Neutrosophic Sets and Systems, {Special Issue: Artificial Intelligence, Neutrosophy, and Latin American Worldviews: Toward a Sustainable Future (Workshop – March 18–21, 2025, Universidad Tecnológica de El Salvador, San Salvador, El Salvador)}, Vol. 84, 2025

University of New Mexico

Comparative analysis of machine learning platforms to optimize DevOps: application of the Neutrosophic OWA-TOPSIS model

Miguel Angel Quiroz Martinez*[1], Keyko Garces Salazar [2], Joshua Montesdeoca Soriano [3] and and Mónica Gomez-Rios [4]

[1 4] GIIAR, Universidad Politécnica Salesiana, Guayaquil, Ecuador. mquiroz@ups.edu.ec , mgomezr@ups.edu.ec
[2 3] Computer Science Department, Universidad Politécnica Salesiana, Guayaquil, Ecuador. kgarcess@est.ups.edu.ec , jmontesdeocas1@est.ups.edu.ec

Abstract. As software systems and their associated information become increasingly complex within DevOps environments, Machine Learning (ML) platforms are growing in importance for optimizing development and deployment processes. This article presents a comparative analysis of two leading ML platforms, Amazon Web Services (AWS) and Microsoft Azure, to evaluate their suitability for optimizing DevOps. A quantitative methodology based on an experimental comparative method was employed, applying the neutrosophic multi-criteria OWA-TOPSIS model to assess and select the best alternative based on specific criteria such as scalability, integration, performance, and cost-benefit. The results from the OWA-TOPSIS model, derived from controlled experimental assessments, indicate that Microsoft Azure offers greater advantages over AWS for DevOps optimization and software deployment in the studied use cases. However, it is acknowledged that the optimal platform choice may vary depending on the specific needs of each project and organization.

Keywords: DevOps, Machine Learning, Azure, AWS, OWA-TOPSIS, Neutrosophic Sets, Single-Valued Neutrosophic Linguistic Sets, Multi-criteria Decision Making, Platform Selection

1. Introduction

Software development is a common term in our daily lives due to technological advances; likewise, it is common to use software development methodologies because they help to efficiently carry out all the processes involved in software development; considering this, we have the DevOps methodology, by using this methodology it is possible to reduce the development life cycle, increasing the deployment frequency and releasing secure products that meet business requirements, thanks to the fact that the " development " and "operations" teams when using this methodology improve their communication and interact more frequently these two teams [1]. Machine learning platforms can be incorporated into this "DevOps" software development methodology to optimize DevOps processes since it allows the use of techniques to analyze data and logs generated during DevOps practices and automatically detect anomalies [2].

Since there is a wide variety of machine learning platforms applicable to DevOps, there is a need to determine the best platform that favors the DevOps methodology, taking into account processes such as the detection of anomalies in the data since DevOps is used, the information generated is deepened, becoming a great job to analyze the data and records generated by this practice [3].

Choosing a platform that optimizes DevOps is not something that can be taken lightly, given that the platform may not be accurate or effective enough when dealing with the data or logs; this is due to the quality and quantity of data that has been generated when using the DevOps methodology [4].

Therefore, this research seeks to analyze machine learning platforms to optimize DevOps environments through a multi-criteria comparative analysis to efficiently improve software development and deployment processes. The specific objectives are to identify machine learning platforms applicable to DevOps environments, select the main platforms through a systematic literature review, and define the criteria of machine learning platforms for their applicability in optimizing DevOps environments using the OWA-TOPSIS model. Validate the results obtained to guarantee the efficiency of the software development and deployment processes through an evaluation.

2. Background

In software development, we rely on development methodologies such as DevOps, which focuses on improving collaboration between development and operations teams; thanks to this, we can have shorter and more reliable software product release cycles, thus improving product quality and customer satisfaction [5].

When using DevOps, system information is deepened and new functions emerge, which makes the methodology complex when analyzing data or records, so machine learning becomes essential for the processes mentioned above to be carried out efficiently and effectively [3].

When using DevOps, aspects such as collaboration between development and operations teams to achieve common goals, automation of processes such as continuous integration and delivery (which allows new changes to be implemented quickly), automation testing that allows the software life cycle to be accelerated, and finally, monitoring and feedback, which allow the identification of areas for improvement, resolution of problems and optimization of development and operations processes, stand out [6]. The stages in which it intervenes range from software development to implementation and, finally, the maintenance stage within the software life cycle [7.

An example of optimization is Jenkins, which is often used in Continuous Integration and Continuous Delivery (CI/CD) workflows, facilitating the creation, testing, and debugging of software projects in an automated way, locally or in the cloud [8].

An example of the relationship between machine learning and DevOps is that with the help of machine learning, it is possible to automate repetitive tasks, thus reducing the workload of development and operations teams [3]. Furthermore, when detecting problems in software projects using DevOps with machine learning, these can be automatically tagged as a bug report, a working solution, or a question [4].

Azure is a platform that has a Machine Learning (ML) section that allows us to create ML models with the information generated in software development using the DevOps methodology, transforming the information into a data set to train the model and predict possible errors in development and deployment, and this is achieved because it allows us to generate a model endpoint by implementing it as a web service that receives input requests and returns the prediction in real-time [9].

On the other hand, we have the AWS (Amazon Web Service) platform, an Amazon cloud service that allows configuring servers and interacting with them to prevent companies from creating their own data centers [10]. Therefore, it allows the development of a machine learning model to optimize the processes of the DevOps methodology, carried out through Amazon Sagemaker , a service that works with machine learning. In this way, the services are configured to deploy the model and make data predictions [11]. Finally, an endpoint is generated in the AWS Sagemaker control panel where the responses to requests can be observed in real-time.

