ABSTRACT

In Government incentive schemes, the issue generally faced by authorities are duplicity and fraudulent claims. The various investigative agencies try to analyses the data for such claims and employ various heuristic algorithms in detecting anomalies from the data, which is generally referred to as forensic accounting. The Director General of Foreign Trade (DGFT) register all exporters and importers of goods with IEC Code. Various incentives are provided to exporters to encourage exports for earing foreign exchange. It was observed that these schemes were misused by some exporters by making multiple claims for same export with different identification details. A beneficiary is entitled to get only one Import/Export Code. However, network analysis revealed that a single beneficiary was enrolled using different attributes such as mobile number, email, PAN, bank account number, etc. and obtained multiple IECs. In in the proposed system data points analysed are Unique IE Code Other Identification attributes, such as Mobile number, E-mail, PAN Number. More attributes viz GST No etc. can be added to further such duplicate registrations. In the proposed system project of network analysis gets data into the proper structure by cleansing and transforming the data and then creates specific object classes for the network package, as well as for igraph and tidygraph, which is based on the igraph implementation. Then the creation of interactive graphs with the viz Network and network D3 packages is done. Finally, the effectiveness of the deduction using network analysis is tested on the test data by calculating the confusion matrix with the deducted outcome and the actual outcome. Since it an attribute analysis, the accuracy between the analysis outcome and the actual outcome was calculated. The accuracy value is 94.6% indicating the efficiency of this technique.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACRONYM	ABBREVIATION
DGFT	Director General of Foreign Trade
IEC	Import Export Code
PAN	Permanent Account Number
IE	Import Export
GST	Goods and Services Tax
GUI	Graphical User Interface
ETL	Extract Transform and Load
ReLU	Rectified Linear Unit
RUS	Random Under Sample
SMOTE	Synthetic Minority Oversampling Technique
GOI	Government of India
ITC(HS)	Income Tax Commission
CN	Complex Network
DC	Data Compression
IEEE	Institute of Electrical and Electronic Engineers
CA	Cellular Automata
IISE	Impact Increments-based State Enumeration
SFN	Stochastic Flow Network
MADM	Multiple Attribute Decision Making
RASS	Regenerative Adaptive Subset Simulation
UML	Unified Modeling Language
ML	Machine Learning
BFS	Breadth First Search
SQL	Structured Query Language
DAG	Directed Acyclic Graph
SNA	Social Network Analysis

DFS	Depth First Search
CRAN	Comprehensive R Archive Network
FP	False Positive
TP	True Positive
TN	True Negative
FN	False Negative
FNR	False Negative Rate
TPR	True Positive Rate

CHAPTER 1
INTRODUCTION

Over a wide range of fields network analysis has become an increasingly popular tool to deal with the complexity of the interrelationships between data elements and attributes. The promise of network analysis is the placement of significance on the relationships between attributes, rather than seeing data items as isolated entities. The emphasis on complexity, along with the creation of a variety of algorithms to measure various aspects of networks, makes network analysis a central tool for data analytics especially in forensic accounting. This proposed system ties to apply networks implementation in R to locate interconnects between record sets.

There are a number of applications designed for network analysis and the creation of network graphs such as gephi and cytoscape. R has developed into a powerful tool for network analysis. The strength of R in comparison to stand-alone network analysis software is three fold. In the first place, R enables reproducible research that is not possible with GUI applications. Secondly, the data analysis power of R provides robust tools for manipulating data to prepare it for network analysis. Finally, there is an ever growing range of packages designed to make R a complete network analysis tool. Significant network analysis packages for R include the statnet suite of packages and igraph. In addition, recently released the tidygraph and ggraph packages leverage the power of igraph in a manner consistent with the tidyverse workflow. R can also be used to make interactive network graphs with the html widgets framework that translates R code to JavaScript.

1.1 OVERVIEW

This proposed system begins with the process ETL (Extract Transform and Load) for getting data into the proper structure for network analysis. The network analysis packages have all implemented their own object classes. In this proposed system, the specific object classes for the statnet suite of packages with the network package, as well as for igraph and tidygraph, which is based on the igraph implementation is used for network analysis. Finally, the creation of interactive graphs with the vizNetwork and networkD3 packages is used for visual analysis and introspection. The output dataset is written to CSV file using R Studio code. Finally, the accuracy is calculated using Python tools. The accuracy metrics such as specificity, sensitivity, cohen's kappa and recall are calculated.

1.2 PROBLEM STATEMENT

Authorities typically deal with deceit and false statements in government incentive programmes. The many investigative agencies attempt to analyze the data for such claims and use a variety of heuristic methods to find anomalies from the data, which is typically referred to as forensic accounting. All merchandise exporters and importers are given an IEC Code by the Director General of Foreign Trade (DGFT). To encourage exports for the purpose of earning foreign currency, exporters are given a variety of incentives. It was discovered that some exporters were abusing these methods by filing multiple claims for the same export with various forms of identification. A beneficiary may only receive one import/export code. Yet, network analysis showed that one beneficiary was registered using a variety of characteristics such as E-mail id, Permanent Account Number, Mobile number, IE Code. Any person can hold multiple values of these, in essence, a person may have more than one mobile number or one PAN number.

Using these, one beneficiary could register multiple times. To sum up the above statement, it is in essence a fraudulence or an attempt at fraudulence. In order to deduce this, we use the classification algorithm Network Analysis from Graph Theory in order to attempt to classify the people who have registered using the same identification details.

Overall, the classification algorithm is applied on the dataset and the efficiency is verified by comparing the result of the Testing dataset.

1.3 EXISTING SYSTEM

Detecting tax fraud is a top objective for practically all tax agencies in order to maximize revenues and maintain a high level of compliance. Data mining, machine learning, and other approaches such as traditional random auditing have been used in many studies to deal with tax fraud. The goal of that study was to use Artificial Neural Networks to identify factors of tax fraud in income tax data. The results showed that Artificial Neural Networks performed well in identifying tax fraud with an accuracy of 92%, a precision of 85%, a recall score of 99%, and an AUC-ROC of 95%.

All businesses, either cross-border or domestic, the period of the business, small businesses, and corporate businesses, are among the factors identified by the model to be more relevant to income tax fraud detection. That study was consistent with the previous closely related work in terms of features related to tax fraud where it covered all tax types together using different machine learning models. To the best of our knowledge, that study is the first to use Artificial Neural Networks to detect income tax fraud in Rwanda by comparing different parameters such as layers, batch size, and epochs and choosing the optimal ones that give better accuracy than others. For this study, a simple model with no hidden layers, soft sign activation function performs better. The evidence from this study will help auditors in understanding the factors that contribute to

income tax fraud which will reduce the audit time and cost, as well as recover money foregone in income tax fraud.

1.3.1 Materials and Methods

a. Dataset

The used data in this research was obtained from the Rwanda Revenue Authority, the national body in charge of tax collection in Rwanda. The dataset consisted of 7840 audited taxpayers from across the country, with 1655 (21.1%) found guilty of fraud and 6185 (78.9%)cleared of any tax fraud. Each taxpayer was identified by features such as Province, Business scale, Sector, Business origin Department, Operation time and others. For feature exploration, provides a comprehensive view of the dataset and the important features in it. There were no missing or duplicate values in our data. There were no outliers since our data set was composed of categorical variables except one variable that indicates the difference in time from when a business was registered to the time of audit that was composed of integers.

