5-2022

Optimized Damage Assessment and Recovery through Data Categorization in Critical Infrastructure system.

Shruthi Ramakrishnan
University of Arkansas, Fayetteville

Optimized Damage Assessment and Recovery through Data Categorization in Critical Infrastructure system.

Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Science

by

Shruthi Ramakrishnan
Anna University
Bachelor of Engineering in Computer Science and Engineering, 2016

May 2022
University of Arkansas

This thesis is approved for recommendation to the Graduate Council.

_____
Brajendra Nath Panda, PhD.
Thesis Director

_____
Ukash Nakarmi, PhD.
Committee Member

_____
Lu Zhang, PhD.
Committee Member

ABSTRACT

Critical infrastructures (CI) play a vital role in majority of the fields and sectors worldwide. It contributes a lot towards the economy of nations and towards the wellbeing of the society. They are highly coupled, interconnected and their interdependencies make them more complex systems. Thus, when a damage occurs in a CI system, its complex interdependencies make it get subjected to cascading effects which propagates faster from one infrastructure to another resulting in wide service degradations which in turn causes economic and societal effects. The propagation of cascading effects of disruptive events could be handled efficiently if the assessment and recovery are carried out as quickly as possible. To be an efficient system, it should reduce the impact by reducing the number of nodes undergoing service degradation. In general, the damage assessments include accessing and assessing log information which is very costly in terms of time spent and IO reads. A generic model thus should be very optimal in suggesting smaller number of assessments as possible and at the same time reduce the number of nodes undergoing unnecessary service degradations. This thesis investigates the CI systems in depth to optimize the damage assessment and recovery process so that it could help in resuming the operations of as many safe data items as quickly as possible. It also focuses on reducing the load imposed in terms of number of nodes towards damage assessment and recovery procedures through the proposed optimization model. The quick identification and categorization of the type of data items as damaged, undamaged, or skeptical within the impacted CI system is the key factor which makes this model highly efficient and helps this model to project better performance. The developed model and its algorithm have been implemented on a simulated data and environment whose results shows that the proposed model performs well in terms of time, speed, accuracy, complexity, efficiency, and performance.

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

Critical infrastructure (CI) systems are large scale distributed systems providing wide variety of services to large scale of areas which are of high importance contributing towards the wellbeing of people and nations. The economy and security are increasingly dependent on critical infrastructure spectrums [9]. They are highly coupled, complex, interdependent, and interconnected systems which are highly vulnerable to the disruptive events which cascade the damage from one system to another resulting in the contamination of damage among multiple infrastructures [1]. The critical infrastructures when exposed to attacks or damage results in cascading spread of damage throughout its multiple interconnected infrastructures resulting in devastating effects economically. There are numerous past incidents and recent reports such as [31], [32], [33], [34], [35] which shows that there are attacks which are frequently occurring in many critical infrastructure systems.

The initial damage when assessed and recovered by consuming long period of time results in increased damage and service degradations from its multiple infrastructures. Hence, quick assessment and recovery of the damage within a critical infrastructure holds greater significance. The initial damage and its contamination should be quickly identified and recovered so that the cascading effects of the disruptive events could be reduced to a greater extent resulting in the reduced service degradation or downtime. The contamination from the initial damage or failure in a critical infrastructure is due to the interdependencies among multiple infrastructures with in a single critical infrastructure. The contaminated or damaged data when used for the update in another infrastructure makes it get its own data damaged and becomes the source of damage towards its dependent infrastructures. Thus, the damage spreads throughout the dependent infrastructures of the initial damage in a critical infrastructure system like a chain reaction if not

controlled and handled at earlier stage. Thus, quick identification of damage and recovery holds vital importance in critical infrastructure system protection.

The understanding and support of management of interdependencies among multiple infrastructures in critical infrastructures should be improved to handle large scale, disruptive events. The significance of the understanding interdependency relationships and their managements could be illustrated from the World Trade Center in New York City on September 11, 2001 [1]. This paper proposes that the critical infrastructure system and its data items could be modeled in a particular way to understand its structure and then the damage assessment and recovery methodologies could be implemented on them based on the CI system understanding.

As described in [24], critical infrastructures are like the body's vital organs which needs to perform their own roles so that human body could function efficiently and painlessly. The US Department of Homeland Security [25] declares that such systems are "so vital to the United States that their incapacity or destruction would have a debilitating impact on our physical or economic security or public health or safety". Critical infrastructure protection holds an important role in ensuring uninterrupted delivery for the economic and social wellbeing in many sectors. [6] emphasizes on critical infrastructure protection and addresses concerns related to the nation's critical computer-dependent infrastructures protection from computer-based attacks and disruption. The interdependencies and the increasing reliability among multiple infrastructures makes the critical infrastructure constitute an unbounded system which might be subjected to failures or faults which in turn might propagate from one system to another. In such cases, the critical infrastructures should impose security which is a real challenge and thus it requires novel tools and methodologies [8].

The interdependencies among multiple infrastructures makes the disruptive events to cast cascading effects from one system to another. A good understanding of interdependencies could help in interpreting the risks accumulated with the cascading effects [11]. The multi order dependency related risk assessment [12] among the critical infrastructures contributes towards their protection and resilience. This paper also suggested that the cascading effects usually ends after reaching certain nodes or data items in CI system due to the presence of contingency and countermeasures. But mostly these does not account for critical infrastructures operational mode changes which includes instances such as stressed, crisis and recovery modes [13]. The cascading effects of disruptive events in CI systems has devastating consequence to multiple infrastructures one such incident includes domino effects of the electric power disruptions in California in 2001 [14]. The dependencies vary as per analysis levels and their examinations has different approaches which finds its applications in many areas. One such instance would be the application within same or different sectors of a country [17]. The types of multi-dependency structures for both cyclical and linear dependencies are demonstrated in [20] which focuses on CI components. There are special types of dependencies which are difficult to identify, one such includes social dependencies whose example includes a disruption in the transportation sector which may cascade in wireless communication networks [18]. The related work in identification and modelling of dependencies uses sector specific methods. Such interdependency models fall into six broad categories as per [19].

This thesis investigates the problems related to delayed or increased load on assessments when addressing a failure in CI system. It proposes an optimized damage assessment and recovery which is generic to any critical infrastructures subjected to attacks or disruptive events. There are many models developed for damage assessment and recovery in critical infrastructure protection.

These models include log-based models, which relies on the log information of the impacted CI system for the complete assessment and recovery process. The other such model includes dependency graph model, where the CI system modeled as a transaction tree structure helps to identify the dependent data items of the damaged data item(s) which are then in turn subjected to the assessment and recovery process. The above-mentioned models have their own benefits and drawbacks when compared to the proposed model which is explained in detail in the further sections.

The proposed generic optimized damage assessment and recovery model implements quick identification of the damaged data items, its respective contaminated data items and the undamaged data items using its zoning technique before subjecting the data items of the impacted CI system to the damage assessment and recovery. This categorization of nodes and then subjecting the respective nodes to the assessment and recovery process helps to save lots of time, and reduces load imposed on costly assessment procedures. This proposed model includes zoning process which is performed even before accessing and assessing log information for determining if a data item is damaged or not. Accessing and assessing log information is time consuming and involves costly IO reads from storage. This zoning process helps to reduce the load subjected to such costly operations and thus optimizes the damage assessment and recovery process in an impacted CI system. This model ensures that only guaranteed to be damaged items to be subjected to damage recovery by skipping the assessment process and only guaranteed to be undamaged items to be free from unnecessary assessment and recovery process. The efficiency of this model totally relies on its optimized damage assessment and recovery process when addressing an impacted CI system and ensuring that the CI system is free to resume its operations only when it is guaranteed to be safe to operate. The goal of this proposed model is to make as many data items

available as possible which are safe to operate and as soon as possible in an impacted CI system, thus preventing many data items or nodes to undergo unnecessary service downtimes.

The rest of the thesis is organized as follows. Chapter 2 addresses the background and related work. Chapter 3 explains and illustrates the proposed model and its algorithm. Chapter 4 discusses the simulation experiments and their respective results followed by Chapter 5 which compares the proposed model against log model and dependency graph model. Chapter 6 presents conclusion and future work.

2 BACKGROUND AND RELATED WORK

This thesis aims to propose optimized and generic damage assessment and recovery model for critical infrastructure systems. This section describes some of the related work in some of the publications that are relevant to our proposed framework and model.

Critical Infrastructure systems are highly coupled, interconnected systems vulnerable to large scale disruptive events that propagate from one system to another by degrading their services resulting in deviations in delivering services far apart from prescribed levels. Hence, critical infrastructure protection plays a vital role. [1] discusses that the impact of disruptive events could be minimized by either mitigating against or responding to them through understanding the interdependencies, modelling the interdependencies, and incorporating them to support for the management of CI interdependencies by restoring loss or restoring services after disasters. It also includes the analysis concerning eight infrastructures, as defined in the [2] President's Commission on Critical Infrastructure Protection (1997) such as emergency services, transportation, information and communications, electric power, banking and finance, gas and oil production, storage, and transportation, water supply systems, and government. [3], [4] presents an incident by illustrating impacts on electrical and telecommunications infrastructures due to World Trade Center attack. [5] focuses on the understanding of the vulnerabilities of the critical infrastructures which could help in better way of controlling the cascading failures due to the disruptive events. It also discusses the significance of services offered by CI systems being of real value to public.

There are papers which focused on modelling of the critical infrastructures, implementing risk assessment and handling of cascading effects of failures. [23] presents six broad categories of dependency models which includes sector- specific and more general methods. [7] focuses on the issues in Information and communication technologies infrastructures as critical infrastructures

are highly reliant on them. It also focuses on system modelling methodologies and proposes an implementation model that enhances CI protection and tackle interdependencies through policy-based management, canonical architecture approach and inter-dependencies with the virtualization of CI. [10] presents a model which includes assessment of the cumulative security risk of cascading threats due to high order dependencies between infrastructures and assessment of risks arising from complex situations involving multiple cascading failures triggered by major or concurrent common-cause events. This model identifies and evaluates common cause threats for studying the cascading effects caused by large scale common cause events. This model helps in assisting decision makers in identifying optimal approaches to mitigate risk. This paper focused on common cause failures because [15] provides statistics indicating the cascading effects of disruptive events are not very well documented and are rare. [16] proposes a method to identify and assess multi order dependencies. It assesses the effects of disruptive events to consequent infrastructures; it also identifies and prevents security threats of very high impact from a macroscopic view which are hard to identify when only first order dependencies are identified. [22] examines the risk and criticality relationship by analyzing the similarities and differences in terms of scope, aim, impact, threats, and vulnerabilities and proposes a generic risk-based criticality analysis methodology. It also provides detailed list of impact criteria for criticality level of infrastructures assessments. The holistic criticality assessment technology for the development of an infrastructure plan in multi sector or national level is proposed in [21] whose proposal is used to assess sector wide or intra sector security risks.

As the critical infrastructures has become vulnerable to attacks in recent years especially on their data systems, post attack the system should be capable of restoring its services from downtime as quickly as possible. The critical infrastructures which use cyber physical systems can

result in new vulnerabilities as described in [26]. The analysis of disruptive events by characterizing them through different impact factors and risk assessments are discussed in [27] using system dynamics modeling. [28] proposes a model of infrastructure system which quantifies damage by using elements with varying criticality and linkages to indicate dependencies and interdependencies. These models aim to mitigate the probability of attacks in CI system prior the attacks, there are models which helps in recovery during attack. [29] proposes such algorithm which breaks down demand flows into simple problems to restore the damaged element paths. To prioritize which damaged nodes need to be recovered first they have used a centrality metric for ranking them in [30] which also proposed to make use of this centrality metric for making repair decisions as future work.

There are papers which discussed the damage assessment and recovery process for impacted CI system due to disruptive events. [37] proposes a new logging protocol which records all appropriate and required information for the complete recovery and repair of the damaged or impacted database post affected transactions. This information includes different predicates and various statements used in the transaction. This paper proposes an algorithm to incorporate these in the log. The damage assessment and recovery algorithm are based on this new log and is performed concurrently. [38] provides techniques to make an assessment on the damaged data and then recover the impacted data to return to their consistent state post the discovery of attack. Data dependency model is used to perform damage assessment to obtain the precise information about the damaged portion of the database. This paper proposed two algorithms, the first one performs damage assessment and recovery simultaneously. The second algorithm improves efficiency by separating the two processes. The two algorithms allow blind writes on data items thus allowing damaged items to be recovered automatically. [39] proposes a phase which uses software or device

to observe the system for any malicious activity or policy violation. As detection system fails sometimes to detect several malicious transactions on time, it leads to data damage. Hence, this paper introduces another phase namely damage assessment and recovery which ensures integrity and system data availability. This phase also identifies further affected transactions and ensures database returns to consistent state. This paper introduced a novel architecture model for applying fog technology to smart cities by using unique method for assessing and recovering damaged data. [40] presents a method to repair data objects that prioritizes quick recovery for important system components. This paper includes three graphs to represent entire system, what changes the system made after an attack and cascading damages because of those changes. This paper proposes an algorithm to optimally schedule repairs to identify damage paths using those graphs that impact most critical system nodes. This proposed algorithm is widely applicable for critical infrastructure system protection where services need to be quickly restored to avoid societal or economic disruptions.

3 PROPOSED MODEL

This section includes the detailed explanation and illustration of the proposed model. It includes contributions of this model, input, outcomes, assumptions, components of the model such as matrix, directed graph, zoning process, algorithm of the proposed model, and its detailed functions.

3.1 Contributions

The primary goal and contributions of this proposed model includes the following:

1. To make as many undamaged data items available as possible before assessing the log information to determine the damage spread which is time consuming especially with heavy IO reads on storage.
2. To declare the data items into one of the three categories as damaged or undamaged or skeptical as soon as possible before imposing intense damage assessment on specific data items.
3. To develop a generic model of quick and optimized damage assessment and damage recovery that could be applicable to any kind of critical infrastructure systems.
4. To reduce the service downtime of all the data items in a critical infrastructure system to a greater extent by quick categorization of nodes and performing respective actions on each category.
5. To immediately identify the undamaged data items and resume their operations quickly as they are guaranteed to be safe to operate after the location and time of initial damage is reported.
6. To immediately identify damaged or contaminated data items and reduce the cascading effects of the disruptive effects of the damaged data items in the impacted CI system.

7. To make sure that only the rest of the skeptical data items which are either contaminated or safe data items to be subjected to damage assessment. In other words, damage assessment is only performed on skeptical nodes which are a narrow set of data items rather than imposing damage assessment on all data items of the impacted CI system.
8. To make use of the instantly available latest learned data which includes the most recent updates about the critical infrastructure systems for damage assessment and recovery.

3.2 Assumptions

The following assumptions are being made for this proposed model:

1. Update formula repository.
2. Log Information.

3.2.1 Update Formula Repository

This assumption for this proposed model includes the information collection regarding the updates and the respective formula used by the user for making updates towards a receiver data item by making use of one or more of the other data items. This information is vital for damage recovery process. For instance, let's assume a data item A whose initial value before damage was 20 and post damage the data item was updated by assigning it with wrong value of 50 using damaged value of data item B. During damage recovery process it is really a difficult task in determining the formula used to make the update that results in the expected value. In other words, it is a difficult task to determine f(B) to get the expected value of A. A dedicated system or team which retains the formula of updates being made by the user on a data item as a function of the dependent data items based on the latest and current CI structure maintained by a system is expected. The assumption about this system is made in such a way that whenever a part of the graph structure in terms of nodes and edges given to the system, it readily delivers the formula of updates being made

by using them towards their receiver nodes respectively. This system also maintains the highly adaptive and dynamic data based on modifications of the graph structure. In other words, whenever a new node or edge is being added or removed from the CI system or time of last update are modified in the CI system, their respective details regarding the update formulae are also being updated accordingly in the repository. This ensures a complete synchronization of data between the current Critical Infrastructure system and update formula for them. Hence this makes sure that there is no redundancy in the data being maintained. It is also assumed that the update formula of a requested node could be retrieved at any instant of time when requested to the team for the respective CI system with the required input data.