In a comparative analysis of the two machine learning platforms, AWS, with its wide range of services and market leadership, stands out for its robust infrastructure and flexibility. On the other hand, Azure stands out for its integration with Microsoft products and offers robust tools for business development. Both platforms are successful in various sectors but differ in their approach and strengths, which influences users' choices based on their specific needs and integration preferences [12].

3. Methodology

This research was conducted using a quantitative approach using the experimental comparative method, whereby we worked with numerical values acquired through experimentation and then used them in a multi-criteria model to determine the best machine learning platform to optimize DevOps environments.

Through a systematic literature review, criteria for machine learning platforms that optimize DevOps were selected, such as a wide range of tools, configuration flexibility, integration with other services, scalable pricing model [13], variety of machine learning and data analysis tasks, compatibility with Python frameworks, scalability, and distribution [14]. Therefore, the platforms chosen based on these criteria were Microsoft Azure and Amazon Web Service.

Microsoft Azure is a multi-tool platform that enables the development of DevOps environments and machine learning models. In terms of scalability and deployment, it allows for quick and easy resource addition, meeting the demands of ever-evolving applications. Its pricing model is based on pay-per-use, offering users the advantage of reducing costs and paying only for the resources used. Its configuration flexibility allows users to modify the cloud infrastructure according to the specific needs of their application, which is crucial for efficient and productive deployment [15]. Various machine learning tasks can be performed, such as building, evaluating, and deploying predictive models and analyzing data to make predictions [16].

AWS (Amazon Web Service) is a cloud services platform that offers a wide range of solutions, facilitating the development of DevOps-based models. This platform enables efficient data collection, analysis, and visualization, standing out for its real-time monitoring capabilities. Thanks to its advanced services, AWS maximizes the use of data storage and processing, as demonstrated by the automatic disaster alert system [17].

In terms of its pricing structure, AWS adapts to user consumption, offering a cost-effective and flexible alternative to traditional physical data centers. Users benefit by paying only for the services used, which can translate into significant savings [12].

Furthermore, AWS demonstrates its ability to manage and process large volumes of CSV data efficiently, highlighting its scalability and optimization [18]. Furthermore, AWS supports various machine learning models, including binary and multiclass classification, as well as regression models, underlining its strength in the field of predictive analytics [19].

The use of a machine learning platform in DevOps development environments can be evaluated using key metrics that demonstrate its effectiveness, importance, and economic viability. The first criterion is Prediction Accuracy, which evaluates the accuracy of the predictive model in identifying error-prone areas [20]. The second criterion was Ease of Use as a fundamental variable in choosing the machine learning platform that optimizes DevOps [21]. As a third criterion, the Operating Cost is taken into consideration, referring to the price to pay for the use of the platform [22]. One criterion to consider is Development Time, which refers to how long it takes to develop and integrate new functionalities, and Ease of Integration, based on how easy it is to integrate the ML model into the current work environment. These metrics comprehensively show how machine learning platforms can optimize processes in DevOps environments.

These metrics provide a basis for a comprehensive assessment. For the subsequent multi-criteria evaluation using the OWA-TOPSIS model, these metrics were operationalized within the four main criteria (C1-C4) presented to the experts for the final evaluation. Specifically, the Performance criterion (C3) was assessed by considering results related to Prediction Accuracy and Development Time. The Integration criterion (C2) directly reflected the Ease of Integration with existing DevOps tools and workflows. The Cost-benefit criterion (C4) was primarily informed by the Operating Cost analysis of using the platform during the experiment. Finally, the Scalability criterion (C1) was evaluated based on the platform's inherent features and capabilities related to adapting to different workloads, as perceived through its Ease of Use and configuration flexibility during the experimental setup. This framework allowed for a structured evaluation suitable for the OWA-TOPSIS model.

Once the platforms were selected, we reviewed each one along with the documentation to perform controlled experimentation. To do this, we selected and trained a machine learning (ML) model. Microsoft Azure offers "Azure ML" and AWS, the " SageMaker " tool. A workspace was created on each platform. On Azure ML, the "STANDARD_DS3_V2" compute clusters were used to train the model, and on AWS, SageMaker, " ml.g 5.48xlarge". The model was trained on each platform with data from a real software project through a Flask project clone. Cyclomatic complexity, maintainability index, and the number of lines of source code were extracted using the Python tool " radon ", and each feature was tagged on GitHub based on its involvement in the project's issues. Once the prediction model was trained, we proceeded to deploy it. The chosen ML platforms allowed us to create endpoint APIs to integrate the model into a testing process in a DevOps environment, thereby anticipating potential bugs in new features in the code. Data was then collected based on the defined metrics, along with expert opinions to assess each of the established metrics.

3.1 OWA-TOPSIS METHOD

This section provides a brief overview of the fundamental principles related to SVNS and SVNLS, covering definitions, operating principles, and metrics for measuring distances.

Definition 1 [23]. Let x be an element in a finite set, X. A single-valued neutrosophic set (SVNS), P, in X can be defined as in (1):

$$P = \{ x, T_P(x), I_P(x), F_P(x) | x \in X\}, \quad (1)$$

where the truth membership function, $T_P(x)$,the indeterminacy membership function $I_P(x)$, and the falsehood membership function $F_P(x)$clearly adhere to condition (2):

$$0 \le T_P(x), I_P(x), F_P(x) \le 1; \; 0 \le T_P(x) + I_P(x) + F_P(x) \le 3 \quad (2)$$

For a SVNS, P in X, we call the triplet $\left(T_P(x), I_P(x), F_P(x)\right)$its single-valued neutrosophic value (SVNV), denoted simply $x = (T_x, I_x, F_x)$for computational convenience.