b. Libraries

Sigmoid, ReLu, softmax, softsign, linear, hard-sigmoid, softplus, and others were tested using the grid search to determine which activation function is best suited for income tax fraud detection. Sigmoid is the most commonly utilized activation function because it is a non-linear function that may convert and squash numbers in the range of 0 to 1. It is critical to have a sigmoid activation function on the output layer in cases of binary classification. A rectified linear unit (ReLU), also known as a piecewise linear function, works in such a way that it will output the input directly if it is positive, or zero otherwise. This function is widely used in neural networks and is widely assumed to be more efficient than others since neurons are not activated all at once, but instead, a subset of neurons are activated at a time. It is recommended that this function

be tried first because numerous studies have shown that it performs well in a variety of tasks

c. Data Cleaning and Preparation

There was an imbalance in our data because, out of 7840 data points of taxpayers,6185, or 78.8%, were labeled as non-fraudulent, while only 1655, or 21.2%, were labeled as fraudulent. To address such issues, a combination of both under-sampling and over- Future Internet 2022, of 14sampling methods were used, namely the Random Under sampling(RUS) and the synthetic minority over-sampling technique (SMOTE). This is because the dataset is quite small and dropping a lot of majority observations would result in a significant loss of information needed in a model while only oversampling would increase the likelihood of overfitting

d. Data Exploration

Using Grid Search for hyper-parameter tuning, combinations of many parameters mentioned above were tried to see which combination is the most accurate in classifying fraudulent and non-fraudulent taxpayers. In total, 576 different combinations were tested and even if we cannot show them all in this paper, we can illustrate some of those parameter combinations, as well as how accurate each one was in regards to the training accuracy, validation accuracy and test accuracy. It also indicates how many layers are there, with each layer having a different number of neurons

e. Training

RUS is straightforward under-sampling method that randomly removes data points from the majority class in order to balance the dataset. While using SMOTE, each minority class sample is over-sampled by introducing synthetic examples similar to k minority class nearest neighbors. Neighbors from the

knearest neighbors are chosen at random depending on the amount of oversampling required. After resampling, the remaining dataset was of 4000 data points which was divided into training and testing ratios of 80% and 20%, respectively.

f. Classification Metrics

To check how well our model, we use some metrics to find the accuracy of our model. There are many types of classification metrics available in Scikit learn, Confusion Matrix, Accuracy Score Precision, Recall, F1-Score

1.4 PROPOSED SYSTEM

Authorities generally face difficulty in detecting false claims and duplicate claims in government incentive programmes. The various investigating agencies attempt to analyze the data for such claims and use a variety of heuristic methods in spotting irregularities from the data, a process known as forensic accounting.

Government of India (GoI) has framed out various schemes of export incentives to exporters to promote exports and earn foreign exchange. The Directorate General of Foreign Trade (DGFT) is the agency of the Ministry of Commerce and Industry of the Government of India responsible for administering laws regarding foreign trade. DGFT provides a complete searchable database of all exporters and importers of India. The search can be completed only if full IEC code and first three letters of company name are entered.

The Central Government appoints the Directorate General of Foreign Trade. Normally a senior member of the Indian Administrative Service having more years is appointed to the post of the Director-General of Foreign Trade. The Director-General heads an attached office under the administrative control

of the Ministry of Commerce and Industry of the Government of India. The Director-General is an Ex-Officio Additional Secretary to the Government of India. The Director-General advises the central Government in the formulation of Foreign Trade Policy and is responsible for carrying out that Policy. At present, the Director-General formulates Foreign Trade Policy and Procedures of Foreign Trade Policy and ITC (HS) Classifications of Import and Export Items.

Keeping in line with liberalization and globalization and the overall objective of increasing of exports, DGFT has since been assigned the role of "facilitator". The shift was from prohibition and control of imports/exports to promotion and facilitation of exports/imports, keeping in view the interests of the country.

All exporters and importers of commodities are given an IEC Code by the DGFT. Some exporters were found to be abusing these schemes by filing multiple claims for the same export using various forms of identification. A beneficiary is only permitted to get one import/export code. Yet, network analysis showed that a single beneficiary was registered using many characteristics.

This network analysis proposed system transforms and cleans the data to give it the right structure before creating specialized object classes for the network package, igraph, and tidygraph, which is based on the igraph implementation. Then interactive graphs are made using the *viz* network and graph programmes. Lastly, by constructing the confusion matrix with the deducted outcome and the actual outcome, the effectiveness of the deduction using network analysis is tested on the test data. Given that it was an attribute analysis, the accuracy between the results of the analysis and the real results was determined.

CHAPTER 2
LITERATURE REVIEW

Au SK et al. (2016) [1] came up with an idea for a new simulation approach, called 'subset simulation', is proposed to compute small failure probabilities encountered in reliability analysis of engineering systems. The basic idea is to express the failure probability as a product of larger conditional failure probabilities by introducing intermediate failure events. With a proper choice of the conditional events, the conditional failure probabilities can be made sufficiently large so that they can be estimated by means of simulation with a small number of samples.

Billinton R et al. (2019) [2] put forth a hybrid approach using Monte-Carlo simulation and an enumeration technique for the reliability evaluation of large scale composite generation-transmission systems is presented. The Monte-Carlo method is based on combining the basic random sampling technique with a direct analytical approach for system analysis and the utilization of a minimization model for load curtailment. The method presented is suited to the analysis of large systems and can be used to include multi-state representation of generating units.

Bompard E et al. (2019) [3] proposed a promising approach for the structural analysis of transmission grids with respect to their vulnerabilities is to use metrics and approaches derived from complex network (CN) theory that are shared with other infrastructures such as the World-Wide Web, telecommunication networks, and oil and gas pipelines. These approaches, based on metrics such as global efficiency, degree and betweenness, are purely topological because they study structural vulnerabilities based on the graphical representation of a network as a set of vertices connected by a set of edges. Unfortunately, these approaches fail to capture the physical properties and operational constraints of power systems and, therefore, cannot provide meaningful analyses.

Chaturvedi SK (2016) [4] suggested that in Engineering theory and applications, we think and operate in terms of logics and models with some acceptable and reasonable assumptions. The present text is aimed at providing modelling and analysis techniques for the evaluation of reliability measures (2-terminal, all- terminal, k-terminal reliability) for systems whose structure can be described in the form of a probabilistic graph.

Chen G et al. (2012) [5] proposes a hybrid approach for structural vulnerability analysis of power transmission networks, in which a DC power flow model with hidden failures is embedded into the traditional error and attack tolerance methodology to form a new scheme for power grids vulnerability assessment and modeling. The new approach embodies some important characteristics of power transmission networks. Furthermore, the simulation on the standard IEEE 118 bus system demonstrates that a critical region might exist and when the power grid operates in the region, it is vulnerable to both random and intentional attacks. Finally, a brief theoretical analysis is presented to explain the new phenomena.

Chui KT et al. (2019) [6] proposed that there are many complex networks like World-Wide Web, internet and social networks have been reported to be scale-free. The major property of scale-free networks is their degree distributions are in power law form. Generally, the degree exponents of scale- free networks fall into the range of (2, 3). The purpose of this paper is to investigate other situations where the degree exponents may lie outside the range.

Crucitti P et al. (2017) [7] The concept of network efficiency, recently proposed to characterize the properties of small- world networks, is here used to study the effects of errors and attacks on scale-free networks. Two different kinds of scale-free networks, in essence, networks with power law P (k), are considered:(1) scale-free networks with no local clustering produced by the Barabasi–Albert model and (2) scale-free networks with high clustering properties as in the model by Klemm and Eguíluz.

He L Shang et al. (2016) [8] proposed a paper that concentrates on the performance evaluation of networks, whose arc failure rates are not deterministic numbers, but imprecise ones. Conventional literatures analyze the network reliability assuming that the failure rates of all components in networks following the same membership function. However, most real-world networks do not abide by this regulation, especial complex ones. Therefore, in this paper, a new method is developed based on cellular automata (CA) and fuzzy logic

Hou K et al. (2016) [9] in his paper proposes an impact increments-based state enumeration (IISE) reliability assessment approach, specially designed for transmission systems. Firstly, the reliability index calculation formula of the traditional state enumeration technique is transformed into an impact increments-based formation. With the derived formula, the calculation of state probability is simplified and the weight of low order contingency states is increased.

Lin YK et al. (2013) [10] gave service-level agreements for data transmission often define criteria such as availability, delay, and loss. Internet service providers and enterprise customers are increasingly focusing on tolerable error rate during transmission. Focusing on a stochastic flow network (SFN), this study extends reliability evaluation to considering tolerable error rate, in which network reliability is the probability that demand can be satisfied.

Liu H et al. (2020) [11] put forth as the support network of power grid, the reliability and stability of power communication network structure become more and more important with the continuous expansion of power grid scale. Identifying and protecting important nodes in the power communication network in advance is an effective method to improve the robustness of the network. Therefore, this paper proposes a comprehensive evaluation method for the importance of nodes in the power communication network based on MADM (Multiple Attribute Decision Making).

