

Review

Not peer-reviewed version

Systematic Review of Graph Neural Network and Consensus Algorithm-Based Approaches for Proactive Deadlock Detection in Distributed Systems

[W.A.T.G.R. Wijethunga](#)^{*} and B.T.G.S. Kumara

Posted Date: 19 November 2025

doi: 10.20944/preprints202511.1400.v1

Keywords: Consensus Diagram; Deadlock; Distributed Systems; Graph Neural Network(GNN); Proactive Dead-lock Detection



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Review

The Systematic Literature Review Proactive Deadlock Detection in Distributed Systems: A Systematic Review of Graph Neural Networks and Consensus Algorithm-Based Approaches

W.A.T.G.R.Wijethunga * and B.T.G.S. Kumara

Sabaragamuwa University of Sri Lanka

* Correspondence: tharushiwijethunga999@gmail.com

Abstract

Aim, This study aims to investigate the enabling technologies and applications of graph neural networks (GNN) and consensus diagrams for deadlock detection in distributed systems; **Background,** A circular wait state can result from a process requesting resources that are being held by another process, which can cause deadlock. The energy usage during offloading may rise as a result of this unforeseen, unplanned cyclic wait. In real-world applications that are sensitive to no buffering, deadlock is an undesirable occurrence. Additionally, a deadlock is an occurrence that occurs in a programming environment when multiple processes vie for a limited quantity of resources. In this case, when a process asks for resources that aren't available at the moment, it goes into a waiting state. The system is said to be in a deadlock if the waiting process is unable to alter his state because another waiting process is holding the necessary resources that he requests. Still, there is limited research on deadlock management in real-time systems. This kind of deadlock can make systems work worse and waste energy because the system can't tell what went wrong. The four main types are ignore, detect and recover, prevent, and avoid. Also, these four things must happen. No preemption, hold and wait, mutual execution, and circular wait. If all four of these things happen at the same time, the system is most likely in a deadlock. The main effect of a deadlock is that the system freezes. In this case, the system stops working, which can make users less happy. The system's efficiency goes down, which wastes resources, aborts transactions, adds to performance overhead, and lowers concurrency. These problems can make cloud services less reliable, less efficient, and less useful overall. But it works to use AI and machine learning algorithms and anticipatory models to find and fix deadlock situations. These kinds of models help figure out when deadlocks are most likely to happen. It can be helpful to set up different ways to deal with them that won't have an effect on the whole system. It also makes the system more reliable and better for everyone who uses it. This research can try to improve the methods, test them out, and suggest the best AI and ML methods, as well as find a way to make cloud management more efficient, even though there are many different kinds of problems, such as data quality and availability, the complexity of cloud environments, real-time processing needs, and integration with existing systems. **Methodology,** A systematic literature review was performed following a standardized guideline for the period from 2020 to 2025. First, a full review protocol was set up and tested. Nine electronic databases were used for the review: IEEE Xplore, ACM Digital Library, Science Direct, SpringerLink, Wiley Online, Tylor & Francis, Elsevier, and MDPI. **Contribution,** This systematic literature review enhances the evaluation of contemporary research on deadlock detection in distributed systems, specifically emphasizing methodologies that utilize Graph Neural Networks (GNN) and consensus diagrams. This study synthesizes the fundamental technologies that support modern GNN-based deadlock detection and delineates essential system and network parameters that guide the design of efficient distributed detection mechanisms. The review further explores the specific roles of consensus diagrams in modeling and facilitating coordination during the deadlock resolution process. The study identifies research gaps and proposes promising future directions for utilizing GNN architectures and consensus-driven

frameworks in resilient, scalable deadlock detection systems by emphasizing current methodologies and their relative efficacy. The findings are intended to support subsequent research by clarifying pivotal challenges and mapping out emerging opportunities in this evolving field. Findings, Graph Neural Networks (GNNs) and consensus-based model are extremely promising models in deadlock detection in distributed systems. These contemporary strategies provide scalability, efficiency, and flexibility in systems of complexity, dynamics, and large scale, which provide options when centralized deadlock management is impossible or too expensive. GNN based and consensus based methods have proven to be now feasible and robust substitutes (not simply complimentary) to the conventional deadlock methods, particularly in systems that are very complex, have dynamic topology and limited centralized coordination. AI and ML models enhance the deadlock prediction and preemption to enable systems to avoid the degradation of performance, minimize the energy wastage and automatic adaptation to prevent critical failures. Working hybrid Petri Net + AI/ML models, reinforcement learning, and consensus improvements have been shown to be effective in areas of manufacturing, multi-robot systems, rail networks, payment networks, and cloud computing. Experimental works show that GNN and deep learning algorithms (e.g., ANN, RL) have performed better than classical algorithms, and in most cases, the accuracy, precision, and recall are up to 100 percent in detection and prediction. Future studies must concentrate on adaptive GNN models, dynamically model changing networks, take advantage of real-time consensus data, and integrate GNN and traditional distributed algorithms. Issues of data quality, benchmarking, interpretability and standardization still persist. Recommendations for Practitioners, Distributed systems Deadlock detection in GNNs and consensus-based models have become promising solutions. As distributed computing advances, GNN-based and consensus-driven deadlock detection schemes are no longer simply complementary to the established monitoring, but rather are also becoming viable and scalable secondary strategies that can perform well even in situations where more traditional, centralized deadlock prevention is not possible or prohibitively expensive. Thus, the distributed system practitioners should use GNN and consensus diagram design techniques especially when the system is complex, changing topology, or where centralized coordination is minimal. Recommendations for Researchers, The continued development of distributed systems and artificial intelligence justifies why researchers should further investigate and merge the new technologies in deadlock detection. It is necessary to pay specific attention in trying new models of graph neural networks and new consensus mechanisms, in particular, their scaling, accuracy, and explainability. It is recommended that researchers explore novel GNNs architecture, dynamic graph modelling, and adaptive consensus mechanisms to even improve deadlock prediction, detection, and resolution in more complicated and large-scale settings. Impact on Society, The implementation of state-of-the-art deadlock detection algorithms based on graph neural networks (GNN) and consensus diagrams in distributed systems is a phenomenon that is likely to have a drastic effect on society. These techniques can be used to avoid expensive downtime and system failure by greatly improving the reliability, availability, and efficiency of distributed computing infrastructures that serve as supports to key services in finance, healthcare, communications, etc. This enhanced resilience does not just protect important data and services to businesses and individuals but also allows innovation and scale of technological ecosystems that support everyday operations of society. Consequently, better deadlock detection leads to the increased confidence in digital systems and hastens greater societal progress in automation, data-driven decision making and resilient computing. Future Research, Future studies in deadlock detection in distributed systems ought to consider the use of graph neural networks (GNNs) and consensus-based algorithms in enhancing accuracy in detection, scalability and efficiency of detection. In particular, there is a need to do additional work to advance adaptive GNN designs capable of dynamically modeling changing distributed environments, incorporating real-time consensus information to better detect context, and reducing communication and computational footprint without compromising timeliness or accuracy. Future research can also consider hybrid methods that integrate GNN-based learning with classical distributed algorithms, and consider robust methods to address false positives and negatives in large-scale, heterogeneous networks. Through dealing with these directions, it will be possible to deal with more robust, self-sufficient, and reliable distributed systems in the upcoming years.

Keywords: consensus diagram; deadlock; distributed systems; graph neural network(gnn); proactive dead-lock detection

Introduction

A deadlock is a state in computing in which two or more processes cannot continue due to one of them holding up the other until it releases a resource, which leaves the processes permanently stuck in their current state without action unless something is done. Deadlocks in distributed systems, processes on different nodes are stalled waiting to get resources held by each other creating a circular dependency that stops further execution. These deadlocks are serious since complexity is increased by the distributed nature: a global clock, central control is absent and resource and communication dependencies involve many machine.

