

A Modular Virtual Environment for Teaching Advanced Topics in Measurement Science

Zivko Kokolanski, Mare Srbinovska, Kiril Demerdziev and Vladimir Dimchev

Abstract – This paper demonstrates the use of a modular virtual environment for advanced teaching in measurement science through the exploitation of virtual instruments. The proposed setup enables the construction of a measurement uncertainty budget within a simulated calibration procedure, real-time data exchange with the ThingSpeak IoT platform, and the implementation of basic machine learning algorithms for data analysis. The virtual instruments operate over synchronized channels, allowing hybrid experimentation with both simulated and real signals. The results confirm that the developed environment provides consistent and validated outcomes when compared with realistic experiments, proving its effectiveness as a valuable tool for university-level education and training in modern metrology.

I. INTRODUCTION

This paper presents the design and implementation of a modular virtual environment intended for teaching and training in advanced measurement science topics, with particular emphasis on measurement uncertainty, telemetry-based data acquisition, and the application of machine learning for metrological tasks. Virtual learning environments have been widely adopted in science and engineering education to improve accessibility, flexibility, and conceptual understanding [1-5], while virtual laboratories have proven especially effective in supporting experimental learning where physical resources are limited or remote access is required [3], [7], [10]. The proposed environment is organised around core virtual instruments (VIs) that emulate typical laboratory equipment. These VIs are interconnected through software-defined (virtual) communication channels that support configurable topologies, bidirectional data exchange, and time-synchronised operation. A dedicated backend layer manages data flow, time stamping, and control signals, ensuring coherent operation of multiple instruments and realistic emulation of distributed measurement systems, in line with modern virtual and hybrid laboratory architectures [8], [9].

Unlike static simulation tools, the proposed virtual environment enables hybrid operation: in addition to synthetic or prerecorded data, it can be coupled with external hardware modules or remote testbeds to acquire real signals. This capability aligns with prior work demonstrating the pedagogical value of interactive and hardware-integrated virtual laboratories in engineering education [7], [10].

Zivko Kokolanski, Mare Srbinovska, Kiril Demerdziev, Vladimir Dimchev, Ss. Cyril and Methodius University in Skopje, Faculty of Electrical Engineering and Information Technologies, Rugjer Boskovikj 18, Skopje, E-mail: kokolanski@feit.ukim.edu.mk.

The hybrid approach extends applicability from concept demonstration to practical validation, allowing students and researchers to experiment with real-world disturbances, latency effects in telemetry, sensor imperfections, and varying measurement conditions. In such a setup, the virtual instruments act as user-facing front ends, while the backend provides secure data transmission, buffering, and protocol abstraction.

Pedagogically, the environment is conceived to support step-wise learning of advanced concepts that are often difficult to demonstrate in a conventional classroom—such as propagation of measurement uncertainty, or the effect of data quality on AI/ML-based estimation. Previous studies highlight that guided interaction, scenario-based experimentation, and adaptive task complexity significantly enhance learning outcomes in virtual environments [4], [6]. By parametrising the scenarios, instructors can tailor exercises to different student profiles and gradually move toward a personalised education approach. The same framework can be reused for remote and blended learning, as all components are software-accessible and can be deployed across institutional networks or cloud infrastructures [1], [5], [8].

The contribution of this work is twofold: (1) a technical architecture for a flexible, instrument-centric virtual measurement laboratory that integrates simulation, real data acquisition, and telemetry concepts, and (2) an educational framework that leverages this architecture to support the teaching of contemporary topics in measurement science and technology, extending and complementing existing virtual laboratory approaches reported in the literature [2], [3], [9], [10].

II. VIRTUAL FRAMEWORK LABVIEW ARCHITECTURE

The developed LabVIEW-based virtual framework is organised around a unified front panel given in Fig.1 that serves as the primary user interaction layer for configuring, operating, and monitoring the virtual measurement environment. The front panel is intentionally structured into two logically separated and functionally complementary sections: Packets and Variables, enabling a clear distinction between instrument-level functionality and data-level interaction. The Packets tab represents the instrument-centric layer of the framework. Within this tab, virtual instruments are grouped and accessed as modular packets, each encapsulating a specific measurement functionality.