Mahadevan S et al. (2001) [12] proposes a methodology to apply Bayesian networks to structural system reliability reassessment, with the

incorporation of two important features of large structures:(1) multiple failure sequences, and (2) correlations between component-level limit states. The proposed method is validated by analytical comparison with the traditional reliability analysis methods for series and parallel systems.

Miao F et al. (2011) [13] suggested that "Regenerative Adaptive Subset Simulation" (RASS) method is proposed for performing the reliability analysis of complex structural systems. Proposed modifications to the classic subset simulation method include the implementation of advanced Markov Chain processes to combine the benefits of a Markov Chain regeneration process, a Delayed Rejection and Adaptive sample selection algorithms and a Component wise sampling model.

Moreno JA (2012) [14] suggested Two cellular automata (CA) models that evaluate the s-t connectedness and shortest path in a network are presented. CA based algorithms enhance the performance of classical algorithms, since they allow a more reliable and straightforward parallel implementation resulting in a dynamic network evaluation, where changes in the connectivity and/or link costs can readily be incorporated avoiding recalculation from scratch. The paper also demonstrates how these algorithms can be applied for network reliability evaluation (based on Monte-Carlo approach) and for finding s–t path with maximal reliability.

Murray L et al. (2013) [15] came up with the idea of Splitting is a variance reduction technique widely used to make efficient estimations of the probability of rare events in the simulation of Markovian models. In this article, splitting is applied to improve a well-known method called the Creation Process used in network reliability estimation. The resulting proposal, called here Splitting, is particularly appropriate in the case of highly reliable networks; in essence, networks for which failure is a rare event.

Ramirez- Marquez JE et al. (2015) [16] presents a new algorithm that can be readily applied to solve the all-terminal network reliability allocation

problems. The optimization problem solved considers the minimization of the network design cost subject to a known constraint on all-terminal reliability by assuming that the network contains a known number of functionally equivalent components (with different performance specifications) that can be used to provide redundancy.

Rebaiaia ML et al. (2013) [17] proposed a paper for Network reliability analysis problem is the center of many scientific productions. It consists of evaluating the all-terminal reliability of networks. Two classes have emerged; exact and approximate methods. The aim of this paper is to present an efficient exact method for enumerating minimal cuts of R-networks. The algorithm proceeds by determining minimal paths set and from which minimal cuts are generated by managing binary decision diagrams.

Shahraki AF et al. (2017) [18] suggested that degradation modeling is an effective approach for reliability assessment, remaining useful life prediction, maintenance planning, and prognostics health management. Degradation models are usually developed based on degradation data and/or prior understandings of physics behind degradation processes of products or systems. Further, the effects of environmental or operational conditions on degradation processes and the knowledge about the dependency between degradation processes help improve the explanatory capabilities of degradation models. This paper presents a comprehensive review of existing degradation modeling approaches commonly used in engineering applications. To assist practitioners in understanding the concept of degradation modelling, the existing methods are classified into two broad categories

Wang J et al. (2018) [19] proposed 30 energy infrastructure models dedicated for the modeling and simulation of power or natural gas networks are collected and reviewed using the emerging concept of resilience. Based on the review, typical modeling approaches for energy infrastructure resilience problems are summarized and compared. The authors, then, propose five indicators for evaluating a resilience model; namely, catering to different

stakeholders, intervening in development phases, dedicating to certain stressor and failure, taking into account different interdependencies, and involving socio-economic characteristics.

Yeh WC et al. (2010) [20] said that network reliability is very important for the decision support information. Monte Carlo Simulation is one of the optimal algorithms to estimate the network reliability for different kinds of network configuration. The traditional reliability estimation requires the information of all Minimal Paths or Minimal Cuts. However, finding all minimum cuts is extremely computationally expensive. This paper has compared and analyzed three Monte Carlo Simulation methods for estimating the two-terminal network reliability.

Yeh WC et al. (2017) [21] evaluates the network reliability is an important topic in the planning, designing, and control of systems. The sum-of-disjoint products technique is a major fundamental tool for evaluating stochastic network reliability. In this study, a new sum of disjoint products based on some intuitive properties that characterize the structure of minimal paths, and the relationships between MPs and sub paths are developed to improved sum of disjoint products. The proposed sum of disjoint products is easier to understand and implement, and better than the existing best- known sum of disjoint products based algorithms under some special situation. The correctness of the proposed algorithm will be analyzed and proven. One bench example is illustrated to show how the network reliability with known MPs is determined using the proposed sum of disjoint products.

Younes A et al. (2015) [22] presents an algorithm for finding the minimal paths of a given network in terms of its links. Then, it presents an algorithm for calculating the reliability of the network in terms of the probabilities of success of the links of its minimal paths. The algorithm is based on a relation that uses the probabilities of the unions of the minimal paths of the network to obtain the network reliability. Also, the paper describes a tool that has been built for calculating the reliability of a given network. The tool has two main phases: the

minimal paths generation phase, and the reliability computation phase. The first phase accepts the links of the network and their probabilities, then implements the first proposed algorithm to determine its minimal paths. The second phase implements the second proposed algorithm to calculate the network reliability.

Zaheri MM et al. (2020) [23] in this paper, a Cellular Automata based simulation-optimization approach is proposed for the optimal design of household sewer networks. A two-phase CA is used as the optimization tool while the EPA's storm water management model is used as the simulator. A splitting method is first used to redefine the sewer network design problem in terms of two simpler sub-problems with diameters and nodal elevations of each pipe as decision variables which are iteratively solved using CA methods in two-stage manner.

Zhai Q et al. (2013) [24] said warm standby sparing is a fault-tolerance technique that attempts to improve system reliability while compromising the system energy consumption and recovery time. However, when the imperfect fault coverage effect (an uncovered component fault can propagate and cause the whole system to fail) is considered, the reliability of a warm standby sparing can decrease with an increasing level of the redundancy. This article studies the reliability of a warm standby sparing subject to imperfect fault coverage, in particular, fault level coverage where the coverage probability of a component depends on the number of failed components in the system.

Zuev KM et al. (2015) [24] propose a stochastic framework for quantitative assessment of the reliability of network service, formulate a general network reliability problem within this framework, and then show how to calculate the service reliability using Subset Simulation, an efficient Markov chain Monte Carlo method that was originally developed for estimating small failure probabilities of complex dynamic systems. The efficiency of the method is demonstrated with an illustrative example where two small-world network generation models are compared in terms of the maximum-flow reliability of the networks that they produce.

CHAPTER 3
SYSTEM DESIGN

In this chapter, the various UML diagrams for the Duplicate Detection using Machine Learning is represented and the various functionalities are explained.

3.1 UNIFIED MODELING LANGUAGE

Unified Modeling language is a standardized modeling languageenabling developers to specify, visualize, construct and document artifacts of a software system. Thus, UML makes these artifacts scalable, secure and robust in execution. It uses graphic notation to create visual models of software systems. UML is designed to enable users to develop an expressive, ready to use visual modeling language. In addition, it supports high-level development concepts suchas frameworks, patterns and collaborations. Some of the UML diagrams are discussed.

3.1.1 Use Case Diagram of Duplicate Detection

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So it can be said that uses cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors.

Actors can be defined as something that interacts with the system. The actors can be human user, some internal applications or may be some external applications.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. The end result of most use cases should be obtained. A successful diagram should describe the activities and variants used to reach the goal.