1. Prevention

The purpose of the deadlock prevention is to make sure that the situation, which pre-conditions the occurrence of a deadlock, never happens at the same time. This is done by having policies that are strict in resource allocation like processes must obtain all the required resources before they can be executed or have policies which preemptively deny a request that may lead to a circular wait. Although such ensures that deadlocks are never created, this seriously impairs system concurrency and efficiency, by forcing the processes to wait before resources they may not actually be required at that time are available to them. Also it is usually infeasible in distributed systems when the future needs of resources cannot be predicted and pre-allocation may result in a situation of underutilization of resource.

2. Avoidance

Deadlock avoidance is a scheme of constantly examining the outstanding and existing requests to ensure that the system does not access a potentially unsafe state where a deadlock may happen. Wait-die or wound-wait are some of the algorithms that are used in deciding on how to grant, delay or deny resource requests. But in the case of distributed systems, deadlock avoidance is seldom employed: it presupposes a system-wide picture of system condition, which is costly to maintain, and occupies a lot of storage capacity and bandwidth. Therefore, deadlocks can be anticipated and avoided, but the costs of running them are usually prohibitive.

3. Detection

Deadlock detection also enables the system to run normally and only intervenes when a deadlock has occurred. The strategy is not limiting in terms of concurrency thus does not impact negatively on system throughput. In order to identify deadlocks, the system keeps a Wait-For Graph (WFG) showing the processes which are waiting to obtain resources occupied by other processes. The key tasks are:

Maintenance of the WFG,

Cycles or knots observed in this graph, indicative of a deadlock, are regularly searched.

In all the fields of operation of the distributed systems, deadlocks have a devastating effect on reliability and performance, including databases, cloud computing, financial services, telecommunications and supply chain management. Deadlocks in these scenarios lead to stagnation of processes and services, resulting in system downtime, low throughput, unutilized resources, inconsistent data and negative user experiences. This may result in loss of money, lost reputation and impact on mission critical application.

It is essential to identify dead locks effectively to ensure that the systems are reliable and efficient. There are effective deadlock detection strategies and convenient deadlock handling strategies like graph based analysis, time out, and distributed consensus which enables the system to quickly detect and overcome deadlocks. This is to guarantee the service is always available, resources are utilized optimally and systems can scale gracefully whilst providing predictable and repeatable behaviors. Fault tolerance and operational excellence in high-availability or large-scale distributed environments rely on a robust deadlock detector.

Background And Related Work

Modern studies of deadlock detection and resolution are indicative of the increasing range and complexity of distributed and multi-agent systems. The contemporary literature is not only trying to identify deadlocks but also pays a lot of attention to avoidance in the proactive mode, real-time recovery, and domain-agnostic constraints in robotics, manufacturing, high-performance computing, blockchain, and others.

New and various algorithms have been proposed. An example, LCL: A Lock Chain Length-based Distributed Algorithm to Deadlock Detection and Resolution (Sharma et al., 2024) extends classical distributed database deadlock detection by allowing the possibility of each process to wait on many different resources a feature important to real-world highly parallel data platforms. Likewise, An efficient deadlock handling model in the context of neutrosophic logic (Lu et al., 2021) also relies on uncertainty reasoning and is applicable to real-time healthcare databases, in which such information as transactions may not be complete or vague and, therefore, it is more resistant to noisy conditions.

The real-world and multi-agent systems present special means of approach. Deadlock-Aware Control of Multi-Robot Coordination with Multiple Safety Constraints (Zhang et al., 2025) provides a distributed control mechanism of cooperative robots, where an analysis of motion is used to dynamically adjust robot trajectories and avoid operational deadlocks of the environment with high safety demands. In the case of automated manufacturing systems, Maximally permissible deadlock and livelock avoidance of automated manufacturing systems using critical distance (Yang et al., 2022) employs Petri nets and the concept of critical distance which defines the minimum steps between safety and risk to design efficient controllers to prevent deadlocks and livelocks.

Other papers also make use of or build on graph-based systems. Large scale data optimum solution strategy to prevent deadlocks in resource assignments (Shanu et al., 2021) system resource allocation is viewed as directed graph and implements cycle of graph and centrality metrics to detect and resolve possible deadlocks in database processes, therefore informing the pragmatic resource management. Graph neural networks are used in Using Deep Learning for Deadlock Detection in Intelligent Software Systems (Romdhani et al., 2025), where deep learning models track demands of software tasks resources, and are found to be highly precise and adaptive to the traditional signature-based detection.

New ideas of chip networks and payment systems emerge with Developing Deadlock-Free Routing Algorithms in Torus NoC: A Formal Approach (Taheri et al., 2022) and Optimal hub placement and deadlock-free routing to payment channel network scalability (Yang et al., 2022). These papers note that there is a need of dedicated graph-based routing and placement algorithms to trade-off between throughput, latency, and ensure deadlock-free operations as on-chip and off-chain networks keep expanding.

Dynamic partial deadlock detection and recovery through garbage collection can be discussed in Dynamic Partial Deadlock Detection and Recovery (Saioc et al., 2025), and performance impact based multi-task distributed scheduling algorithm with task removal inference and deadlock avoidance (Li et al., 2023). The latter implements a sound Liveness-based approach based on Go garbage collector, which makes viable and applicable to real-world cases the detection and recovery of incomplete deadlocks, whereas the latter incorporates a new task-removal policy to guarantee the convergence and resilience of multi-agent mission scheduling.

The importance of consensus mechanisms integration is on the increase. Secure Consensus Control on Multi-agent Systems based on Re-engineered PBFT and Raft Blockchain Consensus algorithms (Zhu et al., 2025) suggests the use of leader grouping and improved signatures to maximize the safety and speed of secure group decision-making, whereas The Evolution and Optimization Strategies of a PBFT Consensus Algorithm in Consortium Blockchains (Yuan et al., 2025) is a review of the improvements to PBFT that focus on its core purpose in current, scale-based, and attack-res

There is also a presentation of new paradigms to resolve and prevent deadlocks in robotics and transportation. Merry-Go-Round: Safe Control of Decentralized Multi-Robot Systems with Deadlock Prevention (Lee et al., 2025) applies a roundabout strategy of proactive avoidance, and Addressing deadlock in large scale, complex rail networks via multi-agent deep reinforcement learning (Bretas et al., 2023) shows that autonomous train agents may coordinate and decongest railway networks with the use of decentralized learning.

Regardless of these contributions, researchers have continued to highlight persistent challenges: scaling methods to industrial applications, integrating ML with graph-based methods and empirically verifying methods across applications and on large-scale, distributed deployments have become key future objectives.

Having consolidated these recent works (2020-2025), this review gives a general and up-to-date outlook of the state of deadlock detection and avoidance with an emphasis on the convergence of AI, graph theory, consensus algorithms, and real-time distributed decision-making at an unprecedented pace being the pillar of the next generation of robust computing and cyber-physical systems. The recent study of 2020-25 years indicates high innovation in deadlock detection, avoidance, and resolution in many fields of distributed systems, robotics, cloud computing, manufacturing, and payment networks. As an example, LCL: A Lock Chain Length-based Distributed Algorithm to Deadlock Detection and Resolution (Sharma et al., 2024) and Partial Orders to Precise and Efficient Dynamic Deadlock Prediction (Oriolo and Russo, 2025) can be considered in solving the theoretical and practical issues by proposing more efficient lock-based models and more accurate prediction tools to reduce false positive and missed deadlocks in large-scale transaction processing and concurrency-intensive applications.

