ARTICLE

Cite This: J. of Artfic. Intell. and Machine. Lear. 2024, 2-2, 1-12

SciForce

Journal of Artificial intelligence and Machine Learning

Journal homepage: www.sciforce.org

The Secure Computation Application Programming Interface Using The ARAS Method

Tejasvi Gorre[1*]

*[*1]Home Depot Management Company LLC, 2455 Paces Ferry Rd SE, Atlanta*

ARTICLE INFO

Article history:
Received : 20240615
Received in revised form : 20240619
Accepted: 20240619
Available online : 20240710

Keywords:
MCDM;
Technology;
Environment.

ABSTRACT

Abstract: An Application Programming Interface (API) entails guidelines, Principles and an array of utilities, it stands as distinct software Applications interacting with one another communication enables A service tailored for developers, of the library or website Intrinsic operations without the need for comprehension in connection with that correspondence Approaches that can be employed and structures of data APIs establish.service or platform. Abstraction Layer: An API acts as an abstraction layer that separates the implementation details of a software component from its usage. It provides a standardized way for developers to access the functionality of another software component or service. Interoperability: The significance of application programming interfaces (APIs) in research is profound and multifaceted, impacting various fields and disciplines. APIs serve as crucial tools that enable researchers to interact with and harness the capabilities of existing software, platforms, and services. Here's why APIs are significant in research: Efficiency and Productivity: APIs allow researchers to access complex functionalities without having to reinvent the wheel. By integrating APIs, researchers can save time and effort, focusing more on the core research objectives. Data Access and Analysis: APIs provide access to vast amounts of data from different sources. Researchers can gather, analyze, and synthesize data from diverse platforms, expanding the scope and depth of their research. Interdisciplinary Collaboration: APIs facilitate collaboration across disciplines. Researchers from different domains can utilize APIs to combine tools and data, fostering interdisciplinary studies that address complex problems. Innovation: APIs encourage innovation by enabling researchers to build upon existing technologies. By integrating APIs creatively, researchers can develop novel solutions that were not feasible before. Replicability and Transparency: APIs enhance research transparency by allowing others to reproduce research procedures easily. The use of APIs ensures that methodologies and data processing steps can be replicated accurately. Customization: APIs enable researchers to customize tools and platforms according to their specific research needs. This flexibility empowers researchers to tailor solutions to their unique requirements.Automation: APIs facilitate the automation of repetitive tasks and data collection, reducing human errors and enhancing the reliability of research outcomes. Real-time Data: Many APIs provide real-time data updates. Researchers can access and analyze current information, making their studies more relevant and timelier. Experimentation and Prototyping: APIs enable researchers to quickly prototype ideas and experiment with different functionalities. This rapid iteration helps in refining research methodologies. Cross-platform Integration: APIs bridge the gap between different software and platforms, enabling seamless integration between tools that might not natively work together. Cost-effectiveness: Instead of building tools from scratch, researchers can utilize APIs, which often offer cost-effective solutions for specific research needs. Education and Training: APIs play a role in educating the next generation of researchers and programmers. Learning to work with APIs exposes students to real-world coding practices and software development concepts. The ARAS method for complex decision problems Trying to simplify and appropriate indicator (degree of

application) "excellent" Through alternate exams It is in between alternative and the best solution Reflects difference and is different Eliminates the influence of units of measurement. ARAS technique might be used. A regular MCDM trouble is related to the project of Limited variety of results Ranking the options, each of them Based on various selection criteria are clearly described, in line with the ARAS method, decide a application characteristic fee. The relative effectiveness of the complexity of the viable opportunity. Google Maps API, Twitter API, Stripe API, Spotify API, OpenWeatherMap API and Twilio API.Documentation Quality, Ease of Integration, Functionality, Performance, Community and Support and Security. the Rank Application Programming Interface using the analysis of Addition Ratio Assessment (ARAS) Method. Twilio API is showing the highest value of rank whereas Twitter API is showing the lowest value.

∗Corresponding author. e-mail: **tejasvigr@gmail.com**

Introduction

An Application Programming Interface (API) entails guidelines, Principles and an array of utilities, it stands as distinct software Applications interacting with one another communication enables A service tailored for developers, of the library or website Intrinsic operations without the need for comprehension in connection with that correspondence Approaches that can be employed and structures of data APIs establishservice or platform.Abstraction Layer: An API acts as an abstraction layer that separates the implementation details of a software component from its usage. It provides a standardized way for developers to access the functionality of another software component or service.Interoperability: APIs enable different software systems to work together [1]. If you want your application to utilize the capabilities of another service or system, you don't need to know how that service is implemented; you just need to interact with its API according to its specifications.Methods and Endpoints: APIs offer a collection of methods (functions) that define the actions you can perform on a service or data. These methods are often exposed through specific endpoints, which are URLs that correspond to different actions or resources.Data Exchange Formats: APIs typically use specific data exchange formats, such as JSON (JavaScript Object Notation) or XML (eXtensible Markup Language), to structure the data being sent and received between applications [2].Standardization: APIs provide a standardized way of interacting with a service, making it easier for developers to understand how to use it. This consistency reduces the learning curve and allows for quicker development.Security: APIs often include authentication and authorization mechanisms to ensure that only authorized users or applications can access the provided services. This helps maintain security and control over data and functionality.Updates and Compatibility: APIs can change over time to add new features or improve existing ones. However, APIs are designed to maintain backward compatibility as much as possible, ensuring that applications built using an older version of the API can still function even if the API is updated [3].Ecosystem and Innovation: APIs enable developers to build on top of existing services, libraries, or platforms, fostering innovation by allowing new applications to leverage existing resources. This promotes the development of a broader software ecosystem.Web APIs: These are APIs that allow applications to interact with each other over the internet. For example, social media platforms provide APIs that allow developers to integrate social features into their applications.Library APIs: Programming libraries often come with APIs that provide a set of functions and classes to perform specific tasks. Developers can use these APIs to streamline their development process.Operating System APIs: OS APIs provide a way for applications to interact with the underlying operating system, accessing features like file management, networking, and hardware resources [4].(i) CNA User Interface: Grants entry to CNA's network models Permits content access Available through GUI(ii) CNA Network Analysis: - Explores capabilities of CNA's network analysis - Accessible from the graphical user interface (GUI) - Can be used independently (iii) Graphical Development and Interaction: - Integrates with the GUI for user-friendly interaction - Supports plug-ins for additional functionality CNA's Network Functionality: Organizes projects within CAN Offers novel API functions for these programs Application Scope: Describes using the API functions in applications, developing networks from scratch, Crafting plans and scenariosLoading and saving projects Performing real analyses Applying various methods Encompasses further features and functionalities [5].A fresh API function (GUI not required) is available in the composer, along with input masks. These masks aid in using the Calculation of CNA without complications. API functions enable network utilization, and temporary project variables are employed when necessary, proving their utility. In Section 3 (as seen in the example), CAN will also evolve alongside Network projects, forming the MF network. Additionally, CAN will generate the SF network,