Definition 2 [23]. Let $x = (T_x, I_x, F_x)$yy $= \left(T_y, I_y, F_y\right)$let there be two SVNV. Then

1) $x \oplus y = \left(T_x + T_y - T_x * T_y, I_x * T_y, F_x * F_y\right)$;
2) $\lambda * x = (1 - (1 - T_x)\lambda, (I_x)\lambda, (F_x)\lambda), \lambda > 0$;
3) $x^\lambda = \left((T_x)\lambda, 1 - (1 - I_x)\lambda, 1 - (1 - F_x)\lambda\right), \lambda > 0$

Let l be $S = \{s_\alpha | \alpha = 1, \dots, l\}$a finite, totally ordered discrete term with an odd value, wheres_α denotes a possible value for a linguistic variable. For example, if $l = 7$, then a set of linguistic terms S could be described as follows[24]:

$$S = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\} = \{extremely\ poor, very\ poor, poor, fair, good, very\ good, extremely\ good\}. \quad (3)$$

Any linguistic variable, s_i and s_j, in S must satisfy the following rules:

1) $Neg(s_i) = s_{-i}$;
2) $s_i \le s_j \Leftrightarrow i \le j$;
3) $\max(s_i, s_j) = s_j,\ if\ i \le j$;
4) $\min(s_i, s_j) = s_i,\ if\ i \le j$.

To avoid information loss during an aggregation process, the discrete set of terms S will be extended to a continuous set of terms. $S = \{ s_\alpha | \alpha \in R\}$.Any two linguistic variables $s_\alpha, s_\beta \in S$satisfy the following operational laws [24] :

1) $s_\alpha \oplus s_\beta = s_\alpha + \beta;$
2) $\mu s_\alpha = s_{\mu\alpha}, \mu \geq 0;$
3) $\frac{s_\alpha}{s_\beta} = s_{\frac{\alpha}{\beta}}$

Definition 3 [25] Given X, a finite set of universes, a SVNLS, P, in X can be defined as in (4):

$$P = \{\langle x, [s_{\theta(x)}, (T_P(x), I_P(x), F_P(x))]\rangle | x \in X\} \quad (4)$$

where $s_{\theta(x)} \in \bar{S}$, the truth membership function $T_P(x)$, the indeterminacy membership function, $I_P(x)$and the falsehood membership function $F_P(x)$satisfy condition (5):

$$0 \leq T_P(x), I_P(x), F_P(x) \leq 1, 0 \leq T_P(x) + I_P(x) + F_P(x) \leq 3. \quad (5)$$

For an SVNLS, P, in X, the 4-tuple $\langle s_{\theta(x)}, (T_P(x), I_P(x), F_P(x))\rangle$is known as the Single-Valued Neutrosophic Linguistic Number (SVNLN), conveniently denoted $x = s_{\theta(x)}, (T_x, I_x, F_x)$for computational purposes.

Definition 4 [25] . Let there be $x_i = \langle s_{\theta(xi)}, (T_{xi}, I_{xi}, F_{xi})\rangle$ $(i = 1,2)$two SVNLN. Then

1) $x_1 \oplus x_2 = \langle s_{\theta(x1)} + \theta_{x2}, (T_{x1} + T_{x2} - T_{x1} * T_{x2}, I_{x1} * T_{x2}, F_{x1} * F_{x2})\rangle$
2) $\lambda_{x1} = \langle s_{\lambda\theta(x1)}, (1 - (1 - T_{x1})^\lambda, (I_{x1})^\lambda, (F_{x1})^\lambda)\rangle, \lambda > 0;$
3) $x_1^\lambda = \langle s_{\theta^\lambda(x_1)}, ((T_{x1})^\lambda, 1 - (1 - I_{x1})^\lambda, 1 - (1 - F_{x1})^\lambda)\rangle, \lambda > 0.$

Definition 5 [25] . Let there be $x_i = \langle s_{\theta(xi)}, (T_{xi}, I_{xi}, F_{xi})\rangle$ $(i = 1,2)$two SVNLNs. Their distance measure is defined as in (6):

$$d(x_1, x_2 v) = \left[|s_{\theta(x1)}T_{x1} - s_{\theta(x2)}T_{x2}|^\mu + |s_{\theta(x1)} I_{x1} - s_{\theta(x2)} I_{x2}|^\mu + |s_{\theta(x1)}F_{x1} - s_{\theta(x2)}F_{x2}|^\mu\right]^{\frac{1}{\mu}} \quad (6)$$

In particular, equation (6) reduces the Hamming distance of SVNLS and the Euclidean distance of SVNLN when $\mu = 1$and $\mu = 2$, respectively.

