Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org

Volume 25, Issues 4, pp. 2711–2720, DOI 10.12694/scpe.v25i4.2981

RESEARCH ON THE APPLICATION OF COLLABORATIVE CO-SIGNING INTELLIGENT VERIFICATION MODELS AND PARALLEL DISTRIBUTED ALGORITHMS FOR SEAMLESS INTEGRATION OF INTERNAL AND EXTERNAL DOCUMENTS

XIGUO HU[*], TAO ZHENG[†], DONGLIANG HOU[‡], AND YUN KANG[§]

Abstract. In today's complex and linked technology environments, attaining business efficiency and innovation is crucial. In this paper, an innovative architecture combining parallel distributed algorithms and collaboratively co-signing intelligent verification frameworks is introduced. By promoting safe and effective interaction among heterogeneous systems, this synergistic method seeks to improve the integration process. Diverse systems are proliferating in both internal and external domains, requiring sophisticated integration solutions. Real-time data interchange, interoperability, and security are common issues with existing methods. By using parallel distributed algorithms and intelligent verification models, the suggested framework aims to overcome these difficulties. The framework presents a novel approach to confirming the integrity and validity of data transferred between systems by utilizing collaborative co-signing. Co-signing increases security and confidence in the integrated environment by having several parties jointly validate the information. Through the introduction of parallel distributed algorithms and collaborative co-signing smart verification models, this research advances a comprehensive strategy for smooth integration of systems. The results highlight how the framework can facilitate safe and effective data interchange between external and internal systems, opening the door for more developments in connected technology environments in the future.

Key words: collaborative co-signing, intelligent verification models, parallel distributed algorithms, seamless integration of internal, external documents.

1. Introduction. The dynamic interchange of knowledge between company operations and external ecological systems, such as suppliers, collaborators, and third-party services, is crucial in today's business climate. The integration issues are tackled by the proposed framework, which highlights the importance of security, flexibility, and real-time response.

Assembly lines are an essential component of today's industrial sector, utilized in the mass production of standardized goods. As a result of shorter product life cycles and more product variations, designing assembly lines becomes a constant challenge. In order to support planers in the line design and setup, effective approaches are required. The design process involves several issues, including determining the best spatial arrangement, balancing the assembly line to ensure equal workload at each station, and selecting appropriate resources to carry out the necessary activities [5, 2]. Planners employ tools to capture their knowledge and expertise. Their goal is to generate design outcomes for each planning activity automatically. Nowadays, each tool typically processes and accesses a certain set of data, which results in [6, 3].

The entirety of all procedures used to put together geometrically defined bodies is referred to as assembly. It is possible to categorize and characterize assembly processes as handling, acceptance, supplementary procedures, and special activities [19, 4, 1, 13]. A workstation's functional sequence of operations essentially consists of connecting, feeding, examining, and supplying. Three functional regions can be used to separate the assembly tasks on an automated assembly line [22].

The parallelism process involves performing multiple operations simultaneously. Unlike traditional sequential algorithms, where tasks are completed one after another, parallel algorithms split tasks into subtasks that are processed at the same time. This approach is particularly useful for large-scale computations and can significantly reduce processing time. Distributed Computing refers to a model where computation is carried out

[*]School of Economics, Zhongnan University of Economics and Law, Wuhan 430000, Hubei, China (xiguohumas@outlook.com)
[†]School of computer science and technology, Harbin Institute of Technology, Haerbin 150000, Heilongjiang, China
[‡]School of Economics, Xiamen University, Xiamen 361000, Fujian, China
[§]College of Modern Economics & Management, Jiangxi University of Finance and Economics, Jiujiang 332000, Jiangxi, China

across multiple physical or virtual machines that are networked together. Each node (machine) in a distributed system can work independently on a portion of the problem and communicate with other nodes as needed.

The required planning periods are decreased by the developed tools. Unfortunately, they either lack basic assembly line information such as the necessary workpiece carrier or material flow via conveyor, or they primarily employ other data models [7]. When logically coupled, human data entry and modifications could be laborious and prone to mistakes. Time and money are greatly wasted since not all required planning data is integrated. In the realm of automated assembly planning, integrated digital description of generated data is consequently regarded as the primary problem [21]. AutomationML (AML) is a widely used data sharing format in the subject under consideration for information modeling.

