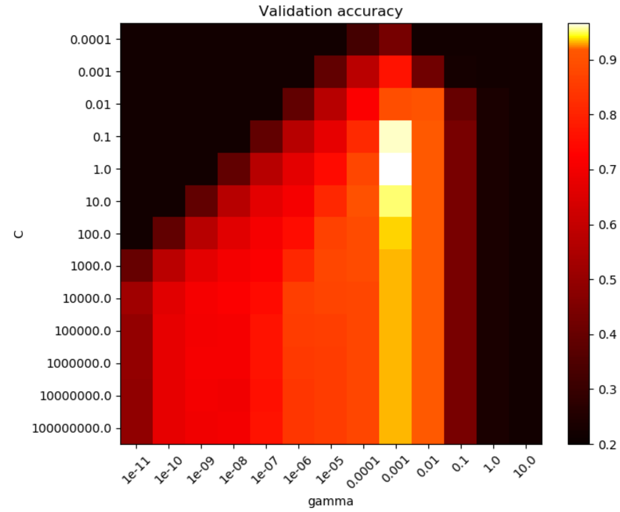


Fig. 6. Heat map of the SVM validation accuracy as a function of γ and C.

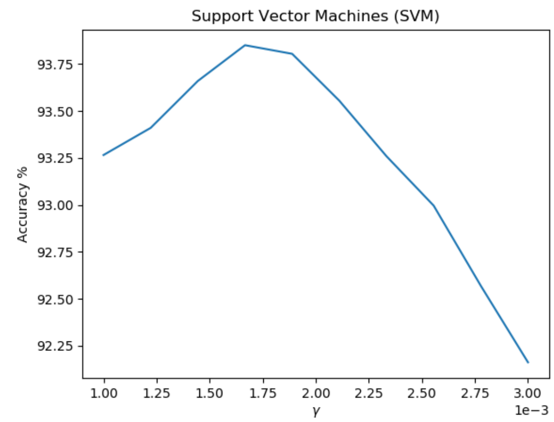


Fig. 7. The effect of the parameter γ on the performance of the SVM classifier.

classification accuracy of KNN for various values of k for each kind of distance. Table IV presents the results for the KNN accuracy, where the $Manhattan$ distance and $k = 1$ proved to be the best parameters with sound type recognition accuracy of 93.88%.

TABLE IV
ACCURACY [%] OF KNN

	Distance		
k	Euclidean	Manhattan	Chebyshev
1	93.46	93.88	90.43
5	88.88	89.42	85.01
10	83.34	84.13	80.58
50	68.20	69.66	67.01

C. Performance of SVM and KNN

In this section we present the performance of SVM and KNN with respect to classification accuracy and time that is needed for training and testing. To examine the accuracy of

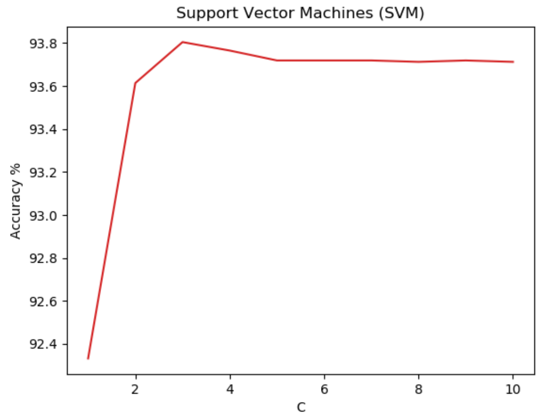


Fig. 8. The effect of the parameter C on the performance of the SVM classifier.

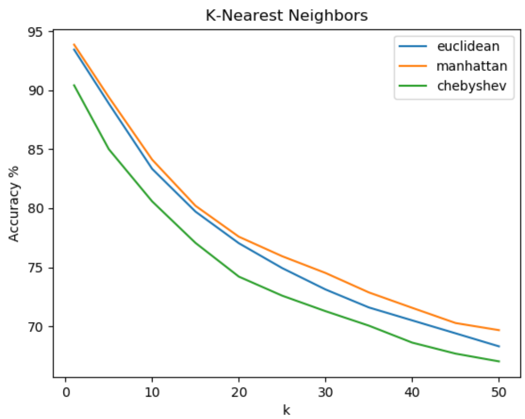


Fig. 9. Performance of the KNN classifier for various values of nearest neighbors k and Euclidean, Manhattan, and Chebyshev distances.

each model we plot the confusion matrix that compares the predicted classes with the true noise classes. Figure 10 and Figure 11 illustrate the confusion matrices of SVM and KNN, respectively, while Table V and Table VI present the time performance of SVM and KNN, respectively, during training and testing on the Raspberry Pi Zero W.

TABLE V
TIME [SECONDS] FOR TRAINING AND TESTING OF SVM MODEL ON PI ZERO W. THE TIME FOR FEATURE EXTRACTION IS NOT INCLUDED.

C	γ							
	0.0001		0.00167		0.01		0.1	
	Train	Test	Train	Test	Train	Test	Train	Test
0.1	8.03	2.37	11.90	2.59	21.98	2.87	31.56	4.64
1	5.00	1.90	11.93	1.98	26.37	2.58	33.00	4.56
3	4.36	1.63	12.29	1.99	26.70	2.65	33.42	4.50
5	4.50	1.62	12.44	1.99	26.76	2.56	33.36	4.51
10	4.29	1.41	12.33	1.98	26.85	2.56	35.32	4.77
100	5.50	1.17	12.29	1.98	26.59	2.58	34.24	4.65

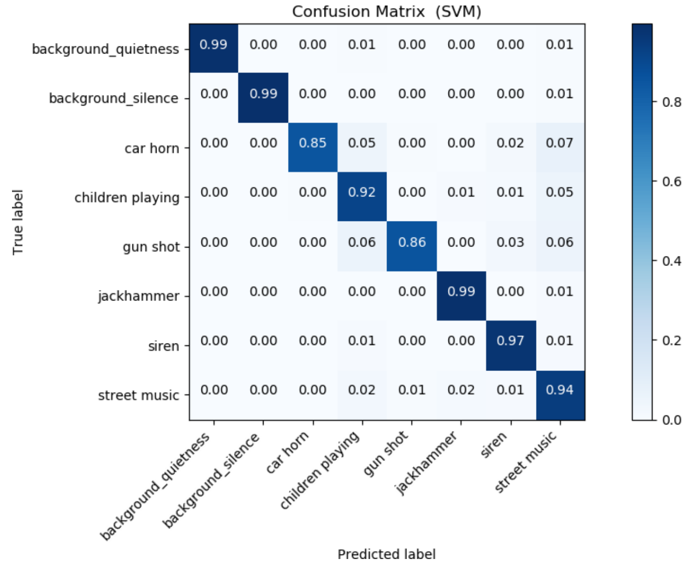


Fig. 10. SVM-based classification of noise.

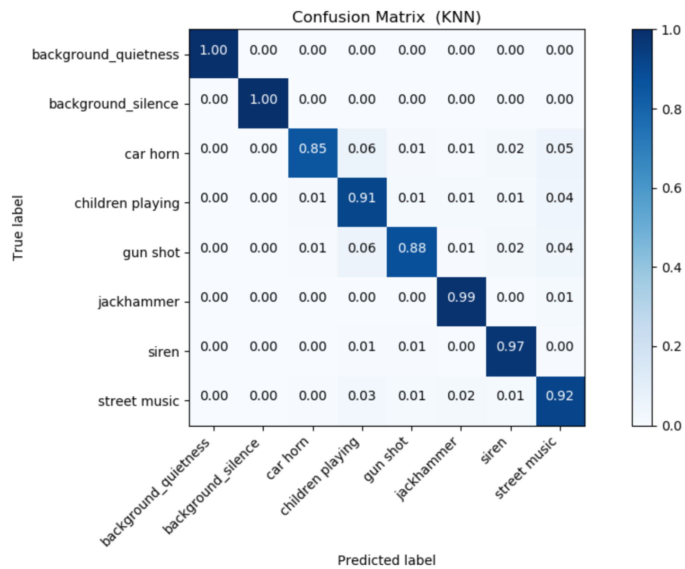


Fig. 11. KNN-based classification of noise.

V. RELATED WORK

In this section we discuss the related work with respect to IoT solutions for noise measurement and machine learning methods for sound classification.

A. IoT Solutions for Noise Measurement

Goetze et al [3] provide an overview of a platform for distributed urban noise measurement, which is part of an ongoing German research project called *StadtLrm*. A wireless distributed network of audio sensors based on quad-core ARM BCM2837 SoC was employed to receive urban noise signals, pre-process the obtained audio data and send it to a central unit for data storage and performing higher-level audio processing. A final stage of web application was used for visualization and administration of both processed and unprocessed audio data.

TABLE VI
TIME [SECONDS] FOR TRAINING AND TESTING OF KNN MODEL ON PI ZERO W. THE TIME FOR FEATURE EXTRACTION IS NOT INCLUDED.

k	*Distance*					
	Euclidean		*Manhattan*		*Chebyshev*	
	Train	Test	Train	Test	Train	Test
1	0.05	0.21	0.05	0.5	0.05	0.14
5	0.05	0.37	0.05	0.92	0.05	0.24
10	0.05	0.47	0.05	1.15	0.05	0.31
100	0.05	0.80	0.05	1.71	0.05	0.57

The authors in [4] used Ameba RTL 8195AM and Ameba 8170AF as IoT platforms to implement a distributed sensing system for visualization of the noise pollution. In [5], two hardware alternatives, Raspberry Pi platform and Tmote-Invent nodes, were evaluated in terms of their cost and feasibility for analyzing urban noise and measuring the psycho-acoustic metrics according to the Zwicker's annoyance model.

In contrast to related work, our approach is not concerned with measuring the noise level in dB using IoT, but with determining the type of noise (for instance, a jackhammer or gun shot).