A number of works indicate that the integration of AI and sophisticated graph-based reasoning is advantageous in detection and avoidance. The article Deeplock Detection with Deep Learning in Intelligent Software Systems (Romdhani et al., 2025) demonstrates that machine learning models, in particular, GNNs can learn complicated dependencies, and deadlocks of software and software interactions can be predicted with a higher degree of confidence than signature- or state-based methods. Likewise in the field of multi-robot and autonomous vehicles, Deadlock-Aware Control for Multi-Robot Coordination with Multiple Safety Constraints (Zhang et al., 2025) and Learning Distributed Safe Multi-Agent Navigation via Infinite-Horizon Optimal Graph Control (Wang et al., 2025) both provide scalable, decentralized, real-time control of agents coordinated and efficient in dynamic or uncertain conditions using distributed control and minimal communication.

Domain-oriented publications provide new algorithms as well as practical frameworks. The ambiguity in real-time healthcare databases is handled by an efficient deadlock handling model that uses neutrosophic logic (Lu et al., 2021), whereas the topology-aware and resource-optimized design is shown to be crucial in both networked payments and high-performance GPU clusters (Optimal hub placement and deadlock-free routing to scalable payment channel networks (Yang et al., 2022) and Comprehensive Deadlock Prevention for GPU Collective Communication (Pan et al., 2025). The Petri nets and circuit models also apply to the real world in manufacturing, either through critical distance with Maximally permissible deadlock avoidance and livelock avoidance of automated manufacturing systems (Yang et al., 2022) or through Event circuit structures with deadlock avoidance of flexible manufacturing systems (Fan et al., 2023).

Decentralized and consensus-oriented policies become more prevalent. Secure Consensus Control on Multi-Agent Systems Based on Enhanced PBFT and Raft Blockchain Consensus Algorithms (Zhu et al., 2025) and The Evolution and Optimization Strategies of a PBFT Consensus Algorithm on Consortium Blockchains (Yuan et al., 2025) ensures improvement of performance and security in blockchain and IoT-scale settings, and Decentralized deadlock prevention in self-organizing industrial mobile robot fleets (Zajac and Malopolski, 2021) shows how efficient

Methodology

A systematic literature review implies a systematic approach to search, select, and summarize the literature related to a particular topic. It has no set of rules to which all people adhere globally, yet there are frameworks that are very commonly applied to ensure that reviews are of a high quality and credible. There are two frameworks, namely the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) and the guidelines of Kitchenham. PRISMA primarily informs the reporting of any systematic review in any field whereas the guidelines offered by Kitchenham were specifically developed based on the requirements of the software engineering research and computer science research (Kitchenham and Charters, 2007). Due to the fact that the investigation is related to computer science, the review methods are approached in accordance with the steps mentioned in the framework by Kitchenham.

The research question is stated as follows:

The process of formulating the research questions is a very important step in any systematic review since the questions are used to define and guide the whole process of the review. The use of clear and focused questions is necessary to make sure that the review can cover the key objectives of the study. This is why it is necessary to develop certain research questions that perfectly reflect on what the research will accomplish.

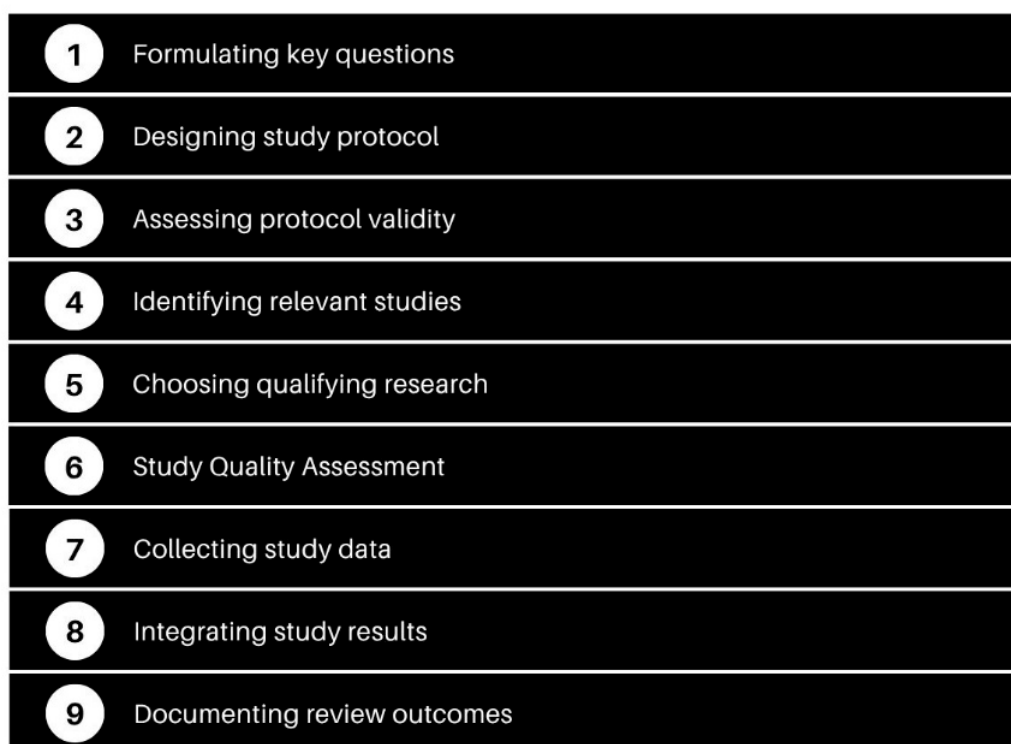


Figure 1. Main steps of a systematic literature review study.

No	Research Question	Rationale
RQ1	What is the current state of research activity and interest regarding the use of Graph Neural Networks (GNNs) combined with consensus-based approaches for deadlock detection in distributed systems?	Understanding the current research landscape is crucial to gauge how much attention the community is giving to GNN and consensus-based methods. This helps identify whether the combination is a growing trend, a niche area, or an

		emerging paradigm, and clarifies recent progress and open challenges that motivate further investigation.
RQ2	Why are consensus mechanisms and GNN approaches considered the most recent or innovative methods for deadlock detection in this field?	Highlighting why these methods are seen as "newest" provides insights into what makes them an innovation over previous techniques. This question will help pinpoint the technical and conceptual advances such as scalability, adaptability, or accuracy that set GNNs and consensus apart from older algorithms.
RQ3	Which experimental benchmarks, datasets, or evaluation frameworks are commonly utilized to test and validate these GNN and consensus-based approaches?	Identifying the empirical tools and datasets used in evaluation is necessary to assess the rigor, comparability, and reproducibility of research in this domain. It also helps future researchers choose appropriate benchmarks and ensures that new methods are tested against recognized standards.
RQ4	In what ways do GNN and consensus-based methods contribute to improved detection and seamless operation (or smooth recovery) in distributed systems experiencing deadlocks?	Examining the practical impact of these modern approaches reveals their real-world benefits and limitations. This question seeks to clarify not only whether deadlock detection is improved, but also if these solutions support the reliable, uninterrupted functioning that is critical in distributed production environments.

Figure 2. lists the research questions along with the reasons they were chosen.

Study Search

The first step when defining a search strategy is to identify specific search terms.

AREA	SEARCH TERM
Deadlock Detection	Deadlock, Deadlock Detection, Blocking, Resource Contention, Circular Wait
Distributed Systems	Distributed System, Distributed Computing, Multi-Agent, Cloud, Peer-to-Peer, Networked Systems
GNN & AI Methods	Graph Neural Network, GNN, Deep Learning, Machine Learning, AI, Neural Network
Consensus Approaches	Consensus, Consensus Algorithm, Blockchain, Distributed Consensus, Raft, PBFT, Paxos

Figure 3. Search Terms.

Subsequently, search strings can be developed. In this study, the search string was formulated based on terms related to the areas of career guidance and computer systems. Search terms and search strings are presented in table.