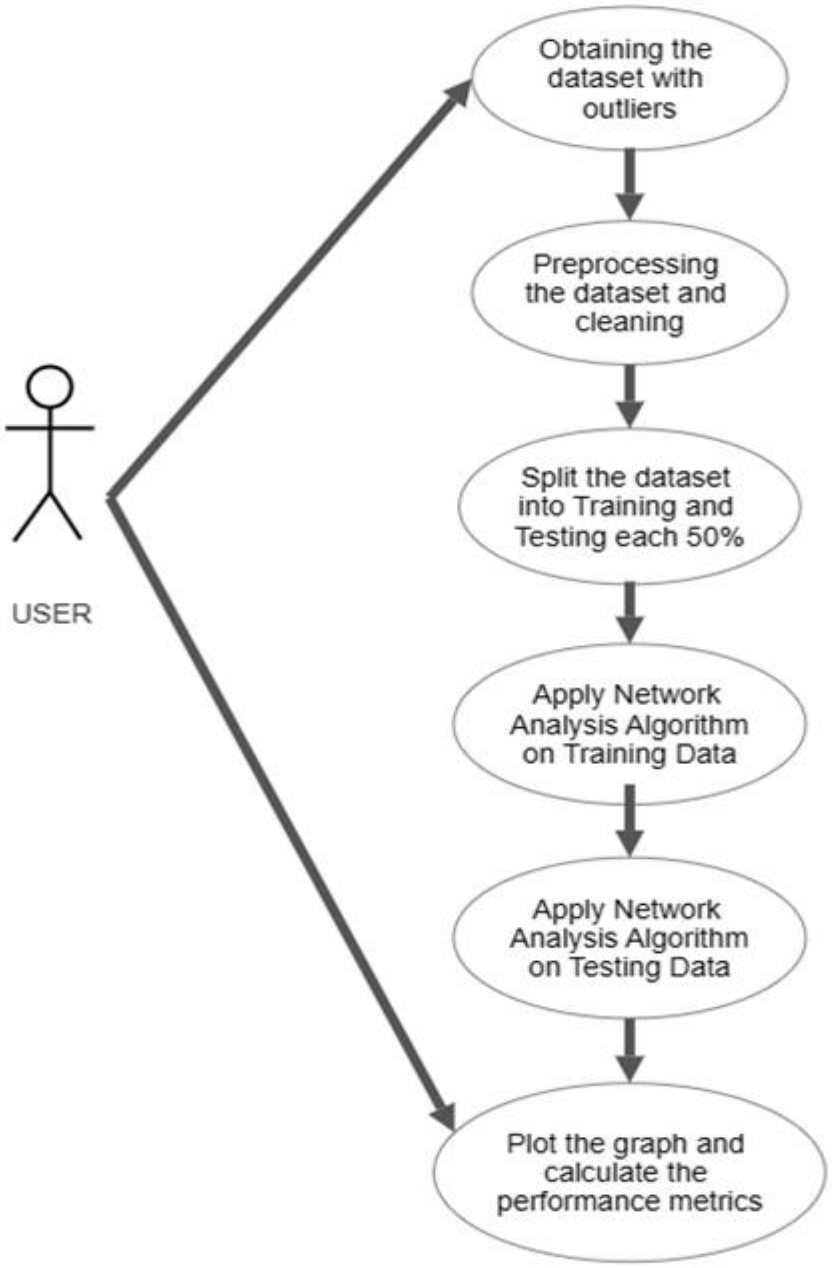

Fig 3.1 Use case diagram of Duplicate Detection

Fig 3.1 shows that the functionalities are to be represented as a use case in the representation. Each and every use case is a function in which the user or the

server can have the access on it. The suggested system's use case diagram flow begins with data collection, followed by cleaning and exploration, supervised machine learning algorithm training, and user input testing. Finally, the performance is analyzed.

3.1.2 Class Diagram of Duplicate Detection

Fig 3.2 shows that class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represents the whole system. The name of the class diagram should be meaningful to describe the aspect of the system.

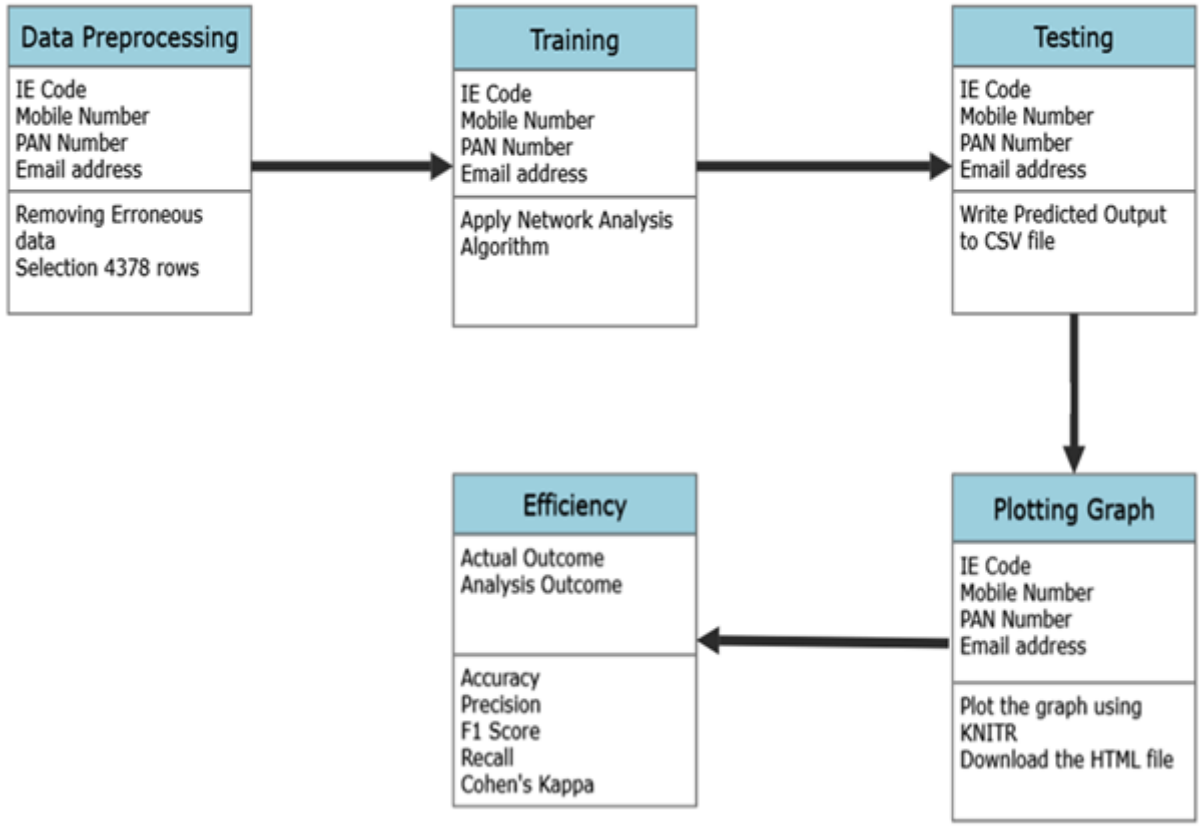

Fig 3.2 Class Diagram of Duplicate Detection

The proposed system's class diagram begins with data collection, followed by cleaning and exploration, supervised machine learning algorithm training, and user input testing.

3.1.3 Sequence Diagram of Duplicate Detection

Fig 3.3 shows that UML sequence diagrams model the flowof logic within the system in a visual manner, enabling to both document and validate the logic, and are commonly used for both analysis and design purposes.