3.1.1. MADM Based on the SVNLOWAD-TOPSIS Method

For a given multi-attribute decision-making problem in SVNL environments, $A = \{A_1, \dots, A_m\}$denotes a set of discrete feasible alternatives, $C = \{C_1, \dots, C_n\}$represents a set of attributes, and $E = \{e_1, \dots, e_k\}$is a set of experts (or DMs) with weight vector $\omega = \{\omega_1, \dots, \omega_k\}$T such that $\sum_{i=1}^n w_i = 1$ and $0 \leq \omega_i \leq 1$. Suppose that the attribute weight vector is $s\ v = (v_1, \dots, v_n)^T$, which satisfies $\sum_{i=1}^n v_i = 1$ and $v_i \in [0,1]$. The evaluation, $\alpha_{ij}^{(k)}$given by the expert, $e_{t(t = 1,\dots,k)}$on the alternative, $A_{i(i = 1,\dots,m)}$,relative to the attribute, $C_{j(j = 1,\dots,n)}$forms the individual decision matrix as shown in equation (7):

$$D^k = \begin{array}{c c} & \begin{array}{ccc} C_1 & \cdots & C_n \end{array} \\ \begin{array}{c} A_1 \\ \vdots \\ A_n \end{array} & \begin{pmatrix} \alpha_{11}^{(k)} & \cdots & \alpha_{1n}^{(k)} \\ \vdots & \ddots & \vdots \\ \alpha_{m1}^{(k)} & \cdots & \alpha_{mn}^{(k)} \end{pmatrix} \end{array} \quad (7)$$

where $\alpha_{ij}^k = \langle s_{\theta(\alpha_{ij})}^k, (T_{\alpha_{ij}}^k, I_{\alpha_{ij}}^k, F_{\alpha_{ij}}^k)\rangle$ is represented by a SVNLN, which satisfies $s_{\theta(\alpha ij)}^k \in \bar{S}, T_{\alpha_{ij}}^k, I_{\alpha_{ij}}^k, F_{\alpha_{ij}}^k \in [0,1] and\ 0 \leq T_{\alpha_{ij}}^k + I_{\alpha_{ij}}^k + F_{\alpha_{ij}}^k \leq 3$.

Geng et al. [26] extended the TOPSIS method to fit the SVNLS scenario, and the procedures of the extended model can be summarized as follows (Figure 1.).

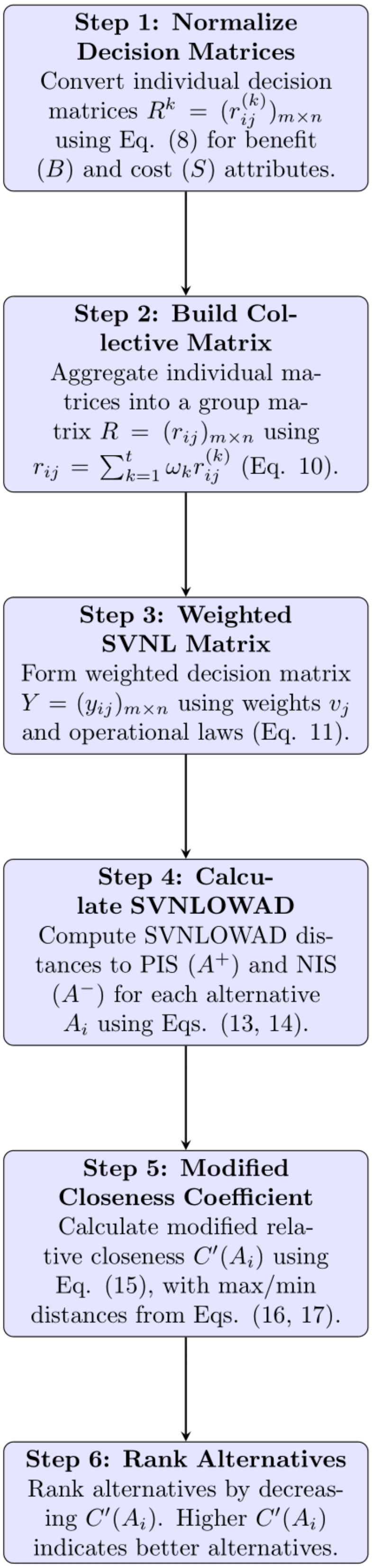

Figure 1. Flowchart of the Multi-Attribute Decision-Making Process with SVNLOWAD

Step 1. Normalize the individual decision matrices:

In practical scenarios, MADM problems can encompass both benefit attributes and cost attributes. Let B and S the benefit attribute sets and cost attribute sets, respectively. Therefore, the conversion rules specified in (8) apply:

$$\begin{cases} r_{ij}^{(k)} = \alpha_{ij}^{(k)} = \langle s_{\theta(\alpha_{ij})}^{k}, (T_{\alpha_{ij}}^{k}, I_{\alpha_{ij}}^{k}, F_{\alpha_{ij}}^{k}) \rangle, & \text{for} \quad j \in B, \\ \quad r_{ij}^{(k)} = \langle s_{l-\theta(\alpha_{ij})}^{k}, (T_{\alpha_{ij}}^{k}, I_{\alpha_{ij}}^{k}, F_{\alpha_{ij}}^{k}) \rangle, & \text{for} \quad j \in S. \end{cases} \tag{8}$$

Thus, the standardized decision information, $R^k = (r_{ij}^{(k)})_{m\times n}$, is set as in (9):

$$R^k = (r_{ij}^{(k)})_{m\times n} = \begin{pmatrix} r_{11}^{(k)} & \cdots & r_{1n}^{(k)} \\ \vdots & \ddots & \vdots \\ r_{m1}^{(k)} & \cdots & r_{mn}^{(k)} \end{pmatrix} \tag{9}$$

Step 2. Build the collective matrix :

All individual DM reviews are aggregated into a group review:

$$R = (r_{ij})_{m\times n} = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mn} \end{pmatrix} \tag{10}$$

Where $r_{ij} = \sum_{k=1}^{t} \omega_k \, r_{ij}^{(k)}$.

