Journal of Ubiquitous Systems & Pervasive Networks
Volume 18, No. 1 (2023) pp. 09-14

Control of autonomous UAV using an onboard LSTM neural network

Bill Karakostas
Independent Researcher, Building 94700, PO Box 4336, Manchester, UK M61 0BW

Abstract
Towards realising autonomous UAVs, this paper investigates one of the fundamental autonomous flying research problems, i.e., the ability of a vehicle to control its flying behaviour autonomously, without reliance on external infrastructure like Instrument Landing Systems or GPS. In this paper we experiment with a physical UAV prototype with embedded intelligent control capabilities, utilising a Long Short Term Memory (LSTM) neural network, in order to learn lift-off control sequences using self-training. The initial results are promising and show potential for embedding LSTMs in the control systems of autonomous UAVs.
Keywords: *Autonomous UAV, Intelligent Control, LSTM neural networks*

1. Introduction

1.1 Autonomous systems

An autonomous system makes the best use of available resources to achieve the specified mission. [1]. Autonomous control systems are designed to perform well under significant uncertainties in the system and environment for extended periods of time, while at the same time be able to compensate for system failures without external intervention [2]. Autonomous control systems therefore, go beyond basic automation: Instead of performing specific tasks repeatedly and without variation, autonomous control systems adjust in real time to changing environments or inputs and optimize towards multiple goals. Intelligent control systems help equipment and machinery adapt in real time to changing inputs or environmental conditions, unlike more static and rigid control systems.
During the past decades, autonomous systems have attracted considerable attention due to their capability of performing various operations with minimal or without human supervision. Because of their salient features of high autonomy and mobility, autonomous systems are vital for numerous civilian and military applications in marine, ground, aeronautics, and aerospace.

1.2 Autonomous aerial vehicles

Autonomous drones are unmanned aerial vehicles (UAVs) that operate without the need for a human pilot (local or remote). These vehicles carry out tasks and make decisions autonomously, regarding taking off landing, and mission performance. Autonomous industrial drones are typically used for monitoring industrial and critical infrastructure sites conduct routine maintenance, oversee safe and secure operations, ensure business continuity after severe weather and other incidents, and maintain compliance.

Autonomous UAVs differ fundamentally from other vehicles that operate different degrees of automation. For instance, many UAVs are able to land automatically; however to do so they rely on external infrastructure like Instrument Landing System (ILS), GPS or visual landmarks. In contrast, autonomous UAVs can achieve that by relying only on on-board systems.
Autonomous UAVs range from small urban air vehicles to large commercial aircraft, and typically utilise computer vision, radar (LiDAR) and machine learning technologies in order to sense their surrounding environment and calculate how best to navigate within it.

1.3 Limitations of current control systems

Current commercial controller systems have several limitations. Traditional control approaches such as Model Predictive Controller (MPCs), Proportional Integral Derivatives (PIDs), and other Advanced Process Controllers (APCs), typically operate on a set of deterministic instructions and in predictable environments [3]. Such controllers must be programmatically retuned for different scenarios, conditions, environments or goals. Additionally, current commercial controllers are only capable of focusing on one optimization goal at a time–for example, maximizing throughput or minimizing energy usage.

1.4 Intelligent controllers

Autonomous control systems are designed to perform well under significant uncertainties in the system and environment for extended periods of time, and they must be able to compensate for system failures without external intervention.
However, with the ever-increasing mission complexity of autonomous systems, the existing control algorithms become inadequate for high control efficiency and enhanced control performance, especially under diverse and challenging environmental conditions, and requirements for cooperative operations with multiple autonomous systems. In addition, autonomous system controllers must comply with system

* Corresponding author. Tel.:
Fax: ; E-mail: bkarakostas@acm.org

DOI: 10.5383/JUSPN.18.01.002

safety, performance optimization, fault detection and human-machine interaction constraints and goals.

1.5 Application of neural networks to intelligent control

Different intelligent control of autonomous systems approaches has been based on technologies including neural networks, fuzzy logic systems, learning and adaptive control, model predictive control, and so on.
In the past few decades, neural networks have gained in popularity in control and in particular in optimal control [3]. We can distinguish two main approaches for training neural networks in the context of control: supervised and reinforcement learning (RL) [4].
In RL, the system follows a trial and error method by interacting with its environment. For instance, in [5], RL is used to derive a controller for the steering angle of an autonomous vehicle from LiDAR measurements.