B. Machine Learning Methods for Sound Classification

In [20], a combination of two supervised classification methods, SVM and KNN, were used as a hybrid classifier with MPEG-7 audio low-level descriptor as the sound feature. The experiments were conducted on 12 classes of sounds. Khunarasal et al [21] proposed an approach to classify 20 different classes of very short time sounds. The study investigated various audio features (e.g., MFCC, MP, LPC and Spectrogram) along with KNN and neural network.

We complement the related work, with a study of noise classification on a low-power and inexpensive device, that is the Raspberry Pi Zero W.

VI. SUMMARY

We have presented a machine learning approach for noise classification. Our method uses MFCC for audio feature extraction and supervised classification algorithms (that is, SVM or KNN) for noise classification. We implemented our approach using Raspberry Pi Zero W that is a low-power and inexpensive hardware unit. We observed in our experiments with various environment sounds (such as, car horn, jackhammer, or street music) that KNN and SVM provide high noise classification accuracy that is in the range 85% – 100%. Experiments with various values of parameter k, which determines the number of nearest data neighbors, indicate that the accuracy of KNN decreases with the increase of k. Experiments with various values of parameter C, which determines misclassification penalty, indicate that SVM had the highest accuracy for $C = 3$ for our dataset. The dataset used in our experiments contains features of about 3000 sound samples and training and testing of KNN ($k = 1$) on Pi Zero W took a fraction of second.

Future work will investigate usefulness of our solution for a large number of Raspberry Pi devices in an environment that combines features of the Edge and Cloud computing systems.

REFERENCES

[1] WHO, *WHO Europe: Data and Statistics*, (accessed Mar. 3, 2018). [Online]. Available: http://www.euro.who.int/en/health-topics/environment-and-health/noise/data-and-statistics

[2] L. Poon, *The Sound of Heavy Traffic Might Take a Toll on Mental Health, CityLab*, (accessed Mar. 9, 2018). [Online]. Available: https://www.citylab.com/equity/2015/11/city-noise-mental-health-traffic-study/417276/

[3] M. Goetze, R. Peukert, T. Hutschenreuther, and H. Toepfer, "An open platform for distributed urban noise monitoring," in *2017 25th Telecommunication Forum (TELFOR)*, Nov 2017, pp. 1–4.

[4] Y. C. Tsao, B. R. Su, C. T. Lee, and C. C. Wu, "An implementation of a distributed sound sensing system to visualize the noise pollution," in *2017 International Conference on Applied System Innovation (ICASI)*, May 2017, pp. 625–628.

[5] J. Segura-Garcia, S. Felici-Castell, J. J. Perez-Solano, M. Cobos, and J. M. Navarro, "Low-cost alternatives for urban noise nuisance monitoring using wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 836–844, Feb 2015.

[6] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven iot ehealth: Promises and challenges of iot in medicine and healthcare," *Future Generation Computer Systems*, vol. 78, pp. 659 – 676, 2018.

[7] D. Perez, S. Memeti, and S. Pllana, "A simulation study of a smart living IoT solution for remote elderly care," in *2018 International Conference on Fog and Mobile Edge Computing (FMEC)*, April 2018, pp. 227–232.

[8] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb 2014.

[9] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22nd ACM International Conference on Multimedia (ACM-MM'14)*, 2014, pp. 1041–1044.

[10] J. Beltran, E. Chavez, and J. Favela, "Scalable identification of mixed environmental sounds, recorded from heterogeneous sources," *Pattern Recognition Letters*, vol. 68, pp. 153 – 160, 2015.

[11] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[12] Raspberry, *Raspberry Pi Foundation*, (accessed May 2, 2018). [Online]. Available: https://www.raspberrypi.org/

[13] Seeed, *ReSpeaker 2-Mics Pi HAT*, (accessed May 2, 2018). [Online]. Available: https://www.seeedstudio.com/ReSpeaker-2-Mics-Pi-HAT-p-2874.html

[14] M. Sahidullah and G. Saha, "Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition," *Speech Communication*, vol. 54, no. 4, pp. 543 – 565, 2012.

[15] J. T. Foote, "Content-based retrieval of music and audio," *Multimedia Storage and Archiving Systems II, Proc. SPIE*, vol. 3229, p. 138147, 1997.

[16] C. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2006.

[17] S. Theodoridis, *Pattern Recognition*, 4th ed. Burlington, MA: Academic Press, 2009.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[19] McFee, Brian, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," *In Proceedings of the 14th python in science conference*, pp. 18–25, 2015.

[20] J.-C. Wang, J.-F. Wang, K. W. He, and C.-S. Hsu, "Environmental sound classification using hybrid svm/knn classifier and mpeg-7 audio low-level descriptor," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 2006, pp. 1731–1735.

[21] P. Khunarsal, C. Lursinsap, and T. Raicharoen, "Very short time environmental sound classification based on spectrogram pattern matching," *Information Sciences*, vol. 243, pp. 57 – 74, 2013.

A Safe Traffic Network Design and Architecture, in the Context of IoT

Angeliki Kalapodi [I], Nicolas Sklavos [I, II], Ioannis D. Zaharakis [II, III], Achilles Kameas [II, IV]

[I] SCYTALE Research Group, Computer Engineering & Informatics Dept. (CEID), University of Patras, Hellas

[II] Computer Technology Institute & Press – "Diophantus" (CTI), Patra, Hellas

[III] Computer & Informatics Engineering Dept., Technological Educational Institute of Western Greece, Hellas

[IV] School of Science and Technology, Hellenic Open University, Patra, Hellas

***Abstract*— Today's life has been simplified by the advent of IoT technology. Smart Homes and Smart Cities tend to be the most frequent subject of study, on this field of science. This work is concentrated on the design and implementation of an IoT network, over smart roads. Car accidents' rate gets higher over the years. A smart road network might offer very useful data for the construction of a real-time accident and traffic preventer. Hardware implementations are also included. The architecture, security and privacy preservation of the network are highlighted. Cryptography could be the tool to the creation of a safe and useful IoT application. A concluding solution to the Road Tragedy phenomenon may be offered by the Academic study and research. Safe and effective smart networks' research and development may simplify daily life and eliminate fundamental issues. All these solutions may be applied to the human society, as very useful and trustworthy approaches.**

Keywords—IoT, Smart Cities, Mobile Ad-hoc Networks, Privacy, Security, Encryption, Tesla Cars.

I. INTRODUCTION

The aim of this research is to study and develop an 'intelligent' Mobile Ad-hoc Network for the detection, identification and recording of events on a given traffic network. The data provided to the manager by the network may lead to case studies, from traffic frequency to accident prevention statistics. In particular, the modern electric cars are equipped with sensors, which could transmit the data to a cloud. Thus, the data could be converted into useful information, under appropriate processing, with the goal of creating secure traffic networks in the cities.

Internet of Things (IoT) is the wide concept of vehicles, home devices etc, which could be connected via software, sensors, activators and networks, that allow these objects to exchange data [1-3]. IoT forms a concept that relates to daily objects, that use built-in sensors to collect data and act on them within a network. In brief, the IoT is the technological future that will make our lives easier [1-3].

Ad-hoc Networks are one of the most modern and challenging research sectors in automation industry. A wide range of applications, such as safety, mobility and connectivity for both the driver and passengers, transport systems in a smooth, efficient and secure way could be exploited by the presence of such networks.

More specifically, this study focuses on the interaction and integration of various critical elements of an Ad-hoc Traffic Network. An Ad-hoc Traffic Network is a wireless network where the communicating nodes are mobile, and the network topology is constantly changing. Wireless sensors can detect any events such as accidents, as well as frozen roads and can forward rescue /warning messages via intermediary vehicles for any necessary help. We therefore propose an Ad-hoc network architecture that uses wireless sensors to detect events and effectively transmit security messages using different service channels. Moreover, a control channel with different priorities may be built.

The purpose of designing this system is to increase driving safety, prevent accidents and effectively use channels by dynamically adjusting the control and service channels' time slots. We will propose a method that can select some driver nodes between vehicles running along a national highway to efficiently transmit data. The method followed can be a guide to managing traffic issues and preventing accidents. The generality of the methodology lies in the fact that the traffic frequency, in existing traffic networks, road behavior, and the availability of electric cars vary by region. However, this work could help in the implementation of a "smart" Ad-hoc traffic network that would be applicable in every state.

This work is organized as follows. First and foremost, the theoretical background is sited. Trust, authentication and Ad-hoc Networks are the necessary terms to be analyzed. MANETs (Mobile Ad hoc NETworks), and more particularly their sub-category VANETs (Vehicular Ad hoc NETworks), are the theoretical model to be implemented. The proposed model, a safe traffic network, is introduced. The network components, as well as the algorithm implemented are shown in detail. Last but not least, the benefits and drawbacks of the proposed model in our daily life are listed and highlighted. The positive effect of the implemented model and the significance of academic research in human life issues are underlined. Online simulations and implementations are included.

II. TRUST MANAGEMENT INFRASTRUCTURES

The significance of trust management infrastructures is highlighted. Trust models are implemented only in small, static networks due to their management constraints and memory requirements. A peer-to-peer validation is required by a web-of-trust model [4]. However, it lacks feasibility for non-static networks. At least one trust anchor, that organize on-the-fly connection requests, between network nodes, manages a hierarchical trust model. This system is supposed to be appropriate for static networks. The categorization of hierarchical trust models exists as follows: Trust Center Infrastructures (TCI) (system Kerberos) and Public Key Infrastructures (PKI) (X.509, Card Verifiable Certificates (CVC) [5].

The most vital part of a digital identity certificate is the identification of both peers. The name of a web resource can only be identified by the Uniform Resource Identifiers (URI). Notwithstanding, the URI may be considered as futile, depending on the expected number of IoT devices. Thus, we use IPv6 address as its unique device identifier. Public Key Cryptosystems, are based on a pair of keys, which is authenticated by both peers, each time. Two of the most famous public key cryptosystems are:

- ✓ *Rivest–Shamir–Adleman (RSA): based on the difficulty of factoring the product of two large prime numbers.*
- ✓ *Elliptic Curve Cryptography (ECC): a quite fresh approach to public key cryptography based on the algebraic structure of elliptic curves over finite fields.*

ECC is considered as faster than RSA and has been established as the leading public key cryptosystem of choice, for resource-constrained embedded systems. Therefore, an IoT device contents a single universal certificate, that lasts the same as the expected operational life span of the device [1].

