

A Small REVOLUTION in PYTHON

PhD. Ciprian DRAGNE¹

¹Mechanics and Acoustics Dept, Institute of Solid Mechanics – Romanian Academy, Bucharest, România.

Abstract—Users of PYTHON programs are now facing a new problem - too many versions of applications, often incompatible with each other, and the inability to install each version.

Almost all programmers and users today who write computer programs use the PYTHON language, not only because most libraries are free, but also because of the simplicity of code construction and the multitude of libraries available, and especially for their simplicity in installation.

One of the great advantages brought by the PYTHON software and the growing demand and need for applications is the multitude of users who have become involved in the development of libraries that facilitate the development of subsequent applications and the provision of these new libraries to users.

But this multitude of applications also has a multitude of versions, both for the basic PYTHON engine and for the libraries used.

Currently, it is not possible, at least on Windows computer system, to have different versions of libraries installed on the same version of PYTHON at the same time. This is possible, but only by uninstalling the old ones and downgrading to the versions required by those specific libraries used. However, this means sabotaging other applications that needed the previous libraries with the versions they were developed for.

I wrote this article to draw the attention of developers that it's time for a small Revolution in PYTHON, namely, adding to *.py scripts when importing libraries, also the version required for running. I will present some examples below...

Keywords— PYTHON, scripting, programming .

I. INTRODUCTION

What is PYTHON? It is a programming language, very popular today, widely used in web applications, software development, data science, and even in modern trends in software development - artificial intelligence (AI) and machine learning (ML).

PYTHON is a high-level, interpreted, general-purpose programming language known for its simple and readable syntax that emphasizes code clarity, but is also easy to install and update. It supports multiple programming paradigms, including structured and object-oriented programming, making it extremely versatile for web development, data science, artificial intelligence, and automation. The simplicity of the Python programming language and the free use of many libraries have attracted many users and programmers specialized in application development [1].

Today, there are several programming languages, but none have as many users and followers as those who use the Python software.

Python was created in 1989 by Dutch programmer Guido van Rossum (Figure 1), who said the application was more of a Christmas present, as a hobby project [2],[3][4].

Developers use Python because it is efficient and easy to learn and can run on many different platforms.

Python software is free to download, integrates well with all types of systems, and allow increases speed in developing user routines and new advanced concurrent applications.

Python it is used for:

- web development (server-side);
- software development;
- mathematics and for any other science;
- easy to use for image and video processing;
- advanced manipulation of database structures with minimal programming code;
- easy integration with many other software;

- system scripting.



Fig 1. The Creator of Python – Guido Van Rossum.

Guido van Rossum held the title of "Benevolent Dictator for Life (BDFL)" of Python, meaning he had the final say in disputes within the community until he stepped down from that role in 2018 [5].

II. LITERATURE REVIEW

The software PYTHON was released in 1991 as a version (0.9.0). Guido Van Rossum is still today a leader in the community of software developers working to improve the Python language and search for new ideas.

The name "Python" is a tribute to the British comedy troupe Monty Python and is meant to be as simple and fun as that comedy troupe.

Advantages in using PYTHON:

- Python is free to download and use;
- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc);
- Python has a simple syntax similar to the English language;
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages even in mathematical data transformations;

- in Python is easy to develop web application on a server;
- Easy to creating processes that may be utilised with software;
- Reaching out to database systems;
- Examining and changing files;
- Analysing huge data;
- Speedy prototyping;
- Producing software that is ready for use because runs as an interpreter system;
- It is more difficult to be corrupted by viruses, because its interpretation methods and text scripting do not allow the transmission of computer viruses from one user to others;
- Like other programming languages, Python also has the ability to be managed in a functional, procedural, or object-oriented manner [6][7].

Python has gained widespread use in the AI and machine learning user community [8].

Python is also widely taught as an introductory and educational programming language [9].