The main contribution of the proposed method is given below:

1. Standardizing communication protocols and data formats to improve compatibility.
2. Putting strong security measures in place to protect private data both in transit and at rest.
3. Data synchronization and real-time communication are made possible by the integration of processes.

Remaining sections of this paper are structured as follows: Section 2 discusses about the related research works, Section 3 describes the Seamless Integration, Parallelization and Deep Learning methods, Section 4 discusses about the experimented results and comparison and Section 6 concludes the proposed optimization method with future work.

2. Related Works. AML is an XML-based data interchange format designed for production system design [20]. Given that the structure attempts to facilitate the exchange of design outcomes for every pairing of instruments in a design sequence, it can be seen as an illustration of an integrated IT landscape [16, 15, 18]. The object-oriented method of AML, which facilitates the mathematical representation of plant parts, is one of its benefits [14].

AML makes use of the four primary CAEX meta-model components to express a plant topology [17]. A case study of hierarchy depicts a plant's topology and organization. InternalElements (IE), which are modelled as objects, can also be saved as reusable parts in a SystemUnitClassLib. Potential roles that IEs could be assigned are listed in role libraries. A role is typically used to specify the semantics of things, describing abstract physical or logical concepts. ExternalInterfaces (EI) are used to define interfaces to external or internal objects. InternalLinks are used to actualize relationships between two IEs. These succinctly stated ideas aid in the proper modeling of the resources, process, and product domain items as well as their relationships [11, 12].

Three stages are involved in the planning of assembly systems. The definition of the products comes first, then the methods needed for assembly are specified, and lastly the assembly system design. In the latter case, the assembly planners' expertise and experience will have a significant impact on the result. Many scholars are working on automated assembly planning-related subjects to aid planners. The digital definition and modeling of the Product, Process, and Resource domains is where many planning systems begin [10, 9]. Product-process-resource triples (PPR-Triples) are created using mapping algorithms [8] that match the needs of processes and products with the resources that are available [8]. PPR-Triple linking allows systems to be defined for assembly. Automated methods consider several parameters for this conceptual design, which are included.

For organizational efficiency and innovation, the seamless integration of internal and external systems is critical in an era marked by complex and interconnected technological landscapes. This paper presents a novel framework that blends parallel distributed algorithms with collaborative co-signing intelligent verification models. By improving the integration process, this synergistic method seeks to guarantee safe and effective communication between diverse systems.

3. Proposed Methodology. An overview of the applied methodology for product, process, and resource domain modeling in AML is provided in this chapter. In addition, other elements that are necessary for assembly line planning are offered, such as supply pallets and workpiece carriers. Here the data is transferred between AML. For seamless data transmission Parallelization and co-signing verification is used. An summary of the introduced elements in relation to one another finishes this section. In figure 3.1 shows the architecture diagram of proposed method.

3.1. Bill-of Material (BoM). Based on the semantics that are attributed to the objects, the product domain is grouped into four levels. References to external documents are defined using external data interfaces

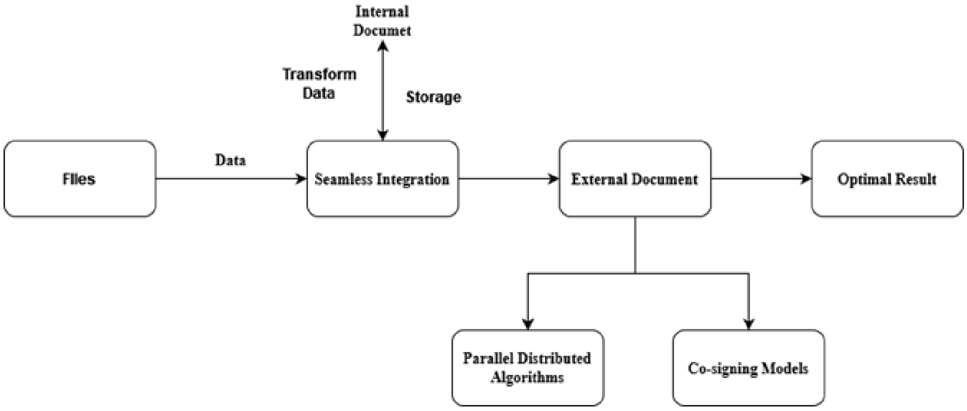

Fig. 3.1: Architecture Diagram of Proposed Method

of type ExternalDataReference. Every IE has an attribute called UniqueID, which is used to identify the same objects across different AML files.