Customized domain-specific Object Identifier (OID) extensions should be defined due to the lack of a standardized framework for the encoding of device attributes entailing authorization credentials in a certificate. Concerning the Trusted Authentication Protocols, one or more nodes may be connected by a device with multiple simultaneous peer-to-peer connections. Transmission Safety Protocol (TLS) refers to the application level protocol in an IP-based environment [6].

A. Trust in the Internet of Things

The individual devices of any trust management system should be protected by the IoT (Figure 1). Encapsulation via memory virtualization, usually fails to be processed by a trustworthy firmware. Consequently, the individual components firmware trustworthiness determination, are not enough. Thus, the firmware overall image should be validated. An integral component to maintain security may overpass the obstacle of the lack of a secure device firmware updating or patching mechanism. Otherwise, several systems can be compromised by a foible. A network-wide update mechanism should be included in an effective patching process. By this mechanism, integrity robustness and authenticity checks, service outages minimization, and a version rollback permission -if necessary- may be goaled.

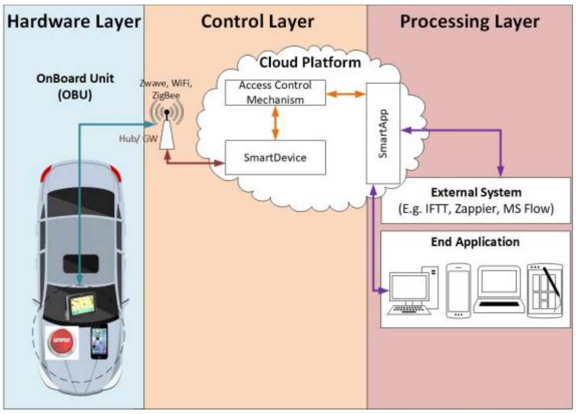


Figure 1 Visualization of an IoT Network

The system should process as follows:

- ✓ *Trust tokens exchange and validation or new session tokens creation.*
- ✓ *Data integrity assurance, optionally combined with data confidentiality via encryption, for the data suggested trustworthines.*
- ✓ *Implementation of data confidentiality via symmetric encryption, often directly in hardware; usually, data integrity is provided via message authentication codes, or cryptographic hashes, attached to the payload data.*

In this way, we reassure the construction of a viable mechanism, protected against fabrication [7-8].

B. Security Protocols for IoT Access Networks

Nowadays, the main pillars that represent the basic technologies are listed as four. They preserve the most common vertical applications related to automation or machine interaction formulate IoT architecture [9]:

1. Radiofrequency ID (RFID); with target to the objects' identification and tracking through tags, spared in the environment or attached to an object, is considered to be the most disseminate technology.
2. Machine-to-Machine (M2M) communications.
3. Wireless Sensor Networks (WSN); a constitution of several sensors widely split in the environment, with the ability of monitoring physical values and wireless communication in a multi hop mode. Its reference standard is the IEEE 802.15.4 [10].
4. Supervisory Control and Data Acquisition (SCADA); a real-time smart monitoring autonomous system. It preserves heterogeneity of terminals and the necessary guarantee for the data security [11].

The analysis would be incomplete, with the elimination of the vast amount of data management, due to the billions of information, from the environment to the Internet. A cloud platform's responsibility includes data storage, computation, visualization, and transforming into useful information. The providence of specific services and the necessity of each object's address could be preserved by a standardized platform. Some issues arising from the diffusion of an IoT are the heterogeneity of terminals, and the need for data security guarantee, from their collection to their transmission.

Finally, the cognitive security is introduced and applied to the time-based security solution. It highlights the main parameters that need to be monitored and measured by actors to strengthen the security in a parti-colored and variable scenario like the IoT [12].

C. *Authentication in IoT Networks*

The parties involved in the entity authentication are:

- ✓ *Claimant (that declares its identity as a message).*
- ✓ *Verifier (that is preventing impersonation).*
- ✓ *Trusted Third Party (mediates between two parties to offer an identity verification service as a trusted authority).*

Transferability and impersonation are included in the entity authentication objectives. The factors of entity authentication are classified, as follows: something known, something possessed and something inherent. These techniques have now been extended beyond authentication of human individuals to device fingerprints. The levels of entity authentication are categorized as weak authentication, strong authentication and Zero-Knowledge (ZK) authentication.

The reciprocity of identification, the computational efficiency, the communicational efficiency, the third party and the timeliness of involvement entity, are the authentication properties that are of interest to users. A central authority (CA) often runs offline to edit public-key certificates. The nature of trust, the nature of security guarantees and the storage of secrets, constitute the most important components.

D. *Ad-hoc Networks*

Hereby, we are focused on the interaction and integration of various critical elements of a Mobile Ad-hoc Network (MANET). A MANET is a wireless network, where the communicating nodes are mobile, and the network topology is constantly changing. Wireless sensors can detect any events such as accidents and can proceed warning messages, via intermediary vehicles for any necessary assistance. The proposed architecture is an ad-hoc network that incorporates wireless sensors to detect events and effectively transmit security messages, using different service channels and a control channel with different priorities. [13].

For security applications, the best routing protocol should be selected. The three most common routing protocols used in the MANET are: Dynamic Source Routing (DSR), Ad Hoc On-Demand Distance Vector (AODV) and Destination-Sequenced Distance Vector (DSDV). Indeed, it is important and necessary to test and evaluate the different routing protocols related to the MANET, before implementing them in the real environment. This can be done through MANET simulation tools. Our goal is to measure the performance of the routing model, for city scenarios. The main objective is to find the appropriate routing protocol, in a high-density traffic area.

A MANET is a self-tuning and wireless network of mobile devices, connected via wireless links, (Figure 2). Every device in a MANET is free to move to any direction, and therefore often changes its links with other devices. Each of them should promote the data circulation, that is not related to its own use, and thus act as a router. The main challenge for building a MANET is to supply each device, so that it always maintains the necessary information to proper route traffic. These networks can either operate autonomously or connect to the Internet. MANETs are a kind of wireless ad-hoc network with a routable network environment at the top of the Open Systems Interconnection (OSI) Reference Model Data Link Layer.

One of the main types of MANETs is Vehicular Ad-hoc NETwork (VANETs). VANETs are used to ease the communication among vehicles and among vehicles and equipment en route. More specifically, this work will be dealt with by InVANETs (Intelligent Vehicular Ad hoc NETworks - Intelligent VANETs). It is a kind of artificial intelligence that helps vehicles behave intelligently during vehicle-related crashes, accidents, driving under the influence of alcohol, etc. The node eviction in VANETs forms the main cause of interest thereby [16]. A Vehicular networking features include high-speed mobility, short-lived connectivity, and infrastructureless networking constitute the formation of a VANET.

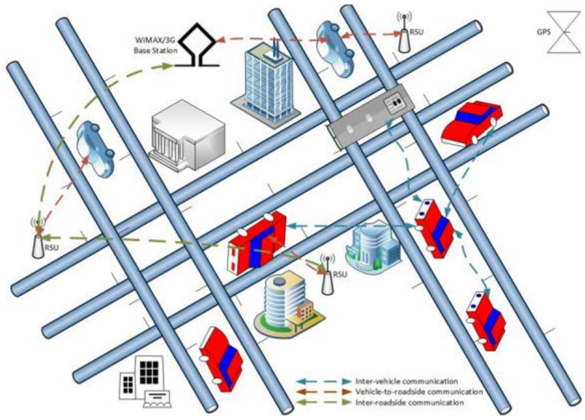


Figure 2. Visualization of a VANET

VANETs consist of vehicles equipped with wireless gadgets [14]. Communication in VANET occurs between vehicle and vehicle operation, and the road with which an intelligent traffic system gets formed. Routing plays an important role in promoting the required data to nodes or vehicles. Some reactive routing protocols, such as AODV and DSR protocols and proactive routing protocols such as Optimized Link State Routing (OLSR) in urban traffic scenarios are examined. Simulation of Urban Mobility (SUMO) and network performance using Network Simulator 3 (NS3) to find an appropriate protocol using network parameters, and delay are being used. The simulations have shown that AODV proceeded well with other routing protocols in VANET scenarios [15].

III. Proposed Model

VANET is an exemplary IoT, with vehicles as things connected to the IoT [17]. Intentionally, faulty messages get inserted to VANET with the potential of massive destruction by malicious nodes. Other than faulty nodes, malfunctioning Onboard Units (OBU) with fatal aftermaths in safety applications obstruct VANET's performance [18]. Moreover, massive destruction may be caused by faulty messages inserted to VANET by malicious nodes. Errant nodes should get removed anyway from VANET as fast as possible. Traditionally, an errant node's certificate gets revoked by a centralized CA. Nevertheless, CA-based approaches become ineffective due to the nature of VANET. Nodes are allowed to decide and act against other errant nodes both distributed and locally by current node-eviction schemes in VANET (Figure 3). Local node-eviction schemes can be classified into four categories: Reputation, Vote, Suicide Abstinence and Police. Various factors may affect the performance of node-eviction schemes. It gets strong in model behaviors and goals of single nodes by the richness in flexibility and emergence of an agent-based simulation. The simulation scenario is formed by a circular road setup in the grid, where vehicles at different speeds cycle around the road and communicate with each other or with the RoadSide Unit (RSU) when nearby.