3.2.2 Log Information

Another assumption regarding the readily available information includes system log data which is being constructed based on the logging protocol as described. A logging protocol is to be followed to track the changes or activities being performed by any data item in the critical infrastructure system. [36] uses such dynamic log pattern creation with good accuracy and it is also adaptive as per changes in the CI system. It should be maintained by a team by tracking all the latest observations or activities of the data items in CI system. As per this logging protocol, it maintains information such as the parent data item ID whose data was used to make the update on this data item, the ID of the data item which was updated or modified or upon which any activity was performed, the old value or initial value of the data item before receiving the update and the time of the last update being made on it. The time of last update depicts the time until which the data item might have been updated. For instance, if a data item has time of last update as 5, then the data item has been updated until 5. In other words, there might have been an update at 1, 2, 3, 4 and last update happened at 5. The log information made until the reported point of time of initial

damage by IDS in the CI system is to be collected and worked upon. The log information contains the history of transactions data as well. The history of data includes the same source data item, recipient data item on which update was made using the source data item, initial value, and the time of each update. In other words, for every transaction that happened in the Critical infrastructure system, there is an entry for each time of update until the time of very last update of the same transaction. It is also assumed that the log data of the respective CI system is to be readily available when requested for any specific activity by a specific data item or at any specific time to the team especially when performing damage assessment and recovery.

3.3 Input and outcomes of the proposed model

This section discusses the type and details of the input data expected or required for the proposed model. It also includes the details of the outcomes that could be expected from the proposed model through its optimized damage assessment and recovery.

3.3.1 Input Data to the model

The IDS report includes the details of the process such as Process ID related to the initial damage(s) in the impacted CI system. The process ID could be used to retrieve the transaction(s) from the respective log file that includes the below mentioned details related to the initial damage(s) which are used for the proposed model implementation.

1. Time of initial damage and its details.
2. Point of initial damage representing the location details of the data item or node where the initial attack happened. It includes information about initially attacked node or data item information.

3.3.2 Outcomes of this model

The outcomes expected from this proposed model for a given CI system includes the following.

1. Quick identification of as many undamaged data items as possible to resume their operations upon discovery and data item categorization as damaged or skeptical as soon as possible even before accessing and assessing log information by saving time through optimized analysis and categorization.
2. The resumption of the quickly analyzed guaranteed to be undamaged nodes instantly after the initial and in-depth analysis based on the reported initial attack location and time.
3. Prevention of cascading effects of the initially reported attack to greater extent.
4. The identification of the damaged nodes and recovery of the nodes affected or contaminated by the spread of initial damage or by the counter effects of the initial damage.
5. The recovery of damaged nodes by resetting their data values as expected and enabling the proper functioning of the CI system as it would have been before damage. In other words, updating the damaged nodes in such a way that the system state would have been as if there was no attack.
6. The service downtime of the data items in the impacted CI system to be reduced to greater extent and increase service availability as much as possible.
7. In case of skeptical nodes, a detailed analysis is made through damage assessment to categorize them further as damaged or undamaged nodes and take actions on them accordingly.

3.4 Directed Graph of critical infrastructure system

To understand and interpret the more relevant, accurate, optimized, and efficient damage assessment and recovery there is a very high dependency on the most recently updated and latest knowledge of the critical infrastructure system. To understand the proposed model of damage assessment and recovery, the directed graph structure composed of nodes and edges of the

respective critical infrastructure system is used for explanation and better interpretation to visualize the analysis. In other words, the directed graph represents the interactions or dependencies among the nodes using edges. The nodes represent the data items in the CI system and the directional edges depict the relationship between them, the flow of information or update between them or dependencies between them. The edge weights are the time of last update made by the user by making use of data item(s) on the other data items based on the direction. The time factor is the main essence behind the entire representation. The complete flow includes the timely updates being made by the nodes from the root of the directed graph. The directed graph equivalent is being developed based on the current functioning of the critical infrastructure system. It is updated after every transaction being carried out anywhere within or around the critical infrastructure system. In other words, whenever an update is being made or if any node or edge is added or removed from the CI system, such changes are instantly recorded in the directed graph with respective associated factors. Th directed graph shows the updated graph by reflecting the changes such as addition or deletion of nodes or edges, change of the latest time of last update made by using the data items over the edge weights in the directed graph accordingly when a node is added or removed as a part of the CI system. The directed graph includes nodes with a single or multiple or without parent node and every node may or may not have a child node. The directed graph structure includes connected nodes with directed edges representing the flow or direction of every update made using a data item towards the child or receiver nodes at some recorded timestamp called time of last update which is represented as edge weight.

Nodes: Each node holds the details of the name of the data item it represents. As shown in figure 1 directed graph, it includes nodes with their respective data items details. Every node is connected to another node in the directed graph of CI system. For instance, a node P is connected to another

node Q which in turn is connected to the node S. The nodes P, Q, S here represents their respective data items. After the analysis is made as per the proposed model, the entire set of nodes in the directed graph gets allocated to one of the three zones. The point of initial damage nodes and the contaminated nodes which gets allocated to damaged zone are represented with filled in nodes or referred to as red nodes, the benign nodes in undamaged zone are represented as nodes unfilled or referred to as green nodes and the skeptical nodes which may or may not have been contaminated in gray zone are represented using nodes filled with straight line patterns or referred to as gray nodes.

Edges: The directed graph includes the directional edges depicting the direction of flow of data or information at the respective time frames. The edges also depict the dependency among the data items or nodes. The directed graphs include edges which could be bidirectional (edge connecting nodes: U, X) edges, self-updates (edge for node: N) and it also includes the cycles among the nodes (edges connecting nodes: Z, T, S) within the graph. The edges are not only analyzed based on the flow of information but also highly characterized based on the time frame of last updates being made by using the nodes towards their respective child or receiver nodes. The edges do not depict the flow of damage as mentioned earlier. The edges in other words also define that there may have existed some dependencies before the time frame of attack but not after that. Every node may have a single parent, multiple parents, or no parents as well. At the same time, every node may have no child or single child or multiple children.

Edge weights: The edges in the directed graph of the CI system are being assigned with the edges weights which represent the time stamps of the last update being made on the receiver node by using the data of the source nodes. The time of last update is the time until which the data item might been updated or the last time the data item was updated. The time of last update is a

timestamp represented using increasing sequential numbers irrespective of the day or nighttime. For instance, the time of last update includes values starting from 1 and includes the incremental values such as 2, 3, …, 30, etc. For instance, in the figure 1 the edge PQ has been assigned with the value 8 which represents that the node Q has been lastly updated by the user at timestamp 8 using the data value of node P. In other words, there might have been updates on node Q until 8 and last update was made at 8 by using P. It could also interpret the meaning behind the edges among the nodes. In figure 1, the node O has been updated based on the data values of the node R and A at the same time stamp 8 such that O = f (R, A). This depicts that an update on a data item could be dependent on multiple data items and at the same timestamp.

Zones: The directed graph equivalent of the CI system includes multiple zones post analysis using this proposed model before damage assessment. It contains different sections including various kinds of nodes determined based on the damage spread or contamination by performing quick and detailed analysis. There are three types of zones into which the nodes of the directed graph of the CI system under consideration is classified into for damage assessment and recovery process in this model. It includes undamaged, damaged, and gray zones. Only damaged nodes or contaminated nodes are in damaged zone, skeptical nodes which are unsure of being damaged or not are in gray zone and benign nodes which are safe to resume their operations in undamaged zone which would be explained in detail in further sections.

Adaptability: The directed graph structure is highly adaptive as it gets remodeled every time a new node or edge is added or discarded from the system. The updates are reflected instantly so that all the relationships between the nodes and their respective details such as time of last update as edge weights are maintained as per the current transactions in the CI system. In other words, after every transaction within the CI system, its respective related information needed for this

proposed model such as matrix equivalent of CI system under consideration, update formula repository and log information are updated accordingly.

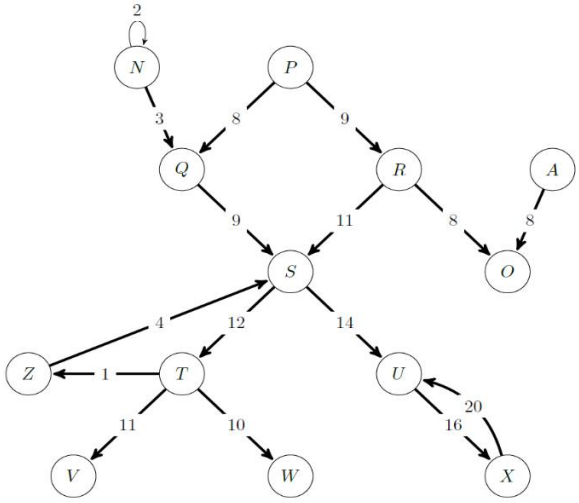

Figure 1. Directed graph of the CI system

The time of last update represents the last time an update is being made on the node by the user using other data item values or in other words, it is the last time until which the update has been made by the user on the data item using other data item or node's values. For instance, as shown in figure 2 children Y and Z of node X are being lastly updated using the data item X at 8 and 10 respectively. It represents that the last time an update has been made on data item Y based on data item X was at 8 and similarly the time of last update made on node Z using data item X was at 10. Based on the factor 'time of last update' the graph interpretation and analysis would take place in this model which would be explained in further sections. It could also be seen that a data item could be updated using a single or multiple data items at the same time or different times. In other words, a data item could be updated using multiple data items at the same time and a data item could also be updated using one parent data item at a specific time and by using the other parent at a different time. For instance, as shown in the figure 2, node Y has been updated using both the data items M and X at the same time 8 or in other words we could say update of data item or node

Y is a function of both the data items M and X represented as Y = f (M, X). It is also shown in the figure 2 that the data item Z has been updated using one of its parent node X at 10 and by using its other parent data item W at 9 which could be represented as Z=f(X) and Z=f(W).

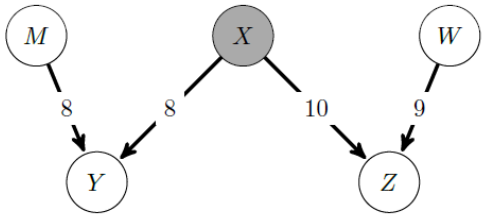

Figure 2. A simple directed graph

The entire directed graph structure from top to bottom represents that there was a dependency in the past, which is prior to the time of attack, but it does not depict the order of occurrence or the flow of the information at current instant of time. For instance, in the figure 3 with time of reported initial attack being at 6, the edge connecting the nodes O and R, depicts that there was a dependency among them before the time of attack, at this time frame post attack, there is no flow of information between them. Let's assume if node O was declared to be damaged at time 6, but node O has updated node R well ahead before time of attack at 5 and there might be no update on R using damaged values of O after that time frame, hence the edges may also describe that there existed a dependency between them prior but not now. The damaged value of damaged data item O after it has been declared to been damaged at 6 has been used to update node N which made it a damaged item with its time of damage being at 10. Hence, initially damaged node O has contaminated node N through update which is past the time of initial damage of O. With this understanding about the edges, it could be understood that the damage from damaged nodes O and N has not been spread to its immediately connected nodes at that time frame of analysis. Or in other words, the node R has not been contaminated by damaged nodes O and N because they made updates before the time of damage of O and time of damage of N. Hence, when nodes are

connected to a damaged it does not mean that they get the cascaded damage from their connected damaged nodes. The damage spreads from them if and only if there is a flow of update between them or the damaged data item has been used to update them post the time of damage of the dependent data item.

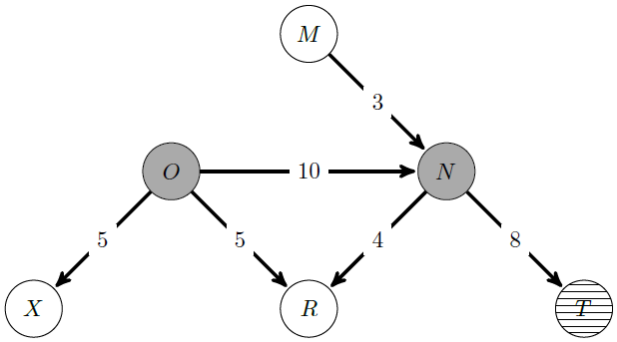

Figure 3. Time of last update for node categorization

It could be observed in the figure 3 that node N even though was not the initial point of damage, but was declared to be damaged, this is because the node N has been contaminated by initially damaged node O whose time of initial damage is at 6. Node N has been contaminated because O was used to update node N post the time of damage of its damaged parent O. Node R which is immediately connected to both the damaged nodes O and M has not been declared to be damaged because node R has been updated by using node O prior to the time of damage of node O and node R has been updated using damaged node N prior to the time of damage of node N. Thus, node R even though directly dependent upon the damaged node O and M for updates has been declared safe to resume its operation. It is also to be observed that the node T whose update depends on node N but has been declared to be a skeptical node as its unsure whether it has been contaminated from damage of node N or not. It is because node T has been updated lastly by node N at 8, but node N has been declared to be damaged anywhere until 10, hence it is unsure whether damaged value of node N has been used or not to update Node T. Let's consider two scenarios for

determining the state of node T. Scenario 1, if node T was updated using node N until 8, and since time of last update which made N to be damaged was 10 by O with time of initial damage at 6, let's assume if node N was damaged exactly at 7 by O and the damaged N was used to update T at 8 which is past the time of damage of N, in that case node T is guaranteed to have been contaminated by N, thus node T could have been a damaged node in this case. Scenario 2, if node T was updated using node N until 8, and since time of last update which made N to be damaged was 10 by O with time of initial damage at 6, let's assume if node N was damaged exactly at 9 by O and N was used to update T at 8 which is prior to the time of damage of N, in that case node T is guaranteed to not have been contaminated by N, thus node T could have been an undamaged node in this case. Since node T has both cases of being a damaged and undamaged node in two different scenarios, a detailed assessment is needed based on the exact time of damage of N which is to be accessed from log information during damage assessment. That is why node T has been declared to be skeptical at this moment and is to be assessed later. Thus, the directed edges do not entirely depict the perception of the flow of data or damage, but the flow entirely includes the time of last update as crucial factor in their definition. In other words, even though the entire graph structure looks connected, when the attack is identified at a particular node, it need not be cascaded from it towards all its connected nodes. There is a scenario where the connected nodes of the damaged node might have been updated well ahead the time the damage was identified at the source. Hence, time of last update is treated as a prime factor in our proposed model during categorization, damage assessment and recovery. There are multiple cases for the time of last update-based analysis for declaring the type of nodes as damaged nodes or not which would be described in detail in the following sections. It is to be observed that this proposed model does not make use of the history of the transactions during the analysis of node categorization, it only

considers the latest transaction with time of last update. This makes the proposed model to perform time efficient node categorization. If the entire history of transactions has been used for analysis, it would have been equivalent to accessing and assessing the log information which is time consuming and is not an optimized option for efficient damage assessment and recovery.