The ProductVariant level is the first. This is one IE that has broad details about the finished product that are contained in a single AML file. For the sake of clarity, it is recommended that each variant have its own file. Model constraints, to which IEs with semantic ProductionConstraint are associated, supply data that informs the definition of the assembly system.

Jointing Elements and ProductParts are found in the following hierarchical level. The tiny part that cannot be dismantled any farther without causing damage is called a ProductPart. A higher value state is achieved by two ProductParts through assembling actions.

The necessity of optional joining aids varies based on the type of joining. Additionally, they are units that can't be dismantled any further. The elements include general details like material and weight in addition to abstract descriptions of their dimensions and shape. A crucial feature is the explanation of part supply. Thus, we distinguish between sorted, partially-sorted, and bulked based on [3]. To specify certain geometrical data in advance for handling via a resource, utilize a HandlingFeature. A planner is aware of the proper way and location to grab a thing beforehand.

There is a Frame definition for each HandlingFeature in relation to the overall product Frame. It is possible to designate a feature as either a HandlingFeatureLine or a HandlingFeatureArea, which will aid in choosing of gripper supplies. On the basis of the specified feature type, one can then choose between vacuum and magnetic grippers or finger grippers.

Components with the semantic roles SubAssembly and IntermediateAssembly are found in the final two hierarchies. An intermediate assembly represents the product's status following the inclusion of a new part. An assembly subgroup, on the other hand, depicts a single unit made up of optional JoiningElements and several ProductParts. But it's handled as a single part for the assembly operation. A gear motor is one example of this; it is typically supplied straight as a fully assembled component for assembly.

A Bill-of-Material structure is constructed with EIs isPartOf and consistOfPart inherited from the class ProductConnector. Each and every object that derives from the Product class has these interfaces attached. The BoM is defined as a graph by establishing InternalLinks, as seen in Fig. 3.2.

3.2. Material Flow and Supply. This research define a workpiece carrier using an abstract class named 'Pallet.' This class inherits attributes like size, weight, and material. The interface 'EI transfersPart' is used to link instances of the product domain's 'IntermediateAssembly'. As it moves down the assembly line, the product's basic portion goes through many intermediate phases before reaching its final condition. When an assembly object cannot be transported directly on a conveyor, the use of a workpiece carrier is required to ensure that it moves smoothly along the assembly line.

A ProductPart's orientation and position are determined in all degrees of freedom if it is supplied sorted.

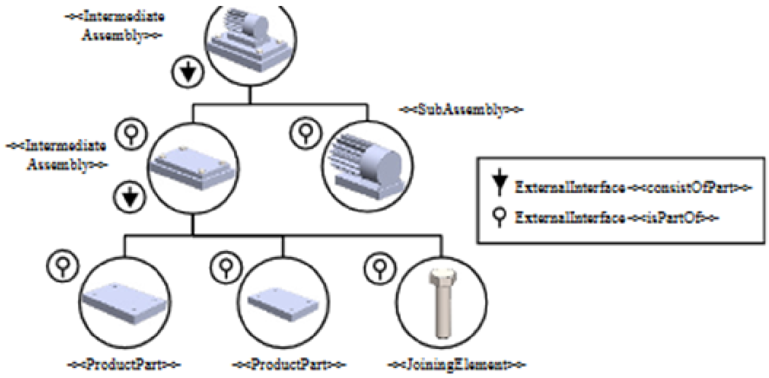

Fig. 3.2: Co-signing Methods

On an assembly line, supply pallets are used to provide materials in a sorted manner. A maximum capacity of hosted components attribute is added to the class Pallet's attributes. Product domain items can be connected to via the Interface SuppliesPart.