Since 2003, Python has consistently ranked in the top ten most popular programming languages in the TIOBE Programming Community Index, which ranks searches across 24 platforms, and in recent years has consistently ranked first [10] (see Figure 2).

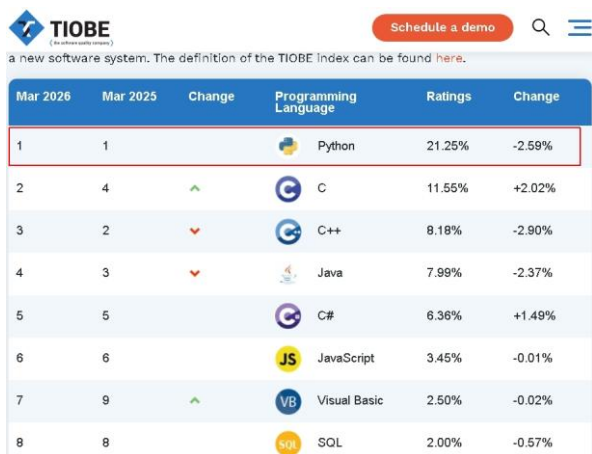


Fig 2. TIOBE index at March 2026.

Python has also few specific tools to build GUI (Graphic User Interface) like Tk, PyQt5 or PyQt6 as most used. These libraries have most tools that can help to develop a complete GUI similar to any standalone applications.

Integration of Python code in web applications can also be studied in some specific web applications with advanced interactivity tools, parametric models, multi-language interface and graphic results frame (Figure 3) [11][12].

Use the next weblink to view this web application:

<https://educatia-la-superlativ.blogspot.com/2026/02/mate-lectia-003-geometrie-demonstratii.html>

There is a lot of documentation on using the Python programming language, most of it directly on the internet, with suggestive examples for anyone who wants to get started.

But, even for advanced users, there are discussion forums about the most pressing errors, problems, ideas for development, or latest libraries for PYTHON [13][14][15].

The official PYTHON webpage announces that as of March 23, 2026, Python 3.15's is now back on track.

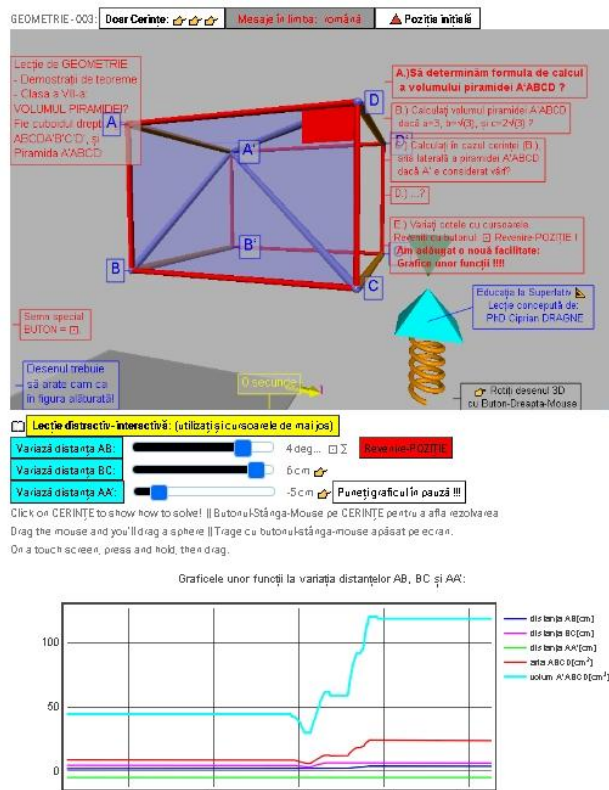


Fig 3. GUI of an example for a math problem with statement and solutions with interactive features.

III. METHODOLOGY

This paper shows some research by the author to develop educational tools and programming codes using PYTHON software with advantages and disadvantages explained at each step.