3.5 Matrix equivalent of Critical infrastructure system

The critical infrastructure system under consideration with its equivalent information is described using directed graph structure as mentioned in the previous section with data items as nodes and the relationship or flow of information of update or dependencies as the directional edges and the time of last update as edge weights. The other equivalent information related to the CI system would be the Matrix representation of the entire CI system. The matrix data for a particular CI system is generated using the tool or component which makes use of details such as number of nodes, edges, timestamps, history of transactions. This tool makes sure that the matrix it creates for a CI system is in sync with the directed graph structure of the CI system which is used for the interpretation. The matrix generation tool makes sure that the latest transactions or updates are recorded in the matrix with the relevant details. Whenever a new node (data item) or edge(dependency) or edge weight (time of last update) has been created or removed, such changes must be reflected on to the matrix data being maintained. The tool is expected to deliver the latest matrix data when requested for a particular CI system when IDS has reported the attack. There might be multiple disjoint directed graphs for a single Critical Infrastructure system. For each directed graph there exists an equivalent matrix for it. Hence for a single critical infrastructure system, there might be multiple disjoint matrices. In other words, there might be one or matrices which are not related or dependent with each other for a single CI system. The respective matrices for a CI system are generated instantly by the tool with the latest changes or transactions being

updated and would be delivered when requested at a specific time frame such as when IDS report has been received regarding the initial damage in a CI system. For this proposed model, the matrix equivalents of the directed graph containing the reported to be damaged node is to be analyzed and are gathered once the IDS report regarding the initial damage has been received. As mentioned before, there might be multiple matrices for a single CI system and the matrices to be considered for analysis using this proposed model are the ones containing the reported initial damages. The figure 4 represents a directed graph structure of a specific CI system whose equivalent matrix as shown in figure 5, is reported to have the initial damage by IDS.

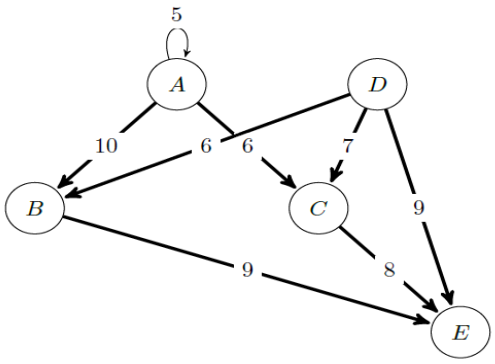

Figure 4. Directed graph of a CI system

The matrix data equivalent of the CI system has parents as rows and children as columns. For instance, in the figure 5 for a node A its entire rows represent its parents such as A (self-loop), similarly for node B its entire row represents its parents such as node A and D. Similarly, entire column for a node represents its respective children. For node A its entire column represents its children such as nodes A, B, C. These could be observed in the directed graph as shown in figure 4. The matrix values, M[i][j] for node i with parent j is the time at which node i has been lastly updated by using the data values of node j. For instance, the matrix value M[C][A] represents that the node C has been lastly updated by using the data values of node A at time stamp 4. Hence, matrix values are the time of last updates made on every node by other nodes respectively in the

CI system. The figures 4 and 5 show the directed graph structure and its equivalent matrix. For every CI system, there might be multiple equivalent matrices. For instance, if a CI system has in total of three separate set of clusters of connected data items, then there would be three directed graph structures and for every directed graph there is a respective equivalent matrix, thus in this case there exists three equivalent matrices for that CI system.

Parents					
Children	A	B	C	D	E
A	5				
B	10			6	
C	6			7	
D					
E		9	8	9	

Figure 5. Matrix equivalent of the directed graph

3.6 Zoning and its categorization process

This is the most significant process of this proposed model which helps us to optimize the time spent and load imposed on the damage assessment and recovery. This step includes the zoning or categorization of the nodes of the directed graph structure based on the initial point of damage location in its matrix. Once the initial damage is reported and the equivalent matrices containing the initial damages of the impacted CI system is gathered, the nodes of the matrix or matrices under consideration are split into three zones based on the time of last update being made on a node, type of nodes, its location, association, outgoing links and spread of damage through directed edges from the damaged nodes based on time of last updates. The resultant of zoning process includes three zones of the latest matrix data of CI system under consideration which are described in detail in the below sections. It is important to note that the zoning of the nodes of the matrix under consideration is being made once IDS has reported the damage and CI related information as mentioned before are collected. This process categorizes the nodes of the directed graph or

matrix into the respective zones based on the time of last update of the nodes or flow of information prior and post the reported time of damage. Hence, most of the nodes which even though directly or indirectly related to damaged nodes are still marked as undamaged as they would have been updated at the earlier time frames well ahead before the initial damage. This process helps us to minimize the service downtime of the data items in the impacted CI system. Its primary goal is to relieve or release as many data items as possible and as soon as possible which are confirmed to be safe nodes from being undergoing unnecessary service downtime after making logical and detailed analysis based on gathered information. To achieve all these goals the zoning process is to be carried out before damage assessment and recovery so that as many data items as possible are allowed to resume their operations or in other words, the data items availability could be increased. Before understanding the process of zoning, the below sections help to understand what the three different types of zones are, what are the nodes that gets allocated to them, based on which analysis they are allocated and what are its properties.

3.6.1 Factors considered for Zoning process

1. Time of initial damage reported by the IDS in the matrix of the CI system under consideration.
2. Time of the last updates made on every data item using specific data items in the CI system.

3.6.2 Damaged Zone

This zone primarily comprises of the initially damaged node(s) reported by the IDS in the matrix. At the end of zoning process and damage assessment, all the damaged or contaminated nodes of the analyzed matrix of the impacted CI system are added to this zone. This zone includes the nodes which are ensured or guaranteed to be damaged and contaminated by the initial damaged node of that matrix. The node in this zone remains to be the primary focal point for only damage recovery

and does not go through damage assessment process as they have already been declared to be damaged based on detailed analysis without accessing log information. This helps to prevent multiple nodes which are damaged to go through damage assessment unnecessarily to determine if they are damaged or not which is time consuming. Thus, it helps in saving the time spent on damage assessment procedures on already confirmed or guaranteed to be damaged nodes. All the nodes which are in or added to this zone always remains to be in freeze state and are not released to resume their operations until they go through damage recovery process. Disabling damaged zone nodes functionalities prevents the cascading effects of the damage or prevents further rapid contamination from initial damage. This in turn enables benign nodes to stay safe while resuming their operations. Thus, service downtime is imposed on the damaged nodes but not all. It is to be noted that the nodes directly or indirectly connected to the primarily damaged node reported by IDS may not be included to this zone or suspected to be damaged as the time-based updates are considered for categorizing the nodes in the impacted CI system. Hence, the flow of damage purely depends upon the time of damage and node's time of last update being made. In other words, this zone includes the nodes which are being updated at the time of damage and post the time of damage of its parent node whose data was used for update, as they are ensured to be contaminated or damaged. In the figure 6, it includes nodes in damaged zone with filled in nodes in the directed graph structure of CI. It includes damaged or contaminated nodes R, A, T, X and Y. For instance, if the IDS has reported that node A has been attacked at time stamp 1. Node A has been used to make updates on Node X and Y at 3 and 4 respectively. The contamination could be seen in nodes X, Y because of the updates being made on them by using the damaged data from damaged node A post the time frame that the node A has been reported to be damaged by IDS. Hence, nodes X and Y gets allocated to the damaged zone and remains to be in freeze state. Nodes R and T has

been declared to be contaminated or been subjected to cascading effects from damaged node A as they were updated using the damaged nodes X and Y which were in turn damaged by node A. Nodes R and T are being updated at timestamps 5 and 6 which are past the time of damage of their damaged parents X and Y.

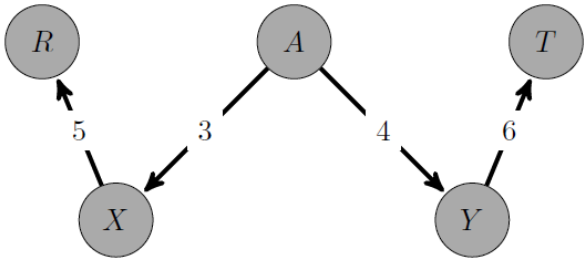

Figure 6. Damaged nodes in Damaged Zone

Definition 1: A Damaged zone includes a group of nodes, such that every node which belongs to this zone are guaranteed to be a damaged node or a contaminated node.

$$DZ = \{x \mid x \text{ is a damaged or contaminated node}\}$$

For instance, if a node X belongs to a set of nodes in damaged zone, DZ, then it is guaranteed to be a damaged node. This zone is the primary focal point which needs to be immediately addressed and recovered during damage recovery post the zoning process. This zone is subjected to only damage recovery phase and no damage assessment is needed for the nodes in this zone as it is guaranteed that the zone only includes nodes which are confirmed to be damaged or has been contaminated after in depth analysis in zoning process. The nodes in this zone need to be subjected to damage recovery process which would be explained in further sections and always remains to be in freeze state until recovered. This prevents the cascading effects of the disruptive events to a greater extent. Hence, the damaged zone contains the nodes which are reported to be damaged by IDS, and includes the nodes contaminated by the updates being made using the damaged nodes in the time frames post parent's time of damage.

3.6.3 Undamaged Zone

The undamaged zone only includes the nodes which are ensured or guaranteed to be completely safe or benign. These nodes are never responsible for any contamination or spread of damage towards the other nodes in the directed graph of CI system. Once a node is added to this zone either during green zone determination or during damage assessment carried out on gray nodes, these nodes are allowed to come out of freeze state and are allowed to resume their operations immediately as they are safe to operate. This categorization helps to resume quickly as many data items in the CI system which are completely safe to operate and prevents unnecessary imposition of service downtime on them at the time of damage discovery or analysis. These nodes are never subjected to damage assessment or recovery, thus saving a lot of time which could be focused to recover damaged nodes. This zone is quickly formed based on the analysis of the node's association with the damaged node and based on the analysis of the time of last update made on the nodes. For instance, as shown in the figure 7, if the node X is reported to be attacked at 6 by IDS in that matrix and node A has been updated by using node X at 5. During green zone determination, node A would be declared to be a safe node as its time of last update using its parent X was prior to the time of damage of node X. At the same time, nodes C and B which are updated by using the data of its safe parent node A well ahead of time of damage of initially damaged node X in that matrix, they are declared to be safe during the green zone determination. Once nodes C and B are declared to be a safe node and added to the undamaged zone, they could resume their operations. Even though the graph structures show a flow from the damaged node X to all the other nodes, the damage is not spread from X to others as they have used the undamaged values of X prior to the time frame it has been declared to be damaged. In the figure, the undamaged nodes A, B, C, D and E are added to the Undamaged zone and are represented as unfilled nodes.

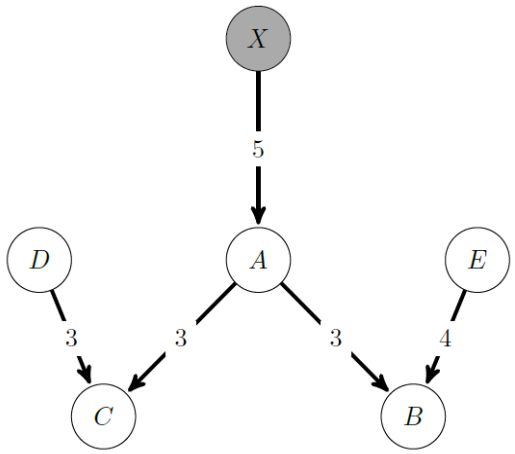

Figure 7. Undamaged nodes in Undamaged Zone

Definition 2: An Undamaged zone includes a group of nodes, such that every node which belongs to this zone are guaranteed to be an undamaged node or safe node or a benign node.

UDZ = {x | x is a safe or benign or undamaged node}

For instance, if a node X belongs to a set of nodes in Undamaged zone, UDZ, then it is guaranteed to be a benign node and is safe to resume its operation without undergoing service downtime or to remain in freeze state. This zone primarily includes the nodes which despite of being connected directly or not with the damaged node but has been updated well ahead of the time at which the initial damage has been reported in that matrix. The nodes belonging to this zone are never subjected to service downtime or never remains in freeze state. Hence, it purely includes the nodes which has not received any updates from the damaged node post the time of initial damage or time of damage of its dependent node, which makes them uncontaminated. Hence, they are ready to resume their functions as they are confident enough of not being contaminated by the damaged node and ensures no further damage or spread of damage to its related dependent nodes. The resumption of this zone bears a lot towards the efficiency of this model as it reduces the unnecessary downtime and increases the resumption of the undamaged nodes for functioning. This

zone includes set of nodes which are represented as unfilled in figure 7 and do not go through service downtime, damage assessment or recovery and are safe to resume their operations.

3.6.4 Gray Zone

The nodes which are skeptical, or which are not sure of being damaged or not are added to gray zone especially during red and gray zone determination. The nodes which are being added to this zone remains to be in freeze state until they are declared to be a safe or undamaged during damage assessment. The nodes which are added to this zone goes through damage assessment process for further categorization of them to be either a damaged node or undamaged node. In this proposed model, the only nodes which goes through or are subjected to damage assessment are nodes in gray zone. The assessment made on these helps us to identify further damaged nodes which were uncaught during the previous node determination processes and helps in determining the safe nodes which could be resumed to operate immediately. This zone is suspected to hold the nodes which could be contaminated or damaged nodes from initial point of damage or benign nodes, that is why they are subjected to detailed analysis using log information in damage assessment process. As it is not sure that the entire zone is ready for the resumption of their functioning, the damage assessment is initially being made in this zone. The assessment includes detailed analysis of the spread of any damage, detection of any abnormalities in the data values from the expected values and classification of node as either a damaged or a benign node which is explained in detail in further section. Once any damaged nodes are identified during damage assessment on gray nodes, they are allocated to the damaged zone which in turn goes through damage recovery process. Similarly, during the damage assessment on gray nodes when no contamination on any node is identified that node is declared to be a safe node by adding it to the Undamaged zone. For instance, a matrix has the initial time of damage as 8 and reported to be at node X. As node X has been used

to update node Y lastly at 11 which is past its time of damage, node Y would have been damaged as well. Now the damaged node Y with time of damage at 11, is used to update node P anytime until 10. The node P state could be declared under two scenarios. In scenario 1, Node Y would have been damaged at 9 and when Y was used to update P at 10, then contamination would been spread to node P, thus making P a damaged node. But in scenario 2, node Y would have been damaged at 11, but node P was updated using Y at 10 which is prior to the time of damage of Y, in such case node Y would have been an undamaged node. Hence, considering the two scenarios in which node P would have been, it is unsure that whether node P was updated by using node Y before Y has been damaged or after it was damaged since the time of last update being made on node Y is anywhere until 11. Thus, node P is declared to be a skeptical node and is being assigned to Gray zone as shown in figure 8. To know the exact state of node P a detailed analysis is needed which could be later performed using damage assessment. In the same way, the nodes Q and R are being updated using skeptical node P data at 9 which is past the time of initial damage at X but before the last damage update of node P. This shows the doubt of whether the damage has been spread to them or not. Hence, nodes Q and R are added to Gray zone as well since they need further assessment to categorize.

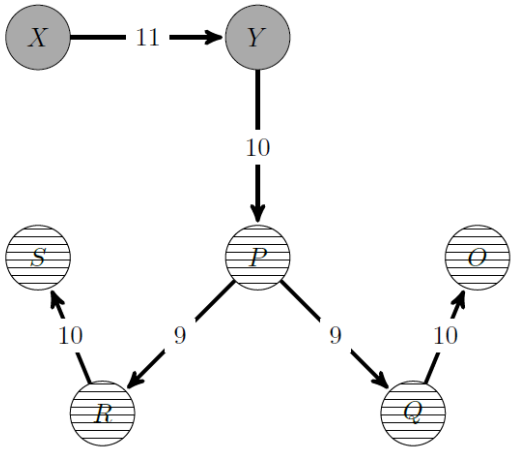

Figure 8. Skeptical nodes in Gray Zone

Definition 3: A Gray zone includes a group of nodes, such that every node which belongs to this zone are either a damaged node or an undamaged node and needs damage assessment for further categorization.

$$GZ = \{x \mid x \text{ is either an undamaged node or a damaged node}\}$$

For instance, if a node X belongs to a set of nodes in gray zone, GZ, then it is either a damaged node or an undamaged node. Damage assessment needs to be carried out on the nodes in this zone to categorize them further as either benign or damaged nodes to take actions accordingly. Post damage assessment, the undamaged nodes are added to Undamaged zone to resume their functionalities instantly and damaged nodes remain to be in freeze state which are added to the Damaged zone. This prevents the nodes from undergoing unnecessary service downtime without being damaged. The nodes belonging to this zone undergoes both damage assessment and recovery processes. Thus, this model ensures that every data item's states are identified, damage assessments are only made in areas suspected to be damaged, service downtime is only imposed on damaged, gray zones and only undamaged zone are allowed to resume their operations as they don't support contamination of damage. The figure 9 shows the directed graph of the impacted CI system after undergoing zoning process. As it could be seen from the figure 9, the entire directed graph's nodes are split into three zones containing undamaged, damaged, and skeptical nodes as explained in the previous sections.