Fig. 1. Front panel of the LabVIEW-based virtual framework, showing the instrument-centric Packets tab for launching specialised virtual instruments, and the Variables tab for visualising data exchanged through software-defined virtual channels.

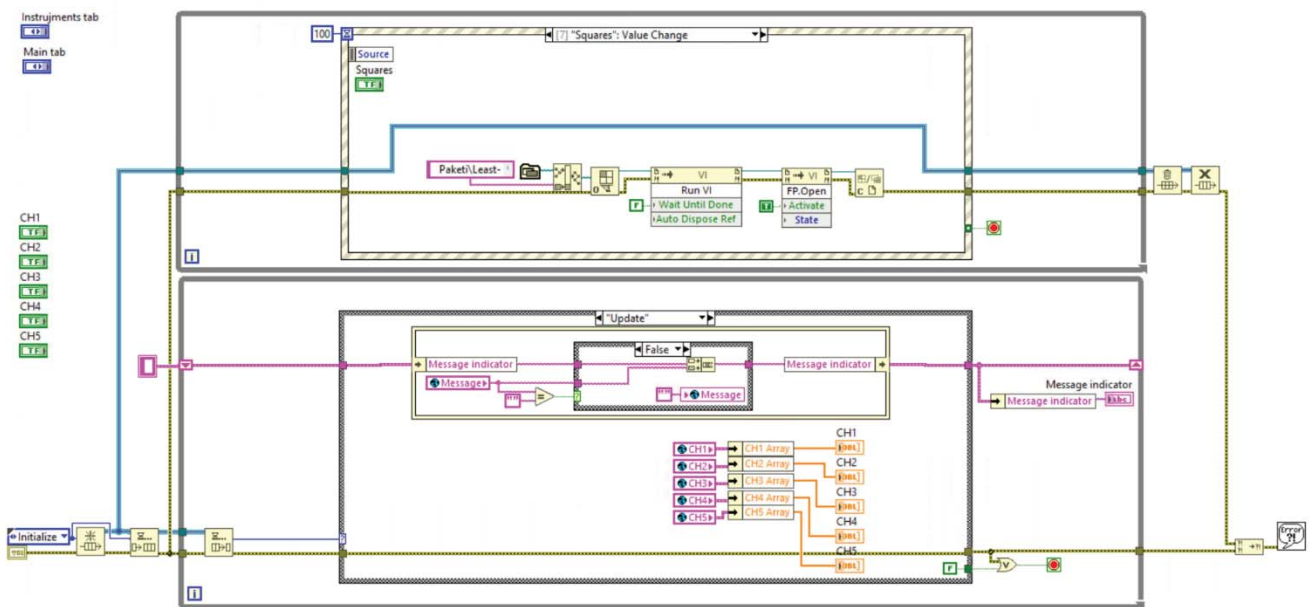


Fig. 2. Block diagram of the virtual framework based on a producer-consumer architecture for event-driven instrument control and data exchange through virtual channels

Each packet is represented by an intuitive graphical icon, allowing users to activate, configure, or interconnect instruments in a manner analogous to selecting and wiring physical laboratory equipment. This modular organisation facilitates extensibility of the framework, as new instruments or application-specific modules can be added without altering the core architecture.

The Variables tab provides a data-centric view of the virtual framework and acts as the shared communication

space between the individual virtual instruments. In this section, the user can visualise, control, and store data associated with five predefined virtual channels. These channels function as software-defined variables that enable mutual exchange of information between different virtual instruments and packets. Each channel can be independently configured in terms of data type, update rate, and storage behaviour, supporting both real-time monitoring and offline analysis. The variables can

represent measured signals, intermediate processing results, control parameters, or metadata, thereby enabling complex interaction scenarios such as feedback loops, data fusion, or distributed measurement workflows.