Kitting is a unique method of material supply used in assembly lines. As a result, at a workpiece carrier, all parts needed to assemble an assembly object are situated alongside the basic product part. Semantically, KittingPallet describes this workpiece carrier. There is one WorkpieceFixture and a predetermined number of KittingSlots in it. Every slot is reduced to a rectangular space inside the KittingPallet that has a frame. Every Intermediate Assembly that the KittingPallet processes throughout the assembly line is contained in the WorkpieceFixture. KittingSlot elements are assigned to required ProductParts, SubAssemblies, or JoiningElements. As a result, an EI named hostsPart serves as an interface between each element and the product domain.

3.3. Process and Modeling. All of the processes needed to put the finished product together are contained in the process domain. We define the element ProcessStep at the highest level. One ProcessStep is necessary for each time the product's state (Intermediate Assembly) changes. ProcessSteps need to be executed in a particular order and in relation to one another.

They include auxiliary or assembly procedures. ProcessTask is obtained by these processes as abstract semantic. There can only be one value-adding assembly process per ProcessStep. ProcessTasks' actual content can be used to better specify them. There are features and process parameters specific to each ProcessTask.

One is AutomationCapabilityFeature [7], which describes the degree to which a ProcessTask is appropriate for automated resources to do. ProcessParameters define the criteria for executing resources, just like a screwdriver requires torque.

A process's smallest unit is referred to as an operation. Given a set of resources and their unique properties, sequences can be used to estimate execution durations. Handling [3] and joining [19] actions provide the foundation of semantics. For instance, Operations describes a JoiningTask for the actual joining process and the movement of a joining tool. Processing times are calculated using velocity limitations or attributes for both horizontal and vertical distances. MotionType is an additional semantic that distinguishes between Pre-, Net-, and PostMotion to indicate how an Operation should be taken into account when determining a ProcessTask's total execution time.

3.4. Resource Composition Model. A resource is the smallest component of an assembly line.

Certain jobs, such as Robot, Vibratory Feeder, or Screwing Tool, are assigned based on the actual type. Attributes contain information about the manufacturer as well as other fundamental identifiers and skill descriptions. There are too many distinct qualities to go into detail about resources. Most characteristics simulate geometric data. The FootprintFeature is one of them. It gives an abstract description of the necessary space for a resource. Every FootprintFeature inherits the semantics and properties of the ShapeFeature that it is assigned. This characteristic distinguishes between polygon, rectangle, and circle abstract shapes. Everybody has a different description model. Certain resources, such as robots, also have a workspace. To specify a reach, a ReachabilityFeature that additionally inherits properties from a ShapeFeature is used.

Sometimes a single resource is insufficient to complete a ProcessTask. It is necessary to assemble the resources.

A ProcessingUnit is the clustering unit for these resources. A gripper and a robot are two examples. An abstract feature called ConnectionFeature is introduced, along with its fundamental attribute Frame, to represent a relationship between two resources. A more detailed feature called AttachmentFeature is used since a gripper is fastened to a robot flange. To define an InternalLink between two AttachmentFeatures, it offers an interface called AttachmentConnector. A MountingFeature can be used to model the type of mounting since some resources are mounted on a table. Use an EnergyConnectionFeature if an energy connection is necessary.

Sometimes a single resource is insufficient to complete a process task. It is necessary to assemble the resources. The clustering unit for these resources is a Processing Unit. Two examples are a gripper and a robot. To express a relationship between two resources, an abstract feature named Connection Feature is developed, along with its fundamental attribute Frame. Since a gripper is attached to a robot flange, a more detailed feature called Attachment Feature is employed. It provides an interface named Attachment Connector for defining an Internal Link between two Attachment Features. Because certain resources are mounted on a table, a Mounting Feature may be used to model the type of mounting. If an energy connection is required, use an Energy Connection Feature.

Workstations can be allocated to and grouped with different Processing Units. An axis-aligned bounding box defines the necessary space for all of a workstation's resources. A Frame property describes a resource's location in relation to a Workstation's bounding box. However, workstations can also be grouped together to form ProductionCells. In terms of our present modeling scope, this is the highest considered cluster.

ProductionCells can be thought of as particular line segments. The locations of the Workstations it contains define the bounding box. A cell's stations are related to one another in a particular way as predecessor and successor. As a result, the classes isSuccessorWorkstation and isPredecessorWorkstation of ResourceConnector are shown. There is a Frame for each Workstation in relation to the cell enclosing box. Connections Among Production Cells.