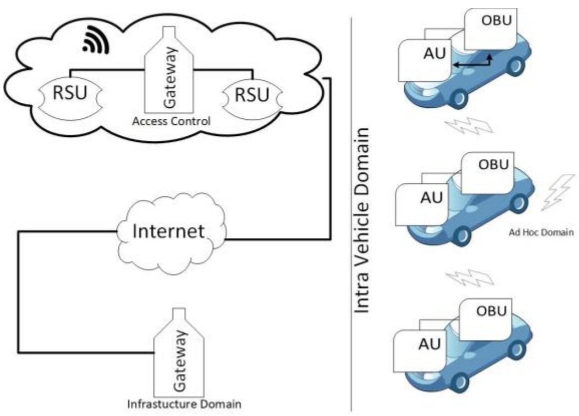


Figure 3. Visualization of a VANET

The RSU transfers the information to the CA. In our model, the node-eviction scheme and frequency of contact are implicit. Any node eviction scheme should be able to optimize the average time, risk, and utility measures under dynamic environment conditions. The node eviction process gets modeled as a set of states and transitions. Eventually, two subnets get formed, separating all nodes, depending on their good or bad state. A state transition occurs as long as a node moves from Subnet I to Subnet II. Finally, Subnet I or Subnet II will converge into the same kind of nodes. A network message exchange, certificate-controlled model, form the final system. Each node formulates a List of other nodes' Valid Certificates (LVC). As long as good and bad nodes are separated with insignificant risk, the procedure terminates. However, it gets complicated the individual police node to capture all the bad nodes on time. In parallel, as the percentage of bad nodes increases, multiple bad nodes pop up simultaneously at different spots. Moreover, possibly some of the bad nodes never being caught, meaning a high risk [9].

The VANET applications are based on the precise information, providence to the drivers. Nevertheless, VANET content delivery includes serious security threats. Common metrics cannot be precisely measured, according to the effectiveness of different techniques. Thus, consumers cannot be reassured, especially with regards to the critical road safety concerns. However, security measurement is difficult and differs from other kinds of measurement, like quality of service in wireless multimedia. An Asymmetric Profit-Loss Markov (APLM) model, constructs a security metric. Briefly, profits are considered to be incidents of detecting data disasters, and the ones of accepting corrupted data as damages.

A. Case of Study

Houston is the capital of the American State of Texas, located southeast and bordered by the Gulf of Mexico. It has population of over 6,000,000 inhabitants and an expanse of 1,558 km^2. It is chosen, as the area of study, because it is in the 2nd place of the traffic congestion table, but also in the 6th place of the fatal accidents chart among the USA.

It is very important to bear in mind that in Houston, according to the recorded car events of 2016, a man was killed every 2 hours and 20 minutes. One person was injured every 1 hour and 59 seconds, and a recorded incident took place every 57 seconds [19]. In total, for 2016, the privately-owned vehicles registered by the U.S. Service vehicle registration statistics reach 261.8 million. The daily statistics of Houston's traffic congestion are shown in Figure 4.

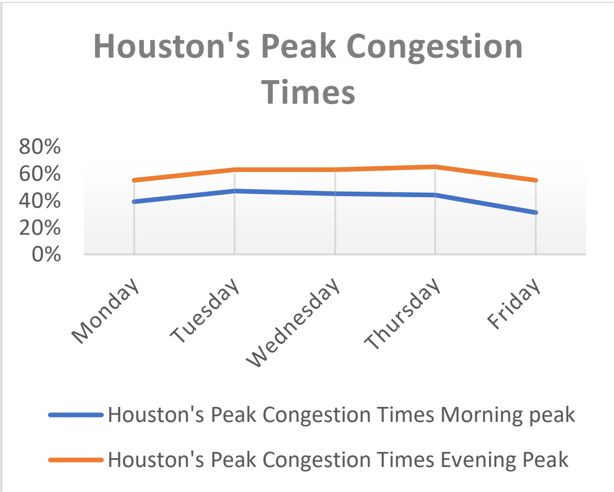


Figure 4. Houston's Peak Congestion Times

The yearly statistics of Houston's traffic congestion are shown in the below Figure 5.

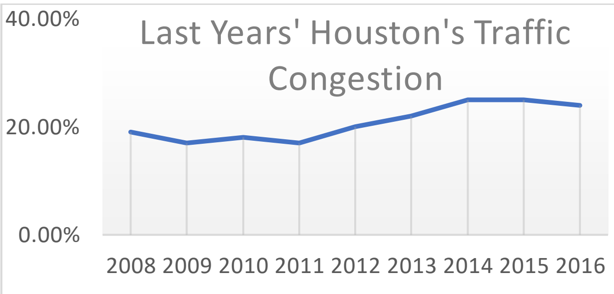


Figure 5. Houston's Congestion

All the mentioned above, prove the usefulness of a smart application for traffic regulation in a state with increased traffic issue. The rate of injuries and deaths in the area of study, necessitate the creation of an ad-hoc network that can provide real-time data for the study, prevention and rehabilitation of the traffic network.

B. Tesla Cars

Tesla cars, with their advanced technology, can provide us with information transfers about what is going on in the street. They are the only candidates to perform the OBU role [23].

Specifically, the Tesla S was designed from the beginning as the safest, most exciting sedan on the road. With outstanding performance delivered through Tesla's unique electric engine, the S-Series accelerates from 0 to 60 mph in just 2.5 seconds. The S model incorporates an Autopilot feature that is designed to make a motorway drive safer, (Figure 6), [23].

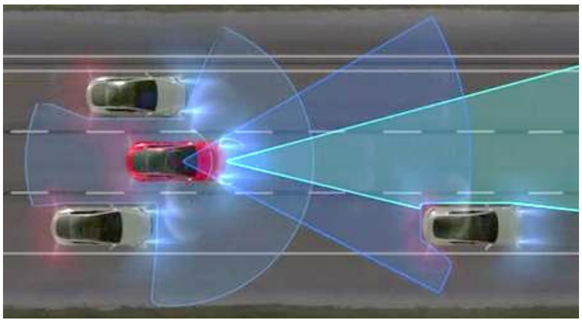


Figure 6. Tesla Autopilot System

The driver's safe driving system is based on the following:

1. *Eight peripheral cameras offer 360 degrees of visibility around the car up to 250 meters.*
2. *Two-time ultrasonic sensors complement this vision, allowing the detection of hard and soft objects almost twice the distance of the previous system.*
3. *A forward-looking radar with improved processing, providing additional data for the world at an unnecessary wavelength that can be seen through intense drop, fog, dust and even the car forward.*

C. The Algorithm

The basic idea of the algorithm is that the essential data are used, to alert the driver for any possible events, throughout the road network. The loop keeps on until there are not essential data to keep the driver vigilant. The following simple algorithm can lead the information of the system to the administrator and each driver for the criticalness of the road events (Figure 7). The daily use of the data produced, may offer useful statistics concerning the special roads, or crucial parts of the street that need attention. Repeating the algorithm, big data can be produced for any necessary road construction works. The algorithm is visualized in Figure 8.

Applying the algorithm, the vehicle data may be collected by the RSU, be processed and used equally. In this way, the driver may be alerted for any kind of danger appearing and the system administrator may be notified to intervene, if necessary. The daily collection and processing may highlight the need for road works or speed limitation for the elimination of road traffic or accidents rate minimization.

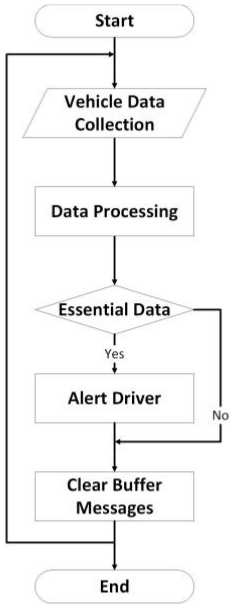


Figure 8. Visualization of the Algorithm

D. A Traffic Simulation Framework

An online simulation was implemented to justify the proposed system. Given real-time data collected from the distributed online simulations, necessary information for near real-time traffic decisions get provided by the IoT traffic system. The traffic IoT network is divided into dynamic overlapped sections, and a simulation processor mapped to each section. Nearby RFIDs and sensors supply each simulation with real-time data, enabled to run continuously. A collection of segment simulations formulates the overall distributed simulation. In this, each small segment of the overall traffic IoT network is modeled based on local criteria. The information exchange among vehicles moving from one simulation segment to another is allowed in the simulation. Each simulator's segment locally models current traffic conditions and shares its predictions with other simulation segments. Altogether, they create an aggregated view of both the individual segment ‘s area of interest and the overall of traffic system. current traffic state information and their predictions to the simulation server are published by the simulators' segments. An accurate estimation of a future state of the system is provided by an aggregation of all simulation segments provides. All the mentioned above are reflected in Figure 7, [21].

Significant network bandwidth and amount of computation by each simulator host are required by the current large-scale distributed simulation methodologies. The communications loads placed in the network can be reduced by mobile agents. Agents communicate with a specific simulation segment, providing all the state information sent to the simulator server.

For modeling a collection of adjacent intersections, NetLogo simulator has been used. Different network features are represented by static and mobile agents. Motor vehicles have been modeled individually within NetLogo using mobile agents.

By NetLogo, instructions can be given to many independent agents which could all operate at the same time. Four types of agents are used in NetLogo: patches represent the static agents, turtles represent the mobile agents; links make connections between turtles; and the observer oversees everything going on in the simulated environment [21].

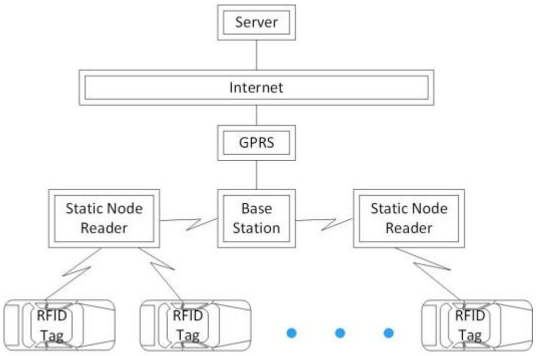


Figure 7. Distributed Online Traffic Simulation Framework

Java is the programming language of the NetLogo environment. In this simulation, the agent entities are vehicle, traffic lights, and sensors of intersections and lanes. Agents are created and randomly distributed over the network of intersections. A random number of vehicles are set to limits defined in the model. Sensors recorded the number of passing vehicles. The traffic lights' action is based on vehicles' waiting time minimization and vehicles successful pass through intersections throughput increase. The following indicators are bore in mind per run: not moving vehicles, average waiting time and average speed of the vehicles in a time step. Usually, the driver's behavior is unpredictable. Drivers' behavior modeling has been performed based on techniques proposed by. The simulation has 'setup' and 'go' switches. The 'setup' switch sets a procedure to reverse the model to the initialization state. The 'go' switch initiates a procedure that carries out all the necessary actions for each simulation. The interface and performance evaluation of the simulation results are shown in Figure 8 [21].