1	("Deadlock" OR "Deadlock Detection" OR "Blocking" OR "Circular Wait") AND
2	("Distributed System" OR "Distributed Computing" OR "Multi-Agent" OR "Peer-to-Peer") AND
3	("Graph Neural Network" OR "GNN" OR "Deep Learning" OR "Neural Network") AND
4	("Consensus" OR "Consensus Algorithm" OR "Blockchain" OR "Raft" OR "PBFT" OR "Paxos")

Figure 4. Search String.

Eight large electronic databases were used to carry out the search of the study: IEEE Xplore, ACM Digital Library, Science Direct, SpringerLink, Wiley Online Library, Elsevier, Taylor and Francis, and MDPI. The search string was thoroughly adjusted to the search rules and options of respective databases. Also, the publication date filter was established to cover only 2020-25 years. A total of 274 studies were found in this extensive search exercise. All the records of the digital libraries were catalogued and stored to undergo additional screening and analysis.

Study Section

Exclusion Criteria

CRITERIA	DESCRIPTION
Exclusion Criterion 1	The study is excluded if there is no abstract available.
Exclusion Criterion 2	The study is excluded if it is not written in English.
Exclusion Criterion 3	The study is excluded if it hasn't been published as a complete paper.
Exclusion Criterion 4	The study is excluded if it is an earlier version of a paper already included.
Exclusion Criterion 5	The study is excluded if it is not an original (primary) research article.

Figure 5. Exclusion Criteria.

In this research, a systematic search process was applied to identify and select relevant research papers, followed by a quality assessment. First, duplicate records were removed, leaving 574 unique publications (100%). Titles, abstracts, and keywords of these 574 papers were screened according to predetermined inclusion and exclusion criteria, resulting in 68 publications (11.8% of the original pool, an 88.2% decrease) that qualified for further review. Following a full-text assessment of the 68 papers, 30 studies (44.1% retention from previous, or 5.2% of the original pool) were selected. A snowballing procedure reviewing references and citations of these studies yielded 42 additional

articles. After applying all screening and eligibility criteria to these, 48 studies (combining main and snowballing paths, 8.4% of the original pool) remained for the final in-depth quality assessment.

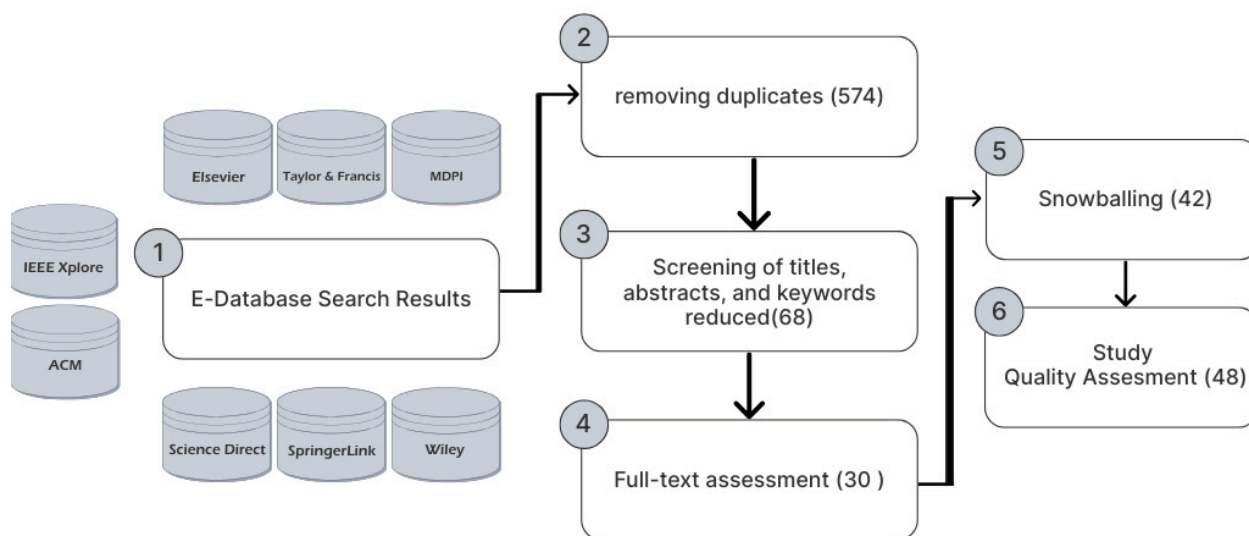


Figure 6. Study selection process.

Study Quality Assessment

The quality assessment in this study was conducted using a checklist of five quality questions, as outlined in Table 4. Each paper received a numerical score based on these criteria. To ensure consistency between reviewers, Cohen's was used to measure agreement. If evaluators disagreed, the Delphi method was employed to reach consensus.

Table 1. Quality assessment criteria.

Criteria	Description
QC1	Are the goals of the research stated in a clear way?
QC2	Does the paper provide a thorough explanation of the relevant theories or foundational ideas?
QC3	Is the chosen methodological approach described with enough detail?
QC4	Are the findings or outcomes of the research clearly and fully presented?
QC5	Does the paper include an assessment or review of the results, or suggest how such an evaluation could be done?

Data Extraction and Synthesis

The data extraction process was completed using a Data Extraction Form, which was developed in the phase of review protocol definition with minor improvements. Extracted data was stored in an electronic spreadsheet. The researcher of the current paper was involved in the extraction of the data on his own. At author assessed each study. Author evaluated any paper where a discrepancy was detected.

Results and Discussion

This section describes the results and implications related to the objectives of each study which are based on the results of the synthesis stage of the data. Here are presented the limitations of the study and recommendations. A new model introduced in the paper "Using Deep Learning for

Deadlock Detection in Intelligent Software Systems" (ROMDHANI et al., 2025) is proposed in order to do real-time deadlock detection in intelligent software systems. This method is based on the combination of Petri Nets, modeling the interaction of the tasks with the resources, and the modelling of the dynamics of the systems, with the artificial neural networks, in order to classify the states of the systems and their deadlock situations with high rates of recovery of accuracy. The training is carried out with different synthetic system instances of normal behaviour and prone to deadlocks. The results show that the model hybrid Petri Net+ANN is being able to improve the classical machine learning solutions (decision trees, random forests, .up to here continue similar.) and pure neural solutions, reaching the values of 100% accuracy, precision and recall in the tests carried out.

"Addressing deadlock in large-scale, complex rail networks via multi-agent deep reinforcement learning" (Bretas et al., 2023) has emerged as a highly effective means of facilitating adaptive deadlock avoidance and resolution in large-scale distributed systems with inherent vigour and dynamism like networks of railways. Intelligent agents (trains) adopt traffic management policies through centralised learning with decentralised execution achieving considerable improvement over the traditional first-come-first-serve (FCFS) heuristic. The introduction of reward shaping and deadlock lists leads to a great reduction of the deadlock frequency and offers much greater robustness to the policy, especially in high density situations.

Demonstrating a fault-tolerant observer-based leader-following consensus strategy based on utilizing virtual actuators and LPV modeling, "Fault-tolerant observer-based leader-following consensus control for LPV multi-agent systems using virtual actuators" (Trejo et al., 2024) illustrates its effectiveness in a cyber-physical domain such as UAV team coordination where the maneuvering of unmanned aerial vehicles in set formations and specific control tasks remains intact even when several actuators (like rotors) cease to function. Actual quadcopter simulation results are used to corroborate this claim.

Regarding consortium blockchains, "The Evolution and Optimization Strategies of a PBFT Consensus Algorithm for Consortium Blockchains" (Yuan et al., 2025) discusses how PBFT (Practical Byzantine Fault Tolerance) consensus can be refined through an improvement in the structure through hierarchical node classification, administrative measures, and hybrid remuneration measures. Such improvements significantly increase throughput and robustness, but the implementation of such improvements requires a balance between communicative complexity and scalability.