Carrier Handling System facilitates material flow via all Production Cells and associated Workstations. It includes resources from the CirculationModule, PalletLoadingModule, and Conveyor classes. A variety of configurations can be modeled. Circulation modules, for instance, are employed when empty workpiece carriers are never removed from an assembly line. Conveyor modules are therefore linked to enable frontflow, and a backflow arrangement returns the pallet to the first cell. Pallet steering from front to backflow is accomplished by a vertical or horizontal Circulation Module. The handling setup's resources have a new functionality known as MaterialTransferFeature.

3.5. Model for Available Space. An assembly line's theoretically accessible space is referred to as a polygon. 2D points are present in the polygon as BorderPoint elements. BorderPoints are established in respect to one another by the interfaces isSuccessorBorderPoint and isPredecessorBorderPoint. The semantic ProductionArea is subsequently assigned to the spanned area. The global coordinate system for each and every other assembly line object is defined in this section as well.

Workpiece carriers are injected and ejected from the assembly line at designated places designated by optional sources (input) and sinks (sinks). They are utilized to locate modules of the CarrierHandlingSystem and are simply referred to as Point2D. The space between these designated sites will subsequently be taken into account for allocating ProductionCells and conveyor resources.

It is possible to simulate so-called interfering contours to account for structural constraints such as reserved zones or pillars. These lessen the area that is available and is indicated by the ProductionArea. A Frame property describes the center of each interfering contour with respect to the global coordinate system. ShapeFeatures are used to describe geometric objects. It is impossible to discover any resource in regions with conflicting outlines.

3.6. Security for Seamless data.

3.6.1. Encryption for Data protection. Policies for granular access can be established according to many criteria, such as content kinds and user qualities. Strict enforcement of the least privilege principle is possible, limiting precise access to the data needed for a certain user role.Comprehensive audit logs are used to track every user access and activities in order to conduct forensic analysis and incident investigations. Anomalies in access patterns can also be the basis for real-time activity policy alerts.By reducing the likelihood of both external data breaches and insider threats, such strong access control protects data integrity within FileNet.

Documents and data are encrypted using AES 256-bit technology by FileNet while they are being sent and are at rest within the content repository. This prevents theft or unwanted access to data. For further security, encryption keys are safely maintained apart from the encrypted data. Hardware Security Modules (HSMs) and FileNet combine to store keys and delegate cryptographic processing.FileNet uses formats like AES-XML to guarantee that encrypted data is fully searchable and analyzable for authorized users.

3.7. Parallelization method for seamless data. For smooth and effective data processing, parallelization techniques are essential, particularly in settings involving huge datasets or intricate calculations. Here's a talk about parallelization techniques to accomplish smooth data processing using map reduce algorithm for distributed environment.

The Map function turns one collection of data into another, where individual items are split down into tuples (key/value pairs). The input data is broken into smaller chunks in a distributed system, and the Map function is applied to each piece individually. The Reduce function takes the Map function's result as input and merges those data tuples into a smaller collection of tuples. Typically, the reduction involves a summary operation such as counting, summing, or calculating the average.

Each document is divided into words during the Map Phase. Each word is processed by the Map function, which returns a key-value pair (word, 1). If the word "apple" appears three times in a document, the Map function will emit (apple, 1) three times. The system organizes all key-value pairings depending on the key (in this example, the word) during the Shuffle and Sort Phase. This step prepares data for the Reduce phase by bringing together all occurrences of a term. The Reduce function sums the numbers for each combination of key-value pairs (per word).

4. Result Analysis . The proposed method for seamless data integration of internal and external documents is evaluated in Python. The parameter metrics such as accuracy, recall, precision and F1-score is evaluated for sharing the files.

The simulation's accuracy, which is expressed as follows in Equation (4.1), indicates how effectively the model works across classes.

$$Accuracy = \frac{Total\ number\ of\ truly\ classified\ samples}{Total\ Samples} \tag{4.1}$$

The precision of the simulations is an assessment of their capacity to detect true positives, and it is computed using Equation (4.2).