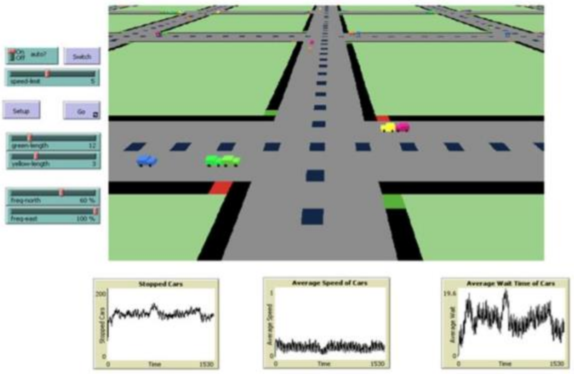


Figure 8. Interface And Performance Evaluation of the Simulation Results

IV. Efficient Hardware Implementations

The system proposed may be implemented by Udoo Kits. Their technologies form a full IoT implementation platform [24]. Actually, it is a single-board computer, Arduino-compatible, that can perform Android or Linux OS. Its benefits are its ease-to-use, with minimum knowledge requirements (Figure 7). Different computing methods, emphasizing on the proper and weak points of each are combined. Educational purposes are the basic reason of Udoo Dual/Quad [23]. A well-trained team that can built-up new applications and projects, using a low-cost and user-friendly platform, may be created for its use. Thus, a useful tool for high-standards implementations may be provided to institutions and companies.

Following the rules of trust and authentication, IoT may be successfully implemented. As the technology evolves, more and more requirements are necessary to networks and systems. IoT systems, are representative of bridging and maintaining complex systems at every appearance of real life.

Figure 9. Udoo Kit: An IoT Implementation Platform

Udoo kits basically consist of touch displays of 7-15 inches, featuring high resolution that makes the content easy to be read, USB gates, USB cables for additional gates and LCD board adapters. The main representative, integrated systems suggested by Udoo are Udoo KIT LCD 15,6" Touch and Udoo KIT LCD 7" - Touch for QUAD/DUAL. The Udoo kits include WiFi technology (as well as ethernet), camera connectors and their capacity may reach up to 2,5 GHz (CPU), 700 MHz (GPU) and 8GB (RAM) [24].

V. Advantages of The Proposed Model

VANETs offer innumerable benefits to organizations of any size. High speed internet access of cars will transform the vehicle's computer from an elegant gadget, into a basic productivity tool, making almost any web technology available in the car. While such a network creates some security concerns, it does not limit the VANETs' dynamics, as a productivity tool. It allows the "dead time", that is lost while waiting for something, to be transformed into "useful time", time used to perform tasks. A passenger can turn a traffic congestion into a productive working time. Even GPS systems can benefit as they can be integrated with traffic reports, to provide the fastest route to run. Finally, it would allow free VoIP services, among the converters, reducing the cost of telecommunications.

On the other hand, while Internet can be a useful productivity tool, it can also turn out to distract enough attention, resulting in security and real-time consuming concerns. Checking emails, surfing the web, or even watching videos, can distract a driver's attention from any danger in the street.

VI. Conclusions & Outlook

While still years away, VANET is a technology that could significantly increase productivity in times that are usually not productive. However, to achieve this, VANET users must first overcome the loose temptations and distractions of Internet. Recent developments in wireless communications technologies and in the automotive industry have generated significant research interest in VANETs in recent years. VANET consists of vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) technologies supported by wireless access technologies such as IEEE 802.11p.

This innovation in wireless communication, is designed to improve road safety, and traffic efficiency, to the close future through the deployment of Intelligent Transport Systems (ITS). As a result, the government, the automotive industry and academia, cooperate to a large extent through various ongoing research projects to establish standards for VANETs. The typical set of VANETs application areas, have made VANETs an interesting wireless domain. This document provides an overview of the current research situation, challenges, VANETs capabilities and the path towards achieving the long-awaited ITS [24-25].

The innovative safety systems such as ABS, seatbelts, airbags, backlight cameras, electronic stability control (ESC) have not reduced the car accidents' rate, which is highly increased. Several studies have argued that 60% of motorway accidents could be avoided if warning warnings were given to drivers just a few seconds before the time of the collision.

The academic community is the one that will play the vital role in the regulation of another social life issue. This implementation may lead to the expunge of traffic problem. Smart systems and intelligent networks may be the tool to this problem's resolution.

The IoT science has evolved throughout the years and daily life has been simplified significantly. Road traffic and car accidents could not be out of IoT science's scope. The real- time preventer that is examined in this paper may be a revolutionary discovery for another side of the daily life. The features of modern implementation platforms may cover the needs of such issues arising.

ACKNOWLEDGMENT

This work is under the UMI-Sci-Ed project. This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 710583.

REFERENCES

[1] N. Sklavos, I. D. Zaharakis, A. Kameas, A. Kalapodi, "Security & Trusted Devices in the Context of Internet of Things (IoT)", IEEE proceedings of 20th EUROMICRO Conference on Digital System Design, Architectures, Methods, Tools (DSD'17), Austria, August 30 – September 1, 2017.

[2] N. Sklavos, I. D. Zaharakis, "Cryptography and Security in Internet of Things (IoTs): Models, Schemes, and Implementations", IEEE proceedings of the 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS'16), Larnaka, Cyprus, November 21-23, 2016.

[3] I. D. Zaharakis, N. Sklavos, A. Kameas, "Exploiting Ubiquitous Computing, Mobile Computing and the Internet of Things to Promote Science Education", IEEE proceedings of the 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS'16), Larnaka, Cyprus, November 21-23, 2016.

[4] G. Guo, J. Zhang, "Improving PGP web of trust through the expansion of trusted neighborhood", IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011, University of Saskatchewan, Canada.

[5] A. Arsenault, S. Turner, Internet X.509 public key infrastructure PKIX roadmap, IETF Roadmap, September 8, 1998.

[6] M. Bourlakis, I. P. Vlachos, V. Zeimpekis (editors), Intelligent Agrifood Chainsand Networks, Wiley-Blackwell, 2011.

[7] N. Sklavos, "Cryptographic Algorithms on A Chip: Architectures, Designs and Implementation Platforms", proceedings of the 6th Design and Technology of Integrated Systems in Nano Era (DTIS'11), Greece, April 6-8, 2011.

[8] N. Sklavos, "On the Hardware Implementation Cost of Crypto-Processors Architectures", Information Systems Security, The official journal of (ISC)2, A Taylor & Francis Group Publication, Vol. 19, Issue: 2, pp. 53-60, 2010.

[9] Arzad Kherani and Ashwin Rao, Performance of node-eviction schemes in vehicular networks, IEEE Transactions on Vehicular Technology, vol. 59, no. 2, pp. 550–558, 2010.

[10] H. Tseng, S. Sheu, and Y. Shih, "Rotational listening strategy (rls) for IEEE 802.15.4 wireless body networks," IEEE Sensors J., vol. 11, no. 9, pp. 1841–1855, 2011.

[11] P. Kasinathan, C. Pastrone, M.A. Spirito, and M. Vinkovits, "Denial-of Service detection in 6LoWPAN based Internet of Things," in Proc. Of IEEE 9th Intl. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob), 2013, pp. 600–607,7–9 October 2013.

[12] M.R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L.A. Grieco, G. Boggia, and M. Dohler, "Standardized protocol stack for the Internet of (important) Things," IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1389–1406, 2013.

[13] J. Tan, and S.G.M. Koo, "A survey of technologies in Internet of Things," in Proc. of IEEE Intl. Conf. on Distributed Computing in Sensor Systems (DCOSS), 2014, vol., no., pp. 269–274,26–28 May 2014.

[14] Sijing Zhang, Enjie Liu. Vehicular ad hoc networks (VANETs): Current state, challenges, potentials and way forward. Elias C. Eze, Centre for Wireless Research, Institute for Research in Applicable Computing (IRAC), Department of Computer Science and Technology, University of Bedfordshire, Luton, LU1 3JU, England.

[15] Viswacheda Duduku. V, Ali Chekima, Farrah Wong, Jamal Ahmad Dargham. A Study on Vehicular Ad Hoc Networks., Univ. Malaysia Sabah, Malaysia.

[16] S. Gao, J. Ma, W. Shi, G. Zhan, and C. Sun. Trpf: A trajectory privacy preserving framework for participatory sensing. IEEE Transactions on Information Forensics and Security, vol. 8, no. 6, pp. 874–887, 2013.

[17] M. Groat, B. Edwards, J. Horey, W. He, and S. Forrest. Enhancing privacy in participatory sensing applications with multidimensional data. In Proc. of 2012 IEEE International Conference on Pervasive Computing and Communications (PerCom '12), pp. 144–152,2012.

[18] Jonathan Andrew Larcom and Hong Liu, Authentication in GPS-directed mobile clouds, in Proceedings of IEEE Global Communications Conference 2013 (IEEE GLOBECOM 2013), pp. 470–475, Atlanta, GA, 9–13 December 2013.

[19] Texas Department of Transpotation, www.txdot.gov, 2018.

[20] Tom Tom Traffic Index , https://www.tomtom.com/en_gb/trafficindex/.

[21] Hasan Omar Al-Sakran "Intelligent Traffic Information System Based on Integration of Internet of Things and Agent Technology", Management Information Systems Department, King Saud University Riyadh, Saudi Arabia, International Journal of Advanced Computer Science and Applications, Vol. 6, No. 2, 2015.