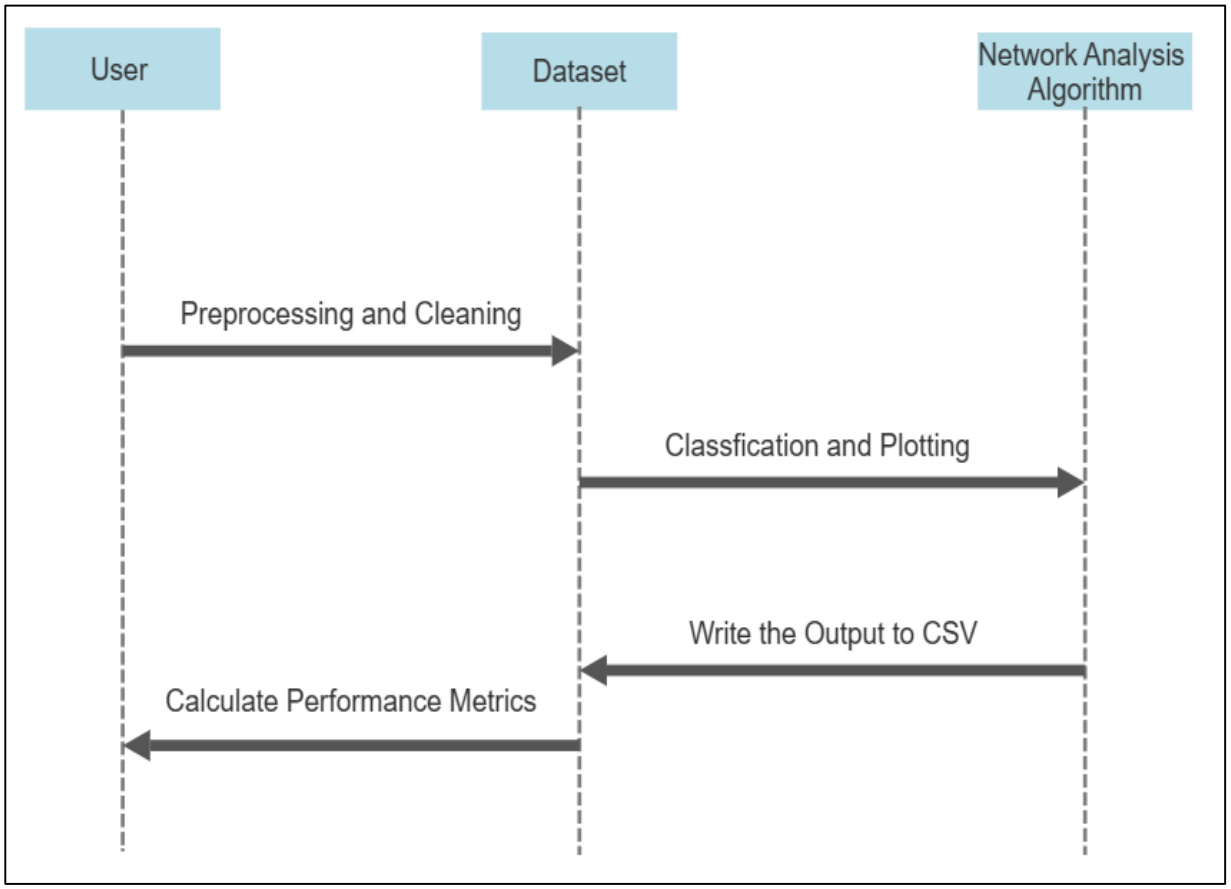

Fig 3.3 Sequence diagram of Duplicate detection

The various actions that take place in the application in the correct sequence areshown in Fig 3.3 Sequence diagrams are the most popular UML for dynamic modeling. The sequence diagram flow in the suggested system begins with dataset collection, cleaning and exploration, training with supervised machine learning algorithms, and testing with user input.

3.1.4 Activity Diagram of Duplicate Detection

Activity diagram is suitable for modeling the activity flow of the system. Activity diagrams are not only used for visualizing dynamic nature of a system

but they are also used to construct the executable system by using forward and reverse engineering techniques.

Fig 3.4 shows that activity is a particular operation of the system.

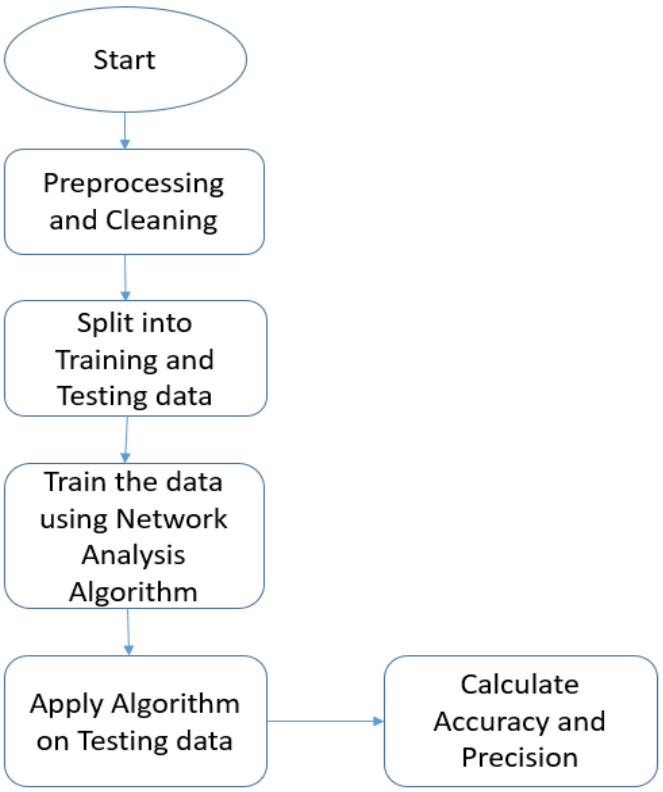

Fig 3.4 Activity Diagram of Duplicate Detection

The only missing thing in activity diagram is the message part. An application can have multiple systems. In the proposed system, activity diagram flow starts from collecting datasets, cleaning and exploration, and training using supervised machine learning algorithm and testing using the user input. The performance metrics such as sensitivity, specificity, F1 score and cohen's kappa are calculated.

Activity diagram is suitable for modeling the activity flow of the system. It does not show any message flow from one activity to another. Activity diagram is sometime considered as the flow chart.

CHAPTER 4

SYSTEM ARCHITECTURE

In this chapter, the System Architecture for the Duplicate Deduction in Availing Export Incentives is represented and the modules are explained.