Step 3. Set the weighted SVNL decision information:

The weighted SVNL decision matrix, Y , is formed as shown in (11), using the operational laws given in Definition 2 above:

$$Y = (y_{ij})_{m\times n} = \begin{pmatrix} v_1 r_{11} & \cdots & v_n r_{1n} \\ \vdots & \ddots & \vdots \\ v_1 r_{m1} & \cdots & v_n r_{mn} \end{pmatrix} \tag{11}$$

The OWA operator is fundamental in aggregation techniques, widely studied by researchers [27] . Its main advantage lies in organizing arguments and facilitating the integration of experts' attitudes in decision making. Recent research has explored OWA in distance measurement, generating variations of OWAD [28] . Taking advantage of the benefits of OWA, the text proposes a SVNL OWA distance measure (SVNLOWAD). Given the desirable properties of the OWA operator, an SVNL OWA distance measure (SVNLOWAD) is proposed in the following text.

Definition 6. *Let $x_j, x_j^{'}$ $(j = 1, \dots, n)$ the two collections be SVNLN. If*

$$SVNLOWAD((x_1, x_1'), \dots, (x_n, x_n')) = \sum_{j=1}^{n} w_j d(x_j, x_j'), \tag{12}$$

Therefore, step 4 of this method can be considered as follows:

Step 4. For each alternative, A_i the SVNLOWAD is calculated for the PIS, A^+ and the NIS A^-, using equation (12):

$$SVNLOWAD(A_i, A^+) = \sum_{j=1}^{n} w_j \, \dot{d}(y_{ij}, y_j^+), i = 1, \dots, m \tag{13}$$

$$SVNLOWAD(A_i, A^-) = \sum_{j=1}^{n} w_j \, \dot{d}(y_{ij}, y_j^-), i = 1, \dots, m \tag{14}$$

where $\dot{d}(y_{ij}, y_j^+)$ and $\dot{d}(y_{ij}, y_j^-)$ are the j - largest values of $\dot{d}(y_{ij}, y_j^+)$ and $\dot{d}(y_{ij}, y_j^-)$, respectively.

Step 5. In the classic TOPSIS approach, the relative closeness coefficient, Ci , is used to rank the alternatives. However, some researchers have highlighted cases where relative closeness fails to achieve the desired objective of simultaneously minimizing the distance from the PIS and maximizing the distance from the NIS. Thus, following an idea proposed in references [30] , in equations (15)–(17), we introduce a modified relative closeness coefficient, $C\ '(\ Ai\ )$, used to measure the degree to which the alternatives, Ai ( ) = 1,..., m =1,...,), are close to the PIS and also far from the NIS, congruently:

$$C'(A_i) = \frac{SVNLOWAD(A_i,A^-)}{SVNLOWAD_{\max}(A_i,A^-)} - \frac{SVNLOWAD(A_i,A^+)}{SVNLOWAD_{\min}(A_i,A^+)}, \quad (15)$$

where

$$SVNLOWAD_{\max}(A_i, A^-) = \max_{1\le i\le m} SVNLOWAD(A_i, A^-), \quad (16)$$

and

$$SVNLOWAD_{\min}(A_i, A^+) = \min_{1\le i\le m} SVNLOWAD(A_i, A^+). \quad (17)$$

It is clear that $C'(A_i) \le 0$ $(i = 1, \dots, m)$the higher the value of $C'(A_i)$and , the better A_i the alternative. Furthermore, if an alternative A^* satisfies the conditions $SVNLOWAD(A^*, A^-) = SVNLOWAD_{\max}(A^*, A^-)$and $SVNLOWAD(A^*, A^+) = SVNLOWAD_{\min}(A^*, A^+)$, then $C'(A^*) = 0$and the alternative $A^*$is the most suitable candidate, since it has the minimum distance to the PIS and the maximum distance to the NIS.

Step 6. Rank and identify the most desirable alternatives based on the decreasing closeness coefficient $C'(A_i)$obtained using Equation (15).

4. Case Study: Comparative analysis of machine learning platforms to optimize DevOps.

Given the increasing complexity of software systems developed using DevOps methodology and the consequent need to identify optimal machine learning platforms for their implementation, this study comparatively evaluates AWS (Amazon Web Services) and Microsoft Azure as the leading machine learning platforms for DevOps environments. For this multi-criteria evaluation, the neutrosophic OWA-TOPSIS model was used. Three DevOps and machine learning experts participated, evaluating the platforms according to specific criteria, applying the neutrosophic single-valued linguistic sets (SVNLS) approach to capture the uncertainty inherent in their evaluation3.3 Evaluation criteria.

The criteria selected to evaluate the platforms were:

- C1: Scalability (ability to adapt to different workloads)
- C2: Integration with DevOps tools (ease of integration with CI/CD pipelines)
- C3: Performance (speed and efficiency in data processing)
- C4: Cost-benefit (relationship between investment and results obtained)

The experts assigned the following weights to the criteria:

- C1 (Scalability): 0.30
- C2 (Integration with DevOps tools): 0.25
- C3 (Performance): 0.25
- C4 (Cost-benefit): 0.20

Alternatives evaluated:

- A1: Amazon Web Services (AWS)
- A2: Microsoft Azure

The following set of linguistic terms was used: S = {s_1 = "extremely poor", s_2 = "very poor", s_3 = "poor", s_4 = "fair", s_5 = "good", s_6 = "very good", s_7 = "extremely good"}

Below are the SVNL decision matrices provided by each expert (DM = Decision Maker):

Table 1. Evaluation of alternatives according to Criterion 1 (Scalability)

Alternatives	**DM1**	**DM2**	**DM3**
AWS (A1)	S_6(0.7,0.1,0.2)	S_6(0.8,0.1,0.1)	S_5 (0.6,0.2,0.2)
Azure (A2)	S_5 (0.6,0.2,0.2)	S_5 (0.5,0.3,0.2)	S_6(0.7,0.1,0.2)