[22] Tesla Cars, www.tesla.com, 2018.

[23] Udoo Kits, www.udoo.org, 2018.

[24] R. Piquepaille, "Turning Cars into Wireless Network Nodes", ZDNet, Vehicular Network Lab @ UCLA – Implementing the First Campus Vehicular Testbed, Vehicular Lab, 2007.

[25] P. McCloskey, "The Mobile Internet: Your Car Could Save a Life", medGadget, 2007.

Algorithm Selection for Non-Linearly separable Algorithms in Computer Vision

Martin Lukac, Nadira Izbassarova
and Albina Li
School of Science and Technology
Nazarbayev university
Astana, Kazakhstan, 010000
Email: martin.lukac@nu.edu.kz

Michitaka Kameyama
Ishinomaki Senchu University
Ishinomaki, Miyagi, Japan
Email: michikameyama@isenshu-u.ac.jp

***Abstract*—In this paper we experimentally analyze the problem of single step algorithm selection in the field of computer vision. For this we introduce a data set based on the VOC2012 that allows to evaluate different algorithm selection approaches. We study the algorithm selection problem formulated as the multi-class classification by analyzing the feature selection, feature compression and data augmentation. We evaluate three different classification algorithms on the benchmark data set. The algorithms used for creating the dataset were selected so that both diversity in performance as well as implementation is represented. We show that while the presented accuracy of the evaluated algorithm selection method is at maximum 44.96% for five algorithms, increasing the algorithm selection accuracy can lead to significant improvement in task result score.**

I. Introduction

Current state of art in many areas of real-world problem solving relies on a large amount of algorithms. Many of these algorithms are very specific to a problem sets or problem instances. For instance in computer vision each sub-problem is represented by literally hundreds of algorithms: object recognition [1], [2], [3], [4], image segmentation [5], semantic segmentation [6], [7], [8], [9], [10], [11], classification [12], [13], etc.

Some of the algorithms are very task specific and results in very high accuracy of the task within its domain. This is in particular the case of the many approaches based on Deep-Learning (DL) and convolutional Neural Networks (CNN). Others algorithms' domain is wider but their average accuracy is lower. This is in general the case of algorithms using engineered features; these features are less specific but are property preserving or resisting.

Because most of the Machine Learning (ML) approaches are data dependent and sensitive, a large number of these algorithms are constantly in development. Therefore a method for optimizing the average algorithm accuracy should be designed with benefits in both performance improvement as well as in the generalization of the domain of the overall approach.

Recently, with the advent of GPGPU technology, the acceleration of DL, CNN, Big Data (BD) and Reinforcement Learning (RL) allowed for the design of particular class of algorithms as well as to recombine existing algorithms for certain applications. While designing algorithms with RL is an appealing approach, the extremely large data and time to obtain solution is in most of cases unrealistic and not achievable [14], [15], [16], [17], [18].

Instead of designing algorithms from scratch, one can gather the already available very focused algorithms, and exploit their strength on a case by case basis by an algorithm selection mechanism [19]. Algorithm selection (AS) is an approach where from a set of algorithms the best one is selected on a case by case basis. Mostly successful on synthetic or logic problems [20], [21], [22] recently the algorithm selection was also applied to real world problems such as computer vision or image processing [23]. Some success was also obtained in more advanced tasks of computer vision such as scene understanding and semantic segmentation [23]. However, for the more advanced tasks a system based approach was required.

In this paper we present a set of experiments that estimate the accuracy of algorithm selection in the semantic segmentation. We estimate the accuracy of algorithm selector using various features selection, machine learning parameters adjustment and synthetic data generation. Additionally we also evaluate the algorithm selection with higher level regional features and semantic annotations. Finally, a benchmark dataset based on selected algorithms results on the VOC2012 dataset [24] is introduced.

We show that while the algorithm selection approach is extremely appealing a direct and only machine learning approach is not the most convincing approach.

This paper is structured as follows. Section II provides the necessary background into the algorithm selection and related topics. Section III describes the data set used and Section IV presents the individual experimental settings. Section V discusses the results. Section VI concludes the paper.

II. Background

Let $A = \{a_0, \ldots, a_{k-1}\}$ bet a set of algorithms, all of them solving a problem defined by the mapping $P : I \to L$, with $I = \{i_0, \cdots, i_{n-1}\}$ being the set of input images and $L = \{l_0, \ldots, l_{l-1}\}$ being a set of labels. Each label represents a distinct object or category. The mapping P assigns to each pixel $p_{xy} \in I$ a label $l_{xy} \in L$.

Let there be two ground truth sets of labels: $C = \{c_0, \ldots, c_{n-1}\}$ and $s = \{s_0, \ldots, s_{n-1}\}$. The set C is the set of target labels associated with each input image for the classification task such that $j = 0, \ldots n-1,\ c_j \in L$. The set S contains a set of sets $s_j = \{s_j(0,0), \ldots, s_j(x-1, y-1)\}$ such that $for\ j = 0, \ldots, n-1, (a,b) = (0,0), \ldots, (x-1, y-1),\ s_j(a,b) \in L$. Each element of s_j represents the labels for each pixel $i_j(a,b)$ of the associated input image i_j of size $x \times y$ for the semantic segmentation task.

The process of algorithm selection can be described according to Figure 1. The algorithm selection process starts from an initial image from which a set of features is extracted. The features and if available additional information is used as input to the algorithm selection mechanism. The selection outputs the identifier for a single algorithm which is then used to process the input image and generate output result. The

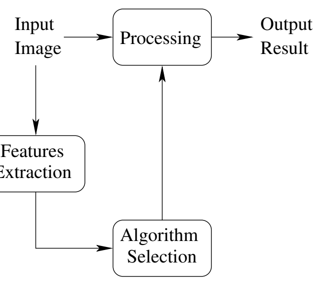


Fig. 1

process of labeling can be divided into to main classes: scene classification and semantic segmentation. In scene classification, the output of each algorithm is a single label as shown in eq. 1. it is the extreme case of of labeling, where every pixel has the same label.

$$c_j = a_i(i_j) \tag{1}$$

with $c_j \in L$, $a_i \in A$ and $i_j \in I$. For semantic segmentation, each algorithm assigns label to each pixel of the input image such as shown in eq. 2

$$s_j(x,y) = a_i(i_j(x,y)) \tag{2}$$

with $s_j(x,y) \in L$, and $i_j(x,y)$ is a pixel located at coordinates x, y in image $i_j \in I$.

The result of any algorithm a_i is evaluated using an error function. In computer vision one of the common measure to evaluate algorithms is the f-measure. For the classification task, the f-measure is reduced to the ratio of correctly classified images over all available images (eq. 3).

$$m_c(a_i, I) = \frac{\sum_{j=1}^{n} \iota(a_i(i_j) == l_j)}{n} \tag{3}$$

with $\iota(\cdot)$ is an indicator function defined as shown in eq. 4

$$\iota(a_i(i_j) == l_j) = \begin{cases} 1 & \text{if } a_i(i_j) == l_j \\ 0 & \text{O.W.} \end{cases} \tag{4}$$

In semantic segmentation, each pixel can have a different label. In general to measure the accuracy of semantic segmentation a more appropriate measure is used. One of the common measures is the f-measure can be defined: here we use the Intersection of Union (IOU). Let $s_i^j(a,b)$ and $s_j(a,b)$ be the label for pixel generated by algorithm a_i and the desired label from ground the truth respectively. As shown in eq. 5 the IOU is the ratio of correctly labeled pixels over the number of all pixels that have been labeled a) correctly as label $s_j(a,b) = l_j$ called *true positive (TP(a,b))*, b) incorrectly as label l_k while $s_j(a,b) = l_j$ (called *false positive (FP(a,b))*) and c) incorrectly as label l_j while $s_j(a,b) = l_k$ (called *false negative (FN(a,b))*).

$$m_s(a_i, I) = \sum_{j=1}^{n} \sum_{a=0,b=0}^{x,y} \frac{TP(a,b)}{TP(a,b) + FP(a,b) + FN(a,b)} \tag{5}$$

The $m_s(a_i, I)$ will be referred to in this paper for simplicity as Semantic Segmentation Accuracy (SSA).

Similarly to the classification problem the algorithm selection can be specified as a binary decision problem. Let there be a set $T = \{t_0, \ldots, t_{n-1}\}$ with elements defined by eq. 6.

$$t_j = arg\ max_k\ m_c(a_k, i_j) \tag{6}$$

Then the average accuracy of any algorithm selector can be simply given by analogy to $m_c(\cdot)$:

$$\hat{s}_c(A, I) = \frac{\sum_{j=0}^{n-1} \iota(s_c(A, i_j) == t_j)}{n} \tag{7}$$

The $\hat{s}_c(A, I)$ will be referred to in the paper as Algorithm Selection Accuracy (ASA).

III. Data Set for Algorithm Selection

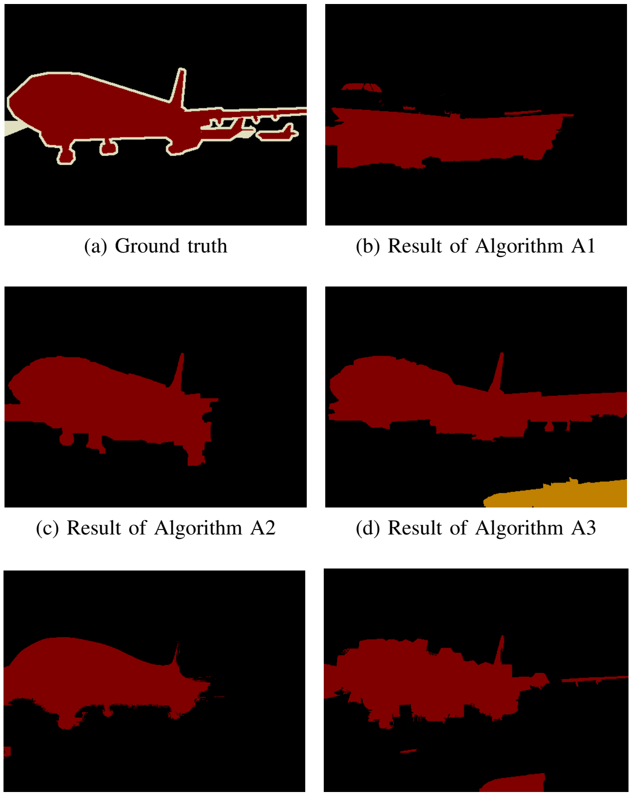
The data set prepared for this experiments is based on the validation data set of the VOC2012 challenge dataset [24]. The reason for using the validation dataset is two folds: a) the algorithms were designed and learned on the training dataset and thus the results can be strongly biased due to learning convergence and b) the validation data set allows to directly evaluate the task accuracy as in most cases the test data is not provided with the ground truth.