1.5.1 Long Short Term Memory neural networks

LSTM [6] are deep neural network architectures that belong to the family of recurrent neural networks (RNN). In RNNs, and in contrast to feedforward networks architectures, the neuron connections form cyclical graphs that correspond to temporal paths. RNNs however, suffer from what is known as the problems of *vanishing gradient* (the network stops learning), and *exploding gradient* (the network never converges to the point of minimum cost) [6]. LSTM however, eliminate both the above problems and hence are more suitable for the processing of complex sequential data. LSTM layers consist of cells that store historical state information, and of gates that control the flow of information through these cells. There are three types of gates in LSTMs: *forget*, *update*, and *output*. The forget gate decides what information should be kept. Information from the previous hidden state and information from the current input is passed through a sigmoid function that outputs a number between 0 and 1, where value closer to 1 cause the retaining of the data, while, values closer to 0 cause the discarding of the data. The update gate chooses which new data will be stored in the cell. The gate passes the previous hidden state and current input into a sigmoid function that transforms the values in the range 0 to 1. 0 means not important, and 1 means important. First, a sigmoid layer chooses which values will be changed and then a tanh layer creates a vector of new candidate values that could be added to the state. Finally, the output gate computes the output of the LSTM cell as a combination of the cell state and the new data, to decide what the next hidden state should be. First, the previous hidden state and the current input are passed through a sigmoid function. Then the newly modified cell state is passed to the *tanh* function. The *tanh* output is multiplied with the sigmoid output to create the new hidden state. The new cell state and the new hidden is then carried over to the next time step. The hidden state is used for predictions.
Because of their inherent capabilities in processing long sequences of inter-dependent data, LSTM networks have found applications in robotics [7], traffic prediction [8], time series prediction such as stock and share prices [9] and others. As a type of recurrent neural network (RNN), the long short-term memory (LSTM) neural network has been proven to be very effective in solving the time series problems, and thus has been widely used in pedestrian trajectory prediction, intersection vehicle destination prediction, and highway vehicle trajectory prediction.

1.6 Main research challenges addressed in this paper

Ongoing research aims to increase the control efficiency, robustness, and self-adaptation of autonomous systems to unexpected internal and external environmental changes, to improve the operational intelligence of single or multiple autonomous systems; and to explore novel control strategies and mission planning.
Under the above research agenda, this paper researches the feasibility of implementing autonomous capability in UAVs by means of low cost, non-proprietary and off-the shelf hardware and software components, that support onboard machine learning capability and do not require external computing or networking infrastructure. Thus, we propose that an autonomous UAV must possess on board (machine) learning capabilities that help it to implement intelligent, self learning and adaptive control. At the same time, the machine learning mechanisms implemented must be efficient as the computing capacity available onboard is usually limited. Hardware limitations are an impediment to use, for example, high performance model predictive controllers (MPCs) whose optimization process is computationally expensive [3].
In this paper we describe the experimental results from training a physical UAV autonomous prototype that we have developed and which we have trained to carry out initially simple tasks such as lift-off, with assistance from an LSTM neural network. Thus, the learning strategy we employed is similar to that of Reinforcement Learning techniques (RL) of learning through trial and error. The autonomous UAV follows an initial control strategy for lift off which it repeats several times, in order to learn from successful lift off control sequences with the use of a LSTM neural network.

1.7 Structure of the paper

The paper is organised as follows. The next section presents the high level architecture of the autonomous UAV controller and then the more detailed hardware and software architectures, including that of the onboard LSTM neural network. Section 3 describes the experiments with the physical prototype of the autonomous UAV and discussed its findings. Section 4 discusses related work, the contributions of this research as well as plans for future research.

2. Architecture of the autonomous UAV controller

2.1 Control Model

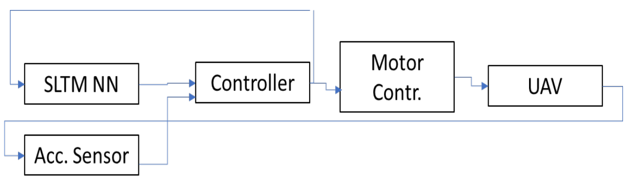

Fig. 1. Control Architecture.