Together, the Packets and Variables tabs form a layered architecture in which virtual instruments operate as functional entities, while shared variables provide a flexible and transparent communication mechanism. This separation enhances usability and pedagogical clarity, allowing learners to focus either on instrument behaviour or on data flow, depending on the educational objective. At the same time, the architecture supports advanced experimentation scenarios, including multi-instrument coordination, telemetry-based data exchange, and hybrid operation with real hardware sources, making the framework suitable for both teaching and research in advanced measurement science.

The internal operation of the proposed virtual framework is implemented using a producer–consumer programming architecture given in Fig.2, which provides deterministic execution, scalability, and clear separation between user interaction and data processing tasks. The block diagram is structured around two main execution loops: an event-driven producer loop and a data-oriented consumer loop, both operating continuously during system runtime.

The producer loop is implemented using an Event Structure and is responsible for handling all user interactions originating from the front panel, particularly from the Packets tab. Each user action—such as clicking on a virtual instrument icon—generates a corresponding event. Upon detection of an event, the producer loop identifies the selected instrument and dynamically launches the associated virtual instrument (VI) module. This is achieved through programmatic VI invocation mechanisms, allowing each module to be opened, activated, and executed independently while maintaining control over its execution state. This approach enables modular operation, where multiple instrument VIs can be instantiated or reactivated without blocking the main execution flow.

Once launched, individual virtual instrument modules operate as autonomous processes that interact with the shared system resources through predefined communication channels. These channels are abstracted at the software level and are independent of the specific instrument implementation, ensuring uniform interaction across standard instruments (e.g., multimeters, function generators, DAQ interfaces) and specialised modules (e.g., uncertainty evaluation, machine learning-based outlier detection, IoT communication).

The consumer loop constitutes the data management and communication backbone of the framework. Its primary role is to manage information exchange through the virtual channels defined in the Variables tab. These channels are implemented as global variables providing a controlled and synchronized mechanism for data sharing

between concurrently running VIs. The consumer loop continuously monitors channel updates, performs buffering and type handling, and propagates updated values to all subscribed modules. In addition to numerical measurement data, the consumer loop also handles control flags, status indicators, and textual messages used for coordination and diagnostics.

Message passing between virtual instruments is supported through a lightweight messaging mechanism integrated into the consumer loop. Each instrument can publish messages—such as state changes, warnings, or completion notifications—which are processed centrally and forwarded to relevant modules or visualised through a dedicated message indicator. This design supports loose coupling between instruments, allowing them to exchange information without direct references to each other's internal implementation.

By decoupling event handling from data processing, the producer–consumer architecture ensures responsive user interaction even when multiple instruments are active or computationally intensive tasks (e.g., machine learning inference or uncertainty propagation) are executed. Furthermore, the use of virtual channels as shared communication entities enables the framework to emulate distributed measurement systems, where instruments interact through well-defined interfaces rather than direct wiring.

III. VIRTUAL FRAMEWORK USE-CASE: MACHINE-LEARNING-BASED OUTLIER DETECTION

To assess different approaches to outlier detection in measurement applications, three representative methods are examined within the proposed virtual framework. Dixon's Q-test and Grubbs' test are applied as classical statistical procedures directly to measurement datasets, while the machine-learning-based Isolation Forest algorithm is implemented and executed as a dedicated virtual instrument operating on the same data channels. This unified setup enables a consistent comparison of statistically driven and instrument-based data-driven approaches under identical experimental conditions

A. STATISTICAL OUTLIER DETECTION USING DIXON'S Q-TEST

This section addresses the application of Dixon's Q-test [11] as a classical statistical method for detecting single outliers in small-sample measurement datasets. Within the virtual framework, Dixon's test is used as a reference baseline, reflecting traditional approaches commonly applied in laboratory-scale measurements and calibration procedures.