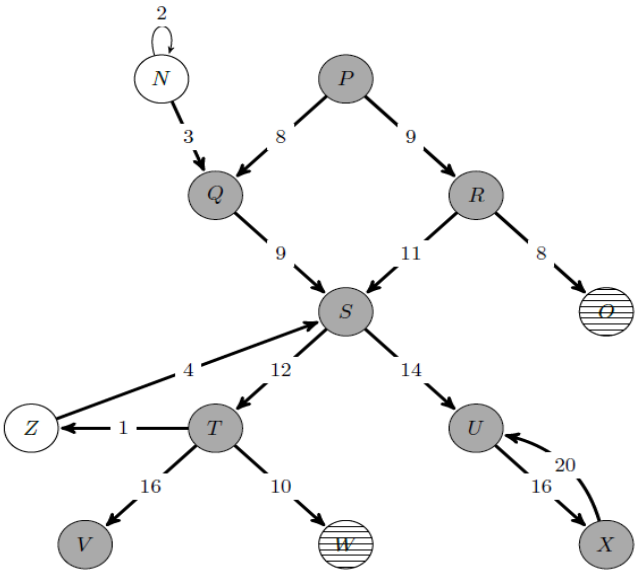

Figure 9. Directed Graph post Zoning

3.7 Optimized damage assessment and recovery model

The implementation of the proposed model starts from the point when the initial damage report from IDS has been gathered until the recovery of the damage in the CI system with optimized damage assessment and recovery which is realized through categorization of the data items in zoning process. If there are multiple matrices with the initial reported damaged nodes by IDS, all matrices are analyzed by using this proposed model simultaneously as there is no dependencies among them that might impact the analysis. The following process are followed one after the other for optimized recovery of the attack in the CI system.

1. Data items subjected to Freeze state.
2. Information collection.
3. Release of disjoint data items.
4. Zoning or data item categorization or zone determination.
5. Damage Assessment.
6. Damage Recovery.

3.7.1 Algorithm of the Proposed Model

Algorithm 1 Optimized Damage Assessment and Recovery through Data Categorization

1: Receive IDS report with attack details and time of initial damage, t_d.

2: Freeze all data items or nodes in the impacted CI system.

3: Collect information respective to CI at t_d: Matrix equivalent, log information, update formula.

4: Release matrices of impacted CI with no reported initial damage data items.

5: Create sets Damaged zone (DZ), Undamaged zone (UDZ), and Gray zone (GZ)

6: **for** every matrix with initial damage at t_d from matrices of the impacted CI system **do**

7: Green zone determination

8: Red and Gray zone determination

9: Damage Assessment

10: Damage Recovery

3.7.2 Freeze state

Once the report from IDS has been received stating that a specific data item or multiple data items has been damaged in a CI system, immediately the entire CI system's data items goes into the freeze state. If a data item is in a freeze state, it is not used to update or push the information towards its receiver nodes or be updated. This makes sure that the damage is not further spread by contaminating the other benign nodes in the CI system. Thus, the cascading effects from the initial damaged node is prevented to a greater extent. The node or data item remains to be in a freeze unless or until it has been declared to be an undamaged node during zoning process. A data item or node when declared to be damaged or contaminated is also subjected to remain in the freeze

state until the data item is recovered and made as a safe node in damage recovery process to resume its operations. In other words, a node or data item in damaged and gray zone remains to be in freeze state and a node when added to the undamaged zone is released from the freeze state to resume their operations.

3.7.3 Information collection

After IDS report regarding the initial damage(s) has been gathered, the next step would be information collection for that impacted CI system from the dedicated teams as discussed earlier. The initial damage might be at one or many data items which might be in different directed matrices within the same CI system. Or in other words, the initial damaged nodes might be distributed among different matrices of the same CI system. From the tool which generates the equivalent matrices of a CI system, the respective matrices for the affected CI system are collected. It is to be observed that there might be a single or a set of matrices for a given CI system. The matrix data containing the initial damaged data item(s) as per the IDS report in the impacted CI system is used for zoning process which helps to make as many nodes available as possible prior to the actual analysis. Post zoning, for damage assessment and recovery, log information and update formula from repository as mentioned earlier are gathered and utilized.

3.7.4 Release of disjoint data items

Once the respective matrices are collected from the dedicated team for the impacted CI system, the next step would be to identify the matrices containing the initially damaged data items as per the IDS report. The matrices with the initial damaged nodes or data items are focused upon for the further processes. The matrices of the affected CI system independent of the initial reported damaged node(s) are thus totally disjoint from the affected matrices. This depicts that the nodes in the disjoint matrices independent of reported damaged data item(s) are totally uncontaminated and

are safe to resume their operations as they are completely independent of the traces of the initial damage. Also, that there is zero probability of the damage being spread from the initial reported damaged nodes to the other disjoint matrices. Thus, disjoint matrices free of reported initially damaged nodes are allowed to resume their operations. This process releases huge number of data items from the freeze state and are allowed to resume their operations instantly. This is one of the major advantages of this proposed model as it releases huge number of data items which are assured to be safe to resume their operations. Each matrix under consideration for zoning process may have a single or multiple initially damaged data items reported by IDS.

3.7.5 Zoning

Once the matrices which contains the initial reported damaged nodes are collected with their data items still being in freeze state, the next step would be to impose zoning procedures on them. The input to the zoning process includes the matrices with the initial reported damaged nodes and the outcome of this process includes the nodes of the impacted matrices to be allocated into the three zones, Damaged, Undamaged, Gray zones. As mentioned earlier, Damaged zone contains only damaged or contaminated nodes, Undamaged Zone contains only safe or benign nodes and Gray zone contains skeptical nodes which could be either damaged or undamaged nodes. There are two steps followed in this process which are listed in order below. Each process is carried after the previous step was completed. Each step would determine nodes to each of the zones after making in depth analysis on the data available in the collected matrix with the initial reported damaged nodes or data items.

1. Green zone determination.
2. Red and Gray zone determination.

3.7.5.1 Green zone determination

This is the first step carried out in the zoning process. The goal of this step is to release the safe nodes immediately through row wise scanning and analysis made in the matrices with initial reported damaged node(s). In other words, the objective of this step in the zoning process is to construct the Undamaged zone or to quickly release safe nodes so that they could resume their operations as they have been guaranteed not to hold any traces of damage or contamination.

Algorithm 1.1 Green Zone determination Algorithm

1: Create a set of dependent nodes of the initially damaged node(s) of matrix M

2: Deduct the dependent nodes from the set of all nodes in the matrix M

3: Assign the remaining nodes post deduction to the Undamaged Set, UDZ to resume its operations

4: **for** every node x in row wise matrix analysis of matrix M with time of initial damage, t_d **do**

5: **if** x's time of last update made by all its parents, i is prior to time of damage, M[x][i] < t_d **and** if x is not the initial damaged node as per IDS report **then**

6: add x to set UDZ, Undamaged zone to resume its operations

This helps to reduce the load imposed on the damage assessment and damage recovery procedures post zoning which in turn prevents benign nodes from unnecessary service downtime. The matrix with the reported initial damaged node is analyzed row wise, in other words, every node is being analyzed in terms of the last updates made on them by using each of its parent's data, time of last update (row wise in matrix). The below mentioned step is executed and then the condition is checked against each node and for its respective parents (row data of that node in matrix), to determine if a node is benign node or not. There are two basic steps followed in this green zone

determination. The first step is performed initially and then the condition mentioned in the second step is imposed on all the nodes being analyzed row wise in the matrix.

1. The dependent nodes of the initially damaged nodes as per IDS report are collected and are deducted from the set of all nodes of the matrix under consideration. The remaining nodes are allocated to the Undamaged zone to resume their operations.
2. For every node being analyzed row wise, if the time of last update being made on the node by using every of its parent's data is well ahead or prior to the reported time of damage by IDS in that matrix and the node is not the initially reported damaged node as per IDS, then the node is assigned to the Undamaged zone to resume their operations.

The first step helps to declare as many safe nodes as possible and adds them to the Undamaged zone because the dependent nodes of the initially damaged node(s) have a very high probability of being contaminated and hence when they are deducted from the set of all nodes of the matrix under consideration, the remaining nodes are guaranteed of not being subjected to have received any damage or contamination. This is because they are very sure of having no connection with the damaged nodes and have or will never have been updated using the damaged data item's value. The first step by itself relieves a very good possible number of safe nodes to resume their functions very quickly. The second condition when imposed on every node row wise and when satisfied ensures a node to be a benign node because if none of its parents have been used to update the node past or at the reported time of initial damage in that matrix, then there would have been no damaged data being traversed from the parents as they were not yet damaged at the time they have been used by these nodes for update. The second condition also checks if the node is not the initially damaged node as per IDS report. The nodes of the matrix after completion of first step undergoes check against the second condition. The nodes of the matrix which surpasses the above-

described steps are added to the Undamaged zone. Then as mentioned earlier, the safe nodes when assigned to undamaged zone are released from freeze state and are allowed to resume their operations immediately without any further delay. Hence, green zone determination helps to relieve the safe nodes as quickly as possible and as much as possible. It also prevents them from going through service downtime unnecessarily. If in case any nodes are left out when determining green or safe nodes, they would somehow be subjected to in detail analysis in the next steps. This ensures that none of the nodes in the matrices having initial damaged nodes are left unanalyzed and uncategorized.

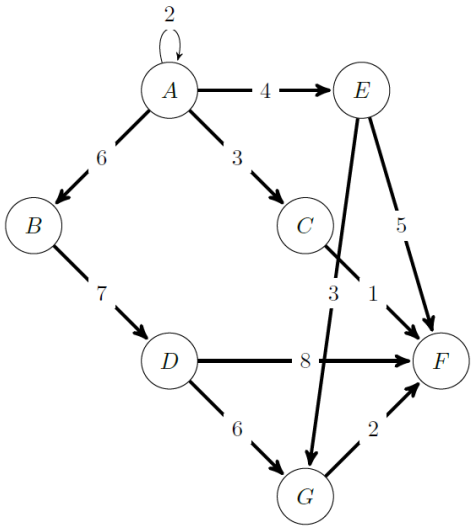

Figure 10. Green zone determination (Directed graph)

For instance, if the IDS has reported damage at the data item C at time stamp 5 for a CI system, the relevant matrices of that CI system are collected. The next step would be to identify the matrix of that CI system that contains the initial reported damaged data item C as node C. The figure 10 represents the directed graph structure of the matrix containing the initial damaged node C as reported by the IDS. Once the matrix with the initial reported data item node C has been identified as shown in figure 11, the next analysis would be green zone determination. For this analysis, the equivalent matrix is subjected to in depth analysis based on two steps as explained before.

Parents							
Children	A	B	C	D	E	F	G
A	2						
B	6						
C	3						
D		7					
E	4						
F			1	8	5		2
G				6	3		

Figure 11. Green zone determination (Matrix)

As explained before, during green zone determination, the first step is to compute the set of dependent nodes of initially damaged node C which includes only node F as its directly connected to and is dependent on node C. The dependent node F and C are deducted from the set of all nodes in the matrix which results in set containing nodes A, B, D, E, G. Now these nodes are assigned to the Undamaged zone, UZ. Now second condition is imposed against each node when analyzing row wise. For instance, when analyzing node, A, its parent includes node A itself because of the self-update it makes. The time of last update on itself was prior to time of initial damage, which was at 5, thus node A is safe node and could be assigned to undamaged zone. It is to be observed that node A has already been allocated to Undamaged zone through step 1. Similarly, for analysis node B, its time of last update is past the time of initial damage of this matrix, hence gets ignored by this condition. It is to be observed that node B has been added to Undamaged zone through step 1 and it could be observed that node B is not dependent on initially damaged node and thus it does not matter if it has been lastly updated past the time of initial damage for this case. When analyzing any node row wise, it's all parent nodes should satisfy condition 2. If any of it its parents did not, it is not a safe node. In other words, a node should have been updated by using all its parent node's data values respectively well ahead of the time of initial damage in that matrix to be declared as a

safe node during green zone determination. In the above matrix analysis, during green zone determination, the nodes A, B, D, E, G are allocated to the Undamaged zone and are then allowed to resume their operations.

3.7.5.2 Red and Gray zone determination

After completing row wise analysis in determining safe nodes based on the two steps as mentioned in the above section, this step helps in determining the damaged or contaminated nodes and skeptical nodes in the matrix under consideration. The goal of this step is to quickly identify the damaged or contaminated nodes to retain them in the freeze state and allocate skeptical nodes into the Gray zone for further damage assessment. This process includes column wise analysis starting from initial reported to be damaged node of that matrix and next their damaged children until all damaged children of damaged children has been analyzed recursively.

Post green zone determination, the matrix is now analyzed column wise starting from the initial damaged node(s) as reported by IDS. The time of damage of the initial damaged node is retrieved from IDS report. Now the time to be considered for comparison would be the initial damaged node's time of damage for column wise analysis. Then using the initial damaged node's column, all its children are analyzed to declare if any damage is being spread to them from this initial damaged node or not. For every child of the initial damaged node, compare their time of last update made by using this damaged parent against the initial damaged node's time of damage. If the last time of last update of a child is past its parent's time of damage, then the child node is guaranteed to be contaminated by its initially damaged parent node. This is because the parent node (initial damaged node) was already damaged by the time it was used to update its child node if its time of last update is past it's time of damage.

Algorithm 1.2 Red and Gray Zone determination Algorithm

1: Create set C to hold all damaged children and initialize to null (next node to analyze).

2: Assign the initial reported to be damaged node in matrix M to Damaged zone set, DZ

3: Assign the initial reported to be damaged node to set C with value t_d

4: **while** set C is not empty **do**

5: Consider damaged node x with time of damage, t_d which is recently added to set C

6: **for** every child, c with time of last update, M[c][x] of damaged node, x **do**

7: **if** time of last update of c is past its parent time of damage, M[c][x] > t_d **then**

8: add c to set DZ, Damaged zone

9: add c with its time of damage, t_d= M[c][x] to set C

10: remove node x from set C

11: Add rest of the nodes in Matrix, M which haven't been categorized yet, into Gray zone, GZ

Hence, all such child nodes which are declared to be damaged are added to the Damaged zone where it is remains to be in freeze state until damage recovery process. Once this comparison is being made on all child nodes of the initial damaged node, the next node to be considered are the damaged children of the initial damaged node who are declared as damaged in this analysis and their time of damage is also noted respectively for further analysis.

After the column analysis is being made on the initial damaged node, its damaged children are next nodes to be focused upon. This process continues until all damaged children of the current considered damaged parent node are analyzed. In other words, the above similar analysis is performed recursively on all damaged children of damaged children of initial damaged node. The

same process is being implemented for the damaged children of the initial damaged node. The next node for analysis is now a damaged node declared in previous column wise analysis, in other words the current node under consideration is the damage child of the previous node which was analyzed to be damaged. The next node considered for column wise analysis is also the one which is not yet allocated to any zones yet. This helps in unnecessary analysis of already declared to be damaged nodes. For every child node of the damaged parent node under consideration whose time of damage is compared against its every child's time of last update being made using it. If a child's time of last update is past the parent node's time of damage, the node is declared to be a damaged node. This is because, as the parent node was damaged when it was used to make an update on this child node as it is clear when child's time of last update is past the parent's time of damage. The declared damaged node is then added to the Damaged zone. The node in the damaged zone is retained to be in the freeze state until damage recovery. The next node to be considered for analysis becomes the damaged children identified in the current column wise analysis. The same process is continued recursively until no node is left for next analysis.