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

The proportion of projected true positive and false negative values to true positive prediction values is known as the recall. Equation (4.3) represents the calculation.

$$Recall = \frac{TP}{TP + FP} \tag{4.3}$$

The model's total accuracy, or F1 score, strikes a positive class balance between recall and precision. Equation (4.4), which represents the calculation, is used.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.4}$$

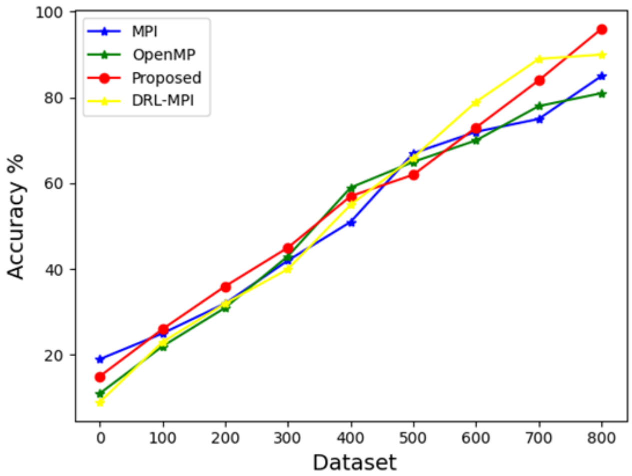

Fig. 4.1: Accuracy

A combination of computational factors, continual monitoring, and data quality assurance are used to ensure correctness in seamless data processing. Finding the right balance between automation and human monitoring is crucial for producing accurate and dependable outcomes in a variety of data processing circumstances. In the context of data processing, ”accuracy” typically refers to the precision and correctness of the processed data or the outcomes of data analysis. In several fields, including machine learning, analytics, and scientific research, accuracy is essential. In figure 4.1 shows the accuracy of proposed method. The proposed method is compared with existing methods such as MPI, OpenMP and DRL-MPI. The dataset uses 800 files for evaluation.

When accurately identifying affirmative cases is prioritized or the cost of false positives is substantial, precision proves to be a useful indicator. A thorough assessment of the overall effectiveness of a seamless data processing system can be obtained by balancing precision with other metrics, such as recall and accuracy. In data processing, precision is a critical parameter, particularly in situations where the accuracy of positive predictions is particularly important. The ratio of real positive predictions to all positive predictions a system makes is known as precision. When it comes to activities like classification, where the objective is to precisely identify particular categories or patterns in the data, precision becomes important in the context of smooth data processing. In figure 4.1 shows the evaluation of Precision.

In situations like medical diagnosis or fraud detection, when missing positive occurrences can have serious effects, recall is very important. Finding a balance between recall and precision is crucial, though, as boosting recollection could result in an increase in false positives. The evaluation criteria selected should be in line with the particular objectives and limitations of the data processing task. In data processing, recall—also referred to as sensitivity or true positive rate—is an essential measure, especially in situations where gathering as many pertinent examples as possible is the aim. The ratio of true positive predictions to all real positive instances in the data is known as recall. When the cost of missing positive cases (false negatives) is deemed large, attaining good recall is crucial for seamless data processing. In figure 4.3 shows the recall of peoposed method.

Combining recall and precision into a single number, the F1 score offers a fair evaluation of a model's performance. When there is an imbalance between positive and negative occurrences, it is very helpful, and it is important to take into account both false positives and false negatives. The harmonic mean of recall and precision is known as the F1 score. The F1 score is a useful metric for situations where a balanced approach to false positives and false negatives is essential since it offers a thorough evaluation by taking both precision and recall into account. Selecting the evaluation metric that fits the particular objectives and limitations of the data processing operation is crucial. In figure 4.4 shows the fi-score of proposed method.