Often, it is necessary to downgrade the versions of certain libraries in order for a program to continue to run, with the risk that other scripts will no longer run, and which in turn require the upgraded versions of required libraries.

Allowing different site-packages versions to co-exists is impossible with the operating systems we have today. Exist some virtualenv file definition, but sometimes is also confuzing when have unsolvable dependencies.

The amount of problems encountered because of version conflicts is enormous now. If some package A imports B and C libraries, and A import a different version of C, then the version of C is not consistent with both A and B, even could be more propriate to run the initial script.

However, multiple versions can also cause some unintuitive behavior when it comes to which version will take priority when a library can work with both versions of a dependency.

Today, by allowing multiple versions to co-exist at runtime, that means that you can pass in objects returned internally are going to cause endless amount of problems, an exponential explosion of testing matrix, and it'll become

much, much harder to deprecate anything because new versions of the library would continue to be expected to be able to handle objects created from older versions indefinitely.

Such problems I observed to numpy libraries and other object-oriented libraries. The main downgrade is going to be a major headache to automate some procedure to fix the problems.

The best solution it could be to make the old version compatible with the newest version, but this means to study by yourself all the necessary codes and make changes.

Another way is to do-it like in the next sentence:

1. Install the newer library version via pip;
2. Download the prebuild whl for the other older version from pypi and unpack it. Rename the directory library to library_old. Put this directory into your site-packages directory of your venv/Python installation;
3. Code, which should use the older version, must import library_old.

Recommendations here is to make a small revolution in Python syntax by import modules in the way presented in the next sentence, directly in Python script:

```
import mylib==1.7.0 (1)
```

or even by specifying the PYTHON version from where to import:

```
import mylib==1.7.0 from PYTHON==3.9.1 (2)
```

The ideas used here was to create a new IDE (Integrated development environment) software for running PYTHON scripts by interpreting the codes in the ways presented in previous paragraph. This IDE can be developed also in PYTHON using capabilities for GUI and file processing. This new IDE application, is splitting the Python code, create temporary files and run the last scripts using genuine PYTHON interpreter.

Advantages of using this IDE:

- Automatically creates specific scripts for specific version of PYTHON;
- Possible to use older versions of existing scripts;
- Any classic line indentation can be used, as the IDE automatically converts the file to the correct code and the same for entire script. The application also allows the use of non-officially implemented line code indentation methods, defined internally by the local user where commands are delimited by any character sequence. This method also has advantages in having more secure code;
- The IDE application also has the possibility to encrypt Python code or a sequence of code, for creating more secure scripts;
- Easier to use various scripts for different Python version in the same application;
- Automatically copy scripts or sequences of code from one script to another; Management of scripts based on various parameters: version, goal, focus, main, results, speed, resources needed, type, etc;
- An interface that gathers a multitude of Python code sources into an interface where they can be much easier to exploit them;

- This IDE have access to Google using browsers for searching new ideas and put questions;
- Step-by-step script execution and preview of results – this implementation is very rare for most IDE applications for Python;
- Save in local files the results of the Python execution for future study;
- Multiple sources execution and running external applications – this IDE can use MATLAB codes, other external C++ and compiled sources to have more interconnectivity with other applications;
- VBA interconnectivity with SOLIDWORKS software for advanced CAD design – automatically create parts and assemblies, running Solidworks solvers and extract results, deploy AI trainings, make design optimizations, and procedures for robots [16];
- Automatically create Load-cases based engineering specifications for various needed solvers – SOLIDWORKS; ABAQUS, CalculiX, etc;
- Accessing databases with materials definition – convert databases in various formats;
- Automatically creates image and video captures from animatronic results;
- Automatically create results with advanced 3D visualisation methods and save them in the necessary formats, even in HTML;
- Management of projects based on various parameters: creation date, last modifying date, goal, various keywords, type, results, etc;
- Open user folders using the same GUI;
- Running external applications with various arguments and necessary environments files and settings;
- Accessing operating system applications – Regedit, MS-Config, App.Data folders, Notepad, Calculator, Paint, etc;
- Direct ARDUINO connectivity [17];
- Create parts for 3D printing – preview and define setting;
- Multiple sequential and parallel processing;
- Resources management and estimations;
- These IDE has its own work frame to directly run Python codes and edit scripts with multiple tools for automatically creating subsequent versions, and running scripts in various sentences;
- This IDE has the possibility to test parametric code design for managing and validating immutable parameters - objects whose value is unchangeable once they are created [18];
- Advanced parametrization and recommendations for code design in producing clean, Pythonic code—namely, using templates, good decorator, bad consequences, object-oriented programming, etc;
- And last but not least, code design proficiency techniques-based sustainability requirements for long-time processing codes and energy consumption of resources, memory/resource usage, time taken, and bottlenecks [19] [20]. When performing this analysis is called “Profiling” [21][22][23].