Table 2. Evaluation of alternatives according to Criterion 2 (Integration with DevOps tools)

Alternatives	**DM1**	**DM2**	**DM3**
AWS (A1)	S_5 (0.6,0.2,0.2)	S_4 (0.5,0.3,0.2)	S_5 (0.7,0.1,0.2)
Azure (A2)	S_6(0.8,0.1,0.1)	S_6(0.7, 0.2, 0.1)	S_6(0.8,0.1,0.1)

Table 3. Evaluation of alternatives according to Criterion 3 (Performance)

Alternatives	**DM1**	**DM2**	**DM3**
AWS (A1)	S_6(0.7, 0.2, 0.1)	S_6(0.6,0.2,0.2)	S_5 (0.6,0.3,0.1)
Azure (A2)	S_5 (0.6,0.2,0.2)	S_5 (0.5,0.2,0.3)	S_6(0.7,0.1,0.2)

Table 4. Evaluation of alternatives according to Criterion 4 (Cost-benefit)

Alternatives	**DM1**	**DM2**	**DM3**
AWS (A1)	S_4 (0.5,0.3,0.2)	S_4 (0.4,0.4,0.2)	S_5 (0.6,0.2,0.2)
Azure (A2)	S_5 (0.7,0.2,0.1)	S_5 (0.6,0.3,0.1)	S_6(0.7,0.1,0.2)

Applying the operations defined for SVNLS, the collective decision matrix was calculated considering an equal weight for each expert ($\omega_1 = \omega_2 = \omega_3 = 0.333$).

Table 5. SVNL Collective Decision Matrix

Alternatives	**C1 (Scalability)**	**C2 (Integration)**	**C3 (Performance)**	**C4 (Cost-benefit)**
AWS (A1)	S_5.67(0.704,0.129,0.162)	S_4.67(0.610,0.191,0.200)	S_5.67(0.636,0.232,0.132)	S_4.33(0.507,0.294,0.200)
Azure (A2)	S_5.33(0.607,0.193,0.200)	S_6.00(0.774,0.129,0.100)	S_5.33(0.607,0.166,0.232)	S_5.33(0.669,0.193,0.132)

By applying the criteria weights to the collective decision matrix, the weighted matrix is obtained.

Table 6. Weighted collective SVNL decision matrix

Alternatives	C1 (Scalability)	C2 (Integration)	C3 (Performance)	C4 (Cost-benefit)
AWS (A1)	S_1.70(0.324,0.614,0.704)	S_1.17(0.212,0.720,0.718)	S_1.42(0.224,0.734,0.686)	S_0.87(0.136,0.792,0.760)
Azure (A2)	S_1.60(0.275,0.675,0.721)	S_1.50(0.297,0.614,0.668)	S_1.33(0.210,0.719,0.755)	S_1.07(0.197,0.675,0.686)

In the context of the OWA-TOPSIS model, the PIS (Positive Ideal Point) and the NIS (Negative Ideal Point) were determined:

- PIS (A^+): (S_1.70(0.324,0.614,0.704), S_1.50(0.297,0.614,0.668), S_1.42(0.224,0.734,0.686), S_1.07(0.197,0.675,0.686))
- NIS (A^-): (S_1.60(0.275,0.675,0.721), S_1.17(0.212,0.720,0.718), S_1.33(0.210,0.719,0.755), S_0.87(0.136,0.792,0.760))

The OWA weight vector used was W = (0.35, 0.30, 0.20, 0.15), reflecting the attitude of the decision-makers. Applying equations (13) and (14), the SVNLOWAD measures between each alternative and the ideal points were obtained.

Table 7. Relative distances between each alternative and the reference points

Alternatives	SVNLOWAD(A_i ,A^+)	SVNLOWAD(A_i ,A^-)	C'
AWS (A1)	0.231	0.142	-2.89
Azure (A2)	0.105	0.237	-0.78

Figure 2. Relative distances between each alternative and the reference points

4. Results and Discussion

The results of the analysis using the neutrosophic OWA-TOPSIS method show that:

- Microsoft Azure (A2) has a higher C' value (-0.78) compared to AWS (-2.89)
- Azure has a smaller distance from the positive ideal point (0.105 vs 0.231)
- Azure shows a greater distance to the negative ideal point (0.237 vs 0.142)

According to the OWA-TOPSIS methodology, the higher the C' value, the more desirable the alternative being evaluated. In this case, Microsoft Azure emerges as the most suitable platform for implementing machine learning in DevOps environments.

Detailed analysis by criteria reveals that:

1. **Scalability (C1)**: AWS scored slightly higher (S_5.67) than Azure (S_5.33), demonstrating a perceived greater ability to adapt to different workloads.
2. **Integration with DevOps tools (C2)**: Azure significantly outperformed AWS (S_6.00) over AWS (S_4.67), representing a considerable advantage in terms of ease of integration with CI/CD pipelines and other DevOps tools.
3. **Performance (C3)**: AWS was rated slightly better (S_5.67) than Azure (S_5.33) in terms of speed and efficiency in data processing.
4. **Cost-effectiveness (C4):** Azure scored significantly higher (S_5.33) than AWS (S_4.33), suggesting a better perception of the relationship between investment and results obtained.

A comparative analysis of machine learning platforms for optimizing DevOps environments, using the neutrosophic OWA-TOPSIS model, has determined that Microsoft Azure represents the most suitable alternative for this purpose. This conclusion is based on a comprehensive evaluation of relevant criteria such as scalability, integration, performance, and cost-effectiveness.