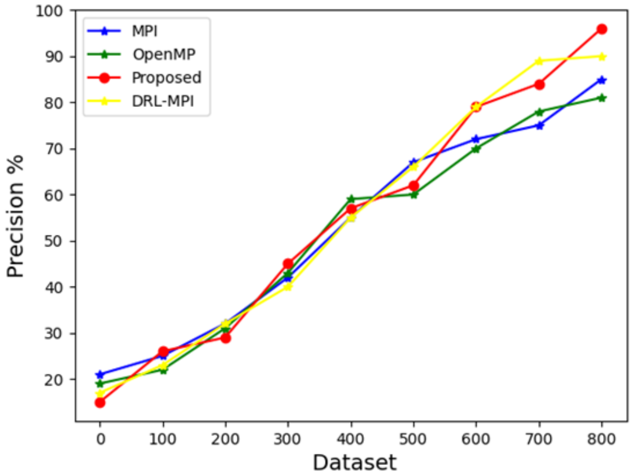

Fig. 4.2: Precision

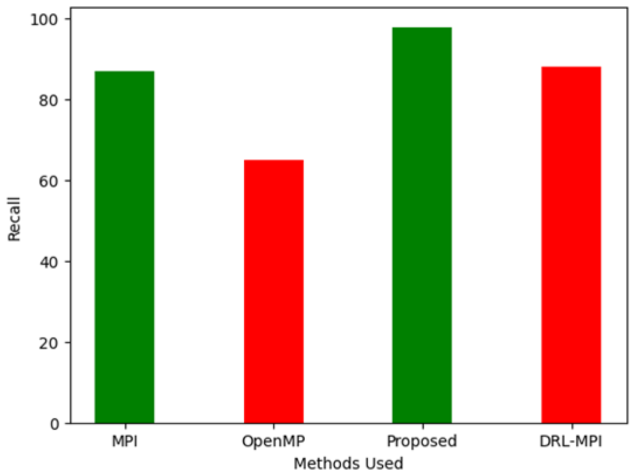

Fig. 4.3: Evaluation of Recall

5. Conclusion. The seamless integration of internal and external systems is essential for organizational effectiveness and innovation in a time of complex, interconnected technology environments. This paper presents a novel architecture that combines cooperatively co-signing intelligent verification frameworks with parallel distributed algorithms. By facilitating safe and effective interaction among heterogeneous systems, this synergistic strategy strives to improve the integration process. Diverse systems are developing in both internal and external sectors, needing sophisticated integration solutions. Common problems with current approaches include interoperability, security, and real-time data transfer. The proposed approach seeks to address these challenges through the use of intelligent verification models and parallel distributed algorithms. Through the use of collaborative

Acknowledgment. This work was supported by technology Project of State Grid Co., LTD. (SGHEXT00 YJJS2250335)

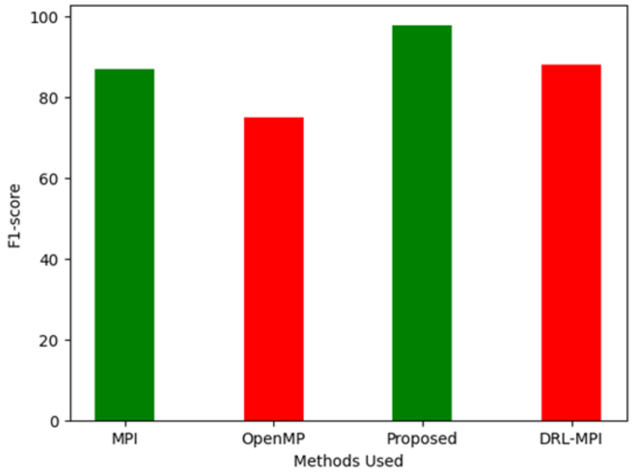

Fig. 4.4: F1-Score

REFERENCES

[1] M. Albus and C. Seeber, *Linear optimization for dynamic selection of resources in constrained assembly line balancing problems*, Procedia CIRP, 104 (2021), pp. 134–139.

[2] J. Backhaus and G. Reinhart, *Digital description of products, processes and resources for task-oriented programming of assembly systems*, Journal of Intelligent Manufacturing, 28 (2017), pp. 1787–1800.

[3] J. Backhaus, M. Ulrich, and G. Reinhart, *Classification, modelling and mapping of skills in automated production systems*, in Enabling Manufacturing Competitiveness and Economic Sustainability: Proceedings of the 5th International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2013), Munich, Germany, October 6th-9th, 2013, Springer, 2014, pp. 85–89.