which can be saved on the disk. If needed, these networks can be stored on the disk. Subsequently, they can be reloaded using the GUI or without it, and might be expanded to encompass various projects. Numerous CNA programs are stored in a specific ASCII format through special files. Remember that these can be loaded into the CAN network. It's possible to retrieve support for API functions MF, acquiring both the network and SF Network for reverse activity generation [6].An Application Programming Interface (API) essentially serves as a means to fulfill the information needs of software developers who rely on its documentation. These information requirements are extensive and must be met to facilitate software development. Existing research implies that these requirements are complex and not entirely comprehended. This article employs a semi-structured approach, involving a series of interviews and a subsequent questionnaire to explore strategies and information resources that software developers employ for API documentation [7]. The obtained results indicate that developers aim to comprehend the overall purpose and key global features of an API initially. However, their approach varies based on their opinions and whether they prefer a concept-driven or code-driven strategy. Both strategies highlight the significance of API documentation as a learning tool. Our findings emphasize the importance of criteria like completeness and clarity in API documentation quality. Furthermore, the research underscores the necessity of involving expert expertise in the creation and maintenance of API documentation, as this process involves communication and knowledge transfer [8].Application Programming Interfaces (APIs) are tools that offer services and enable data access or disclosure, including predefined methods and resources. These APIs facilitate interaction with software applications using objects or URIs. These functionalities were highlighted by Stylos, Faulring, Yang, and Myers in 2009. APIs eliminate the need for creating new applications, allowing data access and services through basic objects and implemented procedures. APIs are crucial in modern software, simplifying programming tasks and supporting distributed, modular design. Myers and Stylos (2016) suggested that most modern software relies heavily on APIs instead of being built from scratch [9]. APIs have become central, replacing functions and becoming a common task for software developers, often referred to as "stitching together." Comprehensive documentation, such as Programmer's Guides, cookbooks, exercises, and API reference documents, assists developers in learning and using APIs. The effectiveness of early learning efforts for API usage is a broad question that requires investigation. Some APIs are more complex than others, influenced by domain coverage, intrinsic properties, and specific formatting features. Research has aimed to identify features that facilitate easy learning and usage of APIs [10].Developers' experience in matching their expectations to API functionality plays a crucial role in effective API utilization. Apart from features, the growing interest in API documentation underscores its impact on usability and task-solving efficiency. Literature findings emphasize API documentation's role in learning and using APIs effectively. Interviews and questionnaire surveys conducted among software developers indicate a significant scarcity of documentation resources as a challenge in learning new APIs [11].Imaging Spectroscopy is a potent method for mapping the Earth's surface remotely.It has proven successful across various environments, encompassing managed and unmanaged forests, agricultural areas, urban zones, mining sites, deserts, snowy regions, and aquatic systems.The technique involves utilizing data with a broad range of applications, employing various analytical approaches such as mapping, observation, empirical modelling, and physical modelling, often utilizing radiative transfer models.Its utility becomes evident in applications where additional spectral information enhances results with fewer data points, leading to improved spectral dimensions [12].Germany initiated the Environmental Mapping and Analysis Program (EnMAP) in 2018, a spaceborne imaging spectroscopy mission that offers a wealth of imaging spectroscopy data for multiple contexts and applications.Efforts are underway, including research projects and EnMAP Summer Schools for doctoral researchers, to unlock the full potential of EnMAP data across diverse fields of application.The NMap-Box, a toolbox for processing imaging spectrometer data, has been developed since 2010, aiming to improve algorithms and make them accessible to the wider community.The EnMAP Toolbox, distributed freely with data, serves three main objectives: (i) enabling data analysis with common software, adhering to standards and providing extension functions; (ii) offering powerful approaches for advanced image processing; and (iii) enhancing accessibility through interfaces and scripting languages, contributing to a rapidly growing resource pool [13].OMPT functions as a performance tool tailored for multiple interface designs, aiming to achieve specific objectives in the context of OpenMP. These objectives are listed in descending order of significance. The OpenMP API for runtime system control requires tools to gather adequate information about the functioning of applications running under it. Both the application source code and runtime system contribute to the overall cost. The API offers low overhead and necessitates a suitable interface for an asynchronous model that aligns with performance tools. Consistency between API calls and procedures within OpenMP is crucial, which necessitates identifying frames in the stack and utilizing unwinding for profiling [14].The OpenMP runtime system connects a thread's state with its function, offering insights into performance by enabling the activation of tools during program execution. Some API routines define synchronization signals, ensuring safety through the implementation of asynchronous events that are processed by specified handlers. OpenMP runtime must support the API to establish connections with the system, but this interface should add minimal overhead when not used by a tool. Support for trace-based performance tools must be outlined by the API. Integrating the API within OpenMP implementation should not disproportionately burden runtime implementers. While developing the tool, careful consideration is essential to prevent unfair development and undue burdens on implementers [15].