The validation data set contains exactly 1441 images, and for the experiments it was divided into train set of 1152 images, and test set of 289 images. The dataset consists of 20 classes in 4 categories: Person, Animal, Vehicle, and Indoor. The categories of of objects contained in the data set are the standard VOC2012 categories such as car, train, people, etc.

The data set contains the results of the evaluated algorithms as well as the input images. Example of the semantic segmentation obtained by the five algorithms for the same image is shown in Figure 2. The five used algorithms are very different:

A1 [25] is based on the use of object co-occurrence statistics to refine a graph-cut based segmentation. The co-occurrence statistics allow to indicates the chances of several classes to occur together in the image.

A2 [26] is based on generating multiple figure-ground hypotheses using machine learned region scores.



(a) Ground truth (b) Result of Algorithm A1

(c) Result of Algorithm A2 (d) Result of Algorithm A3

(e) Result of Algorithm A4 (f) Result of Algorithm A5

Fig. 2: Illustration of examples of ground truth and of the outputs of the five different algorithms

A3 [27] is the algorithm which consists of four steps, namely are 1) proposal generation; 2) feature extraction; 3) region classification; 4) region refinement.

A4 [28] is the algorithm that uses the existing convolutional neural networks with fine-tuning such as AlexNet, VGGnet, and GoogLeNet.

A5 [29] is a CNN with the architecture which is based on the use of feed-forward multi-layer neural network trained with asymmetric loss.

The average SSA of each of the algorithms evaluated on the VOC2012 validation dataset are for information shown below:

- A1: 48.473%
- A2: 47.048%
- A3: 67.637%
- A4: 50.089%
- A5: 69.873%

To verify how effective the algorithm selection can improve the semantic segmentation by using the five algorithms A1~A5, the initial experiment measures the semantic segmentation accuracy as a function of algorithm selection accuracy. For this experiment each image was broken into regions according to each algorithm segmentation result. Each region was scored and then selected proportionally to the score and to the accuracy of algorithm selection as shown in Table I.

TABLE I: Per class segmentation accuracies over all the images in VOC2012 with 100%-50% selection accuracy

Label	100% ASA	90% ASA	70% ASA	50% ASA
background	94.649%	94.649%	94.426%	92.753%
aeroplane	87.363%	87.363%	87.363%	84.237%
bicycle	39.996%	39.996%	39.838%	38.693%
bird	90.811%	90.811%	88.266%	86.394%
boat	81.360%	81.360%	81.360%	77.778%
bottle	80.827%	80.827%	80.795%	78.496%
bus	92.474%	92.474%	91.532%	90.605%
car	87.655%	87.655%	87.655%	86.164%
cat	92.404%	92.404%	92.312%	88.786%
chair	53.704%	53.704%	51.137%	48.948%
cow	91.060%	91.060%	91.060%	84.677%
diningtable	79.888%	79.888%	79.895%	76.061%
dog	89.636%	89.636%	89.730%	86.591%
horse	87.995%	87.995%	87.578%	84.298%
motorbike	84.106%	84.106%	83.647%	80.161%
person	85.143%	85.143%	85.048%	80.867%
pottedplant	73.632%	73.632%	73.632%	63.464%
sheep	88.885%	88.885%	86.556%	79.957%
sofa	75.154%	75.154%	74.969%	64.473%
train	90.041%	90.041%	90.041%	83.006%
tvmonitor	83.409%	83.409%	83.409%	76.555%
Average accuracy	82.390%	82.390%	81.869%	77.76%

Table I shows the results of the experimentation by evaluating semantic segmentation for each object class and for all classes in average. The first column in Table I indicates the object class, second to last columns shows semantic segmentation accuracy (SSA).

The statistical accuracy experiment was conducted as follows. For a given accuracy of algorithm selection (ASA) θ, perform a sampling by selecting the algorithm with highest ASA proportionally to θ. Thus for instance, 100% ASA means that for each class object on each image, the algorithm with highest θ is chosen to make segmentation. The ASA for each category of objects is the average ASA of the highest SSA across the five algorithms.

Third column (90% ASA) shows the result of semantic segmentation accuracies for each class, when 90% of times the algorithm with highest θ is chosen for a particular class to make segmentation. For the remaining 10% of times the algorithm for a class segmentation is chosen randomly among the other remaining four algorithms excluding the best one from the pool.

The best algorithm among the five algorithms used for semantic segmentation based on the highest accuracy is A5 with 69.873% SSA. Note that according to the results of statistical approach in the Table I, 50% ASA resulted in average 77.76% SSA, which is higher than the top accuracy among the five algorithms (69.873%). Thus, even with a relatively low ASA the resulting average SSA is higher than the SSa of algorithm A5!

IV. Experiments

In these experiments we focus on a slightly simpler evaluation of SSA. Instead of selecting algorithm for each segmented region, we only select algorithm based on average SSA for the whole image.

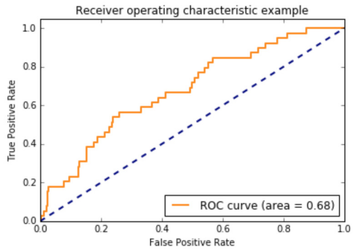


Fig. 3: ROC curve for classification using ResNet18 features

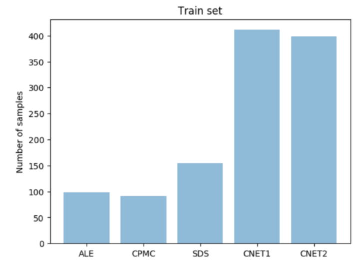


Fig. 4: Algorithm distribution histogram for train set

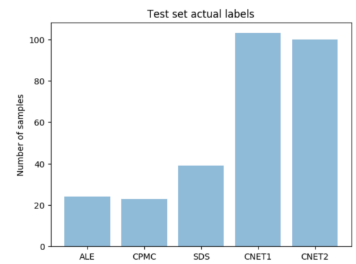


Fig. 5: Algorithm distribution histogram for test set

The first step required for algorithm selection is the features extraction. In order to have an accurate algorithm selection, we need to obtain distinctive feature set [23]. We use the following features for the experiments on VOC2012:

- Feature set 1: features obtained from the fourth convolutional layer of AlexNet.
- Feature set 2: features obtained from the fifth convolutional layer of AlexNet.
- Feature set 3: features obtained by concatenating output of the convolutional layer four to the output of the convolutional layer five of AlexNet.
- Feature set 4: features obtained from ResNet18.
- Feature set 5: visual bag of words using SIFT descriptors.

A. *Experiments with Features and Data Augmentation*

The first set of experiments were conducted using Feature sets 1-3, which are extracted from pretrained AlexNet. The classification algorithm used at the early stage is SVM because it is a good choice whenever the number of instances is less than the number of features. Moreover, since the number of train instances is 1152, and is low compared to the number of features in Feature set-1, Feature set-2 (both have 43264 features), and Feature set-3 (86528 features), we applied feature selection techniques such as XGBoost and PCA. The results of classification accuracy after using XGBoost is 38.75%, and 34.25% when PCA is applied to Feature set-1 to reduce the number of features to 289.

We conducted several experiments using uncompressed and non reduced features extracted from AlexNet. The best results on the classification of algorithm selection on semantic segmentation are obtained using Feature set-3 (concatenation of the fourth and the fifth layers of AlexNet) with RBF kernel in SVM, which resulted in 43.6% of classification accuracy. Since, the goal accuracy to reach is at least 77.76%, we also performed feature extraction using different pretrained neural network. The choice has fallen to ResNet18, which forms out Feature set-4 that consists of 512 features. ResNet18 is one of the deep networks that has been used in recent semantic segmentation algorithms. The accuracy of the classification results of the experiment with Feature set-4 and SVM is around 34.6%, with the ROC curve illustrated on the Fig. 3.

During some analysis on the dataset, we observed that most instances of the test set were predicted to belong to algorithm A4 and algorithm A5. The main reason for such a classification result is high class imbalance, which can be observed on the histograms Fig. 4, Fig. 5, Fig. 6. The distribution of samples across the different classes both in train and test sets are the same. We have very big number of samples classified to A4 and A5; therefore, all the samples in the test set are predicted to A4 and A5.

28 In order to evaluate the impact of the class imbalance on the ASA, we evaluated different techniques for over sampling and synthetic data generation. First technique we used is called data augmentation. This approach ads copies of the instances from the minority class in order to approximate the counts of samples over all the algorithms. The histogram of the class distribution of the train set after the oversampling technique applied is illustrated on Fig. 7, which resulted in the accuracy of 39.1%

Since the last technique is simply making copies of the already existing instances, there is a better technique that generates synthetic samples of the under-represented class, which is called Synthetic Minority Class Oversampling Technique

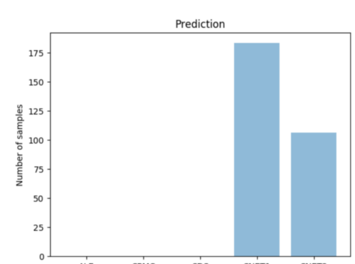


Fig. 6: Algorithm distribution histogram for predicted instances

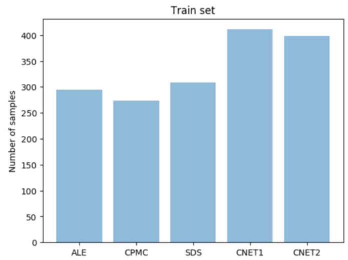


Fig. 7: Algorithm distribution histogram for train set after oversampling

(SMOTE) [30]. SMOTE makes class distribution histogram equal among all the classes. The result of the classification after applying SMOTE to Feature set-4 (ResNet18 features) is 20.41%. The accuracy from this experiment is lower than the previous one when we used pure ResNet18 features, since overall the number of images to the number of features is far from being equal.