4.1 SYSTEM ARCHITECTURE DIAGRAM

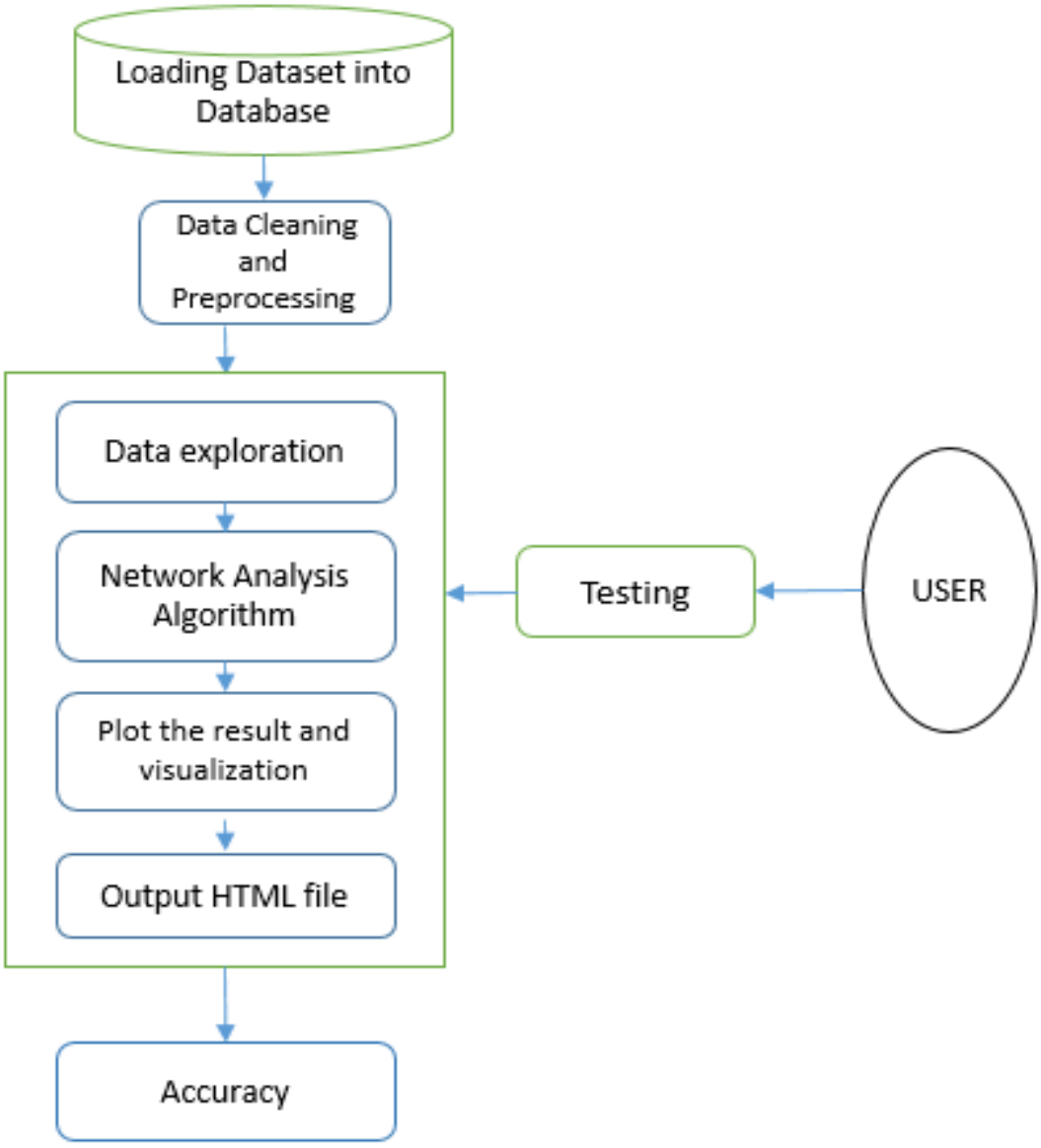

Fig 4.1 System Architecture Diagram

4.2 ARCHITECTURE DESCRIPTION

System architecture of this proposed system provides the detailed description aboutthe datasets, pre-processing of data, extraction of data, training of data using supervised machine learning algorithm and classification of user news as duplicate orreal. The datasets we used in this study are obtained from Government resources and are not available to the general public. The data includes both duplicate and real registrations from Foreign Trade domain. The dataset contains a total of 4738 rows, out of which in order to perform this classification, we need the R Studio, R 4.2.2 and Google Colab. The various libraries needed are downloaded as per requirements. An adjacency matrix is a square matrix in which the column and row names are the nodes of the network. Within the matrix a 1 indicates that there is a connection between the nodes, and a 0 indicates no connection.

Adjacency matrices implement a very different data structure than data frames and do not fit within the tidyverse workflow. The data analysis techniques of the tidyverse is sued to create edge lists, which will then be converted to the specific object classes for network, igraph, and tidygraph. An edge list is a data frame that contains a minimum of two columns, one column of nodes that are the source of a connection and another column of nodes that are the target of the connection. The nodes in the data are identified by unique IDs. If the distinction between source and target is meaningful, the network is directed. If the distinction is not meaningful, the network is undirected. Edge lists contain all of the information necessary to create network objects. The accuracy is calculated using Google Colab. The performance metrics calculated are Accuracy, Recall, F1 Score, Sensitivity, Cohen’s Kappa and Specificity. The accuracy of the proposed system is 94.6% indicating the efficiency of the proposed system.

CHAPTER 5
SYSTEM IMPLEMENTATION

In this chapter, the System Implementation for Duplicate Deduction in Availing Export Incentives is explained in detail.

5.1 IMPLEMENTATION OF DUPLICATE DETECTION USING MACHINE LEARNING

The proposed system is implemented in R Studio and Google Colab. Here, the various functionalities required for the application are implemented by coding them in R language.

Dataset source – Government of Indian Database (Open Government Data)

Data- (IE Code, Mobile Number, PAN number, Email ID)

5.2 MODULES

5.2.1 Dataset

The dataset used in this research is an original dataset obtained from Government sources. The dataset consists of more than 17 lakh rows of data. There were many erroneous data values such as invalid phone numbers, non-existent PAN numbers, insufficient digits in IE Code, and invalid mail address.

All these data values were removed in Oracle database. They were preprocessed and cleaned, thoroughly, removing any adulterated data values. The final, clean dataset consisted of more than 10lakh rows of data, out of which only the top 5000 rows were selected for the analysis of this proposed system.

The data was then separated into training data and testing data by generating random number values for the dataset of 0 and 1. The 0s were considered for training dataset and the 1s were considered for testing dataset.

5.2.2 Libraries

The proposed system was carried out in R Studio. The libraries of R such as tidyverse, knitr, iGraph, networkD3, visNetwork, tidytext, ggtext, ggalt, ggthemes, ggpubr were used. These libraries are mainly used in network analysis and thus are used in plotting the network based on nodes. Knitr library allows us to download the output network graph in the format of an HTML file which will be available for perusal at any time.

Apart from this, Python libraries such as numpy, scikit, sklearn, pandas and matplotlib were used for plotting the confusion matrix and calculating the accuracy.

5.2.3 Data Cleaning and Preparation

The data of 4738 records is split into training data with 2369 records and testing data sets with 2369 records. The records were selected and split using random number generator. The data was then separated into training data and testing data by generating random number values for the dataset of 0 and 1. The 0s were considered for training dataset and the 1s were considered for testing dataset. This was carried out in Oracle Database. Initially there were 17 lakh rows of data out of which 10lakh(approx.) were valid data entries.

Only the valid data entries are considered. Out of these 10 lakh rows of data, only the first 4738 rows of data are considered in the proposed system.

Thus, the training and testing datasets were of equal size with 50% of the dataset each.

5.2.4 Data Exploration

Over a wide range of fields network analysis has become an increasingly popular tool to deal with the complexity of the interrelationships between data elements and attributes. The promise of network analysis is the placement of significance on the relationships between attributes, rather than seeing data items as isolated entities. The emphasis on complexity, along with the creation of a variety of algorithms to measure various aspects of networks, makes network analysis a central tool for data analytics especially in forensic accounting. This proposed system ties to apply networks implementation in R to locate interconnects between record sets.

5.2.5 Training

The data is trained on 2369 rows of data based on the four values of PAN number, IE Code, Mobile Number and E-mail address. This is applied to the classification algorithm of network analysis. The network analysis process classifies the data into nodes. These nodes contain the common data values of other nodes such as two registrations with the same mobile number or PAN number.

5.2.6 Testing

ML testing is more similar to traditional testing: you write and run tests checking the performance of the program. Applying the tests, you catch bugs in different components of the ML program. In this proposed system, 50% of the dataset is used for testing. They contain the parameters- IE Code, Telephone number, Mail Id and PAN number. The output of testing is compared with the actual output.

5.2.7 Algorithm

The algorithm used in this proposed system is Network Analysis from Graph Theory.First we need to compute the edge betweenness of every edge in the graph.

Step 1: Start

Step 2: Select a node X, and perform BFS to find number of shortest path from the node X to each node, and assign the numbers as score to each node.

Step 3: Starting from the leaf nodes, we calculate the credit of edge by (1 + (sum of the edge credits to the node))*(score of destination node / score of starting node)

Step 4: Compute the edge credits of all edges in the graph G, and repeat from step 1. until all of the nodes are selected

Step 5: Sum up all of the edge credit we compute in step 2 and divide by 2, and the result is the edge betweenness of edges.

Step 6: Next, we remove the edges with the highest edge betweenness, and repeat until we find the good community split.

Step 7: Remove the edges with the highest edge betweenness

Step 8: Compute the modularity Q of the community's split

Step 9: Repeat from step 1, if Q is greater than 0.3–0.7.

Step 10: Stop

$$\mathbf{Q} \propto \sum \mathbf{s} \in \mathbf{S}\ [(\#\mathbf{edges\ within\ group\ s}) - (\mathbf{expected\ \#edges\ within\ group\ s})]$$

$$\mathbf{Q(G,S)} = \frac{\mathbf{1}}{\mathbf{2m}} \sum \mathbf{s} \in \mathbf{S} \sum \mathbf{i} \in \mathbf{s} \sum \mathbf{j} \in \mathbf{s} \left(\mathbf{A_{ij}} - \frac{\mathbf{k_i k_j}}{\mathbf{2m}} \right)$$

Normalizing Cost: $-1 < Q < 1$ $\quad\quad$ $A_{ij} = 1$ **if** $i \rightarrow j$; **0 else**

5.2.8 Applications of Graph Theory

Graphs are mathematical structures used to study pairwise relationships between objects and entities.

A graph is a structure with a set of objects, these objects can be related to other objects in pairs. The objects correspond to mathematical concepts called vertices, and the connected or related pairs of vertices are called edges.

A graph can also be thought of as: A binary relation (=edges) between elements of a set (=vertices).

Vertices are also often called nodes or points.

Edges can be called links or lines.

Graphs use the terms vertex and edge commonly, while networks usually use node and link.

When appropriate, a direction may be assigned to each edge to produce what is known as a directed graph, or digraph.

An important number associated with each vertex is its degree, which is defined as the number of edges that enter or exit from it. Thus, a loop contributes 2 to the degree of its vertex.