While AWS demonstrates strengths in scalability and performance, Azure significantly excels in crucial aspects for DevOps environments, particularly in integration with DevOps tools and cost-effectiveness. These factors were decisive in the final evaluation using the OWA-TOPSIS model.

It is important to note that this analysis was conducted in a specific context and with specific criteria. The choice of the optimal platform may vary depending on the specific requirements of each organization and project. Therefore, it is recommended that each implementation consider its specific needs before selecting the most appropriate platform.

This study demonstrates the utility of the neutrosophic OWA-TOPSIS model for decision-making in multi-criteria contexts where there is uncertainty in the evaluations, such as the selection of technology platforms for DevOps environments.

5. Conclusion

Analysis using the neutrosophic OWA-TOPSIS model reveals that Microsoft Azure stands out as the most suitable platform for implementing machine learning in DevOps environments, outperforming AWS in the overall evaluation. This finding is based on Azure's closer distance from the positive ideal point and its greater separation from the negative ideal point, reflecting superior performance across key criteria. The methodology employed allowed for the integration of complex assessments, capturing nuances that conventional approaches might overlook. The practical relevance of these results is significant for organizations seeking to optimize their DevOps processes. Azure's advantage in integration with CI/CD tools and its favorable cost-benefit ratio offer clear guidance for practitioners and decision-makers. These strengths can translate into more efficient workflows, shorter implementation times, and better resource allocation in machine learning projects.

This study offers a notable innovation by introducing the neutrosophic OWA-TOPSIS model as a robust tool for evaluating technological platforms in contexts of high uncertainty. By combining criteria such as scalability, performance, integration, and cost, the research not only enriches theoretical knowledge on multi-criteria decision-making but also provides a practical framework for selecting technological solutions tailored to specific needs. However, the study faces certain limitations that should be considered. The evaluation was based on a specific set of criteria and a particular context, which could restrict the applicability of the results to other scenarios. Furthermore, the inherent subjectivity of expert assessments introduces variability that could influence the generalizability of the conclusions. For future research, we suggest exploring complementary approaches, such as artificial intelligence techniques or fuzzy methods, which could further refine the accuracy of the assessments. Expanding the scope of the study to different organizational contexts and additional criteria would allow for the validation and strengthening of the findings. Furthermore, we recommend that organizations conduct customized assessments, considering their specific priorities and requirements, to ensure the selection of the most appropriate platform.

References

[1] M. Gokarna and R. Singh, "DevOps: A Historical Review and Future Works," in Proc. International Conference on Computing, Communication and Intelligent Systems (ICCCIS), Greater Noida, India, 2021, pp. 366–371, doi: 10.1109/ICCCIS51004.2021.9397163.

[2] H. F. A. Hany, K. Wassif, and H. Moussa, "Framework for Automatic Anomaly Detection in DevOps," Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 1, pp. 8–19, Jan. 2023, doi: 10.1016/j.jksuci.2023.02.010.

[3] Z. Wang, M. Shi, and C. Li, "Research and Design of Intelligent DevOps Platforms Based on Machine Learning," in Proc. 8th International Conference on Advanced Cloud and Big Data (CBD), Taiyuan, China, 2020, pp. 42–47, doi: 10.1109/CBD51900.2020.00019.

[4] R. Kallis, A. Di Sorbo, G. Canfora, and S. Panichella, "Predicting Issue Types on GitHub," Science of Computer Programming, vol. 205, Mar. 2021, Art. no. 102598, doi: 10.1016/j.scico.2020.102598.

[5] L. Leite, C. Rocha, F. Kon, D. Milojicic, and P. Meirelles, "A Survey of DevOps Concepts and Challenges," ACM Computing Surveys, vol. 52, no. 6, pp. 1–35, Jan. 2020, doi: 10.1145/3359981.

[6] T. Mboweni, T. Masombuka, and C. Dongmo, "A Systematic Review of DevOps in Machine Learning," in Proc. International Conference on Electrical, Computer and Energy Technologies (ICECET), Prague, Czech Republic, 2022, pp. 1–6, doi: 10.1109/ICECET55527.2022.9873465.

[7] M. A. Q. Martinez, D. T. L. Rugel, C. J. E. Alcivar, and M. Y. L. Vazquez, "A Framework for Selecting Classification Models in the Intruder Detection System Using TOPSIS," in Human Interaction, Emerging Technologies and Future Applications III: Proceedings of the 3rd International Conference on Human Interaction and Emerging Technologies: Future Applications (IHIET 2020), August 27-29, 2020, Paris, France. Springer International Publishing, 2021, pp. 173–179.

[8] A. Wiedemann, M. Wiesche, H. Gewald, and H. Krcmar, "Understanding How DevOps Aligns Development and Operations: A Tripartite Model of Intra-IT Alignment," European Journal of Information Systems, vol. 29, no. 5, pp. 458–473, Sep. 2020, doi: 10.1080/0960085X.2020.1782277.

[9] M. H. Tanzil, M. Sarker, G. Uddin, and A. Iqbal, "A Mixed-Methods Study of DevOps Challenges," Information and Software Technology, vol. 161, Sep. 2023, Art. no. 107244, doi: 10.1016/j.infsof.2023.107244.

[10] A. Rezazadeh, "A Generalized Flow for B2B Sales Predictive Modeling: An Azure Machine-Learning Approach," Forecasting, vol. 2, no. 3, pp. 267–283, Sep. 2020, doi: 10.3390/forecast2030015.

[11] Hudgeon, D., & Nichol, R. (2019). Machine learning for business: Using amazon sagemaker and jupyter. Simon and Schuster.

[12] S. Alla and S. K. Adari, Beginning MLOps with MLFlow: Deploy Models in AWS SageMaker, Google Cloud, and Microsoft Azure. Berkeley, CA, USA: Apress, 2021.