Table 2. Characteristics of the Selected Studies.

Id	Year	Purpose	Summery of Findings
#1	2024	Analyze and prevent deadlocks in payment channel networks using resource allocation graphs.	Identified and mitigated NP-complete deadlock risks, increasing scalability and reliability in blockchain transaction networks.
#2	2023	Enable efficient task scheduling in distributed systems by preventing deadlocks via task removal inference.	Proposed an improved multi-task scheduling algorithm that ensures system convergence and resource utilization during task allocation.
#3	2022	Provide deadlock/livelock avoidance in manufacturing using Petri nets and critical distance method.	Maximally permissive approach efficiently avoids deadlocks in complex manufacturing by reducing computational complexity.
#4	2022	Accelerate MPI deadlock detection through trace compression techniques.	Trace compression reduced detection time, speeding fault identification in message-passing programs.

#5	2021	Detect and recover deadlocks in Flexible Manufacturing Systems via resource flow graphs.	Resource flow graph approach enables fast, accurate recognition and recovery from deadlocks, ensuring smooth FMS operation.
#6	2021	Resolve collisions and deadlocks in multi-AGV transport by zone-decomposition and online control.	Unidirectional zone decomposition improved collision/deadlock handling, outperforming former approaches in transport tests.
#7	2021	Offer adaptive deadlock control in Petri net systems with unreliable resources.	Siphon control method enables robust operation even during frequent machine or resource failures.
#8	2022	Compare deadlock avoidance/prevention in autonomous vehicle resource provisioning over 6G.	Smart cooperative edge control strategies outperformed traditional systems in real-vehicle network tests.
#9	2021	Introduce the Vagabond non-exhaustive algorithm for fast deadlock detection in distributed models.	Distinctly detects and classifies deadlock vs. termination with little parameter tuning or reachability search.
#10	2021	Use graph theory and centrality to optimize resource allocation and avoid deadlocks in databases.	Demonstrated how critical resource requests can be prioritized to maintain deadlock-free large data operations.
#11	2023	Predict and avoid deadlocks in fog computing with multi-module load balancing.	Preemptive deadlock detection maintains load balance, ensuring real-time reliability and system uptime.
#12	2022	Guarantee deadlock-free routing by optimal hub placement in payment channel networks.	Adaptive hub placement improves throughput and guarantees deadlock-free payments as network scales.
#13	2023	Speed up deadlock avoidance in flexible manufacturing with Petri net event circuit structures.	Event circuit methods prevent deadlocks far more efficiently than prior state-intensive techniques.
#14	2022	Develop fault-tolerant, deadlock-free routing for chiplet (2.5D) network-on-chip systems.	"DeFT" algorithm maintains routing and fault tolerance across vertical links, outperforming alternatives.
#15	2022	Deliver a dynamic tool for broad deadlock detection via generalized dependency.	"UnHang" detects lock and condition-variable deadlocks, supporting more real-world concurrency scenarios.
#16	2025	Study synthetic influence group effects on consensus in social networks.	Visibility and noise shape consensus outcomes; findings inform design of robust digital social systems.
#17	2025	Optimize PBFT consensus strategy for blockchain consortia.	Review highlights PBFT optimizations for throughput, scaling, and attack resilience.

#18	2025	Secure scalable consensus for multi-agent systems using PBFT/Raft.	Node grouping and crypto-enhanced consensus lower communication costs while improving security.
#19	2024	Maintain multi-agent consensus under actuator faults with LPV virtual leader model.	Fault-tolerant observer-controller keeps group formation and task execution even when faults occur.
#20	2025	Enable resilient consensus cluster via a dynamic, event-triggered approach under DoS attacks.	Reduces message overhead while safeguarding consensus, protecting against unreliable networks and attacks.
#21	2025	Accelerate blockchain consensus with new express Clique approach.	“ExClique” enables up to 2x-7x faster transaction confirmation than existing Clique-based protocols.
#22	2025	Distribute safe task allocation and motion coordination in networked robot teams.	Local observation and coordination enable robust task execution across changing multi-robot environments.
#23	2025	Apply deep learning (GNN) in software task deadlock detection.	Supervised ANN model accurately detects software deadlocks, outperforming classical methods.
#24	2025	Prevent GPU collective communication deadlocks in DL applications.	“DFCCL” preemption ensures robust, high-performance distributed deep learning.
#25	2023	Use multi-agent deep RL to handle rail network deadlocks.	Decentralized agents reduce deadlock rates and improve traffic flow through learned policies.
#26	2025	Combine IoT, Petri nets, ANNs for deadlock/tool fault management in flexible manufacturing.	Enhanced controller improves uptime and productivity by controlling both deadlocks and tool faults.
#27	2025	Prevent robot deadlocks via decentralized, roundabout-based navigation (“Merry-Go-Round”).	Temporary roundabouts, local peer communication, and preventive intervention avoid standstill.
#28	2025	Optimize distributed, multi-agent navigation via GNN-based graph control.	Infinite-horizon planning enhances safety and goal-seeking, allowing real-time deadlock avoidance.
#29	2025	Efficient two-layer path planning for WMRs in dynamic environments.	Combines ACO and dynamic window to generate fast, energy-efficient robot paths in crowded spaces.
#30	2025	Map the service-oriented evolution of modern AI.	Identifies emerging directions, gaps, and digital transformation drivers in AI service delivery.

#31	2025	Deadlock-free multi-agent pickup/delivery in dynamic, mixed-agent settings.	Local, reactive collision avoidance policy ensures correct task completion even with external disruptions.
#32	2025	Schedule AGV movement/data processing using extreme-edge computing.	Joint scheduling maintains system responsiveness and prevents deadlocks through adaptive task allocation.
#33	2025	Promote safe multirobot coordination using distributed, deadlock-aware control.	Bottleneck detection and navigation priorities preserve team safety and task success.
#34	2025	Use garbage collection liveness for partial deadlock detection/recovery in message-passing programs.	Dynamic GC marking phase detects and recovers partial deadlocks, reducing memory leaks in Go-style applications.
#35	2025	Theoretical study and solution to deadlocks using "weak deadlock sets."	"Wise states" and weak set recognition offer polynomial-time detection and new prevention options in networks.
#36	2025	Plan safe, deadlock-free trajectories for AVs with occlusions and risk constraints.	Phantom obstacle modeling, risk quantification, and improved planning mitigate AV deadlocks in occluded scenes.

Table 3. Application Domain and Technology.

Application Domain	Technology Used	Study IDs
Manufacturing	Petri Net, AI	#5, #13, #26
Blockchain	Deadlock-Free Routing	#1, #12, #17

Table 4. System Scope and Strategy.

System Type	Deadlock Strategy	Study IDs
Cloud/Fog	Prediction/Prevention	#11, #24
Multi-Agent	Consensus-Based	#18, #22

Table 5. Algorithm Category and Results.

Algorithm Type	Key Feature	Study IDs
Graph Neural Net	Deep Learning	#23, #28
Game Theoretic	Decentralized Control	#25, #27

Table 6. Categorization by Reactive vs Proactive Deadlock Management.

Strategy	Description	Study IDs
Proactive	Methods focusing on prediction, avoidance, and prevention of deadlocks before they occur. These include control policies,	#2, #3, #6, #7, #8, #10, #11, #12, #13, #14, #16, #17, #18, #19, #20, #21, #22, #24, #26, #27, #28, #29, #30, #31, #32, #33, #35, #36

	algorithms for avoidance, forecasting, and planning approaches.	
Reactive	Methods focusing on detection and resolution after deadlocks have occurred, such as detection algorithms, recovery techniques, and resolution frameworks.	#1, #4, #5, #9, #15, #23, #25, #34

Limitations of the review

Only studies published in English were included, which may result in missing valuable research published in other languages.