The system architecture of the autonomous UAV is shown in Figure 1. The autonomous controller receives the state of the system from the acceleration sensor and then evaluates a set of constraints in order to decide whether to proceed with the next step of the control sequence or to abort. If the constraints are satisfied, the controller receives the next predicted control command from the LSTM neural network, using the current control sequence as input, and applies the control command to environments that no analytical optimal control model exists. The control architecture is therefore formally defined in terms of:

--A control sequence $\{u^n\}$, n=0,1,… and $u \in U$, the domain of control commands.
--A control prediction function L : $\{u^n\} \rightarrow \hat{u}_{n+1}$ that predicts the next step $\hat{u}_{n+1}$ in a control sequence $\{u^n\}$.
--A logical (Boolean) expression Λ that implements system constraints (safety, physical performance limitations, etc.).

The control algorithm shown below as pseudo-code, begins a take-off sequence by selecting at random an initial thrust (power level) and proceeds with building a control sequence (thrust adjustments) based on the thrust levels predicted by the LSTM network for the current take-off sequence. The take-off sequence continues until the vehicle has taken off, or a constraint condition is no longer satisfied (i.e. the vehicle has crashed, a time-out has occurred, etc.).

```
Set initial control parameter u0 to a value
from the permitted power level range
Let {u0} be the initial control sequence
n ← 0
While eval(Λ)==true:
{
 ûn+1 ←  L({un})
 {un+1} ← {un} + ûn+1
 n ← n + 1
}
```

3 Implementation and experiments

3.1 Physical Architecture

3.1.1 Hardware Architecture

A physical prototype has been constructed in order to validate the feasibility of the LSTM supported autonomous control model described in the previous section. More specifically, a Raspberry Pi 3B single board computer with a HAT (Hardware on Top) Motorshield DC Motor controller, has been used as the platform for implementing the autonomous controller. Table 1 shows the specification of the Raspberry Pi 3B single board computer. A micro quadcopter was connected to the motor controller. An acceleration sensor has been attached to the quadcopter and connected to the single board computer via I2C interface. Both the LSTM neural network and the autonomous controller software run on the single board computer. Therefore, the system is totally autonomous regarding its sensing and computing (predicting, controlling) functions.

Table 1. Technical specifications of the single board computer

SoC	Broadcom BCM2837
CPU	4× ARM Cortex-A53, 1.2GHz
GPU	Broadcom VideoCore IV
RAM	1GB LPDDR2 (900 MHz)

3.1.2 Software Architecture

The Raspberry PI Single Board computer runs the Raspbian GNU/Linux 10 ('buster') Linux version as well as Python 3.7.3. To implement the LSTM neural network we utilised Google TensorFlow 2.2 machine learning library and the KERAS high level ML Python library from which we imported the KERAS Sequential, LSTM and Dense modules.

3.2 Programming the autonomous controller

In this physical prototype, the autonomous controller learns how to make the UAV take off ('lift-off') with assistance from the SLTM neural network. More specifically, the control goal is to lift the UAV off the ground within a certain amount of time. The lift off condition is verified through the readings of the acceleration sensor, i.e. when the sensor records a +- 0.2g vertical acceleration, The +-0.2g tolerance is due to the need to accommodate for sensor errors on the z axis (vertical acceleration).
In general, autonomous vehicles need to be controlled within an envelope of physical constraints that are determined both by the design parameters of the UAV and by their operating status. For instance, the autonomous UAV will need to take off using only the necessary power to minimise use of its fuel /battery reserves.
Although a take off control sequence can be programmed into the controller, in practice each take off session may require adjustment to the control parameters due to environmental and capacity characteristics (e.g. battery level, winds affecting the vessel, the operating status of the motors, propellers etc.). Thus a precise control sequence cannot be modelled and has to be learned through trial and error.
We begin by programming the controller with a simple control strategy (such as from an initial power setting increase throttle gradually by 5% until lift off is achieved), and execute it over a number of successful and failed control sequences. Then we train the LSTM network using the successful control sequences, aiming to improve over the performance of the simple control strategy.
We set therefore an envelope of operating constraints regarding the lift off parameters as follows:

- Lift off must be completed in the shortest possible time
- At each step of lift-off power between 60 and 100% of the motors maximum power must be applied.
- Lift-off must be accomplished within 3 seconds or otherwise aborted.