1. MATERIALS AND METHOD

2.1. Google Maps API:An interface provided Developers possess their own Google Maps within applications feature Enable incorporation Through Google. Google Maps the API is Supported by Google serves as a programming interface, It's accessible to developers enables incorporate the features and functionality of Google Maps into their applications. This API enables developers to integrate maps, location data, geocoding, routing, and other location-based services seamlessly into their websites or applications.

2.2. Twitter API:An interface provided by Twitter that enables the Twitter developers Data and functionalities Programmatically attain access Get in touch. The Twitter API is associated with Twitter Offering resources and constitutes a collection of approaches, this facilitates interaction for developers Permitting entry to. Twitter's platform programmatically. This API enables developers to create applications that can retrieve tweets, post tweets, search for specific content, access user profiles, and perform various actions on the Twitter platform using code. It facilitates the integration of Twitter's data and features into third-party applications and services.

2.3. Stripe API:An interface offered by Stripe, a payment processing company, which developers can use to integrate payment processing and e-commerce functionality into their applications. The Stripe API is an interface provided by Stripe, a payment processing company, which enables developers to integrate online payment processing functionalities into their applications and websites. This API allows developers to programmatically manage payments, handle subscriptions, process transactions, and create customized e-commerce experiences. It provides tools for securely handling payment information and streamlining the payment process for businesses operating online.

2.4. Spotify API:An interface provided by Spotify that allows developers to access and incorporate Spotify's music streaming services and features into their applications. The Spotify API is a set of tools and methods provided by Spotify, a popular music streaming service, that allows developers to access and incorporate Spotify's music catalog and streaming features into their applications. With this API, developers can build applications that retrieve music metadata, play tracks, create playlists, access user libraries, and interact with various aspects of the Spotify platform programmatically. This enables the integration of Spotify's music content and functionality into third-party apps and services.

2.5. OpenWeatherMap API:An interface provided by OpenWeatherMap that offers developers access to weather data and forecasts, allowing them to integrate weather information into their applications.The OpenWeatherMap API is a programming interface offered by OpenWeatherMap, a weather data provider. This API allows developers to access weather information and forecasts programmatically, enabling them to integrate current weather conditions, forecasts, and historical weather data into their applications or websites. By utilizing this API, developers can or preferences.

2.6. Twilio API:An interface provided by Twilio that enables developers to integrate communication capabilities like SMS, voice calls, and more into their applications programmatically. The Twilio API is a set of tools and methods provided by Twilio, a communication platform, that empowers developers to integrate various communication functionalities into their applications programmatically. This API enables developers to send and receive SMS messages, make and receive phone calls, handle video and chat interactions, and manage communication workflows seamlessly within their applications. By utilizing the Twilio API, developers can create interactive and dynamic communication experiences for users, enhancing the way applications engage with their audience.

2.7. Documentation Quality:The clarity, completeness, and accessibility of the provided documentation play a pivotal role in how effectively developers can understand and implement the API's features. Well-organized and comprehensive documentation enhances the developer experience.

2.8. Ease of Integration:The ease with which the API can be incorporated into existing applications is crucial. A well-designed API should have clear guidelines, straightforward endpoints, and logical workflows, simplifying the integration process and reducing the time developers need to get up and running.

2.9. Functionality:The range and depth of features the API offers are essential. Developers need access to a variety of functionalities that align with their application's requirements. A robust and versatile API supports a broader array of use cases.

2.10. Performance:The API's speed and responsiveness are critical for delivering a seamless user experience. Low latency and efficient data processing contribute to the overall performance of the applications that rely on the API.

2.11. Community and Support:A strong developer community and reliable support resources, such as forums, documentation updates, and responsive customer service, provide a safety net for developers facing challenges or seeking guidance during their API implementation journey.

2.12. **Security:** Security mechanisms and practices integrated into the API are vital to protect sensitive data and ensure secure transactions. Strong authentication, encryption, and adherence to industry security standards help establish trust and safeguard user information.

2.13. Method:Complex decision-making involves the process of making choices in situations that are intricate and multifaceted. In the context of handling complications, the ARAS (Additive Ratio Assessment) method simplifies this process. It's both straightforward and suitable for assessing various options.An indicator is used to gauge the degree of utility, where the aim is

to select alternatives that are consistently "better" than the others. This represents an improved and more favourable solution among choices. [16]. The ARAS method finds applicability as a technique. It fits well with a common MCDM issue there's a restricted range having an outcome strategy associated, the choices arranging, they exist each one stands as a distinct selection built on standards are explicitly outlined, following the ARAS approach, the cost for usage establish. Feasible evaluative effectiveness of the random issue a concurrent undertaking [17]. Within transportation firms' evaluation of overall rates(ARAS) quantified metrics for performance evaluation method utilized. a total of 20 performance aspects derived from markers evaluation conductedAttained internally. the acquired outcomes are responsive structured in a three-tier evaluation frameworkImplemented in this fashion [18]. The ARAS methodology,essentially intricate global occurrencesbased on uncomplicated factors grasping the practical implementation the assertion that is capable of

falls within considerationdetails the potential,adjusted and values assigned to weighted amounts extremely advantageous prospects asserted to be all-encompassing alteration in evaluation [19]. Renewable energy sources

such as polysilicon-based systemsPhotovoltaic solar energy, Solid OxideFuel cells,Phosphoric acid fuel cells,

and the utilization of ocean breezesfor energy generationare evaluated by energy specialistswith contributions