The method assumes a normally distributed dataset with a limited number of observations n , typically $3 \leq n \leq 30$, and is designed to identify a single suspect value at either end of an ordered dataset. Given a sorted sample

$x_1 \leq x_2 \leq \dots \leq x_n$, the Dixon test statistic is defined as:

$$Q = \frac{|x_{\text{suspect}} - x_{\text{nearest}}|}{x_n - x_1}, \quad (1)$$

where x_{suspect} is the extreme value under investigation and x_{nearest} is the closest neighboring value. The computed Q statistic is compared against critical values Q_{crit} that depend on the sample size n and the chosen significance level α .

In the proposed virtual framework, Dixon's Q -test is implemented as a standalone statistical method outside the virtual instrumentation framework. Its role is to provide transparent, rule-based outlier identification suitable for didactic comparison. The limitations of the method—namely its restriction to small datasets and inability to handle multiple or masked outliers—are explicitly demonstrated through controlled experimental scenarios.

B. GRUBBS' TEST-BASED OUTLIER DETECTION

The Grubbs' test [12] is a parametric statistical approach for detecting outliers under the assumption of normality. Compared to Dixon's test, Grubbs' method is more flexible in terms of sample size and is widely used in metrology and experimental physics for validating measurement consistency.

For a dataset $\{x_1, x_2, \dots, x_n\}$ the Grubbs test statistic is defined as:

$$G = \frac{\max_i |x_i - \bar{x}|}{s}, \quad (2)$$

where $\bar{x}$ is the sample mean and s is the sample standard deviation. The value G is compared to a critical threshold G_{crit} , which depends on the sample size and significance level and is derived from the Student's t -distribution.

Similarly as the Dixon test, the Grubbs' test is implemented statistically based on the data from the virtual channels written into text files. This approach allows repeated application of the test to iteratively remove detected outliers, enabling analysis of both single and multiple outlier scenarios. The implementation highlights the sensitivity of Grubbs' test to deviations from normality and to the presence of multiple extreme values, which may distort the mean and variance estimates.

Grubbs' test serves as a benchmark parametric method against which the machine-learning-based approach is evaluated, particularly in terms of robustness to non-Gaussian noise and dynamically changing measurement conditions.

C. MACHINE-LEARNING-BASED OUTLIER DETECTION USING ISOLATION FOREST

This section introduces Isolation Forest (iForest) as a data-driven, non-parametric outlier detection method integrated into the virtual framework. Unlike Dixon's and Grubbs' tests, Isolation Forest does not rely on distributional assumptions and is well suited for high-dimensional or complex measurement data.

The Isolation Forest algorithm is based on the

principle that anomalies are more susceptible to isolation than normal observations. It constructs an ensemble of randomly generated binary trees, where each tree recursively partitions the data using randomly selected features and split values. The path length $h(x)$ required to isolate a data point x is used as a measure of normality. The anomaly score is defined as:

$$s(x, n) = 2^{-\frac{E[h(x)]}{c(n)}} \quad (3)$$

where $E[h(x)]$ is the expected path length over all trees and $c(n)$ is a normalization factor dependent on the sample size n . Data points with scores close to 1 are classified as anomalies.

In the proposed virtual framework, the Isolation Forest algorithm is implemented as a dedicated machine-learning virtual instrument that operates on data acquired through the virtual channels. Through a python node, the framework supports configurable hyperparameters, such as the number of trees and subsampling size, and allows real-time comparison with classical statistical methods. This enables systematic evaluation of detection accuracy, false-positive rates, and adaptability to non-stationary measurement processes.

The Isolation Forest use-case demonstrates the potential of machine-learning-based approaches to overcome the limitations of traditional statistical tests, particularly in scenarios involving large datasets, mixed noise sources, or evolving measurement conditions.