We executed 353 lift-off sequences out of which 229 were successful and 74 unsuccessful (i.e. because they failed one of the above constraints).
We split the successful lift-off sequences into 10 datasets of lengths 2,4,6,8,10,12,14,16,18, and 20. The reason for doing so, is that we want to predict the correct motor power to apply at any given control step given the entire previous control sequence. The assumption is that the entire control sequence conveys more information about the state of vehicle and environment compared to that of the previous control action only. This is also the reason why an LSTM rather than a feed forward neural network was used.
Each of the 10 datasets was used to train a separate model in the LSTM neural network. Thus, a dataset of length n is used to train the neural network to predict the *nth* command in a control sequence. Each model was trained for 200 epochs. Total time to train all models was ~835 seconds

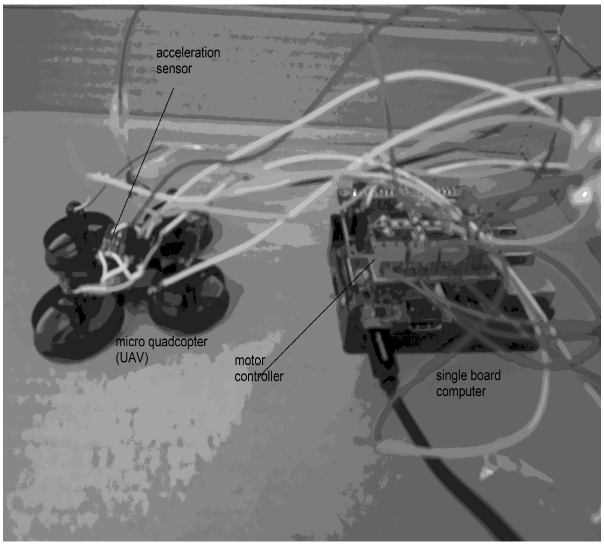

Fig. 2. Autonomous UAV prototype

The architecture of the LSTM neural network developed is shown in Figure 3.

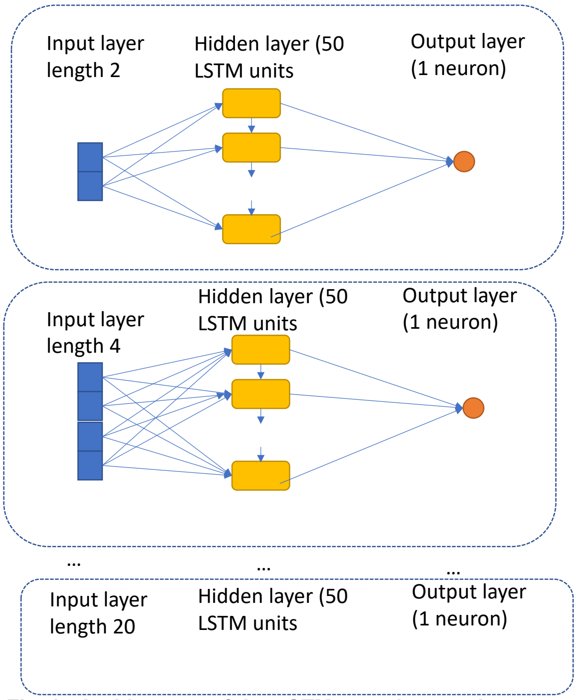

Fig. 3. Architecture of the LSTM neural network.

Table 2 shows examples of training data. The sequences of numbers indicate the power applied to the motors at each control step until lift off was obtained.

3.3.2 Neural network validation

After training of the neural network was completed, the system was tested with the execution of a number of lift off sequences, Each lift-off sequence begins with a random control command (i.e. throttle level), from the permitted range that was set to 60%-100% of total motor power. We compared the performance of the neural network assisted control sequences to that of the training datasets (which was created without assistance from the LSTM neural network), to decide whether the neural network assisted autonomous control improves the performance of the system in a number of metrics, as discussed in section 3.4.

3.3 LSTM configuration and training

3.3.1 Setting the Hyperparameters

According to the bibliography [10], the learning rate followed by the network size of the LSTM are its most crucial hyperparameters. Table 3 shows the key hyperparameters used in the training of the neural network.

Table 3 LSTM training parameters

LSTM units:	50
Input sequence length:	2,4,6,8,10,12,14,16,18,20
Epochs:	200
Activation type:	Rectified Linear Unit (ReLU)
Batch size:	50
Loss function:	Mean Squared Error (MSE)
Optimizer:	Adam

Table 2 examples of training data.