The rest of the nodes in the matrix which are left unassigned to any of the zones during this analysis are added to Gray zone as they are skeptical nodes and are unsure if the damage has been spread to them or not. These nodes are added to the Gray zone because their time of last update makes them totally unsure if they have been contaminated by its damaged parent or not. They are added to Gray zone to go through damage assessment for further clarification on if its contaminated or not. Thus, by this step all the nodes in the matrix would have been analyzed and categorized into one of the three zones. The figure 12 shows the directed graph of the matrix (figure 13) with initial reported damaged node. For instance, let's consider the matrix with initial damaged node P with time of damage as 5. The matrix is initially subjected to green zone determination. The results

of green zone determination conclude and categorizes that nodes N and Z are safe nodes and are allocated to the Undamaged Zone, UDZ to resume their operations.

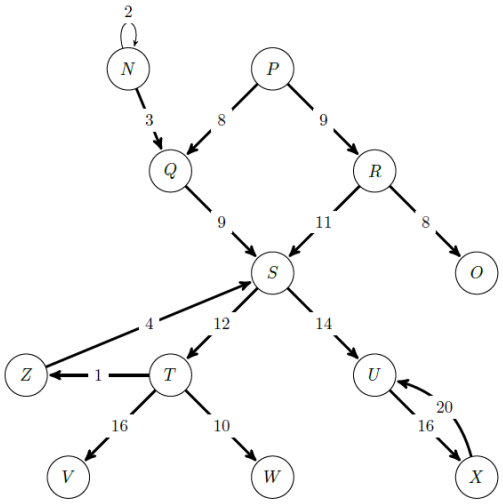

Figure 12. Red and Gray zone determination (Directed Graph)

The next analysis would be red and gray zone determination where the damaged nodes and skeptical nodes are identified. The analysis starts from the column wise analysis of initial damaged node P with time of damage at 5. The child nodes Q and R are allocated to Damaged zone as their time of last updates 8 and 9 are past the time of damage of node P. The next node to be analyzed becomes the node Q as its time of damage is the earliest and it would help to categorize most of the nodes and prevents most of the nodes to be left unanalyzed and end up in Gray zone. The child node S of Q whose time of last update 9 by node Q is past the time of damage of node Q, 8. Hence, node S is assigned to damaged zone, and it becomes the next node to be analyzed column wise. Similarly, the damage child nodes of S are analyzed in order of node T, then node U. Then damage children of node T are analyzed, that is node V which results in no damaged children. Since the children of node T is completely analyzed, the next node left to be analyzed in the queue would be the node U. The same process is recursively carried on all the damaged children of the damaged children of initial damaged node in that matrix which is shown below. If the damaged children

identified during any column analysis is already allocated to Damaged zone, they are not analyzed again. This avoids redundant and unnecessary repeated analysis.

Parents												
Children	P	Q	R	S	T	U	V	W	X	Z	N	O
P												
Q	8										3	
R	9											
S		9	11							4		
T				12								
U				14					20			
V					16							
W					10							
X						16						
Z					1							
N											2	
O			8									

Figure 13. Red and Gray zone determination (Matrix)

Green zone determination:

Undamaged Zone = {N, Z}

Red and Gray zone determination:

Next node set includes every element as ‘Node: time of damage’.

Analysis of node **P**:

Damaged Zone: {P, Q, R}

Next Node: {Q:8, R:9}

Analysis of node **Q**:

Damaged Zone: {P, Q, R, S}

Next Node: {R:9, S:9}

Analysis of node **S**:

Damaged Zone: {P, Q, R, S, T, U}

Next Node: {R:9, T:12, U:14}

Analysis of node **T**:

Damaged Zone: {P, Q, R, S, T, U, V}

Next Node: {R:9, U:14, V:16}

Analysis of node **V**:

Damaged Zone: {P, Q, R, S, T, U, V}

Next Node: {R:9, U:14}

Analysis of node **U**:

Damaged Zone: {P, Q, R, S, T, U, V, X}

Next Node: {R:9, X:16}

Analysis of node **X**:

Damaged Zone: {P, Q, R, S, T, U, V, X}

Next Node: {R:9, U:20}

Analysis of node **U**:

Damaged Zone: {P, Q, R, S, T, U, V, X}

Next Node: {R:9}

Analysis of node **R**:

Damaged Zone: {P, Q, R, S, T, U, V, X}

Next Node: {None} since the damaged children of R are already allocated to Damaged zone and does not need any further analysis.

Post Red and Gray zone analysis:

Damaged Zone: {P, Q, R, S, T, U, V, X}

Gray Zone: {O, W}

Post zoning:

Undamaged Zone = {N, Z}

Damaged Zone: {P, Q, R, S, T, U, V, X}

Gray Zone: {O, W}

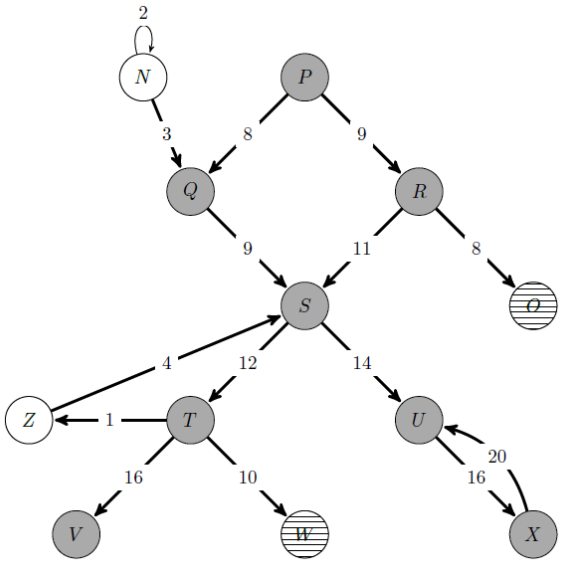

Figure 14. Post Red and Gray zone determination (Matrix)

3.7.5.3 Observations during Red and Gray zone determination

The following observations are made during red and gray zone determination.

Observation 1: If a node X is being updated by using only undamaged parents, then the node X is guaranteed to be an Undamaged node.

This is because since all its parents are undamaged until the time the zoning analysis is being made, there is zero probability of the damage being spread toward their child node. Any updates made by them would contain no trace of damage and hence a node being updated only using the data items which are Undamaged is guaranteed to be Undamaged node.

Observation 2: If a node X is being updated by using the data values from a mixture of only damaged or undamaged nodes, then node X is guaranteed to be either Gray or Damaged node.

The nodes with damaged parents are subjected to column wise analysis and during such analysis, if their time of last update is beyond their parent's time of damage they would have been declared

as damaged node else would have been declared gray node at the end of column wise analyses. Similarly, the node with undamaged parent would have never been subjected to column wise analysis during red & gray zone determination, if they remain to unanalyzed until end of recursive column wise analysis, they would be allocated to Gray zone as they would have been unallocated until then. If they are damaged by any other damaged parent, then this node would be made damaged node and would have been subjected to column wise analysis. The node being updated using data from Damaged and Undamaged parents, could fall into two cases. In case 1, the node would have been contaminated or damaged if the Damaged parent has been used to make a latest update on it, in such case the damage would have been spread making this node to be a Damaged node. In case 2, the node would be or would not be contaminated and remains to be skeptical. This is because it is totally unsure if the last update was made by damaged parent or by undamaged parent, in this case time of last update is vital deciding factor. Thus, the node would be added to Gray zone.

Observation 3: If a node X is being updated by using the data values from only Gray nodes or from a mixture of Gray and Undamaged nodes, then the node X is guaranteed to be a Gray node.

If node was updated by using only Gray parents, then node would be a Gray node, as the state of the parents are still unknown, the updates made using them is also unsure to be damaged or node. If a node was updated by a mixture of Gray and Undamaged parents, then this node's state totally relies on the parent node type who has been used for the latest update on this node. If the latest update on this node would have been made by the Gray node, as the state of its parent which was used for update is unknown, this node is also unsure of its category, thus gets allocated to Gray zone. If the latest update is being made by using undamaged parent, then during the red & gray zone determination, this node would have never been subjected to column wise analysis and would

have been left unanalyzed. Thus, the node would have been allocated to Gray zone for further analysis.

Observation 4: If a node X is being updated by using the data values from a mixture of Gray nodes and Damaged nodes then the node X is guaranteed to be either a Gray node or Damaged node.

If the node is being lastly updated using the Gray node, then this node would also be a Gray node as it has been updated using a skeptical node whose state of damage spread is still unknown. If the latest update made on the node is by using the damaged node, then this node would have been contaminated and hence would be declared a damaged node.

3.7.6 Damage Assessment in Gray zone

All the nodes in the matrix with the initially attacked node would have been categorized by this instance into one of the three zones: Damage, Undamaged, Gray zone by the previous steps. The nodes in the Damaged zone are retained to be in the freeze state, nodes added to the Undamaged zone are released from freeze state immediately to resume their operations whereas the nodes added to the Gray zone are subjected to damage assessment for further analysis. The proposed model is efficient in optimizing the time spent and load subjected to damage assessment, as the number of nodes has been reduced to a greater extent by the previous analysis by declaring them as either damaged or not. The goal of damage assessment is to categorize every node in the Gray zone to be either a damaged node or safe node by assigning them to their respective zones to take further actions. Every node in the Gray zone is subjected to damage assessment so that by the end of damage assessment, there would be no nodes in Gray zone and all of them would have been allocated to either Damaged zone or to the Undamaged zone. Every time a node is declared to be a safe node during damage assessment, they are added to Undamaged node to immediately resume their operations. Every time a node is declared to be a damaged node during damage assessment,

it is retained to be in freeze state and is further subjected to damage recovery. After completing damage assessment all nodes in the matrix analyzed would be only in of the two zones, Undamaged zone, and Damaged zone. For the damage assessment, the algorithm mentioned in [38] is followed. This effective and efficient algorithm is implemented for all the data items allocated to the Gray zone until none is left in the Gray zone. This algorithm starts the damage assessment by scanning the log file information of the CI system for the detailed assessment of the gray node to determine if it is a damaged node or undamaged node. By examining the log information, the exact time of damage of the damaged parent nodes are analyzed, which in turn helps to determine the gray node actual type. This assessment is imposed on all gray nodes and finally results in adding them either to the Damaged zone or to the Undamaged zone. This process is totally dependent on the log information of the impacted CI system for the determination of damaged or benign nodes from Gray zone nodes.

3.7.7 Damage Recovery in Red Zone

This is the next step carried out post zoning process and damage assessment. The goal of this step is to recover all the damaged nodes from Damaged zone and make them safe to resume their operations. Once the damaged nodes undergo damage recovery, they are guaranteed to be safe nodes, hence, they are added to the Undamaged zone and are allowed to resume their operations. This is the final step which ensures that by this time no nodes would be left in Damaged zone. In other words, post damage recovery, there would be no nodes in Damaged and Gray zone, all nodes would have been allocated to Undamaged zone and would have resumed their operations. The final step is to resolve the damage incurred in the contaminated damaged nodes in Damaged zone by recovering them using the existing and collected information regarding the impacted CI system. As discussed earlier the log information until the IDS report has been received regarding the matrix

with the initial damage nodes of the impacted CI system would be collected from the team dedicated to maintaining the log information of the CI system. The log information about the activities or transactions carried out in the CI system would include details such as ID of the data item, the initial value of the data item or node before any update has been made using any of its parent nodes and the time of last update made. The other set of related data of CI system to be gathered would be from the dedicated team which maintains the update formula used by or on all the nodes or data items, called Update formula repository for the matrix with initial damage of the impacted CI system after the damage has been reported. The repository could help for recovery with details such as the update formula used by the data item to update the other data item. The update formula repository and log information are synchronized. The repository includes details such as parent data item, update formulation made and the destination data item. Once, the log data and update formula repository data of the affected CI system specific to the matrix analyzed in zoning with reported initial damage has been gathered, the damage recovery process could be initiated. The Damaged zone includes the nodes which are guaranteed to be damaged and contaminated after making in depth analysis in previous steps. The goal of the damage recovery process is to recover each damaged node so that it could then be declared to be a safe node to resume its operations at the instant it was recovered. Each damaged node from Damaged zone is considered and recovered using the gathered relevant data for that node from log information. The damaged node's data item ID helps to retrieve its respective log information at the time it was damaged. From that retrieved log data, it collects the initial values of damaged node before update has been made by using damaged data at the specific time of last update. At the same time the damaged node's update formulation is to be retrieved from the repository using the source and dependent data item ID. Now, the update formulation is applied on to the initial data of that

damaged node to get the expected normal value of that node. Once the expected value for the damaged node has been computed, it is updated to that node in respective areas and the node is now declared as a benign node by adding to the Undamaged zone. It is because it holds the value as expected and as it should have been if there was no damage. As mentioned earlier, now the node or data item could resume its operations after being released from freeze state. This entire process is repeated until all nodes from the damaged zone has been recovered and released to the Undamaged zone to resume their operations. Now, when the damaged zone and gray zone is empty, the only zone would be Undamaged zone which is filled entirely with only safe nodes which are set of all nodes of the matrix being analyzed for the impacted CI system. Now, post the damage recovery the state of the CI system remains to be in such a state as if there was no damage and all the data items would have been with the values as they are expected to have been.

4 SIMULATION AND ANALYSIS OF RESULTS

The proposed model and its algorithms were implemented on a simulated data and the simulation experiments are carried out for various test cases. The test cases execution and results were analyzed and studied under different circumstances. The goal of the simulation experiments carried out was to analyze and realize how proposed model performed over the other two models such as log and dependency graph model. The experiments were carried out as different test cases for varying number of data items of a CI system under consideration against varying factors of the proposed model such as history of transaction, time stamp range (for time of last update of every transaction), maximum number of children for each data item (number of immediately dependent data items of a node), maximum number of roots or data items of an impacted CI system. Each set of test case includes multiple set of experiments where each experiment was carried out at least 20 times and their average resultants are used for the simulation results and analysis. The simulation makes use of the data generated by the tool by using random values of time of last update for a transaction from a timestamp range, random number of occurrences of a transaction in the log file based on number chosen from history of transactions range, random number of children or dependent data items for a node based on values chosen from range of maximum number of children, and number of roots based on random values chosen from the number of roots range.

The matrix containing the initially reported to be damaged data items is generated using the tool which takes the number of data items as input. The log model totally relies on the history of transactions recorded in it for the reported attack time and attacked location. The number of assessments proposed by the log model is computed starting from the first occurrence of the transaction which has the reported initial attack time and attack location until the last transaction at the end of the log file. The log file for a matrix under analysis of the impacted CI system is being

generated randomly by the tool implemented which takes in the number of data items as input and creates a log file with random number of occurrences for every transaction based on specific range for history of transactions and random value from the specified time stamp range is used as the time of last update for every transaction. The dependency graph model relies on the initially damaged data item and all the dependent data items of it. In other words, all the directly or indirectly connected data items of the initially reported to be damaged data items are the number of assessments suggested by the dependency graph model. The direct or indirect dependent nodes of the damaged nodes are computed using the matrix generated by the tool. The proposed model makes use of the matrix and performs green zone determination to suggest the number of safe nodes which could resume their operations. It performs red & gray zone determination to compute the number of damaged nodes to be recovered and the number of gray nodes as the number of assessments to be made as per the proposed model and its algorithm. The important factor to be considered for analyzing the performance and efficiency of the proposed model and the other existing models of CI damage assessment and recovery is shown below.

4.1 Factors influencing performance

The factor which is used for analyzing and comparing the performance and efficiency of the proposed model and the existing models includes the below one.