Finally, an synthetic samples generation was implemented based on the Gaussian Mixture of Models (GMM). The GMM model is a technique to approximate arbitrary data distribution by fitting a set of Gaussian kernels onto the data. Using this approach we used the training set of samples to build the GMM model. Then the model was sampled for a total of 3000 samples per algorithm. That is the training set for the algorithm selector now contained 15000 data samples. The selector was then tested on the test samples from the original VOC2012 data set. The average accuracy using this method resulted in $ASA = 36\%$.

The experiments, that have been already described, were using the features obtained from the pretrained neural networks. In order to evalute the quality of features engineered features have also been included in the experimentation. The Feature set-5 is formed using the visual bag of words on SIFT descriptors, which are 128 dimensional vectors. Using K-Means clustering, the SIFT descriptors are grouped into K=50 clusters, which build the Feature set-5, and used to train ANN with two fully connected hidden layers of 100 units. Classification results on the test set is 37%.

B. Experiments with Algorithm Selectors

We also considered two stage SVM classification on the Feature set-3 consisting of the following steps:

- 1. First SVM model is trained using the concatenated features from the fourth and fifth convolutional layers of AlexNet. Afterwards, the train set is used to make the predictions and to obtain the confidence scores using the trained model. The confidence scores are used later as the features for the second classifier.
- 2. The design matrix is constructed using the confidence scores from the previous stage. This matrix is fed as an input to train SVM with the same parameters as in the previous stage.

TABLE II: Summary of experiments showing ASA as a function of different features combination.

Method	Accuracy
AlexNet (c4), SVM	28.02%
AlexNet (c5) + PCA, SVM	34.26%
AlexNet (c4 + c5), SVM	**43.6%**
AlexNet (c4 + c5) + SMOTE, SVM	39.45%
ResNet18, SVM	34.6%
ResNet18 + SMOTE, SVM	20.41%
AlexNet (c4 + c5), Oversampling, SVM	39.1%
SIFT, ANN	37%
SIFT, SVM	35.64%
AlexNet (c4 + c5), Two stage SVM	35.6%
All features, GMM, SVM	36%

- 3. Finally, prediction is made on the test set using the second SVM model, which resulted in 35.6% of accuracy.

The summary of the results of the different experiments conducted on algorithm selection on semantic segmentation is outlined in the Table II

C. Attributes and Semantic Labels

In order to determine the sensitivity of the dataset to higher level information another sets of experiments was implemented. For this additional information was generated for the used dataset. The additional information is uses the following components:

- Region attributes extracted from gray images
- Region attributes extracted from black and white images
- Semantic labels (context attributes)

The region attributes represents region properties based on gray intensity. in the case of black and white images, the thresholding used to transform input color image to black and white is the mean intensity of the combined RGB intensities. Additionally to determine whether the main tool for learning algorithm selection, SVM is the most appropriate gradient boosting was also compared with the SVM approach.

TABLE III: Summary of Experiments for determining the impact of Context Attributes and Regional Properties on ASA.

Configuration	SVM Prediction	Gradient Boosting
Alex Net (c4)	37.98%	37.98%
Alex Net (c4), RPG	39.1%	38.4%
Alex Net (c4), RPB	36.67%	38.06%
Alex Net (c4), RPG, Att	43.41%	**44.96**%
Alex Net (c4), RPB, Att	37.6%	35.27%
Alex Net (c4), Att	41.09%	40.31%
Att	37.6%	38.76%

V. Results andd Discussion

resen The initial experiments are described in Table II. As can be seen the accuracy using an SVM classifier is slightly lower than the one achieved with gradient boosting. But the maximal accuracy of 44.96% is far from an average random accuracy obtained by random selector resulting in $\approx 22\%$. Additionally the experiments demonstrated that features from AlexNet seems to be more effective than deeper features from

ResNet50. This is interesting because in general ResNet50 has higher accuracy rather than AlexNet.

Additionally, note that the average ASA is below 50% and thus additional sources of information are required to get more accurate ASA.

Concerning the ASA algorithm evaluation Table III, the most accurate algorithm selection approach is the SVM classifier. The reason is due to the fact, that the data have a relatively small amount of of samples and larger amount of features. And SVM is one of the approaches common.

VI. Conclusion

In this paper we introduced a dataset for the algorithm selection and we evaluated its hardness using a set of simple machine learning methods. We showed that while the dataset is quite difficult a high accuracy algorithm selection can improve the task of semantic segmentation by a up to 13%.

The future work of this approach is to study more in depth relative machine learning; instead of learning one-vs.-all for multiple label classification,. Different approaches using algorithm ranking or stack of classifiers will be explored.

References

[1] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using Convolutional Networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 648–656. [Online]. Available: http://ieeexplore.ieee.org/document/7298664/

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[3] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy, and X. Tang, "DeepID-Net: Deformable deep convolutional neural networks for object detection," *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 2403–2412, 2015.

[4] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, no. Figure 1, pp. 3367–3375, 2015.

[5] J. Yang, B. L. Price, S. Cohen, H. Lee, and M. Yang, "Object contour detection with a fully convolutional encoder-decoder network," *CoRR*, vol. abs/1603.04530, 2016. [Online]. Available: http://arxiv.org/abs/1603.04530

[6] N. Zhang, J. Donahue, R. B. Girshick, and T. Darrell, "Part-based r-cnns for fine-grained category detection," *CoRR*, vol. abs/1407.3867, 2014. [Online]. Available: http://arxiv.org/abs/1407.3867

[7] C. Liu, P. Kohli, and Y. Furukawa, "Layered Scene Decomposition via the Occlusion-CRF," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 165–173, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7780394/

[8] J. Theiler and L. Prasad, "Overlapping image segmentation for context-dependent anomaly detection," *Proc.SPIE*, vol. 8048, pp. 8048 – 8048 – 11, 2011. [Online]. Available: https://doi.org/10.1117/12.883326

[9] V. Jain, H. S. Seung, and S. C. Turaga, "Machines that learn to segment images: A crucial technology for connectomics," *Current Opinion in Neurobiology*, vol. 20, no. 5, pp. 653–666, 2010.

[10] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," *CoRR*, vol. abs/1505.04366, 2015. [Online]. Available: http://arxiv.org/abs/1505.04366

[11] M. Ravanbakhsh, H. Mousavi, M. Nabi, M. Rastegari, and C. S. Regazzoni, "Cnn-aware binary map for general semantic segmentation," *CoRR*, vol. abs/1609.09220, 2016. [Online]. Available: http://arxiv.org/abs/1609.09220

[12] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *2007 IEEE 11th International Conference on Computer Vision*, Oct 2007, pp. 1–8.

[13] F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 4, pp. 2175–2184, April 2015.

[14] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013. [Online]. Available: http://arxiv.org/abs/1312.5602

[15] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *CoRR*, vol. abs/1611.01578, 2016. [Online]. Available: http://arxiv.org/abs/1611.01578

[16] Z. Wang, N. de Freitas, and M. Lanctot, "Dueling network architectures for deep reinforcement learning," *CoRR*, vol. abs/1511.06581, 2015. [Online]. Available: http://arxiv.org/abs/1511.06581

[17] J. X. Wang, Z. Kurth-Nelson, D. Tirumala, H. Soyer, J. Z. Leibo, R. Munos, C. Blundell, D. Kumaran, and M. Botvinick, "Learning to reinforcement learn," *CoRR*, vol. abs/1611.05763, 2016. [Online]. Available: http://arxiv.org/abs/1611.05763

[18] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *CoRR*, vol. abs/1606.04671, 2016. [Online]. Available: http://arxiv.org/abs/1606.04671

[19] J. Rice, "The algorithm selection problem," *Advances in Computers*, vol. 15, p. 65118, 1976.

[20] K. Leyton-Brown, E. Nudelman, G. Andrew, J. Mcfadden, and Y. Shoham, "A portfolio approach to algorithm selection," in *In IJCAI-03*, 2003, pp. 1542–1543.

[21] L. Xu, F. Hutter, H. Hoos, and K. Leyton-Brown, "Satzilla: Portfolio-based algorithm selection for sat," *Journal of Artificial Intelligence Research*, no. 32, pp. 565–606, 2008.

[22] S. Ali and K. Smith, "On learning algorithm selection for classification," *Applied Soft Computing*, vol. 6, pp. 119–138, 2006.

[23] M. Lukac, K. Abdiyeva, A. Kim, and M. Kameyama, "Reasoning and algorithm selection augmented symbolic segmentation,," in *IEEE Technically Sponsored Intelligent Systems Conference*, 2017.

[24] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[25] L. Ladicky, C. Russell, P. Kohli, and P. Torr, "Graph cut based inference with co-occurrence statistics," in *Proceedings of the 11th European conference on Computer vision*, 2010, pp. 239–253.

[26] J. Carreira, F. Li, and C. Sminchisescu, "Object recognition by sequential figure-ground ranking," *International Journal of Computer Vision*, vol. 98, no. 3, pp. 243–262, 2012.

[27] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *European Conference on Computer Vision*, 2014, pp. 297–312.

[28] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *CoRR*, vol. abs/1412.7062, 2014. [Online]. Available: http://arxiv.org/abs/1412.7062

[29] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," *CoRR*, vol. abs/1412.0774, 2014. [Online]. Available: http://arxiv.org/abs/1412.0774

[30] N. Chawla, B. K.W., L. hall, and W. Kegelmayer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–257, 2002.