D. EXPERIMENT SETUP AND ANALYSIS

The experiment was carried out within the proposed virtual framework using two cooperating virtual instruments given in Fig.3 through the virtual channel CH1. A DC voltage generator with additive white Gaussian noise acted as the signal source, while a dedicated machine-learning virtual instrument implementing the Isolation Forest algorithm performed outlier detection. The generator was configured to produce a constant DC voltage of 10 V, to which zero-mean Gaussian noise and random voltage transients were added, and the resulting signal was continuously transmitted to the ML virtual instrument via the shared virtual channel.

In the initial phase of the experiment, the generator operated with a noise standard deviation of 0.5 V, representing nominal measurement conditions. Under these settings, a dataset of 100 samples was acquired and used to train the Isolation Forest model within the ML virtual instrument. During training, the generator output remained stable, and the acquired samples were simultaneously recorded to a file to ensure reproducibility and enable later offline analysis.

After completion of the training phase, the trained model was evaluated using a new dataset obtained from the same signal source, but with an increased noise standard deviation of 0.7 V and new voltage transients.

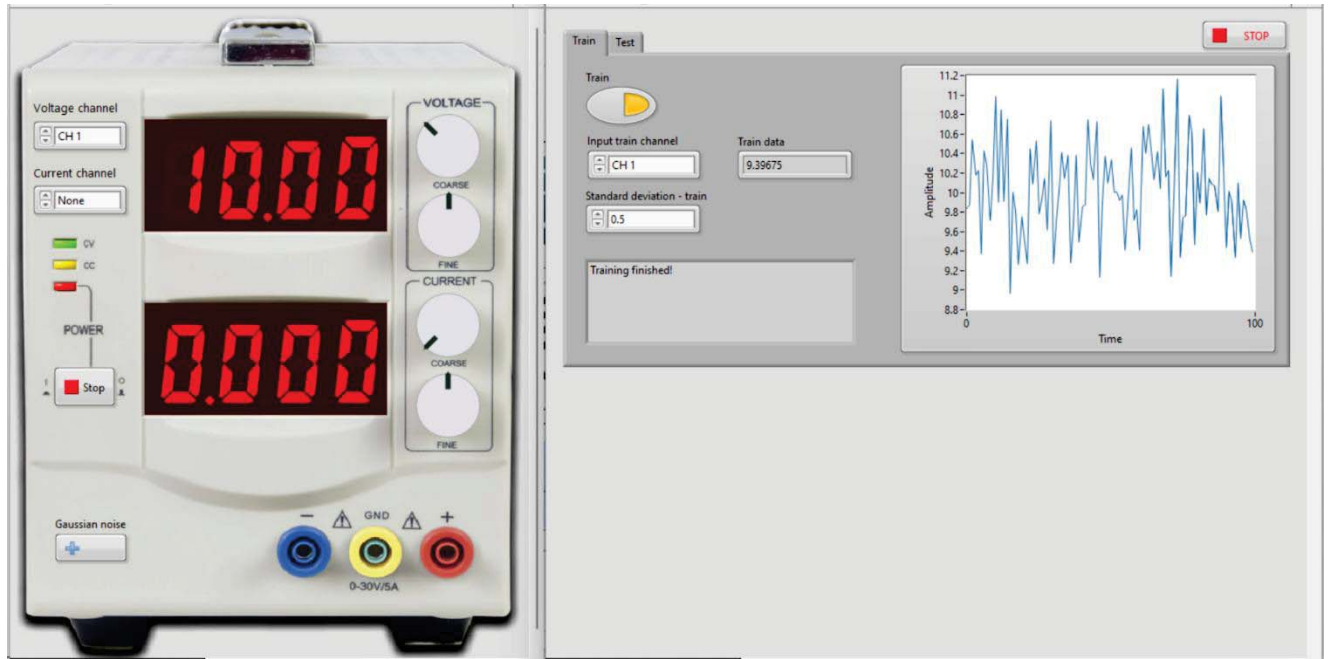


Fig. 3. Front panel of the virtual instrument setup for testing the ML Outlier detection test using the isolation forest approach

This deliberate change in noise level introduced a controlled variation in the statistical properties of the signal, emulating realistic measurement scenarios such as increased interference or degraded sensor performance. The new dataset was processed in real time by the ML virtual instrument, which produced outlier detection results based on the previously trained model. The test data were also stored to a file for further analysis.