- How many assessments are required or how many data items needs to be validated?

When an attack or failure or damage occurs in a CI system at a particular instance of time the performance of a model is analyzed in terms of number of assessments to be made or number of data items to be validated. Each model is being tested for a case where series of 20 experiments is carried out for each varying factor of that test case and its average values are being used for analysis. Each model suggests varied number of assessments for the damage assessment in the

impacted CI system during simulation comparison. The number of assessments to be made in each model is nothing but the number of data items to be validated for damage assessment and recovery procedures. This is a very vital factor to be analyzed as it contributes much towards the efficiency of the model. The number of data items to be validated determines how quick the model handles the damage assessment and recovery in terms of the time and resources spent.

4.2 Measuring assessments for different features

The simulation is carried out for varying values of below mentioned features in the matrix of the CI system of the proposed model for different number of data items.

1. Number of root nodes or data items.
2. Maximum number of children or dependent data items for a data item.
3. Varying range of time stamp for time of last update for every data item.
4. Varying range of history of transactions for every transaction.

4.2.1 Measuring assessments by varying number of roots

This simulation experiment was carried out for different number of data items such as 1000, 2000, 3000, 4000, 5000. For every set of data items, the modified value of number of root nodes of the matrix under consideration is used. The range considered for this experiment was from 1 until 3% of total number of data items, N. The simulation results are plotted as number of data items against the number of assessments suggested by the proposed model and the other two models. As it could be observed in figure 15, the log model is suggesting higher number of assessments when compared to proposed model and dependency graph model. The log model as it entirely depends on the history of transactions, attack time and attack location there are sudden spikes and lows in its pattern.

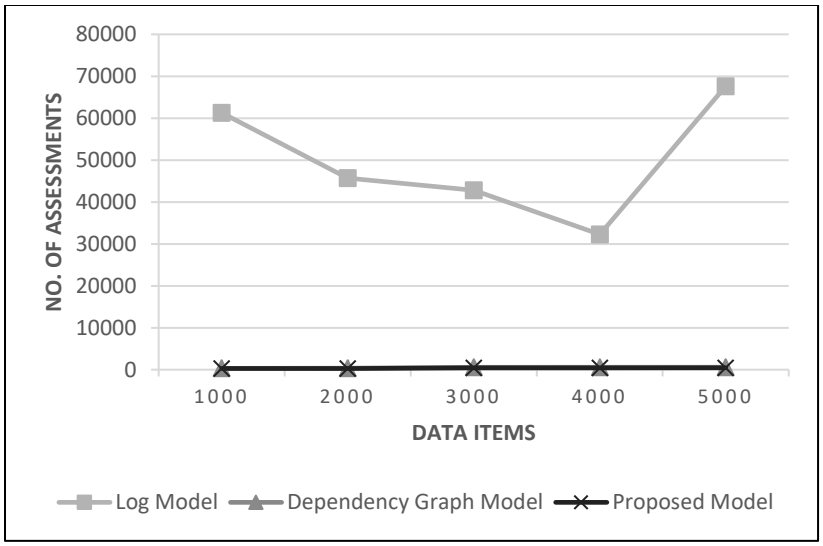

Figure 15. Assessments of 3 models with number of roots as [1 to 3%(N)]

The proposed model and the dependency graph model are quite close to each other in the number of assessments suggested by them. Hence, to analyze their results in depth, their respective simulation results are plotted in figure 16 from which it could be observed that dependency graph model is suggesting a greater number of assessments when compared to the proposed model. Thus, in this simulation experiment, the proposed model outperforms the other two model as it requires the least number of assessments.

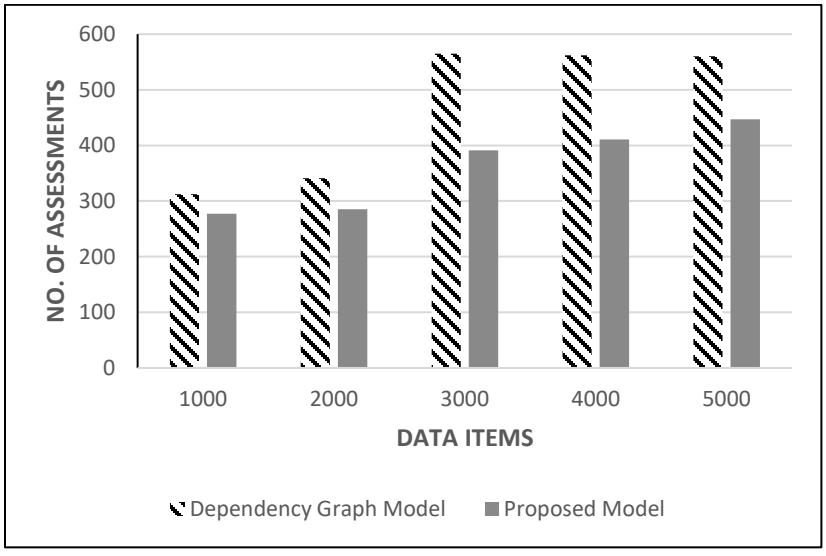

Figure 16. Assessments of 2 models with number of roots as [1 to 3%(N)]

4.2.2 Measuring assessments by varying maximum number of children

This simulation experiment was carried out for different number of data items such as 1000, 2000, 3000, 4000, 5000. For every set of data items, the maximum number of children or dependent nodes for a data item in the matrix under consideration of the impacted CI system is modified. The range considered for this experiment was from 1 until 3% of total number of data items, N. The simulation results are plotted as number of data items against the number of assessments suggested by the proposed model and the other two models.

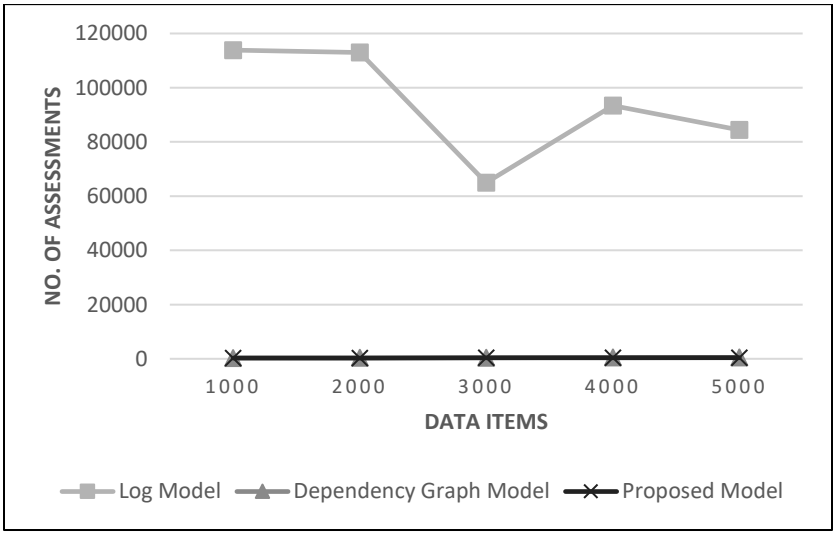

Figure 17. Assessments of 3 models with max. number of children as [1 to 3%(N)]

As it could be observed in figure 17, the highest number of assessments are suggested by the log model when compared to the proposed and dependency graph model. The spikes and lows in the pattern of the log model is because of its reliability on history of transactions, attack time and attack location. The number of assessments suggested by dependency and proposed model are very close, hence they are analyzed separately as shown in figure 18. It could be observed from figure 18 that proposed model outperforms the dependency graph model for varying number of data items and for the specified maximum number of children nodes factor. Thus, it could be understood that the proposed model is more efficient than the other two as it suggests least number of assessments.

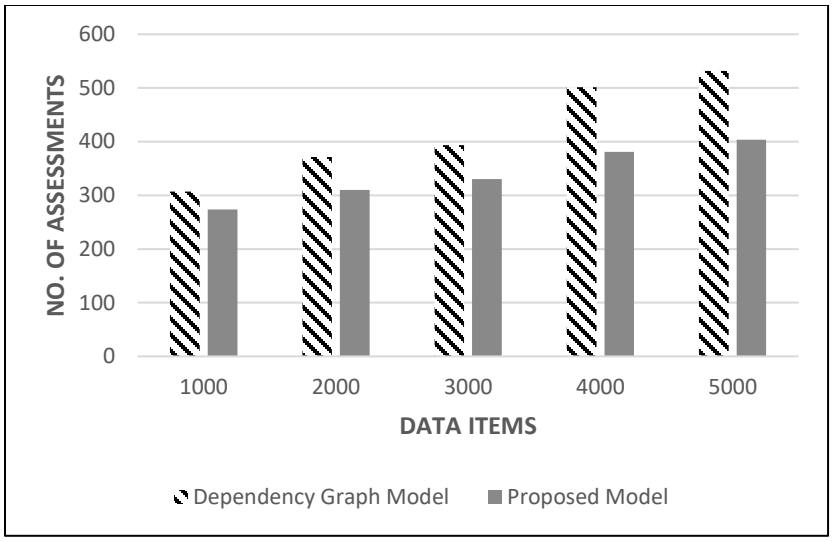

Figure 18. Assessments of 2 models with max. number of children as [1 to 3%(N)]

4.2.3 Measuring assessments by varying timestamp range

This simulation experiment was carried out for different number of data items such as 1000, 2000, 3000, 4000, 5000. For every set of data items, the range of time stamp used for generating time of last update for every transaction in the matrix under consideration of the impacted CI system is modified. The range considered for this experiment was from 1 until 2% of total number of data items, N. The simulation results are plotted as number of data items against the number of assessments suggested by the proposed model and the other two models. It could be observed in figure 19 that the log model is proposing the highest number of assessments. The proposed and dependency graph model is suggesting very close number of assessments each. Hence, they are analyzed separately together as shown in figure 20, from which it could be observed that the proposed model is comparatively proposing smaller number of assessments than dependency model. Thus, in this simulation experiment proposed model suggested least number of data items to validate.

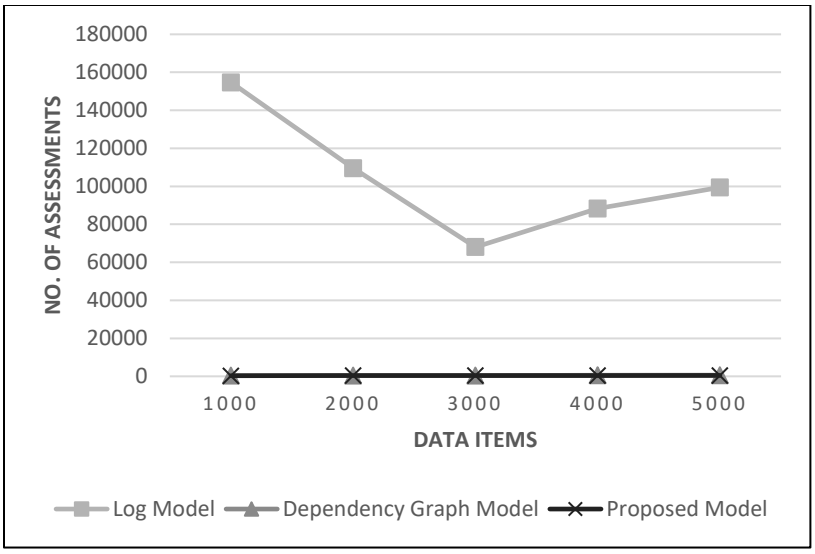

Figure 19. Assessments of 3 models with timestamp range as [1 to 2%(N)]

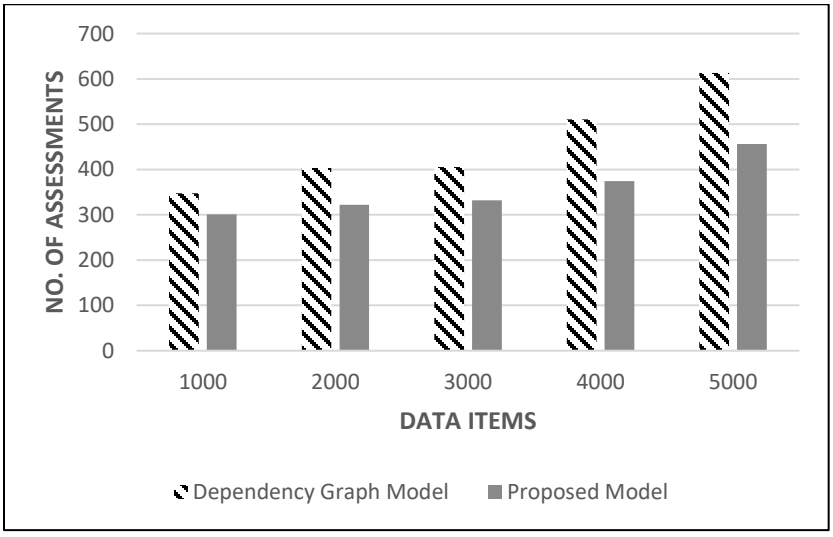

Figure 20. Assessments of 2 models with timestamp range as [1 to 2%(N)]

4.2.4 Measuring assessments by varying history of transaction range

This simulation experiment was carried out for different number of data items such as 1000, 2000, 3000, 4000, 5000. For every set of data items, the history or number of occurrences of transaction among data items in the matrix under consideration of the impacted CI system is modified. The range considered for this experiment was from 1 to 7 times of occurrences. The simulation results

are plotted as number of data items against the number of assessments suggested by the proposed model and the other two models.

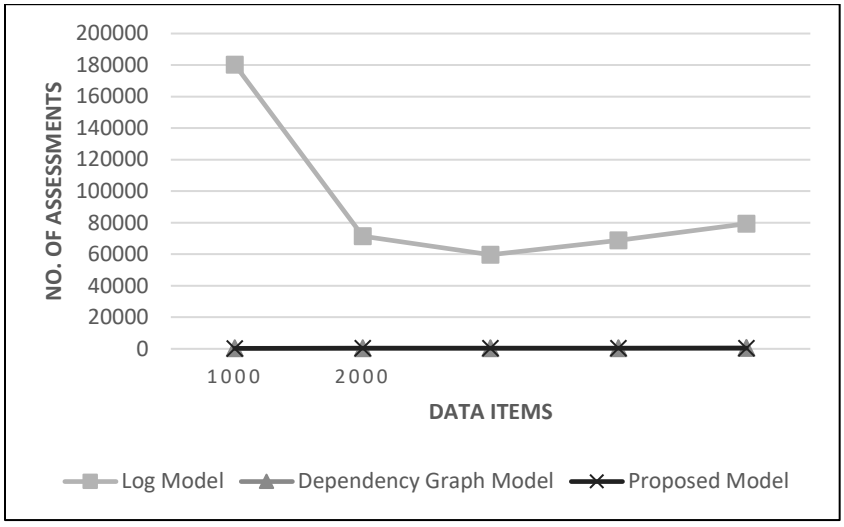

Figure 21. Assessments of 3 models with history of transaction range as [1 to 7]

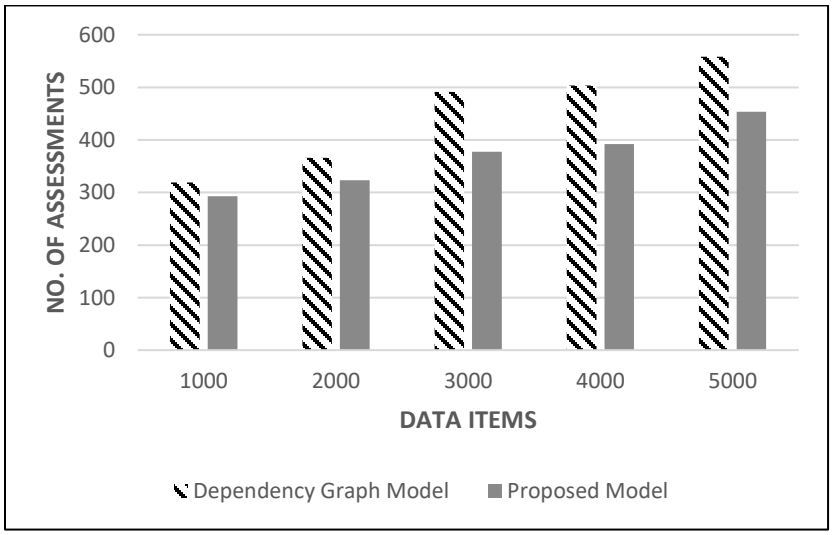

Figure 22. Assessments of 2 models with history of transaction range as [1 to 7]

It could be observed in figure 21 that the greater number of data items to be validated are suggested by only log model. The proposed model and dependency graph model suggested almost closest number of assessments; hence they are observed in another graphical data as shown in figure 22. It could be observed that the proposed model recommends smaller number of data items to be

validated or assessed than the dependency model. Thus, in this simulation experiment proposed model outperformed the other two.