Database & Source Selection: Although the study used several major digital libraries, relevant literature indexed only in other, smaller or regional databases may have been missed.

Publication Bias: Studies with positive results or novel findings are more likely to be published and included, potentially skewing the overall conclusions toward approaches that worked well, while negative or inconclusive results might be underrepresented.

Quality and Heterogeneity of Included Studies: Differences in research methods, scale, datasets, experimental settings, and definitions of ‘deadlock’ across studies can make it difficult to compare findings directly or synthesize results meaningfully.

Time and Resource Constraints: The review window was limited to 2020–2025. Some newer studies may have been missed and resource restrictions may have prevented deeper full-text analysis or expert quality assessment for all retrieved studies.

Rapid Evolution of the Field: AI and distributed systems research evolves extremely quickly; important new approaches, tools, or datasets can be published during or after the review process, leaving gaps in coverage or making some findings quickly outdated.

Limited Experimental Standardization: Benchmarks, metrics, and datasets vary widely. Not all studies used publicly available evaluation frameworks, making comparative analysis challenging and sometimes limiting reproducibility.

Model & Application Diversity: The studies included cover a range of architectures (GNN variants, consensus protocols, reinforcement learning models), application domains (software systems, rail networks, robotics, blockchains), and problem scales. This diversity can complicate cross-study generalization and mean that conclusions apply only to specific settings.

Interpretive Depth: While systematic reviews are powerful for synthesis, they sometimes lack the theoretical insight or nuanced explanation that can be found in detailed primary studies or meta-analyses.

Recommendations

Future reviews should incorporate additional databases and consider studies published in languages other than English to reduce the risk of missing relevant research and minimize language bias.

There is a strong need for widely-accepted experimental benchmarks and datasets tailored to deadlock detection and prevention in distributed systems. This will improve comparability and reproducibility across future studies.

Encourage joint efforts between computer scientists, engineers, and domain experts (such as robotics, networking, AI, and manufacturing) to transfer and adapt best practices and innovative algorithms across different distributed system environments.

Report Negative and Neutral Findings: Journals and conferences in the field should emphasize the importance of publishing negative or inconclusive results. This will provide a complete and unbiased picture of current methods' capabilities and limitations.

Encourage the sharing of code, datasets, and evaluation scripts. Open-source resources will accelerate the adoption, scrutiny, and iterative improvement of cutting-edge techniques.

Given the rapid evolution of AI and distributed computing, regularly updating systematic reviews or maintaining living documents online will help keep research, industry, and policy efforts aligned with the state-of-the-art.

Future work should include results from large-scale deployments and real-world applications to ensure that proposed algorithms are robust, scalable, and practical under realistic operating conditions.

As GNN and consensus-based models become more complex, integrating interpretable modeling and visualization techniques will help users and stakeholders trust, understand, and troubleshoot AI-driven deadlock management systems.

Conclusion

The conclusion of this study encapsulates the comprehensive insights drawn from a systematic literature review on the enabling technologies and applications of Graph Neural Networks (GNNs) and consensus diagrams for deadlock detection in distributed systems. The review has revealed that leveraging AI-driven models particularly GNNs in combination with consensus-based algorithms constitutes a promising direction for addressing the escalating complexity and scale of modern distributed environments. These approaches offer marked improvements in detection accuracy, adaptability, scalability, and efficiency compared to conventional deadlock management techniques.

Through the synthesis of research published between 2020 and 2025, the findings underscore several key advancements. GNNs have demonstrated superior capabilities in modeling complex, dynamic resource dependencies and predicting potential deadlock patterns in heterogeneous systems. Meanwhile, consensus mechanisms ranging from Raft and PBFT to adaptive hierarchical models have proven invaluable for ensuring agreement, fault tolerance, and coordinated recovery across distributed nodes. Together, these techniques form the foundations of proactive and decentralized deadlock management, reducing performance degradation, system stalling, and energy wastage in critical computing infrastructures.

However, the study also highlights persistent challenges that require ongoing research attention. Issues such as limited standardization of benchmarks, data quality variation, model interpretability, and cross-domain generalization continue to hinder widespread adoption. Furthermore, publication bias and linguistic or database coverage limitations may have constrained the comprehensiveness of the review. Despite these constraints, the identified gaps pave the way for future investigation into hybrid GNN-consensus frameworks, explainable AI models, and rigorous, real-world testing to validate scalability and robustness under practical conditions.

In conclusion, this review affirms that integrating GNN and consensus-based strategies represents a crucial evolution in distributed system reliability and autonomy. By fostering predictive, collaborative, and resilient deadlock management architectures, these approaches not only enhance computational performance but also strengthen the dependability of systems underpinning critical sectors such as finance, healthcare, robotics, and cloud computing. Continued research and practical deployment of these methods will be vital for realizing the next generation of intelligent, self-healing distributed systems capable of sustaining the demands of a globally connected digital society.

Appendix

Table A1. Unique Identifier for Selected Studies.

Paper ID	Research Paper	Unique identifier for selected studies
----------	----------------	--

#1	Deadlock Prevention in Payment Channel Networks	Sharma et al.,2024
#2	A performance-impact based multi-task distributed scheduling algorithm with task removal inference and deadlock avoidance	Li et al.,2023
#3	Maximally permissive deadlock and livelock avoidance for automated manufacturing systems via critical distance	Yang et al., 2022
#4	Improving the Efficiency of Deadlock Detection in MPI Programs Through Trace Compression	Huang et al.,2022
#5	An Efficient Method of Deadlock Detection and Recovery for Flexible Manufacturing Systems by Resource Flow Graphs	Lu et al.,2021
#6	Structural on-line control policy for collision and deadlock resolution in multi-AGV systems	Zajac & Małopolski,2021
#7	Adaptive Deadlock Control for a Class of Petri Nets With Unreliable Resources	Zhang et al,2021
#8	A Comparative Analysis of Deadlock Avoidance and Prevention Algorithms for Resource Provisioning in Intelligent Autonomous Transport Systems Over 6G Infrastructure	Ugwuanyi et al.,2022
#9	Non-exhaustive Verification in Integrated Model of Distributed Systems (IMDS) Using Vagabond Algorithm	Daszczuk,2021
#10	Optimal solution approach on large scale data to avoid deadlocks in resource allocations	Shanu et al., 2021
#11	Fog computing effective load balancing and strategy for deadlock prediction management	Talaat et al.,2023
#12	Optimal hub placement and deadlock-free routing for payment channel network scalability	Yang et al., 2022
#13	Event circuit structures for deadlock avoidance in flexible manufacturing systems	Fan et al.,2023
#14	DeFT: A deadlock-free and fault-tolerant routing algorithm for 2.5 D chiplet networks	Taheri et al.,2022
#15	Deadlock prediction via generalized dependency	Zhou et al.,2022
#16	Consensus effects of social media synthetic influence groups on scale-free networks	Porciúncula et al.,2025
#17	The Evolution and Optimization Strategies of a PBFT Consensus Algorithm for Consortium Blockchains	Yuan et al, 2025
#18	Secure Consensus Control on Multi-Agent Systems Based on Improved PBFT and Raft Blockchain Consensus Algorithms	Zhu et al.,2025
#19	Fault-tolerant observer-based leader-following consensus control for LPV multi-agent systems using virtual actuators	Trejo et al.,2024
#20	Dynamic Event-Triggered Cluster Consensus for Multiagent Systems Under DoS Attacks With Antagonistic Interactions	Zhou et al.,2025
#21	ExClique: An Express Consensus Algorithm for High-Speed Transaction Process in Blockchains	Zhao et al.,2025
#22	A Distributed Framework for Integrated Task Allocation and Safe Coordination in Networked Multi-Robot SystemsA Distributed Framework for Integrated Task Allocation and Safe Coordination in Networked Multi-Robot Systems	Miele et al.,2025
#23	Using Deep Learning for Deadlock Detection in Intelligent Software Systems	ROMDHANI et al.,2025
#24	Comprehensive Deadlock Prevention for GPU Collective Communication	Pan et al.,2025