from sustainability measureswith a focus on the hybrid ARAS technique.Economic aspects, Regulatory frameworks,corporate structures, production methods,shape and composition,of regulationsPrimarily through implementation of the ARAS methodare harmonized the suggested approach is enhanced. [20]. Aras valley, Moderate winter temperatures without extreme cold,under cultivation within the valley the majority of fruits belonging to the rosaceae family and the valley's ecosystem encompassing, in its character given the circumstances, Indigenous apricot varietiesoffering various options to humans Instances are typically people obtaining diminutive fruits are infrequent. Top-quality indigenous apricots bushes yield sparingly, and pests and vulnerability to illnesses [21]. Utilizing graynumbers, the ARAS approach. ARAS classical decision-making within from a technical perspectivediffering from, to address MCDM issues as a novel method, choices in this

operate with quantities, at the outset of the procedureof the test producer in contrast to feature expenses, an improved option is identified. this uncertainty entails prudent judgment and with the gray notion ash can be amalgamated incorporating Gray Assessment (ARAS-G)signifies that ARAS integrates gray merging texture and hueconstitutes a technical principle. although recent in ARAS methodology literature, it finds application in numerous domains widely across various fields,employed in numerous inquiries [22].Flash-lamp photolysis, ARAS 1.9, a hundred combined evaluations, of the photosynthetic process, initially following commencement, A testing duration of 150 ps, related to the flash mechanism, Resultant from the flash occurrence, Fluctuations internal to rendered it nonfunctional. In ongoing experiments, Because of the excimer flash's effects, Photo-multiplier tube (PMT) intensity is selectively filtered Extracted via inclusion, and every stimulus, Optical in nature, Separators and the excimer laser stands as an ideal choice for safeguarding purposes through its implementation, Eliminating electronic disruption. [23]. Uncertainty and Personal assessments and/or established truths and/or Insufficient data coping abilities Inapplicable to ARAS. Definitive sound reasoning Benefit derived from utilizing in essence, it signifies Indeterminacy Takes into account unfamiliar and Complicated scenarios in effectively dealing with this technique A valuable notion. [24]. ARAS Method, arranging alternatives and/or for assessing Specific facts are using occurrences. Hence, this methodology Through utilization Those who decide have their own Optimistic, pessimistic, and Rational strategies Conveying. It is, in written form, a numerical instance The process of e-learning route selection appears to be the rationale behind this type of proficiency holds significance. An exceptional electronic cognition in guiding attention, the advantage of a downward path Alongside flaws and opposition Its position is compared Reaching a decision is imperative. In that context, naturally, Which components require enhancement? And which one holds more is deemed adequate the creators of that Have realized. Suggested by means of the integrated approach in this scenario, software Highly cost-effective and evaluated as suitable [25].

Result and Discussion

Table 1. Application Programming Interface

	Documentation Quality	**Ease of Integration**	**Functionality**	**Performance**	**Community and Support**	**Security**
Max or Min	**680**	**315**	**1200**	**760**	**1580**	**0.119**
Google Maps API	220	220	460	360	880	0.342

Twitter API	200	200	330	100	380	0.171
Stripe API	270	250	630	435	590	0.119
Spotify API	270	270	670	540	1190	1.283
OpenWeatherMap API	585	240	1100	680	1580	3.128
Twilio API	680	315	1200	760	1250	4.732

Table 1 shows theApplication Programming Interface in Evaluation parameters Documentation Quality, Ease of Integration, Functionality, Performance, Community and Support and Security.And Alternative ParametersGoogle Maps API, Twitter API, Stripe API, Spotify API, OpenWeatherMap API and Twilio APIGoogle Maps API in Documentation Quality: 220, Ease of Integration: 220, Functionality: 460, Performance: 360, Community and Support: 880 and Security: 0.342. Twitter API in Documentation Quality: 200, Ease of Integration: 200, Functionality: 330, Performance: 100, Community and Support: 380 and Security: 0.171. Stripe API in Documentation Quality: 270, Ease of Integration: 250, Functionality: 630, Performance: 435, Community and Support: 590 and Security: 0.119. Spotify API in Documentation Quality: 270, Ease of Integration: 270, Functionality: 670, Performance: 540, Community and Support: 1190 and Security: 1.283. OpenWeatherMap API in Documentation Quality: 585, Ease of Integration: 240, Functionality: 1100, Performance: 680, Community and Support: 1580 and Security: 3.128. Twilio API in Documentation Quality: 680, Ease of Integration: 315, Functionality: 1200, Performance: 760Community and Support: 1250 and Security: 4.732it is also used in Maximum and Minimum Value.