4.3 Measuring assessments for varied features

The next set of simulation experiments include the analysis of the number of data items to be assessed as suggested by the proposed model and the other two models for a fixed number of data items. The experiments are carried out with the 3000 data items but by just varying the below mentioned factors.

1. Number of root nodes or data items.
2. Maximum number of children or dependent data items for a data item.
3. Varying range of time stamp for time of last update for every data item.
4. Varying range of history of transactions for every transaction.

4.3.1 Measuring assessments for 3000 data items for varied ranges of number of roots

This simulation is carried out for 3000 data items by varying the number of root nodes of the matrix under analysis of the impacted CI system. The number of root nodes are varied in terms of [1 to 2%(N)], [1 to 3%(N)], [1 to 4%(N)], [1 to 5%(N)], [1 to 6%(N)] where N being the total number of data items which is 3000. The number of assessments suggested by the three models are plotted with the varied ranges of root nodes against the number of assessments suggested by each model respectively. The figure 23 presents the results from which it could be observed that the log model is recommending the highest number of data items to be validated when compared to the other two models. The proposed and dependency model were proposing almost close number of data items for assessment in the graphical data. Thus, they are analyzed as shown in the below figure 24 from which it could be interpreted that the proposed model is suggesting the least number

of data items for validation. Hence, in this simulation experiment the proposed model suggested the least number of data items to assess.

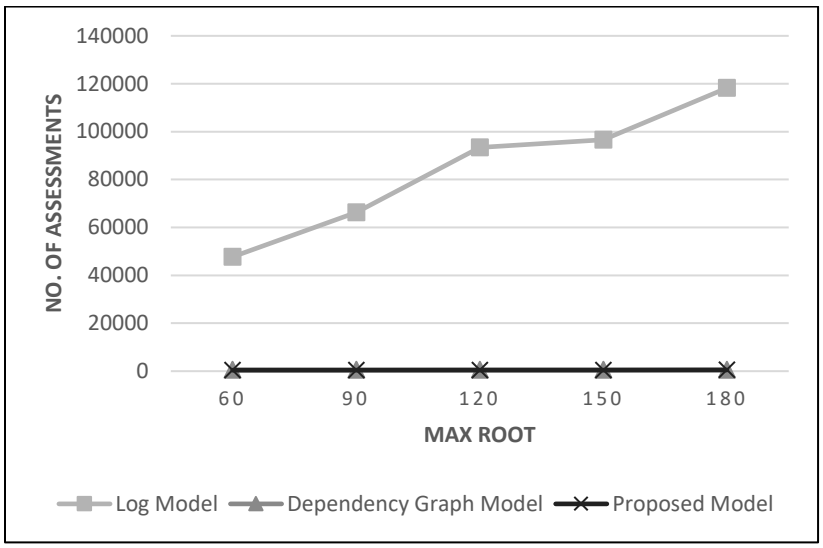

Figure 23. Assessments of 3 models for variations in number of roots

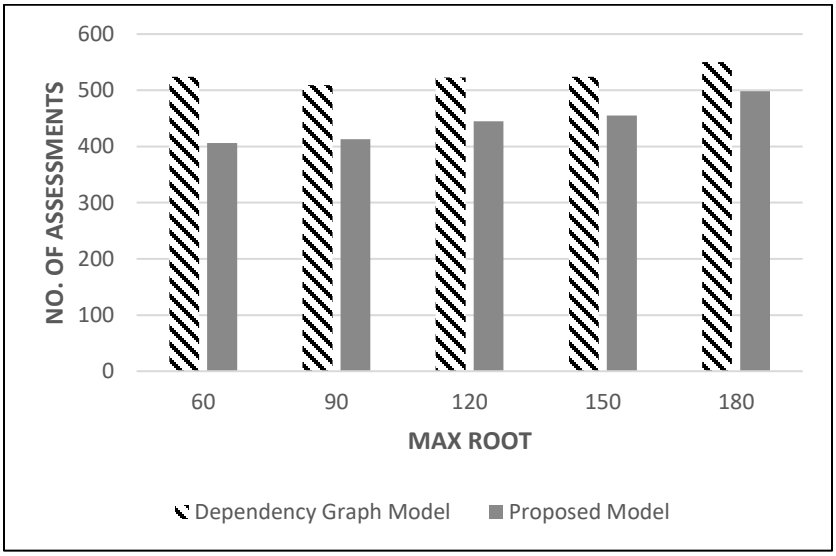

Figure 24. Assessments of 2 models for variations in number of roots

4.3.2 Measuring assessments for 3000 data items for varied maximum number of children

This simulation is carried out for 3000 data items by varying the maximum number of children or dependent nodes for each data item in the matrix under analysis of the impacted CI system. The maximum number of children nodes are varied in terms of [1 to 2%(N)], [1 to 3%(N)], [1 to

4%(N)], [1 to 5%(N)], [1 to 6%(N)] where N being the total number of data items which is 3000. The number of assessments suggested by the three models are plotted with the varied ranges of maximum number of children nodes against the number of assessments suggested by each model respectively.

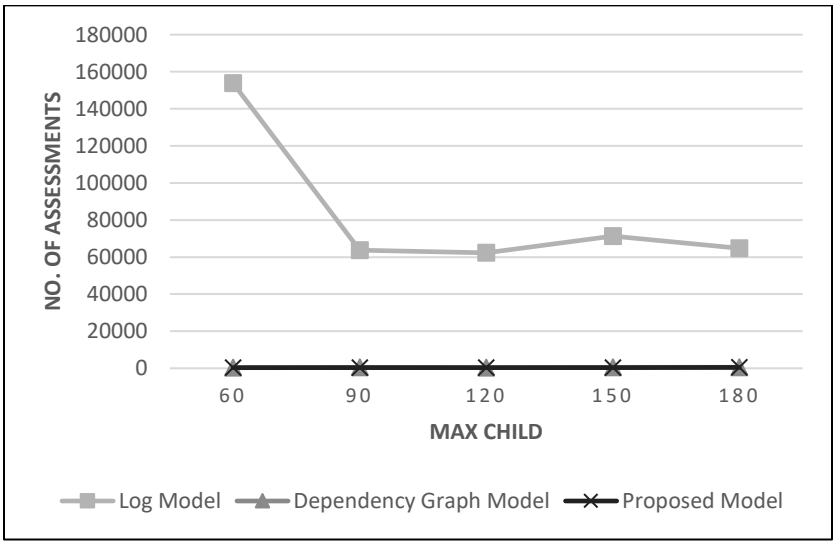

Figure 25. Assessments of 3 models for variations in max. children

As it could be observed from figure 25 the log model is proposing for the highest number of data items to be validated when compared to the other two models.

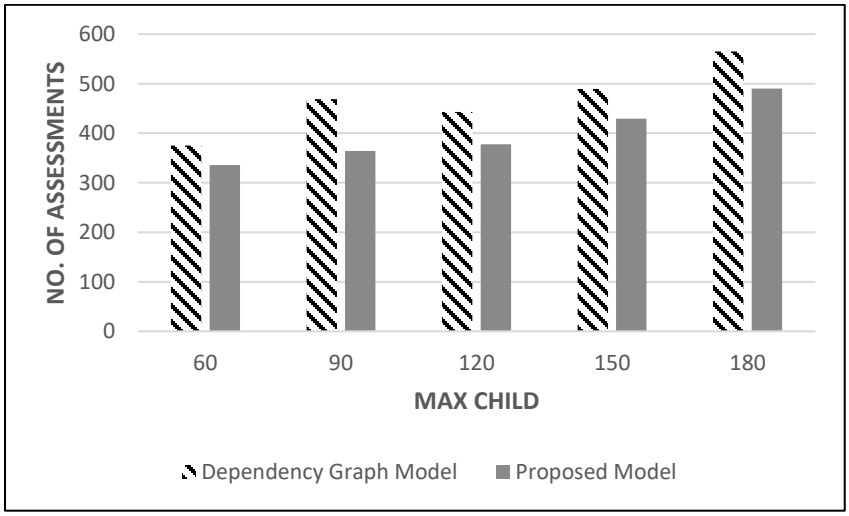

Figure 26. Assessments of 2 models for variations in max. children

The proposed and dependency graph model are showing almost closest number of assessments hence they are analyzed as shown in figure 26 where it could be observed that the proposed model has recommended the least number of data items to be validated. Thus, this simulation results projects that the proposed model has suggested smaller number of data items to be assessed.

4.3.3 Measuring assessments for 3000 data items for varied timestamp ranges

This simulation is carried out for 3000 data items by varying the range of time stamp used for the time of last update of every transaction in the matrix under analysis of the impacted CI system. The timestamp range used for every transaction's time of last update are varied in terms of [1 to 1%(N)], [1 to 2%(N)], [1 to 3%(N)], [1 to 4%(N)], [1 to 5%(N)] where N being the total number of data items which is 3000. The number of assessments suggested by the three models are plotted with the varied ranges of root nodes against the number of assessments suggested by each model respectively.

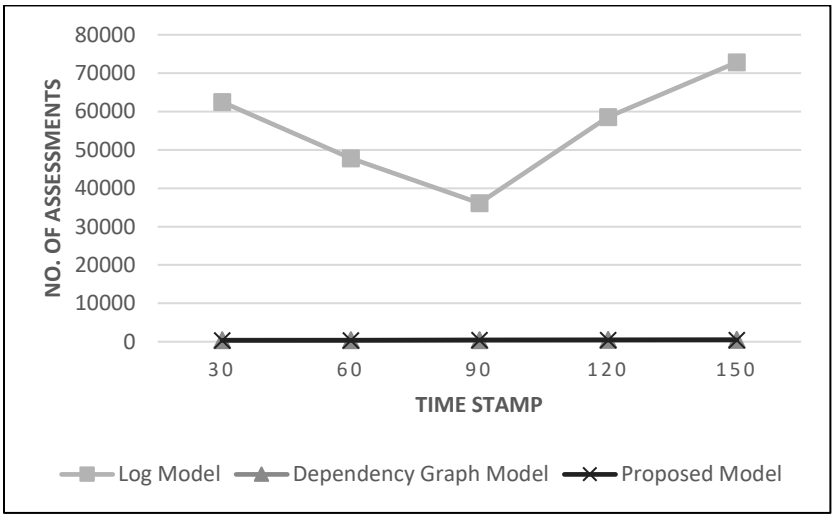

Figure 27. Assessments of 3 models for variations in timestamp range

The figure 27 shows that the log model has suggested the highest number of data items to undergo assessment when compared to the other two models. However, it has also shown that there is very close association in the number of assessments suggested by both dependency and proposed

model. Thus, they have been analyzed together separately to make further observations. In figure 28 it could be observed that the proposed model has proposed smaller number of data items to undergo validation when compared to dependency model. Thus, in this simulation experiment, proposed model has outperformed the other two models.

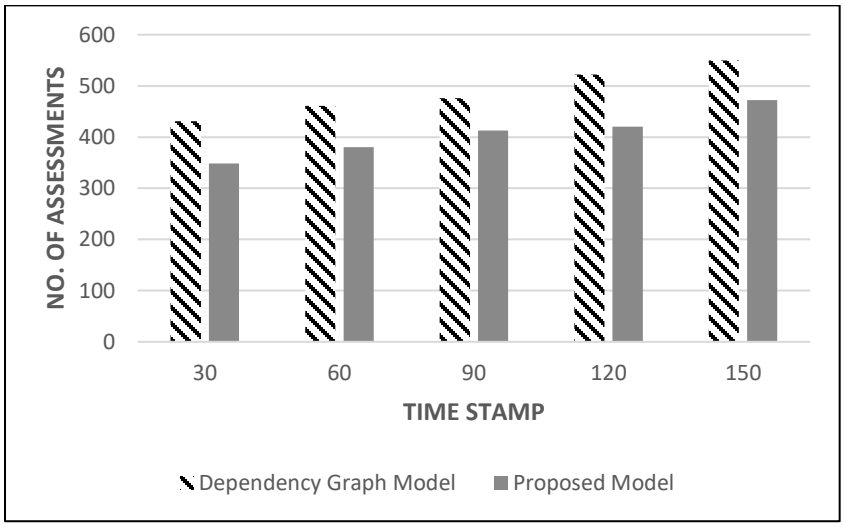

Figure 28. Assessments of 2 models for variations in timestamp range

4.3.4 Measuring assessments for 3000 data items for varied history of transactions range

This simulation is carried out for 3000 data items by varying the range of history of every transaction among data items of the matrix under analysis of the impacted CI system. The number of occurrences of every transaction are varied in terms of [1 to 3], [1 to 4], [1 to 5], [1 to 6], [1 to 7]. The number of assessments suggested by the three models are plotted with the varied ranges of history of transactions against the number of assessments suggested by each model respectively. As it could be observed in figure 29, the highest number of data item assessments are recommended by the log model when compared to the other two models. There is very close association of results projected by both dependency and proposed model. Hence, their results are analyzed separately as shown in figure 30 where it shows that the proposed model has suggested

the least number of assessments. Thus, the simulation result of this experiment projects that the proposed model has recommended smaller number of data items for validations.

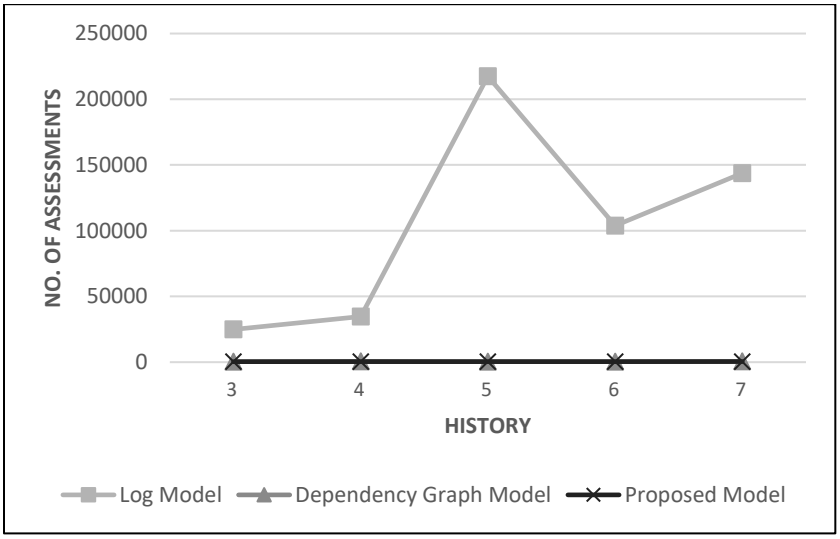

Figure 29. Assessments of 3 models for variations in history of transaction

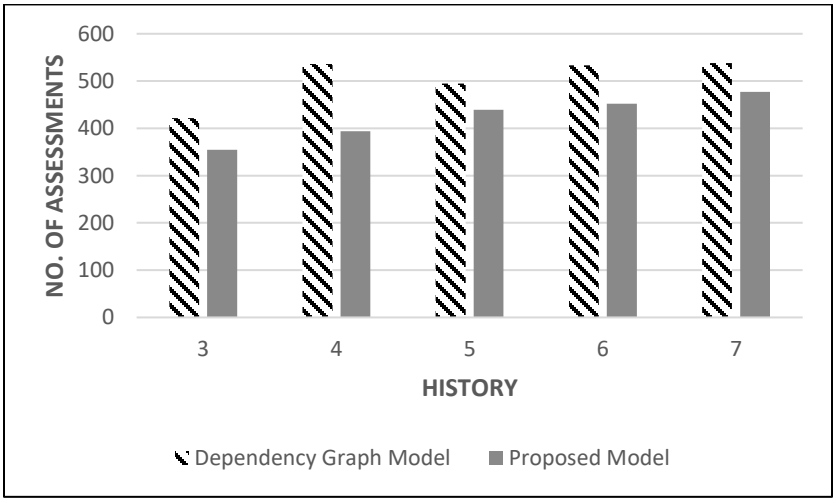

Figure 30. Assessments of 2 models for variations in history of transaction

5 COMPARISON OF DEPENDENCY GRAPH, LOG AND PROPOSED MODEL

The previous sections presented how the log and dependency graph models performed when compared to the proposed model in terms of the number of assessments suggested by them for damage assessment and recovery of the impacted CI system. There are some factors that would make one model perform better when compared to the other among the three models as discussed. The factors include speed, complexity, accuracy, and efficiency.