#25	Addressing deadlock in large-scale, complex rail networks via multi-agent deep reinforcement learning	Bretas et al.,2023
#26	An Internet-of-Things-Based on ILPP, Petri Nets, and Artificial Neural Networks for Controlling Deadlocks and Tool Failures in Flexible Manufacturing Systems under ...	Kaid et al.,2025
#27	Merry-Go-Round: Safe Control of Decentralized Multi-Robot Systems with Deadlock Prevention	Lee et al.,2025
#28	Learning Distributed Safe Multi-Agent Navigation via Infinite-Horizon Optimal Graph Control	Wang et al.,2025
#29	Two-layer path planning framework for WMRs in dynamic environments: Optimized ant colony algorithm and dynamic window approach	Liu et al.,2025
#30	Service-Oriented Evolution of Modern AI: A Position Paper	Li et al.,2025
#31	A Deadlock-Free Solution for Multi-Agent Pickup and Delivery in Presence of External Agents	Flammini et al.,2025
#32	Dynamic Joint Scheduling of Movement and Data Processing Tasks Using Extreme-Edge Computing in Multi-AGV Scenarios	Masoumi et al.,2025
#33	Deadlock-Aware Control for Multirobot Coordination With Multiple Safety Constraints	Zhang et al.,2025
#34	Dynamic Partial Deadlock Detection and Recovery via Garbage Collection	Saioc et al.,2025
#35	Avoiding Deadlocks via Weak Deadlock Sets	Oriolo & Russo, 2025
#36	Occlusion-Aware Trajectory Planning With Quantified Risk Constraint for Deadlock Mitigation in Autonomous Driving	Chen et al.,2025

References

- N. Sharma and K. Kapoor, "Deadlock Prevention in Payment Channel Networks," in *IEEE Transactions on Network and Service Management*, vol. 21, no. 5, pp. 5164-5177, Oct. 2024, doi: 10.1109/TNSM.2024.3435484.
- J. Li, R. Chen, C. Wang, and Y. Chen, "A performance-impact based multi-task distributed scheduling algorithm with task removal inference and deadlock avoidance," Springer, 2023.
- B. Yang and H. Hu, "Maximally Permissive Deadlock and Livelock Avoidance for Automated Manufacturing Systems via Critical Distance," in *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3838-3852, Oct. 2022, doi: 10.1109/TASE.2021.3138169.
- Y. Huang, T. Wang, Z. Yin, E. Mercer and B. Ogles, "Improving the Efficiency of Deadlock Detection in MPI Programs Through Trace Compression," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 1, pp. 400-415, 1 Jan. 2023, doi: 10.1109/TPDS.2022.3218346.
- Y. Lu, Y. Chen, Z. Li and N. Wu, "An Efficient Method of Deadlock Detection and Recovery for Flexible Manufacturing Systems by Resource Flow Graphs," in *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1707-1718, July 2022, doi: 10.1109/TASE.2021.3114498.
- Zajac, Jerzy & Małopolski, Waldemar. (2021). Structural on-line control policy for collision and deadlock resolution in multi-AGV systems. *Journal of Manufacturing Systems*. 60. 80-92. 10.1016/j.jmsy.2021.05.002.
- Z. Zhang, G. Liu, K. Barkaoui and Z. Li, "Adaptive Deadlock Control for a Class of Petri Nets With Unreliable Resources," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 5, pp. 3113-3125, May 2022, doi: 10.1109/TSMC.2021.3062469.
- E. E. Ugwuanyi, M. Iqbal and T. Dagiuklas, "A Comparative Analysis of Deadlock Avoidance and Prevention Algorithms for Resource Provisioning in Intelligent Autonomous Transport Systems Over 6G Infrastructure," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 7444-7461, July 2023, doi: 10.1109/TITS.2022.3169424.

- Daszczuk, Wiktor. (2021). Non-exhaustive Verification in Integrated Model of Distributed Systems (IMDS) Using Vagabond Algorithm. 10.1007/978-3-030-76773-0_4.
- Shanu, Saurabh & Sastry, Hanumat & Marriboyina, Venkatadri. (2021). Optimal solution approach on large scale data to avoid deadlocks in resource allocations. *Materials Today: Proceedings*. 47. 10.1016/j.matpr.2021.06.357.
- Talaat, Marwa & Saleh, Ahmed & Moawad, Mohamed & Zaki, John. (2023). Fog computing effective load balancing and strategy for deadlock prediction management. *Ain Shams Engineering Journal*. 14. 102561. 10.1016/j.asej.2023.102561.
- L. Yang *et al.*, "Optimal Hub Placement and Deadlock-Free Routing for Payment Channel Network Scalability," *2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS)*, Hong Kong, Hong Kong, 2023, pp. 692-702, doi: 10.1109/ICDCS57875.2023.00087.
- X. Fan, H. Hu, B. Yang, Y. Liu and G. He, "Event Circuit Structures for Deadlock Avoidance in Flexible Manufacturing Systems," in *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 1, pp. 597-610, Jan. 2023, doi: 10.1109/TASE.2022.3163854.
- E. Taheri, S. Pasricha and M. Nikdast, "DeFT: A Deadlock-Free and Fault-Tolerant Routing Algorithm for 2.5D Chiplet Networks," *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Antwerp, Belgium, 2022, pp. 1047-1052, doi: 10.23919/DATE54114.2022.9774617.
- Jinpeng Zhou, Hanmei Yang, John Lange, and Tongping Liu. 2022. Deadlock prediction via generalized dependency. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2022)*. Association for Computing Machinery, New York, NY, USA, 455–466.
- Giuliano G. Porciúncula, Marcone I. Sena-Junior, Luiz Felipe C. Pereira, André L.M. Vilela, Consensus effects of social media synthetic influence groups on scale-free networks, *Chaos, Solitons & Fractals*, Volume 197, 2025, 116479, ISSN 0960-0779
- Yuan F, Huang X, Zheng L, Wang L, Wang Y, Yan X, Gu S, Peng Y. The Evolution and Optimization Strategies of a PBFT Consensus Algorithm for Consortium Blockchains. *Information*. 2025; 16(4):268.
- Zhu, Jing & Lu, Chengfang & Li, Juanjuan. (2025). Secure Consensus Control on Multi-Agent Systems Based on Improved PBFT and Raft Blockchain Consensus Algorithms. *IEEE/CAA Journal of Automatica Sinica*. 12. 1407-1417. 10.1109/JAS.2025.125300.
- Vazquez Trejo, J. A., Ponsart, J. C., Adam-Medina, M., & Valencia-Palomo, G. (2024). Fault-tolerant observer-based leader-following consensus control for LPV multi-agent systems using virtual actuators. *International Journal of Systems Science*, 56(8), 1816–1833.
- S. Zhou, Y. Liu, X. Lu, W. Li and J. Long, "Dynamic Event-Triggered Cluster Consensus for Multiagent Systems Under DoS Attacks With Antagonistic Interactions," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 55, no. 6, pp. 4362-4374, June 2025, doi: 10.1109/TSMC.2025.3549726.
- Zhao, Chonghe & Zhou, Yipeng & Zhang, Shengli & Sheng, Quan & Zhang, Yang & Wen, Shiting. (2025). ExCliques: An Express Consensus Algorithm for High-Speed Transaction Process in Blockchains. 10.48550/arXiv.2501.15289.
- A. Miele, M. Lippi and A. Gasparri, "A Distributed Framework for Integrated Task Allocation and Safe Coordination in Networked Multi-Robot Systems," in *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 11219-11238, 2025, doi: 10.1109/TASE.2025.3532023.
- R. ROMDHANI, O. MOSBAHI and M. KHALGUI, "Using Deep Learning for Deadlock Detection in Intelligent Software Systems," *2025 International Symposium on Innovative Informatics of Biskra (ISNIB)*, Biskra, Algeria, 2025, pp. 1-6, doi: 10.1109/ISNIB64820.2025.10983618.
- Lichen Pan, Juncheng Liu, Yongquan Fu, Jinhui Yuan, Rongkai Zhang, Pengze Li, and Zhen Xiao. 2025. Comprehensive Deadlock Prevention for GPU Collective Communication. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys '25)*. Association for Computing Machinery, New York, NY, USA, 541–557.
- Bretas, A. & Mendes, A. & Chalup, Stephan & Jackson, M. & Clement, R. & Sanhueza, C.. (2023). Addressing deadlock in large-scale, complex rail networks via multi-agent deep reinforcement learning. *Expert Systems*. 42. 10.1111/exsy.13315.