Control sequence length	**Control sequence (% of motor full power)**
6	96.0, 97.79388, 97.79388, 96.82182, 96.82182, 98.4
5	62.0, 61.426575, 61.426575, 62.692154, 62.692154
1	100.00

Table 2 shows sample successful (lift achieved) control sequences used as training data. For instance, in the first row of the table, lift off was achieved after 6 steps, while in the last row it was achieved after one step. Although it is intuitive to expect that lift off is easier to achieve on full throttle (100 % of motor power), as explained previously due to system and environment conditions this may not always possible, i.e. a full throttle control sequence will not always achieve lift off.

3.4 Analysis of the experiment results

3.4.1 Performance of the LSTM assisted UAV controller

Regarding the performance of the LSTM neural network, with the set of hyperparameters described in section 3.3.1, total training time (carried out entirely on the single board computer) for the 10 LSTM models was approximately 835 seconds. This indicates that while not prohibitively expensive, training must be carefully scheduled in an operating autonomous UAV so that it does not compete for resources with other critical mission tasks.

Table 4. Performance comparison.

UAV with LSTM	**#Successful lift-offs:** 156 (67.8% of total) **#Unsuccessful lift-offs:** 74 (32.2% of total) **Average length of control sequence**: 9.8
UAV without LSTM	**#Successful lift-offs:** 229 (64.8% of total) **#Unsuccessful lift-offs:** 124 (35.2% of total) **Average length of control sequence**: 9.05

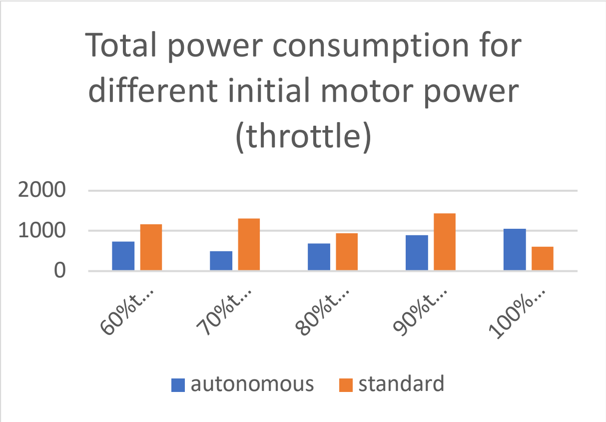

Fig. 4 UAV lift-off power usage.

An autonomous UAV will typically have limited energy resources (battery fuel,…) which it will need to manage efficiently. Figure 4 shows the comparative throttle/power usage (which equates to energy consumption) for the autonomous and 'standard'(without use of neural network) control methods. The values on the y axis show total energy consumption for the different control sequences. For instance, at 70% start throttle, the LSTM based controller uses approximately 3 times less energy for lift-off compared to the non-LSTM control.

From the diagram of Figure 4, therefore, it can be seen that with the exception of full throttle start, the neural network assisted sequences are more optimal regarding energy efficiency.

4. Conclusion

4.1 Related Work

Autonomous vehicles, whether land, air or sea based represents currently an intensive research area. Without reliance on active guidance by humans, autonomous vehicles depend on intelligent control algorithms. Due to the difficulty in navigating in unknown dynamic environments, many autonomous vehicle control algorithms employ neural network techniques for safe navigation, i.e. routing, collision avoidance and so on. For instance, the approach described in [11] predicts future trajectories of surrounding vehicles to ensure safe and reasonable interaction between intelligent vehicles and other types of vehicles. The approach utilises a long short-term memory (LSTM) neural network driven by driving knowledge. Similarly, in [12] the authors present a first step towards consistent trajectory prediction with the use of a LSTM neural network, that is capable of accurately predicting future longitudinal and lateral trajectories for vehicles on highway.

4.2 Main research contributions of this paper

The main research contributions of this paper are as follows:

- It contributes to the ongoing research in autonomous (aerial) vehicles.
- It proposes a novel way of autonomous/onboard learning of autonomous UAVs without reliance on external infrastructure such as Cloud systems. training is carried out on the single board computer rather than offloaded to a faster machine.
- It validates the proposed idea with a physical prototype (not a simulated one) of an autonomous UAV with onboard self-learning capabilities.
- It provides initial evidence that LSTM neural networks can be used for intelligent/self-learning flight controllers of autonomous UAVs.

4.3 Future research

Autonomous controllers can relieve humans from time consuming mundane tasks, thus increasing efficiency, with enhanced reliability (since they monitor the health of the system and possibly self-heal), and self-adaptive performance [13], while protecting the system from internal faults, and with consistent performance in accomplishing complex tasks. Autonomous UAVs for instance can survey large and inhospitable areas and by using onboard sensors, can detect dangerous incidents such as pollution, fires and so on. All these without any guidance from human operators [14].