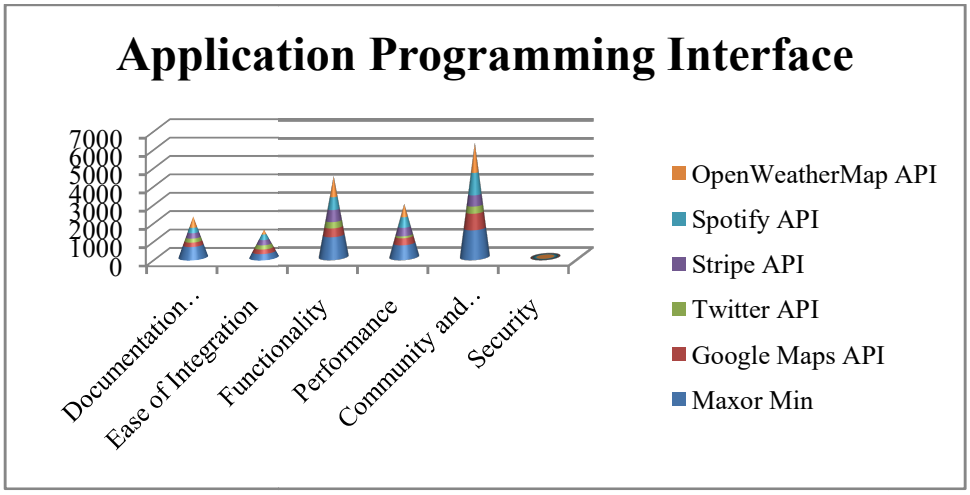

Figure 1. Application Programming Interface

Figure 1 shows the Application Programming Interface in Evaluation parameters Documentation Quality, Ease of Integration, Functionality, Performance, Community and Support and Security.And Alternative ParametersGoogle Maps API, Twitter API, Stripe API, Spotify API, OpenWeatherMap API and Twilio API Google Maps API in Documentation Quality: 220, Ease of Integration: 220, Functionality: 460, Performance: 360, Community and Support: 880 and Security: 0.342. Twitter API in Documentation Quality: 200, Ease of Integration: 200, Functionality: 330, Performance: 100, Community and Support: 380 and Security: 0.171. Stripe API in Documentation Quality: 270, Ease of Integration: 250, Functionality: 630, Performance: 435, Community and Support: 590 and Security: 0.119. Spotify API in Documentation Quality: 270, Ease of Integration: 270, Functionality: 670, Performance: 540, Community and Support: 1190 and Security: 1.283. OpenWeatherMap API in Documentation Quality: 585, Ease of Integration: 240, Functionality: 1100, Performance: 680, Community and Support: 1580 and Security: 3.128. Twilio API in Documentation Quality: 680, Ease of Integration: 315, Functionality: 1200, Performance: 760 Community and Support: 1250 and Security: 4.732 it is also used in Maximum and Minimum Value.

Table 2. Application Programming Interface Maximum & Divided Value						
	Documentation Quality	**Ease of Integration**	**Functionality**	**Performance**	**Community and Support**	**Security**
Maxor Min	**680**	**315**	**1200**	**760**	**1580**	8.40336134
Google Maps API	220	220	460	360	880	2.92397661
Twitter API	200	200	330	100	380	5.84795322
Stripe API	270	250	630	435	590	8.40336134
Spotify API	270	270	670	540	1190	0.77942323
OpenWeatherMap API	585	240	1100	680	1580	0.31969309
Twilio API	680	315	1200	760	1250	0.21132713

Table 2 shows theApplication Programming Interface Maximum & Divided Valueof Evaluation parameters Documentation Quality, Ease of Integration, Functionality, Performance, Community and Support and Security.And Alternative ParametersGoogle Maps API, Twitter API, Stripe API, Spotify API, OpenWeatherMap API and Twilio APIit is also used in Maximum & Divided Value

Table 3. Normalization of DM						
	Documentation Quality	**Ease of Integration**	**Functionality**	**Performance**	**Community and Support**	**Security**
Maxor Min	**0.234079**	**0.174033**	**0.214669**	**0.209078**	**0.212081**	**0.312519**
Google Maps API	0.075731	0.121547	0.08229	0.099037	0.118121	0.108742
Twitter API	0.068847	0.110497	0.059034	0.02751	0.051007	0.217484
Stripe API	0.092943	0.138122	0.112701	0.11967	0.079195	0.312519
Spotify API	0.092943	0.149171	0.119857	0.148556	0.159732	0.028987
OpenWeatherMap API	0.201377	0.132597	0.19678	0.18707	0.212081	0.011889
Twilio API	0.234079	0.174033	0.214669	0.209078	0.167785	0.007859

Table 3 shows theApplication Programming Interface Normalization of DMEvaluation parameters Documentation Quality, Ease of Integration, Functionality, Performance, Community and Support and Security. And Alternative Parameters Google Maps API, Twitter API, Stripe API, Spotify API, OpenWeatherMap API and Twilio API it is also used in Normalization of DM Value.

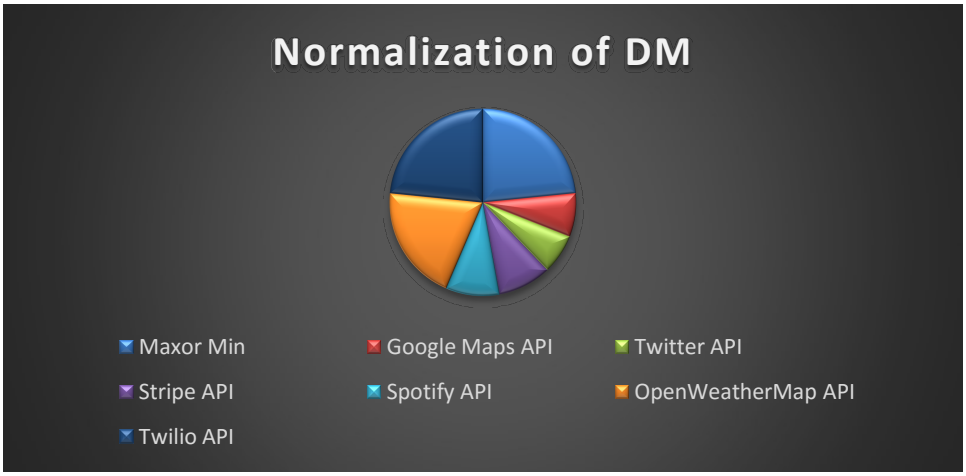

Figure 2. Normalization of DM

Figure 2 shows theApplication Programming Interface Normalization of DMEvaluation parameters Documentation Quality, Ease of Integration, Functionality, Performance, Community and Support and Security. And Alternative Parameters Google Maps API, Twitter API, Stripe API, Spotify API, OpenWeatherMap API and Twilio API it is also used in Normalization of DM Value.