Disadvantages of using this IDE:

- Sometimes it is necessary to correct the format of variables when you want to continue the process through multiple scripts. This may also mean creating new subroutines for data conversion;
- The previous drawback is not always easy to fix when the formats differ greatly following subsequent massive developments for new versions of Python or other libraries;
- Sometimes it is necessary to temporarily save the results in their original format, load the results into new scripts, and restore the data in the final required format;
- The entire process could be slow for a huge amount of data.

A typical example of errors in Python because of various versions of libraries needed and with explanations for solution found, can be viewed in Figure 4.

A specific example with errors related to the current installed version of PyQt6 inside the Python 3.12

No QtMultimedia backends found. Only QMediaDevices, QAudioDevice, QSoundEffect, QAudioSink, and QAudioSource are available. Failed to initialize QMediaPlayer "Not available"
Failed to create QVideoSink "Not available"

Solution found after many test:

1. Not only Downgraded the PyQt6 at lower version
2. But also various subsequent packages at specific versions:

```

pynput==1.7.7
PyQt6==6.6.0
PyQt6-Qt6==6.6.0
PyQt6-sip==13.6.0
PyQt6-WebEngine==6.6.0
PyQt6-WebEngine-Qt6==6.6.0
PyQt6-WebEngineSubwheel-Qt6==6.7.1
pyserial==3.5
requests==2.32.3
                
```

Fig 4. Errors example because different versions needed.

IV. CASE STUDIES

In this chapter will be presented some practical examples implemented by the author.

The best method to understand is to test by yourself the methodology explained herein.

Each example has some explanations about programing level involved and recommendations for better understanding.

In general, all these applications are recommended for anyone who has a low level of programming in Python, but is needed a clear developing of scripts.

A. Application for a new interpretation of Python code

This example (fig. 5) is a good point to start in developing more advanced Python applications. The application was first released in 2014 using the Tk libraries, and later, during development, the PyQt5 and PyQt6 libraries in Python were used. One of the main advantages of using PyQt6 is that we can create textbox with various text colors and fonts, depend on the type and role of the code.

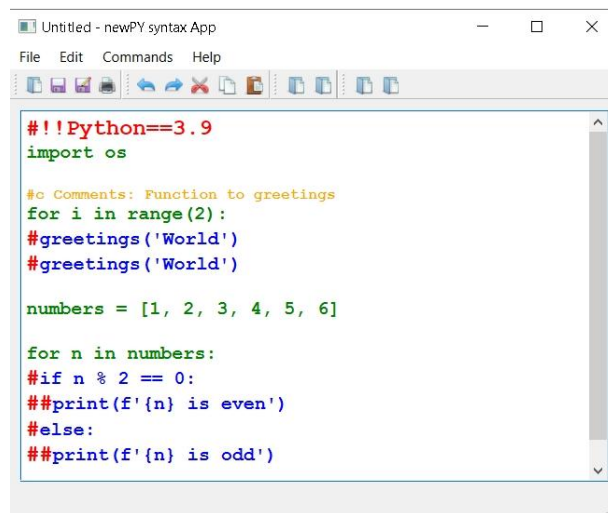


Fig 5. Application for interpreting the new Python codes.