[4] M. Fechter, R. Keller, S. Chen, and C. Seeber, *Heuristic search based design of hybrid, collaborative assembly systems*, in Advances in Production Research: Proceedings of the 8th Congress of the German Academic Association for Production Technology (WGP), Aachen, November 19-20, 2018 8, Springer, 2019, pp. 188–197.

[5] M. Fechter, C. Seeber, and S. Chen, *Integrated process planning and resource allocation for collaborative robot workplace design*, Procedia CIRP, 72 (2018), pp. 39–44.

[6] J. Herzog, H. Röpke, and A. Lüder, *Allocation of pprs for the plant planning in the final automotive assembly*, in 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), vol. 1, IEEE, 2020, pp. 813–820.

[7] N. Kousi, D. Dimosthenopoulos, A.-S. Matthaiakis, G. Michalos, and S. Makris, *Ai based combined scheduling and motion planning in flexible robotic assembly lines*, Procedia CIRP, 86 (2019), pp. 74–79.

[8] A. R. Kunduru, *Artificial intelligence usage in cloud application performance improvement*, Central Asian Journal of Mathematical Theory and Computer Sciences, 4 (2023), pp. 42–47.

[9] ———, *Industry best practices on implementing oracle cloud erp security*, International Journal of Computer Trends and Technology, 71 (2023), pp. 1–8.

[10] ———, *Security concerns and solutions for enterprise cloud computing applications*, Asian Journal of Research in Computer Science, 15 (2023), pp. 24–33.

[11] A. Lämmle and S. Gust, *Automatic layout generation of robotic production cells in a 3d manufacturing simulation environment*, Procedia CIRP, 84 (2019), pp. 316–321.

[12] A. Lämmle, C. Seeber, and E. Kogan, *Automatic simulation model implementation of robotic production cells in a 3d manufacturing simulation environment*, Procedia CIRP, 91 (2020), pp. 336–341.

[13] D. Leiber, V. Hammerstingl, F. Weiss, and G. Reinhart, *Automated design of multi-station assembly lines*, Procedia CIRP, 79 (2019), pp. 137–142.

[14] A. Lüder and N. Schmidt, *Automationml in a nutshell*, Handbuch Industrie 4.0 Bd. 2: Automatisierung, (2017), pp. 213–258.

[15] A. Mazak, A. Lüder, S. Wolny, M. Wimmer, D. Winkler, K. Kirchheim, R. Rosendahl, H. Bayanifar, and S. Biffl, *Model-based generation of run-time data collection systems exploiting automationml*, at-Automatisierungstechnik, 66 (2018), pp. 819–833.

[16] A. Neb and J. Hitzer, *Automatic generation of assembly graphs based on 3d models and assembly features*, Procedia CIRP, 88 (2020), pp. 70–75.

[17] M. Reichenbach, *Entwicklung einer Planungsumgebung für Montageaufgaben in der wandlungsfähigen Fabrik, dargestellt am Beispiel des impedanzgeregelten Leichtbauroboters*, Shaker, 2010.

[18] E. Schäffer, L. Penczek, A. Mayr, J. Bakakeu, J. Franke, and B. Kuhlenkötter, *Digitalisierung im engineering*, 2019.
[19] M. Schleipen and R. Drath, *Three-view-concept for modeling process or manufacturing plants with automationml*, in 2009 IEEE Conference on Emerging Technologies & Factory Automation, IEEE, 2009, pp. 1–4.
[20] G. N. Schroeder, C. Steinmetz, C. E. Pereira, and D. B. Espindola, *Digital twin data modeling with automationml and a communication methodology for data exchange*, IFAC-PapersOnLine, 49 (2016), pp. 12–17.
[21] A. Schyja, M. Bartelt, and B. Kuhlenkötter, *From conception phase up to virtual verification using automationml*, Procedia CIRP, 23 (2014), pp. 171–177.
[22] C. Seeber, M. Albus, M. Fechter, A. Neb, and S. I. Yoshida, *Automated 2d layout design of assembly line workstations through physical principles*, Procedia CIRP, 104 (2021), pp. 1197–1202.

Edited by: Rajanikanth Aluvalu
Special issue on: Evolutionary Computing for AI-Driven Security & Privacy: Advancing the state-of-the-art applications
Received: Dec 6, 2023
Accepted: Jan 4, 2024