Both the training and testing datasets were subsequently reused for the application of Dixon's Q-test and Grubbs' test. By applying the statistical methods to exactly the same recorded data, a consistent and fair comparison between the machine-learning-based approach and classical statistical outlier detection techniques was ensured.

To understand how much the methods agreed, confusion matrices were computed using the Grubbs test as a reference. A true positive (TP) indicates that both methods identified the same observation as an outlier; a false positive (FP) that the method under consideration flagged a point not flagged by Grubbs; and a false negative (FN) that the method failed to flag an observation identified by Grubbs. Table I summarises these results.

TABLE I

Confusion matrices comparing the isolation forest and Dixon test with the iterative Grubbs test.

Grubbs	TP	FP	FN	TN
Isolation Forest	6	0	1	93
Dixon test	2	0	5	93

The isolation forest and Grubbs test agreed on six outliers; the only false negative for the isolation forest was a high-valued observation that Grubbs identified but the machine-learning model did not. There were no false positives, indicating that all six isolation forest detections were also considered outliers by Grubbs. The Dixon test agreed on the two most extreme points but missed five outliers identified by Grubbs. These results demonstrate that the conservatively tuned isolation forest produces a detection pattern comparable to the Grubbs test, whereas the Dixon test captures only the most extreme deviations. They are graphically presented in Fig. 4.

The plot shows that the isolation forest and Grubbs test highlight largely the same observations, while the Dixon test flags only the most extreme low and high values. This analysis shows that a conservatively tuned isolation forest can produce outlier counts comparable to classical tests while retaining the ability to detect moderate deviations. The algorithm's use of random partitioning means that anomalies tend to be isolated quickly, resulting in shorter isolation paths and higher anomaly scores.

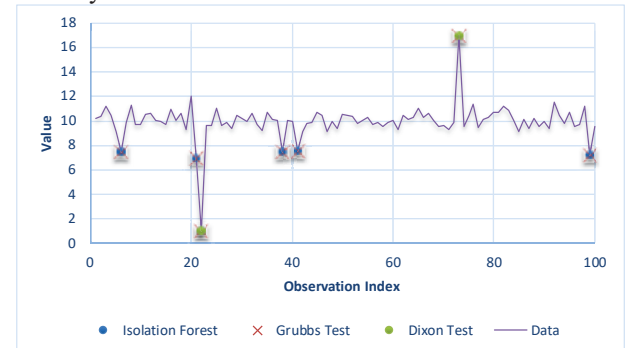


Fig. 4. Outlier detection comparison

The Grubbs test remains a straightforward method for detecting extreme deviations when the data are approximately normally distributed and only a small number of outliers is expected. In this dataset it identified seven points that deviated markedly from the rest, and all but one were also detected by the isolation forest. Because the Grubbs test examines one point at a time and assumes a single outlier, iterative application can be sensitive to the order in which outliers are removed and may not capture more subtle anomalies.

The Dixon test's Q statistic considers the ratio of the gap to the range and is designed for very small samples. Its conservative nature is evident here, as it identified only the two most extreme observations. While this makes Dixon's test reliable when only high errors are of concern, it may overlook moderate anomalies.

IV. CONCLUSION

This work has demonstrated a flexible virtual laboratory environment that integrates simulation, real-time data acquisition, and machine-learning analysis for advanced measurement science education. Using modular LabVIEW-based instruments and a producer–consumer architecture, the framework enables coordinated operation of multiple virtual instruments and systematic exploration of sensor noise, telemetry effects, and outlier detection. The case study comparing the Isolation Forest algorithm with classical Dixon and Grubbs tests highlighted the strengths and limitations of each method. A conservatively tuned Isolation Forest produced outlier counts comparable to the iterative Grubbs test, while both approaches clearly outperformed the Dixon Q-test, which detected only the most extreme values.