Examples:

Nodes = vertices (ex: vertices = {A, B, C, D, E})

Edges = links (ex: edges = {AB}, {AC}, {B, C}, {C, E})

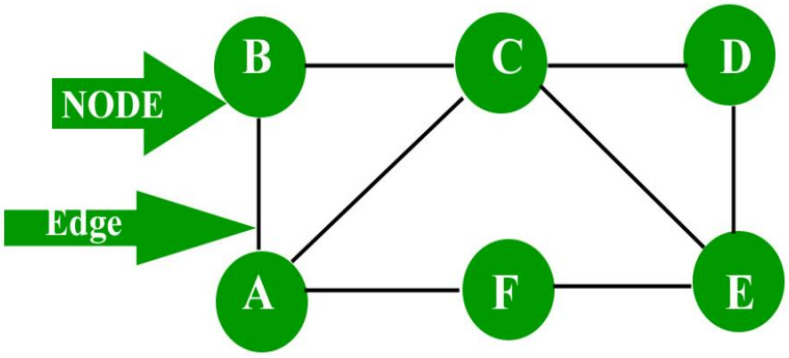

Fig 5.1 Graph – Nodes and Edges

5.2.9 Graph Theory

i. Graphs provide a better way of dealing with abstract concepts like relationships and interactions. They also offer an intuitively visual way of thinking about these concepts. Graphs also form a natural basis for analyzing relationships in a Social context

ii. Graph Databases have become common computational tools and alternatives to SQL (structured query language) and NoSQL databases

iii. Graphs are used to model analytics workflows in the form of DAGs (Directed acyclic graphs)

iv. Some Neural Network Frameworks also use DAGs to model the various operations in different layers

v. Graph Theory concepts are used to study and model Social Networks, Fraud patterns, Power consumption patterns, Virality and Influence in Social Media. Social Network Analysis (SNA) is probably the best known application of Graph Theory for Data Science

vi. It is used in Clustering algorithms – Specifically K-Means

vii. System Dynamics also uses some Graph Theory concepts – Specifically loops

viii. Path Optimization is a subset of the Optimization problem that also uses Graph concepts

ix. From a Computer Science perspective – Graphs offer computational efficiency. The Big O complexity for some algorithms is better for data arranged in the form of Graphs (compared to tabular data)

5.2.10 Key Terms and Concepts

Edges = lines or links, edges can have directions, signs, weights, functional equations, or locations.

Vertices = nodes or points, can have labels, weights, or locations.

Directed Graph = a graph that is made up of a set of vertices connected by edges, where the edges have a direction associated with them

Undirected Graph = a graph with a set of vertices or nodes that are connected, where all the edges are bidirectional, sometimes called an undirected network

Average path length = average number of steps along the shortest paths for all possible pairs of network nodes, measure of efficiency of information or mass transport on a network

Breadth first search (BFS) = algorithm that traverses a graph data structure, starts on a node and explores all neighbor nodes before moving to nodes at the next depth level

Depth first search (DFS) = similar to BFS, starts at a node on a graph data structure, searches as far as possible along each branch before backtracking

BFS and DFS are algorithms to search for nodes in a graph

Centrality = identify the most important vertices (nodes) within a graph, importance needs to be defined prior

Degree centrality = simple centrality measure that counts how many neighbors a node has

Closeness centrality = measure of centrality in a network, calculated as the sum of the length of the shortest paths between the node and all other nodes in the graph

Betweenness centrality = measure of centrality based on shortest paths

Isolation = an isolated vertex in a graph with, if a graph has an isolated vertex then the graph is disconnected

Adjacency/Incident = vertices are called adjacent if they connected by an edge, two edges are called incident if they share a vertex

Network density = a measure of how many edges a graph has

Isomorphic graph/s = two graphs which contain the same number of graph vertices connected in the same way are said to be isomorphic

Bipartite graph = special kind of graph with the following properties:

It consists of two sets of vertices X and Y.

The vertices of set X join only with the vertices of set Y.

The vertices within the same set do not join.

The embedded graph is an example of a bipartite graph because:

The vertices of the graph can be decomposed into two sets.

The two sets are X = {A, C} and Y = {B, D}.

The vertices of set X join only with the vertices of set Y and vice-versa.

The vertices within the same set do not join.

Simple path vs Euler path

A path in a graph represents a way to get from an origin to a destination node by traversing the edges of the graph.

Simple path: a route around a graph that visits every vertex one is called a simple path.

Euler path: a route around a graph that visits every edge once is called an Euler path.

You can tell which graphs have a Euler path by counting how many vertices have an odd degree. The number of vertices of odd degree must be either zero or two, if not it is not a Euler path.

Two vertices are adjacent if they share a common edge (an edge joins the vertices).

The degree of a vertex is the total number of vertices that are adjacent to the vertex.

Before computers graph theory could become quite complex depending on the application.

Recently the use of graphs and graph theory have become increasingly popular due to advancements in computing power and applications in diverse fields.

5.2.11 Graph theory in R

Useful packages for graph theory in R Studio

i. iGraph - a collection of network analysis tools with the emphasis on efficiency, portability and ease of use

ii. Rgraphviz (requires graphiz) - provides plotting capabilities for R graph objects

iii. ggplot2 - one of the leading packages for creating graphs in R

iv. raster - Geographic Data Analysis and Modeling Reading, writing, manipulating, analyzing and modeling of gridded spatial data

v. reshape2 - easily transform data between wide and long formats

vi. ggraph (extension of ggplot) - extension of ggplot, but geared for use in graphs and networks

vii. tidygraph - can help manipulate data used in graph and network applications, and provide tidy interfaces for common graph algorithms

viii. networkD3 (interactive network graphs) - Creates D3 JavaScript network, tree, dendrogram, and Sankey graphs from R

ix. visNetwork (interactive network graphs) - used for network visualization

x. sp - classes and methods for spatial data

xi. dismo - species distribution modeling

xii. NetIndices - Estimating network indices, including trophic structure of food webs in R

First we need to download a package to be able to make and visualize graphs

igraph is on CRAN and the igraph package can be found at the official website.

5.2.12 Expressions in igraph

i. V(dataset) = returns all vertices

ii. V(dataset)[#, #:#] = shows vertices in the specified positions

iii. V(dataset)[degree(dataset) < #] = finds vertices satisfying the specified condition

iv. V(dataset)[nei('vertex name')] = shows vertex neighbors

v. V(dataset)['vertex name', 'vertex name'] = selects given vertices

vi. E(dataset) = returns all edges

vii. E(dataset)[vertexset %–% vertexset] = shows all edges between two vertex sets

5.2.13 Case Examples

• *Marketing Analytics* – Graphs can be used to figure out the most influential people in a Social Network. Advertisers and Marketers can estimate the biggest bang for the marketing buck by routing their message through the most influential people in a Social Network

• *Banking Transactions* – Graphs can be used to find unusual patterns helping in mitigating Fraudulent transactions. There have been examples where Terrorist

activity has been detected by analyzing the flow of money across interconnected Banking networks

• *Supply Chain* – Graphs help in identifying optimum routes for your delivery trucks and in identifying locations for warehouses and delivery centers

• *Pharma* – Pharma companies can optimize the routes of the salesman using Graph theory. This helps in cutting costs and reducing the travel time for salesman

• *Telecom* – Telecom companies typically use Graphs (Voronoi diagrams) to understand the quantity and location of Cell towers to ensure maximum coverage

• *Ecological flow networks* using graph theory to real seqeuntial chains in which energy passes from producers to consumers in complex food webs. -Allesina, S., Bodini, A., & Bondavalli, C. (2005). Ecological subsystems via graph theory: the role of strongly connected components.

• *Landscape connectivity* using focal-species analysis to demonstrate the utility of a mathematical graph as an ecological construct with respect to habitat connectivity. -Bunn, A. Landscape connectivity: a conservation application of graph theory, Journal of environmental management.

• *Network neuroscience* being used with graph theory methods to aid in the detection of network communities or modules, and the identification of central network elements that facilitate communication and signal transfer in brains.

5.2.14 Classification

In machine learning, classification refers to a predictive modelingproblem where a class label is predicted for a given example of input data.

Examples of classification problems include: Given an example, classifyif it is spam or not. After training and testing, our trained model will classify whether the user given news is real or fake. To check how well our model, we use some metrics to find the accuracy of our model. There are many types of classification metrics available in Scikit learn: Confusion Matrix, Accuracy Score, Precision, Recall, F1-Score.

Accuracy refers to the correctness of a single measurement. Accuracy is determined by comparing the measurement against the true or accepted value. An accurate measurement is close to the true value, like hitting the center of a bullseye.