The application has possibility to interpret the revised Python codes with lines defined like in Eq (1) and Eq (2), splitting the code in temporary genuine Python and run resulted files. The application is developed under Python 3.12. Installation of various libraries in various versions should follow the rules presented in this paper.

B. Codes with various new syntaxes

Figure 6 shown a simple code sequence that contain a new type of syntax in Python.

```

#!Python==3.9
import os

#c Comments: Function to greetings
def greetings(name):
    #print("Hello, " + name + "!")

#c Comments: Loop 2 times
for i in range(2):
    #greetings("World")

numbers = [1, 2, 3, 4, 5, 6]

for n in numbers:
    #if n % 2 == 0:
    ##print(f"{n} is even")
    #else:
    ##print(f"{n} is odd")
                
```

Specify the Python version to use

Comments with symbol#c

Unmodified official syntax

Modified official syntax with # as indentation symbol

Modified official syntax with ## as double indentation

Fig 6. Type 01 for new Python syntax.

This new Python syntax has advantages that using the character “#” for indentation in lines, which is a comment marker in official Python syntax, has better arrangements in code structure, and reduces the number of characters in code.

A different Python code defined in type-01 syntax is shown in Figure 7. This example is added to observe code when multiple indentation is needed.

VBA under SOLIDWORKS. I recommend here to read next papers [25].

Figure 12 shown an application under SOLIDWORKS that allow to control a robot and perform many kinematics and dynamics studies.

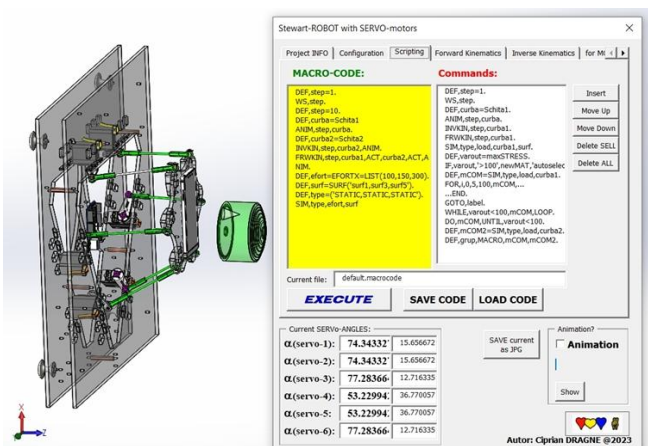


Fig 12. VBA application under SOLIDWORKS to control a robot.

Using such scripts was created a connectivity between Python and SOLIDWORKS and the methodology presented here is for evaluating a parallel hexapod robot in:

1. Inverse and forward kinematics studies;
2. Verifying positioning at specific locations (robot starting position, minimum or maximum actuator positions, etc.);
3. Moving the robot in specific directions in a global or local coordinate system using generalized coordinates;
4. Positioning at a specific location in 3D space;
5. Performing static and dynamic simulations and extract results;
6. And all other facilities developed for a CAX application that perform advanced studies in SOLIDWORKS [26].

Similar connectivity's was developed also with next programs: MATLAB, ABAQUS, ARDUINO, etc.

F. Specialized programming tools

I want to mention herein also some high specialized programming tools for academic and engineering research, related to:

- Professional specifications for methodology and experimental testing;
- Define specialized cycling for experimental testing;
- Global and local variables definition;
- Defining specific user functions;
- Stop conditions;
- Management and panel for connectivity with other projects.

V. CONCLUSION

I hope that now, the purpose, the results obtained so far, and the performance of this study are clear even to a novice programmer. I summarized here all the problems that were observed during the time of using PYTHON applications and tried to develop some scripts that help me in my academic research.