- Kaid, Husam & Al-Ahmari, Abdulrahman & Alqahtani, Khaled N. & Dabwan, Abdulmajeed & Nasr, Mustafa & Alhaag, Mohammed. (2025). An Internet-of-Things-Based on ILPP, Petri Nets, and Artificial Neural Networks for Controlling Tool Failures in Flexible Manufacturing Systems under Complex Operational Conditions. *IEEE Access*. PP. 1-1. 10.1109/ACCESS.2025.3544772.
- Lee, Wonjong & Sim, Joonyeol & Kim, Joonkyung & Jo, Siwon & Luo, Wenhao & Nam, Changjoo. (2025). Merry-Go-Round: Safe Control of Decentralized Multi-Robot Systems with Deadlock Prevention. 10.48550/arXiv.2503.05848.
- Wang, Fenglan & Shu, Xinguo & He, Lei & Zhao, Lin. (2025). Learning Distributed Safe Multi-Agent Navigation via Infinite-Horizon Optimal Graph Control. 10.48550/arXiv.2506.22117.
- Liu, Hongshuo & Yue, Ming & Liu, Minghao & Su, Longfei & Zhao, Xudong. (2025). Two-layer path planning framework for WMRs in dynamic environments: Optimized ant colony algorithm and dynamic window approach. *Transactions of the Institute of Measurement and Control*. 10.1177/01423312241296969.
- Z. Li, C. McKie, H. Wang, H. Shakeel and R. Ranjan, "Service-Oriented Evolution of Modern AI: A Position Paper," *2025 IEEE International Conference on Software Services Engineering (SSE)*, Helsinki, Finland, 2025, pp. 39-49, doi: 10.1109/SSE67621.2025.00014.
- B. Flammini, M. Ç. Sipahioğlu and F. Amigoni, "A Deadlock-Free Solution for Multi-Agent Pickup and Delivery in Presence of External Agents," *2025 European Conference on Mobile Robots (ECMR)*, Padova, Italy, 2025, pp. 1-6, doi: 10.1109/ECMR65884.2025.11163390.
- M. Masoumi, E. Carmona-Cejudo, I. de Miguel, C. Torres-Pérez and R. J. Durán Barroso, "Dynamic Joint Scheduling of Movement and Data Processing Tasks Using Extreme-Edge Computing in Multi-AGV Scenarios," in *IEEE Open Journal of the Industrial Electronics Society*, vol. 6, pp. 1312-1334, 2025, doi: 10.1109/OJIES.2025.3593281.
- Z. Zhang, Y. Zhang, X. Zhao, B. Tao and H. Ding, "Deadlock-Aware Control for Multirobot Coordination With Multiple Safety Constraints," in *IEEE Transactions on Robotics*, vol. 41, pp. 5209-5228, 2025, doi: 10.1109/TRO.2025.3600159.
- Saioc, Georgian-Vlad & Lee, I-Ting & Møller, Anders & Chabbi, Milind. (2025). Dynamic Partial Deadlock Detection and Recovery via Garbage Collection. 244-259. 10.1145/3676641.3715990.
- Oriolo, Gianpaolo & Russo Russo, Anna. (2024). Avoiding Deadlocks via Weak Deadlock Sets. 10.48550/arXiv.2410.05175.
- Z. Chen, W. Liu, L. Xiong, Z. Yu and C. Tang, "Occlusion-Aware Trajectory Planning With Quantified Risk Constraint for Deadlock Mitigation in Autonomous Driving," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 8, pp. 11489-11503, Aug. 2025, doi: 10.1109/TITS.2025.3578974.
- M. H. Abdul-Hussin, "Optimal Solution Implemented for Deadlock Problems of the FMSs Modeled and Simulating with Petri Nets," *2025 IEEE Integrated STEM Education Conference (ISEC)*, Princeton, NJ, USA, 2025, pp. 1-8, doi: 10.1109/ISEC64801.2025.11147245.
- Chuang, Wen-Yi & Ching Yun, Tseng & Tan, Kuang-Hsiung & Pan, Yen-Liang. (2025). Design of a Novel Transition-Based Deadlock Recovery Policy for Flexible Manufacturing Systems. *Processes*. 13. 1610. 10.3390/pr13051610.
- Mordido, Andreia & Pérez, Jorge. (2025). Deadlock-free Context-free Session Types. 10.48550/arXiv.2506.20356.
- Wang, Siyi & Feng, Yanxiang & Li, Xiaoling & Zhang, Guanghui. (2025). DAFSP with limited assembly buffers: A deadlock-free coding-decoding paradigm and hybrid cooperative co-evolutionary approach. *Swarm and Evolutionary Computation*. 99. 102155. 10.1016/j.swevo.2025.102155.
- Chandra, Rohan & Zinage, Vrushabh & Bakolas, Efstathios & Stone, Peter & Biswas, Joydeep. (2025). Deadlock-free, safe, and decentralized multi-robot navigation in social mini-games via discrete-time control barrier functions. *Autonomous Robots*. 49. 10.1007/s10514-025-10194-8.
- W. Lin, X. -Y. Li, J. -M. Chang, D. Xiang and X. Jia, "Link/Switch Fault-Tolerant Hamiltonian Path Embedding in BCube Networks for Deadlock-Free Routing," in *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 5, pp. 4829-4846, Sept.-Oct. 2025, doi: 10.1109/TDSC.2025.3553617.
- Jaramillo, Juan & Pérez, Jorge. (2025). Contrasting Deadlock-Free Session Processes (Extended Version). 10.48550/arXiv.2504.15845.

- Surajit Das, Abhijit Das, and Chandan Karfa. 2025. Developing Deadlock-Free Routing Algorithms in Torus NoC: A Formal Approach. *ACM Trans. Embed. Comput. Syst.* 24, 5s, Article 113 (September 2025), 26 pages. <https://doi.org/10.1145/3762650>
- Soliman, Karim & Li, Chunfeng & Shi, Feng. (2025). Reactive deadlock avoidance based on Focus Routing Graph classification for Triplet-Based Architecture Network-on-Chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. PP. 1-1. 10.1109/TCAD.2025.3579322.
- Baumeister, Tom & Jacobs, Swen & Sakr, Mouhammad & Völz, Marcus. (2024). Automatic WSTS-based Repair and Deadlock Detection of Parameterized Systems. 10.21203/rs.3.rs-4635496/v1.
- Garg, Kunal & Hamilton, Sera & Fan, Chuchu. (2024). Deadlock Resolution of Connected Multi-Agent Systems using Hierarchical Control. 1275-1282. 10.1109/CDC56724.2024.10886421.
- Heuvel, Bas & Sulzmann, Martin & Thiemann, Peter. (2025). Partial Orders for Precise and Efficient Dynamic Deadlock Prediction. 10.48550/arXiv.2502.20070.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.