[13] M. A. Kamal, H. W. Raza, et al., "Highlighting the Features of AWS, GCP and Microsoft Azure That Influence the Choice of a Cloud Service Provider," International Journal of Recent Technology and Engineering, vol. 8, no. 5, pp. 4124–4132, Jan. 2020, doi: 10.35940/ijrte.D8573.018520.

[14] K. Priyadarsini, E. F. I. Raj, A. Y. Begum, and V. Shanmugasundaram, "WITHDRAWN: Comparing DevOps Procedures from the Context of a Systems Engineer," Materials Today: Proceedings, 2020, doi: 10.1016/j.matpr.2020.09.624.

[15] A. Shivanna and D. P. Agrawal, "Defaulter Prediction Using Machine Learning on Azure ML," in Proc. 11th Annual International Conference on Information, Electronics and Mobile Communication Technology (IEMCON), Vancouver, BC, Canada, 2020, pp. 0320–0325, doi: 10.1109/IEMCON51383.2020.9284936.

[16] S. Srithar, E. Vetrimani, V. Vignesh, et al., "Cost-Effective Integration and Deployment of Enterprise Applications Using Azure Cloud DevOps," in Proc. International Conference on Communication and Computing (ICCCI), Coimbatore, India, 2022, pp. 01–05, doi: 10.1109/ICCCI54379.2022.9740876.

[17] Dauletbak, D., & Woo, J. (2020). Big data analysis and prediction of traffic in los angeles. KSII Transactions on Internet and Information Systems (TIIS), 14(2), 841-854.

[18] D. Kim, J. Kim, and H. Kim, "Automatic Disaster Alert System with Amazon Web Service (AWS): Focus on Floods in East Asia," in Proc. IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Kuala Lumpur, Malaysia, 2022, pp. 4335–4338, doi: 10.1109/IGARSS46834.2022.9884309.

[19] A. Gupta, N. Dhanda, and K. K. Gupta, "CSV File Ingestion and Visualization with AWS Platform for Unstructured to Structured Data Transition," in Proc. 11th International Conference on Emerging Trends in Engineering and Technology - Signal and Information Processing (ICETET-SIP), 1 Nagpur, India, 2023, pp. 1–6, doi: 10.1109/ICETET-SIP58143.2023.10151476.

[20] R. Ramesh, "Predictive Analytics on Banking Users' Data Through AWS Machine Learning Cloud Service," in Proc. 2nd International Conference on Computer and Communication Technology (ICCCT), Chennai, India, 2017, pp. 210–215, doi: 10.1109/ICCCT2.2017.7972267.

[21] S. X. S. Muñoz and M. A. Q. Martinez, "A Framework for Selecting Machine Learning Models Using TOPSIS," in Advances in Artificial Intelligence, Software and Systems Engineering: Proceedings of the AHFE 2020 Virtual Conferences on Software and Systems Engineering, and Artificial Intelligence and Social Computing, July 16-20, 2020, USA. , 2020, pp. 119.

[22] M. Á. Quiroz Martínez, G. A. Granda Villon, D. I. Maldonado Cevallos, and M. Y. Leyva Vázquez, "Comparative Analysis to Select an Emotion Recognition Tool Applying Fuzzy Decision Maps and TOPSIS," Dilemas Contemporáneos: Educación, Política y Valores, vol. 8, no. 1, Oct. 2020, doi: 10.46377/dilemas.v8i1.2441.

[23] H. Wang, F. Smarandache, Y. Q. Zhang, and R. Sunderraman, "Single Valued Neutrosophic Sets," Multispace and Multistructure, vol. 4, pp. 410–413, 2010.

[24] M. Leyva-Vázquez, E. Santos-Baquerizo, M. Peña-González, L. Cevallos-Torres, and A. Guijarro-Rodríguez, "The Extended Hierarchical Linguistic Model in Fuzzy Cognitive Maps," in Technologies and Innovation: Second International Conference, CITI 2016, Guayaquil, Ecuador, November 23-25, 2016, Proceedings 2. Springer International Publishing, 2016, pp. 39–50.

[25] E. Zhang, F. Chen, and S. Zeng, "Integrated Weighted Distance Measure for Single-Valued Neutrosophic Language Sets and Its Application in Supplier Selection," Journal of Mathematics, vol. 2020, Art. ID 6468721, 10 pages, 2020. Available: https://www.hindawi.com/journals/jmath/2020/6468721/.

[26] J. J. Geng, H. Y. Wanhong, and D. S. Xu, "A TOPSIS-Based and Distance Measure Method for Single-Valued Neutrosophic Linguistic Sets and Its Application," IAENG International Journal of Applied Mathematics, vol. 51, no. 3, pp. 1–8, 2021. Available: http://www.iaeng.org/IJAM/issues_v51/issue_3/IJAM_51_3_11.pdf.

[27] A. Metwaly and I. Elhenawy, "Predictive Intelligence Technique for Short-Term Load Forecasting in Sustainable Energy Networks," Sustainable Machine Intelligence Journal, vol. 5, no. 4, pp. 1–7, 2023, doi: 10.61185/SMIJ.2023.55104.

[28] S. Xian, W. Sun, S. Xu, and Y. Gao, "Fuzzy Linguistics-Induced OWA Minkowski Distance Operator and Its Application in Group Decision Making," Pattern Analysis and Applications, vol. 19, pp. 325–335, 2016. Available: https://link.springer.com/article/10.1007/s10044-014-0397-3.

Received: December 26, 2024. Accepted: April 7, 2025.