A unique paper is not sufficient to explain all the subroutines developed here, but I hope to give some ideas about what can be done using software, some brilliant ideas and a lot of time for code testing.

These types of applications have helped me a lot in developing software and using old databases, incompatible with current versions of Python, but not only for Python.

This IDE also has AI tools built in. With a dedicated code database, I use this app to deploy Python code in sections, store and perform machine learning and AI training, and use models to detect the best-performing code. The main advantages of this framework are that allow me to start developing applications using older applications that can be detected very fast from a specific database of codes. These tools will be presented in detail in a future paper.

For me, the indentation proposed in this paper creates simpler Python code. Now, it depends on each person how much they use this programming language and how complicated are the applications they develop [24].

However, the disadvantages explained in this paper make me believe that the future of operating systems for computers is versions without installed applications, in which most applications will be completely generated by AI.

This trend in operating system development will be more visible when quantum computers are used massively, which will have a different structure for internal codes, for users' programming codes, for files, and any kind of data stored, and even a new type of necessary memories.

Of course, we can turn to AI for help in updating scripts. I've done these few times before, but AI is still in its infancy. Until you find code that actually works, you have to run a lot of examples and debug it yourself, because a lot of the code received through AI is not completely directly functional in your computer.

Unfortunately, today, a trend of reducing programmer positions has already been reported for most IT companies, which are trying to use artificial intelligence for almost all computer-related tasks. We shouldn't rush here. Even more tasks will have to be solved in the coming years, precisely through the introduction of quantum computers.

Many programmers have invested a lot to reach a certain level. The use of robots more frequently and everywhere will require more and more specialist involvement.

In the future, the illiterate will be the one who doesn't know how to use a computer in more advanced ways than just for fun.

I did used programming language in Python not only for small user applications, but also for connectivity between other software, or as scripting programs for engineering professional applications like ANSA-Beta-CAE together with various FEA solvers to automatically correct FEA databases, create input for solvers and to extract results data [25][26].

Ansys and Abaqus professional software's has the possibility to develop internal tools in Python for the management of databases and solver results [27].

Many other software applications use Python for 3D modeling (Blender, Maya), for graphic performances and image processing (Gimp, Inkscape), or for geographical information system (ArcGIS).

FreeCAD is also another software that uses Python as scripting.

This paper was developed using Python applications [1], JAVA, CSS, HTML, VBA in MS Office Suite, and other author codes and resources [28]. The applications were developed on a computer with Windows 10 64-bit architecture. In other operating systems (Linux, Raspberry Pi), Python could be more versatile in developing such tools.

Future research:

1. Study for more integration of PYTHON in web applications;
2. Study for new ways to present data;
3. Improved GUI for visualization and results;
4. Interconnectivity with other applications.

ACKNOWLEDGMENT

This paper was developed by PhD Ciprian Dragne, București, România, Research Engineer level 3 from Institute of Solid Mechanics – Romanian Academy.

DECLARATION OF CONFLICTING INTERESTS

The author declares that has no conflicts of interest.