Table 4. Weighted Normalized DM

	0.21	**0.18**	**0.22**	**0.15**	**0.13**	**0.11**
	Documentation Quality	**Ease of Integration**	**Functionality**	**Performance**	**Community and Support**	**Security**
Maxor Min	0.049157	0.031326	0.047227	0.031362	0.02757	0.034377
Google Maps API	0.015904	0.021878	0.018104	0.014856	0.015356	0.011962
Twitter API	0.014458	0.01989	0.012987	0.004127	0.006631	0.023923
Stripe API	0.019518	0.024862	0.024794	0.01795	0.010295	0.034377
Spotify API	0.019518	0.026851	0.026369	0.022283	0.020765	0.003189
OpenWeatherMap API	0.042289	0.023867	0.043292	0.028061	0.02757	0.001308
Twilio API	0.049157	0.031326	0.047227	0.031362	0.021812	0.000865

Table 4 shows theApplication Programming Interface Weighted Normalized DMof Evaluation parameters Documentation Quality, Ease of Integration, Functionality, Performance, Community and Support and Security. And Alternative Parameters Google Maps API, Twitter API, Stripe API, Spotify API, OpenWeatherMap API and Twilio API it is also showing theWeighted Normalized DM Value.

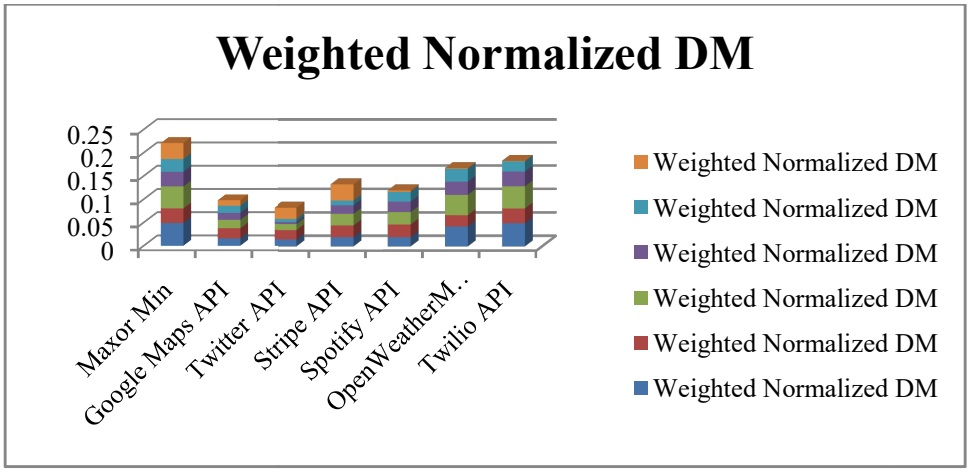

Figure 3. Weighted Normalized DM

Figure 3shows theApplication Programming Interface Weighted Normalized DMof Evaluation parameters Documentation Quality, Ease of Integration, Functionality, Performance, Community and Support and Security. And Alternative Parameters Google Maps API, Twitter API, Stripe API, Spotify API, OpenWeatherMap API and Twilio API it is also showing theWeighted Normalized DM Value.

Table 5. Final Result of Application Programming Interface SI,KI Value

	Si	**Ki**	**Rank**
Maxor Min	**0.221019**	**1**	
Google Maps API	0.098059	0.443666	5
Twitter API	0.082015	0.371079	6
Stripe API	0.131797	0.596315	3
Spotify API	0.118974	0.538299	4
OpenWeatherMap API	0.166387	0.752817	2
Twilio API	0.181748	0.822319	1

Table 5 shows the Final Result of SI, KI Value for Application Programming Interface in Additive Ratio Assessment method. And it shows the SI , KI values In SI method Twilio APIis showing the highest value and Twitter API is showing the lowest value for KI method Twilio API is showing the highest value and Twitter API is showing the lowest value

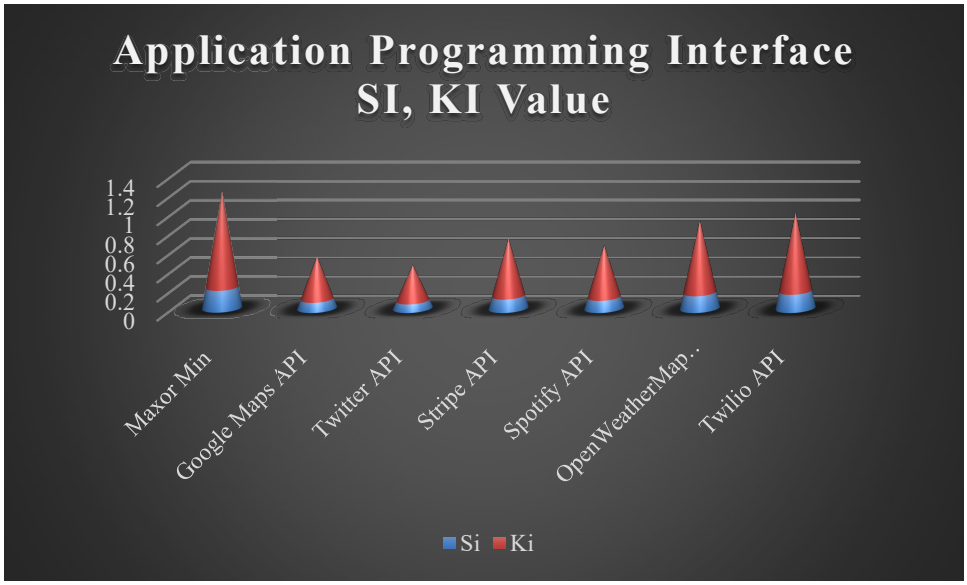

Figure 4. Application Programming Interface SI,KI Value

Figure 4 shows the Final Result of SI, KI Value for Application Programming Interface in Additive Ratio Assessment method. And it shows the SI , KI values In SI method Twilio APIis showing the highest value and Twitter API is showing the lowest value for KI method Twilio API is showing the highest value and Twitter API is showing the lowest value.