5.1 Speed

The factor speed could be examined and compared for the models in terms of the time consumed for the execution of the model for proposing their respective suggested number of assessments in the impacted CI system. The log model consumes much higher time when compared to the other two models as it has numerous transactions recorded daily and the extraction of required specific data of interest from such huge log files consumes more time. The dependency graph model performs the assessment by simplifying the work of log model by automatic transaction tree construction when proceeding with the operations. Despite of the huge time consumption and delay caused in log-based model, it manages the data needed for the entire process. This makes it a highly sensitive data which should be in almost protection as it when subjected to threat could reveal confidential information. When compared among the models discussed, proposed model consumes very less time in categorization, assessments, and recovery as it only inputs comparatively lower load for assessments and recovery.

5.2 Complexity

The log model is comparatively more complex when compared to proposed and dependency graph model when it comes to the number of assessments to be made at the time of damage assessment and recovery. The dependency graph model becomes complex when it is subjected to assess and

recover huge data related to impacted CI system as it must construct and analyze huge transaction tree considering numerous dependencies which might concern certain organizations as it involves huge expense in terms of data structures and effort. Comparatively, proposed model is very less complex as it all relies on simple matrix for quick and automatic categorization of nodes and imposes less load towards costly assessments.

5.3 Accuracy

All the models discussed are accurate in damage assessment and recovery but only varies in terms of efficiency when compared. The proposed model is accurate but also simultaneously faster, efficient, and less complex when compared to log and dependency graph model.

5.4 Efficiency

When compared to log and dependency graph model, the proposed model is comparatively efficient as it ensures quicker and less complex model for optimized damage assessment and recovery. The proposed model makes use of only data such as matrix for a very quick categorization of the type of nodes of the impacted CI system. This quick categorization helps to perform very optimized and performance enhancing methods in damage assessment and recovery. As proposed model comparatively results in quicker identification of comparatively lesser number of nodes to be subjected to the assessments and recovery, and at the same time the quick identification and resumption of guaranteed to be safe node to operate, it makes the proposed model highly efficient model when compared to the other two.

5.5 Simulation perspective

The simulation experiments and results reveal that the proposed model requires comparatively lesser number of assessments when compared to log and dependency graph model. The number of data items to be assessed is comparatively lower and the number of data items subjected to

recovery and assessment are well narrowed by using proposed model when working with the impacted CI system recovery. The simulation results show huge difference in the suggestions of the number of data items to be validated among the proposed and other two models (log and dependency graph model). Thus, proposed model is comparatively efficient model for damage assessment and recovery in critical infrastructures.

6 CONCLUSION AND FUTURE WORK

Critical infrastructures are spread out in wide variety of fields for delivering services of vital importance. They are largely distributed systems which are highly coupled, complex, and interconnected. Multiple infrastructures within a CI system are interconnected which introduces higher dependencies and reliabilities on each other. Thus, when a single infrastructure or data item is subjected to attack or failure or damage, it has huge impact on multiple infrastructures within the same CI system. This is due to the cascading effects of the disruptive events which propagates damage from one infrastructure system to the other resulting in economic or societal effects. Thus, optimized damage assessment and recovery post attacks in critical infrastructure system plays a vital role for preventing unnecessary service downtimes and service degradations.

This thesis proposed a model for optimizing the damage assessment and recovery through zoning procedure. The zoning process is the key factor for higher efficiency of this proposed model when compared to the other models such as the log model and dependency graph model. This model included categorization of node or data item types as damaged, undamaged, and skeptical nodes which helped in quick identification of as many safe nodes as possible and as quickly as possible, so that they could resume their operations without undergoing unwanted service downtimes and degradations. This process also helped in quicker and easy identification of number of damaged nodes which could be directly subjected to damage recovery procedures without going through the damage assessment as they are guaranteed to be damaged or contaminated through in depth, intense and quick analysis made in zoning process. The skeptical nodes or nodes of gray zone are the only narrowed down data items which were subjected to damage assessment which includes extensive costly analysis of log information. As only refined narrowed down nodes are

subjected to damage assessments, it has proven that the proposed model is highly optimized and efficient comparatively.

The efficiency of the proposed model is analyzed through the simulation experiments by comparing proposed model's performance with the performance of other models such as log and dependency graph models. The proposed model implementations on simulated data of matrix of the CI system for varying number of data items and varying factors such as timestamp range, history of transactions range, maximum number of children range and number of roots range, projected its performance and efficiency which made evident that the proposed model outperformed the other two models in terms of number of assessments suggested or number of data items to be validated in damage assessment and recovery in CI system.

As future work, the proposed model could be extended to find its application in wide varieties of large distributed critical infrastructure systems with highly coupled, complex, and complicated dependencies among its multiple infrastructures. The extended model could find its application in wide variety of sectors as well. It could also include concurrent damage assessments implementations to be carried out simultaneously when performing the zoning process for categorization of nodes which could save even more time and be even more optimized.

REFERENCES

[1] Wallace, William & Mendonça, David & Lee, Earl & Mitchell, John & Chow, J.. (2003). Managing Disruptions to Critical Interdependent Infrastructures in the Context of the 2001 World Trade Center Attack. Beyond September 11th: An Account of Post-Disaster Research.

[2] President's Commission on Critical Infrastructure Protection. 1997. Critical Foundations: Protecting America's Infrastructures: The Report of the President's Commission on Critical Infrastructure Protection. Washington, D.C.: U.S. Government Printing Office.

[3] Kennedy, R. 2001. "With City Transit Shut Down, New Yorkers Take to Eerily Empty Streets." New York Times (September 12): A8.

[4] Pristin, T. 2001. "Phone Service Improving, but Many Still Lack Power." New York Times (September 12): A12.

[5] Little, R. 2002. "Controlling Cascading Failure: Understanding the Vulnerabilities of Interconnected Infrastructures." Journal of Urban Technology 9(1): 109–123.

[6] U.S. General Accounting Office. 2001. Critical Infrastructure Protection. GAO-01- 323. Washington, D.C.: U.S. General Accounting Office.

[7] Grand, Gwendal & Riguidel, Michel. (2022). A GLOBAL FRAMEWORK TO ENHANCE CRITICAL INFRASTRUCTURE PROTECTION.

[8] Robert J. Ellison, David A. Fisher, Richard C. Linger, Howard F. Lipson, Thomas A. Longstaff, Nancy R. Mead. Survivability: Protecting Your Critical Systems. IEEE Internet Computing, November/December 1999 (Volume 3, No. 6)

[9] Rinaldi, Steven M., James P. Peerenboom, and Terence K. Kelly. Critical infrastructure interdependencies. IEEE Control Systems Magazine, December 2001.

[10] Kotzanikolaou P., Theoharidou M., Gritzalis D. (2013) Cascading Effects of Common-Cause Failures in Critical Infrastructures. In: Butts J., Shenoi S. (eds) Critical Infrastructure Protection VII. ICCIP 2013. IFIP Advances in Information and Communication Technology, vol 417. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-45330-4_12

[11] P. Kotzanikolaou, M. Theoharidou and D. Gritzalis, Interdependencies between critical infrastructures: Analyzing the risk of cascading effects, Proceedings of the Sixth International Conference on Critical Information Infrastructure Security, pp. 107–118, 2011.

[12] P. Kotzanikolaou, M. Theoharidou and D. Gritzalis, Risk assessment of multi-order dependencies between critical information and communication infrastructures, in Critical Information Infrastructure Protection and Resilience in the ICT Sector, P. Theron and S. Bologna (Eds.), IGI Global, Hershey, Pennslyvania, pp. 153–172, 2013.

[13] A. Nieuwenhuijs, E. Luiijf and M. Klaver, Modeling dependencies in critical infrastructures, in Critical Infrastructure Protection, E. Goetz and S. Shenoi (Eds.), Boston, Massachusetts, pp. 205–213, 2008.

[14] S. Rinaldi, J. Peerenboom and T. Kelly, Identifying, understanding and analyzing critical infrastructure interdependencies, IEEE Control Systems, vol. 21(6), pp. 11–25, 2001.

[15] M. van Eeten, A. Nieuwenhuijs, E. Luiijf, M. Klaver and E. Cruz, The state and the threat of cascading failures across critical infrastructures: The implications of empirical evidence from media incident reports, Public Administration, vol. 89(2), pp. 381–400, 2011.

[16] Kotzanikolaou, P., Theoharidou, M. and Gritzalis, D. (2013) 'Assessing n -order dependencies between critical infrastructures', Int. J. Critical Infrastructures, Vol. 9, Nos. 1/2, pp.93–110.

[17] Aung, Z. and Watanabe, K. (2009) 'A framework for modeling interdependencies in Japan's critical infrastructures', in Palmer, C. and Shenoi, S. (Eds.): 3rd IFIP Int. Conf. on Critical Infrastructure Protection (CIP-2009), pp.243–257, Springer, USA.

[18] Barrett, C., Beckman, R., Channakeshava, K., Huang, F., Kumar, V., Marathe, A., Marathe, M. and Pei, G. (2010) 'Cascading failures in multiple infrastructures: from transportation to communication network', in 5th Int. Conf. on Critical Infrastructure (CRIS), pp.1–8.

[19] Rinaldi, S. (2004) 'Modeling and simulating critical infrastructures and their interdependencies', in 37th Hawaii Int. Conf. on System Sciences, Vol. 2, USA, IEEE.

[20] Svedsen, N. and Wolthunsen, S. (2007) 'Connectivity models of interdependency in mixed-type critical infrastructure networks', Information Security Technical Report, Vol. 1, pp.44–55.

[21] Theoharidou, M., Kotzanikolaou, P. and Gritzalis, D. (2010) 'A multi-layer criticality assessment methodology based on interdependencies', Computers & Security, Vol. 29, No. 6, pp.643–658.

[22] Theoharidou, M., Kotzanikolaou, P. and Gritzalis, D. (2009) 'Risk-based criticality analysis', in C. Palmer and S. Shenoi (Eds.): 3rd IFIP Int. Conf. on Critical Infrastructure Protection (CIP-2009), pp.35–49, Springer, USA.

[23] E. Zio and G. Sansavini, Modeling interdependent network systems for identifying cascade-safe operating margins, IEEE Transactions on Reliability, vol. 60(1), pp. 94–101, 2011.

[24] E. Viganò, M. Loi., E. Yaghmaei, "Cybersecurity of Critical Infrastructure", In Christen M., Gordijn B., Loi M. (eds), The Ethics of Cybersecurity, The International Library of Ethics, Law and Technology, vol 21, Springer

[25] Critical Infrastructure Security, [retrieved: April 7, 2022], https://www.cisa.gov/infrastructure-security.

[26] Ding, Jianguo & Atif, Y. & Andler, Sten & Lindström, Birgitta & Jeusfeld, Manfred. (2017). CPS-based Threat Modeling for Critical Infrastructure Protection. ACM SIGMETRICS Performance Evaluation Review. 45. 129-132. 10.1145/3152042.3152080.

[27] Canzani, Elisa & Kaufmann, Helmut & Lechner, Ulrike. (2016). Characterising Disruptive Events to Model Cascade Failures in Critical Infrastructures. 10.14236/ewic/ICS2016.11.

[28] David Rehak, Jiri Markuci, Martin Hromada, Karla Barcova, "Quantitative evaluation of the synergistic effects of failures in a critical infrastructure system", International Journal of Critical Infrastructure Protection, Volume 14, 2016, Pages 3-17, ISSN 1874-5482, https://doi.org/10.1016/j.ijcip.2016.06.002.

[29] N. Bartolini, S. Ciavarella, T. F. La Porta and S. Silvestri, "Network Recovery After Massive Failures," 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2016, pp. 97-108, doi: 10.1109/DSN.2016.18.

[30] Ciavarella, S. & Bartolini, Novella & Khamfroush, Hana & Porta, T.. (2017). "Progressive damage assessment and network recovery after massive failures." 1-9. 10.1109/INFOCOM.2017.8057042.

[31] Sanger, D. E., Perlroth, N, [retrieved: April 7, 2022], "Pipeline Attack Yields Urgent Lessons About U.S. Cybersecurity", https://www.nytimes.com/2021/05/14/us/politics/pipeline-hack.html

[32] Arampatzis Anastasios, [retrieved: April 7, 2022], "Is the Electric Grid Ready to Respond to Increased Cyber Threats?", https://www.tripwire.com/state-of-security/ics-security/electric-grid-ready-increased-cyber-threats/

[33] Brian Barrett, [retrieved: April 7, 2022], "An Unprecedented Cyberattack Hit US Power Utilities", https://www.wired.com/story/power-grid-cyberattack-facebook-phone-numbers-security-news/

[34] Kate O'Flaherty, [retrieved: April 7, 2022], "U.S. Government Issues Powerful Cyberattack Warning As Gas Pipeline Forced Into Two Day Shut Down https://www.forbes.com/sites/kateoflahertyuk/2020/02/19/us-government-issues-powerful-cyberattack-warning-as-gas-pipeline-forced-into-two-day-shut-down/#5f3061645a95

[35] Morgan Lewis, [retrieved: April 7, 2022], "Cyberattack Forces Gas Pipeline Shutdown", https://www.jdsupra.com/legalnews/cyberattack-forces-gas-pipeline-shutdown-76217/

[36] Du, M., Li, F., Zheng, G. and Srikumar, V., 2017, October. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In Proceedings of the 2017 ACM SIGSAC conference on computer and communications security (pp. 1285-1298).

[37] Rumman Sobhan and Brajendra Panda, "Reorganization of Database Log for Information Warfare Data Recovery", Database and Application Security XV, M. Olivier and D. Spooner (Editors), Kluwer Academic Press, 2002.

[38] Panda B., Giordano J. (1999) Reconstructing the Database After Electronic Attacks. In: Jajodia S. (eds) Database Security XII. IFIP — The International Federation for Information Processing, vol 14. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-35564-1_9

[39] Abdulwahab Alazeb and Brajendra Panda, "Maintaining Data Integrity in Fog Computing Based Critical Infrastructure Systems", In Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence, in Research Track / Symposium on Cyber Warfare, Cyber Defense, and Cyber Security (CSCI-ISCW), Las Vegas, Nevada, USA, December 5-7, 2019, 2019.

[40] Justin Burns, Brajendra Panda, and Thanh Bui, "Modeling Damage Paths and Repairing Objects in Critical Infrastructure Systems", In Proceedings of the Fifteenth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2021), Athens, Greece, November 14-18, 2021.