Confusion-matrix analysis showed strong agreement between Grubbs and the Isolation Forest, whereas the Dixon test missed several anomalies, confirming its conservative character. These results indicate that machine-learning-based techniques can be tuned to closely match classical statistical tests while remaining sensitive to subtler deviations. Overall, the proposed virtual environment provides a robust and accessible platform for teaching and experimentation in modern metrology, effectively combining traditional statistical methods with contemporary data-driven approaches.

ACKNOWLEDGEMENT

This work has been supported by the Erasmus+ Programme of the European Union through the project Personalized Education in Measurement Science and Technology Empowered by Machine Learning - PEMS-ML, grant no. 024-1-MK01-KA220-HED-000248299. The authors acknowledge the support of the project partners and institutions involved in the implementation of the project activities.

REFERENCES

- [1] L. Tibola, F. Herpich, P. Silva, and L. Tarouco, "Experience in teaching science in virtual environment," *International Journal for Innovation Education and Research*, vol. 7, 2019, doi: 10.31686/ijer.Vol7.Iss4.1375.
- [2] F. A. Jalk Duque, S. Valbuena Duarte, and F. J. Racedo Niebles, "Design of a virtual environment with interactive simulations for the teaching of mechanic," *EURASIA Journal of Mathematics, Science and Technology Education*, vol. 22, no. 1, Art. no. em2764, 2026, doi: 10.29333/ejmste/17756.
- [3] C. Ionascu and A. Jipa, "Virtual experiments for measuring fundamental physical quantities," *Romanian Reports in Physics*, vol. 74, Art. no. 911, 2022.
- [4] C. Tokatlidis, S. Rapti, S. Tselegkaridis, T. Sapounidis, and D. Papakostas, "Virtual environment in engineering education: The role of guidance, knowledge and skills development in electronic circuits teaching," *Education Sciences*, vol. 14, no. 12, Art. no. 1336, 2024, doi: 10.3390/educsci14121336.
- [5] A. Vargas-Hernández, S. Robledo, and G. R. Quiceno, "Virtual teaching for online learning from the perspective of higher education: A bibliometric analysis," *Journal of Scientometric Research*, vol. 13, no. 2, pp. 406–418, 2024, doi: 10.5530/jscires.13.2.32.
- [6] S. Lee, Y.-J. Choi, and H.-S. Kim, "The accurate measurement of students' learning in e-learning environments," *Applied Sciences*, vol. 11, no. 21, Art. no. 9946, 2021, doi: 10.3390/app11219946.
- [7] M. K. Pedersen, B. Skyum, R. Heck, R. Müller, M. Bason, A. Lieberoth, and J. F. Sherson, "Virtual learning environment for interactive engagement with advanced quantum mechanics," *Phys. Rev. Phys. Educ. Res.*, vol. 12, Art. no. 013102, 2016, doi: 10.1103/PhysRevPhysEducRes.12.013102.
- [8] C. Illi and A. Elhassouny, "Edu-metaverse: A comprehensive review of virtual learning environments," *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3540944.
- [9] S. Ren, "Development of a virtual experiment teaching system based on LabVIEW," in *Proc. International Conference on Educational and Information Technology*, 2021, doi: 10.1007/978-3-030-62746-1_136.
- [10] Y. Zhou, J.-J. Jiang, and S.-C. Fan, "A LabVIEW-based, interactive virtual laboratory for electronic engineering education," *Int. J. Engng Ed.*, vol. 21, no. 1, pp. 94–102, 2005.
- [11] W. J. Dixon, "Processing data for outliers," *Biometrics*, vol. 9, no. 1, pp. 74–89, 1953.
- [12] F. E. Grubbs, "Sample criteria for testing outlying observations," *Ann. Math. Stat.*, vol. 21, no. 1, pp. 27–58, 1950, doi: 10.1214/aoms/1177729885.