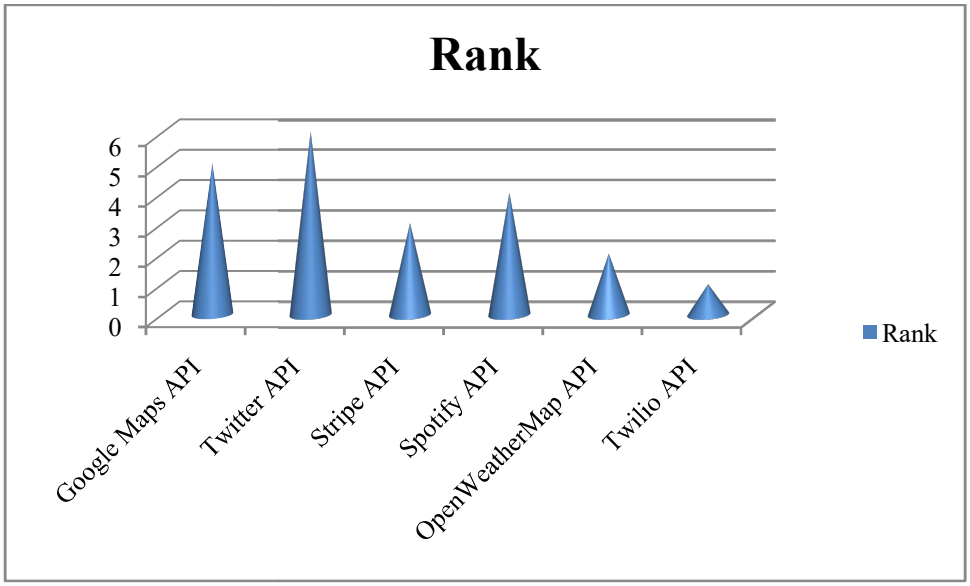

Figure 5. Shown the Rank

Figure 5 Shows the RankApplication Programming Interface using the analysis of Addition Ratio Assessment (ARAS) Method. Twilio API is showing the highest value of rank whereas Twitter API is showing the lowest value.

Conclusion

An Application Programming Interface (API) entails guidelines, Principles and an array of utilities, it stands as distinct software Applications interacting with one another communication enables A service tailored for developers, of the library or website Intrinsic operations without the need for comprehension in connection with that correspondence Approaches that can be employed and structures of data APIs establish. service or platform. Abstraction Layer: An API acts as an abstraction layer that separates the implementation details of a software component from its usage. It provides a standardized way for developers to access the functionality of another software component or service. Interoperability: The significance of

application programming interfaces (APIs) in research is profound and multifaceted, impacting various fields and disciplines. APIs serve as crucial tools that enable researchers to interact with and harness the capabilities of existing software, platforms, and services. Here's why APIs are significant in research: Efficiency and Productivity: APIs allow researchers to access complex functionalities without having to reinvent the wheel. By integrating APIs, researchers can save time and effort, focusing more on the core research objectives. Data Access and Analysis: APIs provide access to vast amounts of data from different sources. Researchers can gather, analyse, and synthesize data from diverse platforms, expanding the scope and depth of their research. Interdisciplinary Collaboration: APIs facilitate collaboration across disciplines. Researchers from different domains can utilize APIs to combine tools and data, fostering interdisciplinary studies that address complex problems. Innovation: APIs encourage innovation by enabling researchers to build upon existing technologies. By integrating APIs creatively, researchers can develop novel solutions that were not feasible before. Replicability and Transparency. Complex decision-making involves the process of making choices in situations that are intricate and multifaceted. In the context of handling complications, the ARAS (Additive Ratio Assessment) method simplifies this process. It's both straightforward and suitable for assessing various options. An indicator is used to gauge the degree of utility, where the aim is to select alternatives that are consistently "better" than the others. This represents an improved and more favourable solution among choices. the Rank Application Programming Interface using the analysis of Addition Ratio Assessment (ARAS) Method. Twilio API is showing the highest value of rank whereas Twitter API is showing the lowest value.