Although autonomous systems are already used for specific missions such as space exploration, we argue that for their usage to be expanded, their capabilities need to be implemented in lower cost configurations, using off the shelf components, in order to reduce costs. Additionally, autonomous, software-based capabilities must be architected as frameworks of micro-services ensuring interoperability between the autonomous UAV subsystems [15]

In future research, we plan to train the autonomous UAV prototype to learn more advanced maneuvers such as, hovering over a specific ground point and so on. This will require more complex control routines as the rotors of the UAV will need to be controlled individually.

More complex control sequences, however are likely to require larger neural networks containing more input and output parameters ('features'). The performance of the onboard neural network must therefore be optimised [16] We will investigate, therefore, efficient training methods regarding the size of training data, the architecture of the LSTM network, batch sizes and other hyperparameters. In particular, to improve the model's accuracy, all model hyperparameters should be tuned carefully. We will also experiment with different batch sizes of training data which although do not affect the accuracy of the prediction can impact the training performance. It is important that the autonomous vehicle carefully balances the allocation of its computing resources between training and performance of other mission tasks. Neural network training could therefore be carried out periodically, and at times when there is spare computing capacity in the autonomous UAV.

References

[1] Figueroa F, Underwood L, Walker MG, Morris J. NASA Platform for Autonomous Systems (NPAS) Proceedings AIAA SciTech 2019, San Diego, CA.

[2] Antsaklis PJ, Passino KM, Wang SJ. An Introduction to Autonomous Control Systems. Proceedings of the 5th I EEE International Symposium on Intelligent Control, pp.21-26, Philadelphia, Pennsylvania, Sept. 5-7, 1990.

[3] Costa Vianna ML, Goubault E, Putot S. Neural Network Based Model Predictive Control for an Autonomous Vehicle. arXiv:2107.14573 [cs.RO]

[4] Lillicrap P, Hunt J, Pritzel A , Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Continuous control with deep reinforcement Learning. Proceedings ICLR 2016

[5] Ivanov R, Carpenter TJ, Weimer J, Alur R , Pappas GJ, Lee I. Case Study: Verifying the Safety of an Autonomous Racing Car with a Neural Network Controller, Proceedings HSCC, 2020.

[6] Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computing 1997; 9 (8): 1735–1780.

[7] Perez EAM, Hitoshi I. LSTM Neural Network-based Predictive Control for a Robotic Manipulator. Proceedings ISEEIE 2021: 2021 International Symposium on Electrical, Electronics and Information Engineering February 2021 Pages 128–135.

[8] Bogaerts T, Masegosa AD, Angarita-Zapata JS, Onieva, E. a Graph CNN-LSTM Neural Network for Short and Long-Term Traffic Forecasting Based on Trajectory Data. Transportation Research Part C Emerging Technologies 112 (March 2020):62-77.

[9] Pramod BS, Mallikarjuna Shastry PM. Stock Price Prediction Using LSTM. January 2021 Test Engineering and Management 83(May-June 2020):5246-5251

[10] Breuel, TM. Benchmarking of LSTM networks. arXiv preprint arXiv:1508.02774 (2015). arXiv:1508.02774. Bibcode:2015arXiv150802774B.

[11] Wang S, Zhao P, Yu B, Huang W, Liang H. Vehicle Trajectory Prediction by Knowledge-Driven LSTM Network in Urban Environments. Journal of Advanced Transportation, vol. 2020.

[12] Altché F, de La Fortelle A. "An LSTM network for highway trajectory prediction," 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017, pp. 353-359.

[13] Ignatious HA, Hesham-El-Sayed, Khan M, Kharbash S, Malik S. Understanding and Comparing Approaches for Performance Engineering of Self-adaptive Systems Based on Queuing Networks. JUSPN, volume-14, Issue 2 (2021), PP 27 – 35.

[14] Percepto Air https://percepto.co/

[15] Arcelli D. A Novel Micro-services Cluster Based Framework for Autonomous Vehicles JUSPN, volume-16 , Issue 2 (2022), PP 77 – 86.

[16] Steuler JL, Beck M, Passow BN, Guckert M. Optimizing the Energy Consumption of Neural Networks. IEEE Symposium Series on Computational Intelligence (SSCI), December 2020.