REFERENCES

- [1] PYTHON website, <https://www.python.org/>, [Accessed Mar. 10, 2026].
- [2] van Rossum, G. (2007). Language Design Is Not Just Solving Puzzles. Artima forums.
- [3] Wikipedia webpage: https://en.wikipedia.org/wiki/Guido_van_Rossum, [Accessed Mar. 19, 2026].
- [4] Sebesta, R. W. (2012). Concepts of Programming Languages - 10th Edition. Pearson Addison Wesley.
- [5] "Benevolent dictator for life". Linux Format. 1 February 2005. Archived from the original on 1 October 2006. Retrieved 1 November 2007.
- [6] Amazon website, <https://aws.amazon.com/what-is/python/>, [Accessed Mar. 15, 2026].
- [7] Nosrati, M. (2011). Python: An Appropriate Language for Real World Programming. World Applied Programming, 1(2), 110-117.
- [8] Healy, J., McInnes, L., & Weir, C. (2017). Bridging the cyber-analysis gap: the democratization of data science. The Cyber Defense Review, 2(1), 109-118.
- [9] Sultana, S. G., & Reed, P. A. (2017). Curriculum for an introductory computer science course: Identifying recommendations from academia and industry. Journal of Technology Studies, 43(2), 80-92.
- [10] <https://www.tiobe.com/tiobe-index/>, [Accessed Mar. 24, 2026].
- [11] DRAGNE C., Educational Tools with WWW-Interactivity, <https://ijses.com/wp-content/uploads/2026/02/120-IJSES-V10N1.pdf>.
- [12] Author web applications, <https://educatia-la-superlativ.blogspot.com/>, [Accessed Jan. 29, 2026].
- [13] "4. Built-in Types – Python 3.6.3rc1 documentation". python.org. Archived from the original on 14 June 2020. Retrieved 1 October 2017.
- [14] Zadka, M., & van Rossum, G. (2001). PEP 237–Unifying Long Integers and Integers. Python Enhancement Proposals.
- [15] Beazley, D. M. (2009). Python essential reference. Addison-Wesley Professional.
- [16] Dragne, C., "Advanced Concurrent Engineering using CAD software - Classic, Avant-garde and AI". *International Journal of Advanced Multidisciplinary Research and Studies*, 2024. DOI: <https://doi.org/10.62225/2583049X.2024.4.4.3038>.
- [17] MicroPython website, <https://micropython.org/>, [Accessed Mar. 24, 2026].
- [18] Kao, C. L. (2025). The Tuple Object and Its Immutability. In CPython: A Complete Guide to CPython's Architecture and Performance (pp. 169-180). Berkeley, CA: Apress.
- [19] Szustak, L., Lawenda, M., Arming, S., Bankhamer, G., Schweimer, C., & Elsässer, R. (2023). Profiling and optimization of Python-based social sciences applications on HPC systems by means of task and data parallelism. *Future Generation Computer Systems*, 148, 623-635.
- [20] Arora, I. (2024). Improving performance of data science applications in python. *Indian Journal of Science and Technology*, 17(24), 2499-2507.
- [21] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., & Glance, N. (2007, August). Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 420-429).
- [22] Alnowibet, K. A., Khan, I., Sallam, K. M., & Mohamed, A. W. (2022). An efficient algorithm for data parallelism based on stochastic optimization. *Alexandria Engineering Journal*, 61(12), 12005-12017.
- [23] Szustak, L., Wyrzykowski, R., Kuczynski, L., & Olas, T. (2021). Architectural adaptation and performance-energy optimization for CFD application on AMD EPYC Rome. *IEEE Transactions on Parallel and Distributed Systems*, 32(12), 2852-2866.
- [24] Abdulsalam, S., Lakomski, D., Gu, Q., Jin, T., & Zong, Z. (2014, November). Program energy efficiency: The impact of language, compiler and implementation choices. In *International Green Computing Conference* (pp. 1-6). IEEE.
- [25] Dragne, C., "Super Digital-Twins for Academic Research", *International Journal of Scientific Advances*, ISSN: 2708-7972, 2025, Volume: XX | Issue: XX | Available Online: www.ijscia.com.
- [26] Dragne, C., "CAX Software—The Next Level in Computer Aided Technology". *Int. J. Adv. Eng. Manag. —IJAEM*, 5, 2023, pp.551-561, *International Journal of Advances in Engineering and Management (IJAEM)*.
- [27] ABAQUS Professional software website, <https://www.3ds.com/products/simulia/abaqus/cae>, [Accessed March 25, 2026].
- [28] GRABCAD author codes, <https://grabcad.com/phd.ciprian.dragne-1>, [Accessed Nov. 29, 2025].