References

1. Klamt, Steffen, and Axel von Kamp. "An application programming interface for CellNetAnalyzer." Biosystems 105, no. 2 (2011): 162-168.
2. Meng, Michael, Stephanie Steinhardt, and Andreas Schubert. "Application programming interface documentation: What do software developers want?." Journal of Technical Writing and Communication 48, no. 3 (2018): 295-330.
3. Baek, Woongki, Chi Cao Minh, Martin Trautmann, Christos Kozyrakis, and Kunle Olukotun. "The OpenTM transactional application programming interface." In 16th International Conference on Parallel Architecture and Compilation Techniques (PACT 2007), pp. 376-387. IEEE, 2007.
4. Van der Linden, Sebastian, Andreas Rabe, Matthias Held, Benjamin Jakimow, Pedro J. Leitão, AkponaOkujeni, Marcel Schwieder, Stefan Suess, and Patrick Hostert. "The EnMAP-Box—A toolbox and application programming interface for EnMAP data processing." Remote Sensing 7, no. 9 (2015): 11249-11266.
5. Ayres, Daniel L., Aaron Darling, Derrick J. Zwickl, Peter Beerli, Mark T. Holder, Paul O. Lewis, John P. Huelsenbeck et al. "BEAGLE: an application programming interface and high-performance computing library for statistical phylogenetics." Systematic biology 61, no. 1 (2012): 170-173.
6. Selby, Peter, Rafael Abbeloos, Jan Erik Backlund, Martin Basterrechea Salido, Guillaume Bauchet, Omar E. Benites-Alfaro, Clay Birkett et al. "BrAPI—an application programming interface for plant breeding applications." Bioinformatics 35, no. 20 (2019): 4147-4155.
7. Eichenberger, Alexandre E., John Mellor-Crummey, Martin Schulz, Michael Wong, Nawal Copty, Robert Dietrich, Xu Liu, Eugene Loh, and Daniel Lorenz. "OMPT: An OpenMP tools application programming interface for performance analysis." In OpenMP in the Era of Low Power Devices and Accelerators: 9th International Workshop on OpenMP, IWOMP 2013, Canberra, ACT, Australia, September 16-18, 2013. Proceedings 9, pp. 171-185. Springer Berlin Heidelberg, 2013.
8. Ong, Shyue Ping, Shreyas Cholia, Anubhav Jain, Miriam Brafman, Dan Gunter, Gerbrand Ceder, and Kristin A. Persson. "The Materials Application Programming Interface (API): A simple, flexible and efficient API for materials data based on REpresentational State Transfer (REST) principles." Computational Materials Science 97 (2015): 209-215.
9. Ejgenberg, Yael, Moriya Farbstein, Meital Levy, and Yehuda Lindell. "Scapi: The secure computation application programming interface." Cryptology ePrint Archive (2012).
10. Elhadi, Ammar AE, Mohd A. Maarof, and Ahmed H. Osman. "Malware detection based on hybrid signature behaviour application programming interface call graph." American Journal of Applied Sciences 9, no. 3 (2012): 283.
11. Wulf, Jochen, and Ivo Blohm. "Fostering value creation with digital platforms: A unified theory of the application programming interface design." Journal of Management Information Systems 37, no. 1 (2020): 251-281.
12. Olsen, Lindsey A., Clifford G. Robinson, Guangrong R. He, H. Omar Wooten, Sridhar Yaddanapudi, Sasa Mutic, Deshan Yang, and Kevin L. Moore. "Automated radiation therapy treatment plan workflow using a commercial application programming interface." Practical radiation oncology 4, no. 6 (2014): 358-367.
13. Kochanov, Roman V., I. E. Gordon, L. S. Rothman, P. Wcisło, C. Hill, and J. S. Wilzewski. "HITRAN Application

Programming Interface (HAPI): A comprehensive approach to working with spectroscopic data." Journal of Quantitative Spectroscopy and Radiative Transfer 177 (2016): 15-30.

14. Li, Jingming, Nianping Li, Kereshmeh Afsari, Jinqing Peng, Zhibin Wu, and Haijiao Cui. "Integration of Building Information Modeling and Web Service Application Programming Interface for assessing building surroundings in early design stages." Building and Environment 153 (2019): 91-100.
15. Fujiwara, Toyofumi, Jae-Moon Shin, and Atsuko Yamaguchi. "Advances in the development of PubCaseFinder, including the new application programming interface and matching algorithm." Human Mutation 43, no. 6 (2022): 734-742.
16. Liu, Nana, and Zeshui Xu. "An overview of ARAS method: Theory development, application extension, and future challenge." International Journal of Intelligent Systems 36, no. 7 (2021): 3524-3565.
17. Zavadskas, EdmundasKazimieras, ZenonasTurskis, and TatjanaVilutiene. "Multiple criteria analysis of foundation instalment alternatives by applying Additive Ratio Assessment (ARAS) method." Archives of civil and mechanical engineering 10, no. 3 (2010): 123-141.
18. Radović, Dunja, ŽeljkoStević, Dragan Pamučar, EdmundasKazimierasZavadskas, Ibrahim Badi, JurgitaAntuchevičiene, and Zenonas Turskis. "Measuring performance in transportation companies in developing countries: a novel rough ARAS model." Symmetry 10, no. 10 (2018): 434.
19. Turskis, Zenonas, and EdmundasKazimierasZavadskas. "A new fuzzy additive ratio assessment method (ARAS-F). Case study: The analysis of fuzzy multiple criteria in order to select the logistic centers location." Transport 25, no. 4 (2010): 423-432.
20. Ghenai, Chaouki, Mona Albawab, and MaamarBettayeb. "Sustainability indicators for renewable energy systems using multi-criteria decision-making model and extended SWARA/ARAS hybrid method." Renewable Energy 146 (2020): 580-597.
21. Gecer, Mustafa Kenan, TuncayKan, MuttalipGundogdu, SezaiErcisli, GulceIlhan, and Halil Ibrahim Sagbas. "Physicochemical characteristics of wild and cultivated apricots (Prunusarmeniaca L.) from Aras valley in Turkey." Genetic Resources and Crop Evolution 67, no. 4 (2020): 935-945.
22. Yildirim, BahadirFatih, and BurcuAdiguzelMercangoz. "Evaluating the logistics performance of OECD countries by using fuzzy AHP and ARAS-G." Eurasian Economic Review 10, no. 1 (2020): 27-45.
23. Davidson, D. F., and R. K. Hanson. "High temperature reaction rate coefficients derived from N-atom ARAS measurements and excimer photolysis of NO." International Journal of Chemical Kinetics 22, no. 8 (1990): 843-861.
24. Zamani, Mahmoud, Arefeh Rabbani, AbdolrezaYazdani-Chamzini, and ZenonasTurskis. "An integrated model for extending brand based on fuzzy ARAS and ANP methods." Journal of Business Economics and Management 15, no. 3 (2014): 403-423.
25. JaukovicJocic, Kristina, Goran Jocic, DarjanKarabasevic, GabrijelaPopovic, DragisaStanujkic, EdmundasKazimierasZavadskas, and PhongThanh Nguyen. "A novel integrated piprecia–interval-valued triangular fuzzy aras model: E-learning course selection." Symmetry 12, no. 6 (2020): 928.