Precision and recall are two terms used in machine learning. Precision is the amount of data that can be classified as a result of a prediction. Recall is the number of correct predictions made.

The F-score, also called the F1-score, is a measure of a model's accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into 'positive' or 'negative'.

Accuracy = True Positive +True Negative/True Positive+FP+FN+TN.

Precision=True Positive/True Positive + False Positive

Recall=True Positive/True Positive + False Negative

F1=2*Precision*Recall/Precision + Recall

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{FNR}$$

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{N}} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} = 1 - \mathrm{FPR}$$

CHAPTER 6
CODING AND SCREENSHOTS

6.1 SAMPLE CODE

```
dgft <- fread('IEC_DATA.csv')
names(dgft)
dgft <- as.data.frame(dgft)
dgft <- dgft[, c(1,2,3,4)]
dgft %>%
mutate(TELEPHONE = ifelse(nchar(str_trim(TELEPHONE)) >= 8,
str_trim(TELEPHONE), NA),
EMAIL = ifelse(str_detect(str_trim(EMAIL), '@'), str_trim(EMAIL), NA),
PAN= ifelse(nchar(str_trim(PAN)) == 10, str_trim(PAN), NA) -> dgft_m
```

The data of 4,738 records is split into training data with 2,369 records and testing data sets with 2369 records. The records were selected and split using random number generator.

Load libraries: Then load the training data set with 2369 records for analysis.

```
library(tidyverse)
library(knitr)
library(igraph)
library(networkD3)
library(visNetwork)
library(tidytext)
library(ggtext)
library(ggalt)
library(ggthemes)
library(ggpubr)
```

LOADING DATA AND ASSIGN TO GRAPH NODE:

```
IEC_data <- read.csv('IEC_DATA_TRG.txt')
knitr::kable(IEC_data)
#Change in long format
IEC_data %>%
  mutate(across(everything(), as.character)) %>%
  pivot_longer(!IE_code, names_to = 'Attribute',values_to = 'Attrib_value') %>%
  relocate(Attribute, .after = 3) -> long_data
# graph object
g <- graph_from_data_frame(long_data)
# Long format all columns
long_data2 <- IEC_data %>%
  mutate(across(everything(), as.character)) %>%
  pivot_longer(everything(), names_to = 'Attribute', values_to = 'Attrib_value') %>%
  distinct()
# Re-order using joins
V(g)$att_type <- V(g)$name %>%
  as.data.frame() %>%
  set_names('name') %>%
  inner_join(long_data2, by = c("name" = "Attrib_value")) %>%
  pull(Attribute)
V(g)$color <- c('steel blue', 'orange')[1 + (V(g)$att_type == 'IE_code')]
V(g)$shape <- c("square", "circle")[(V(g)$att_type == 'IE_code')+1]
```

PLOT THE GRAPH OBJECT

```
#Plot the object
echo=FALSE
fig.align='center'
fig.width=7
fig.cap="Network Diagram of Duplicate Importers Exporters detected through network analysis"
visIgraph(g)
```

NETWORK OUTPUT CODE

```
#Network output-
#{r}
output <- g %>%
  components() %>%
  pluck(membership) %>%
  stack() %>%
  set_names(c('Group_id', 'IE_code')) %>%
  right_join(IEC_data %>%
               mutate(IE_code = as.factor(IE_code)),
             by = "IE_code")
knitr::kable(output)
```

The analysis of the training data set revealed that the were 88 duplicates. There were 52 data sets with 2 duplicate, 21 data set with 3 duplicates and so on.

Analysis Summary- Training DATA Set	
No of duplicate	**Count Of duplicate sets**
1	52
2	21
3	4
4	5
5	6
6	88

Table 6.1 Analysis Summary- Training Data Set

The analysis on the Test data set revealed 91 duplicates as detailed below:

Analysis summary – TEST DATA SET	
No of duplicates	**Count of duplicate sets**
1	52
2	24
3	4
4	5
5	6
6	91

Table 6.2 Analysis Summary- Testing Data Set

6.2 SAMPLE OUTPUT:

The visigraph visualization is as below:

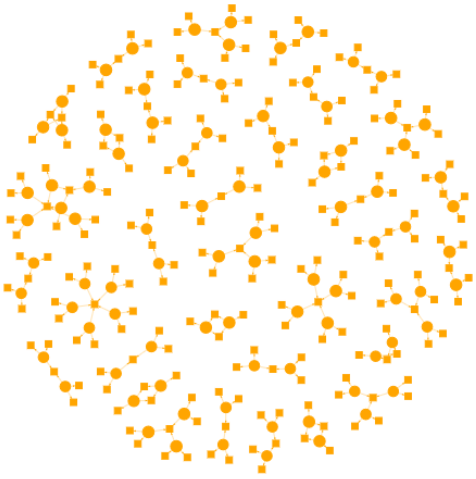

Fig 6.3 Visual Analysis of the training set

Zooming in on any particular group gives further details of interconnect

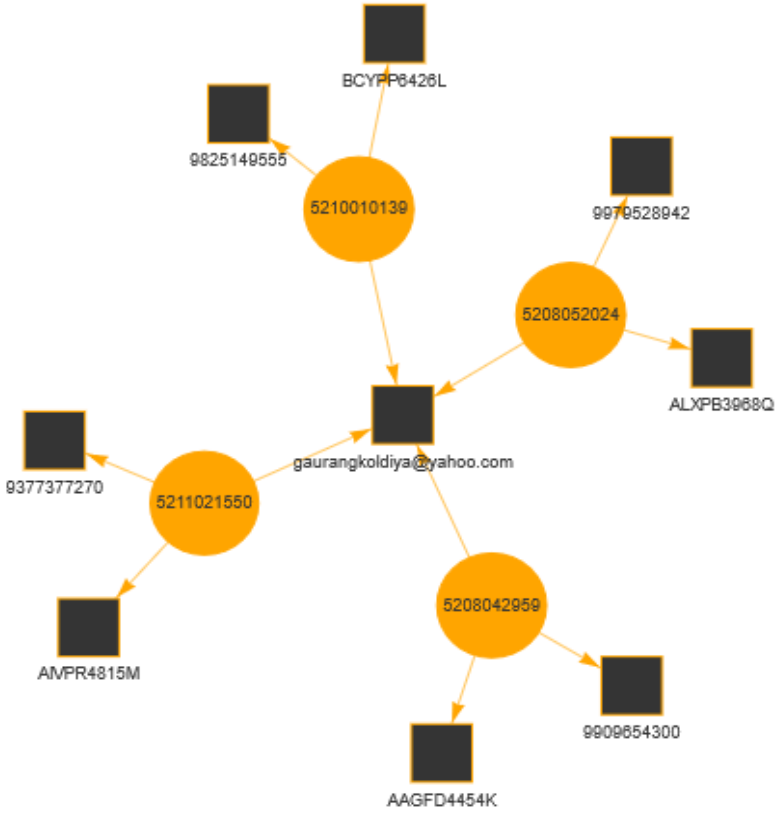

Fig 6.4 Zoomed image of Training Data Set

VISUAL NETWORK DISPLAYS WITH IGRAPH OF THE TEST DATA SET

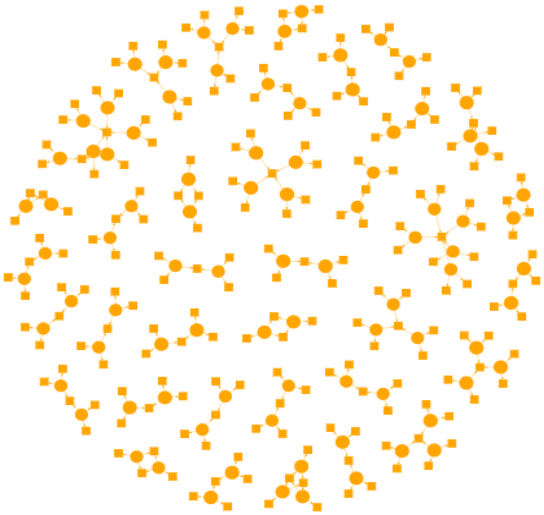

Fig 6.5 Visual Analysis of Testing Data Set

When we zoom on a particular interconnected set of nodes we see

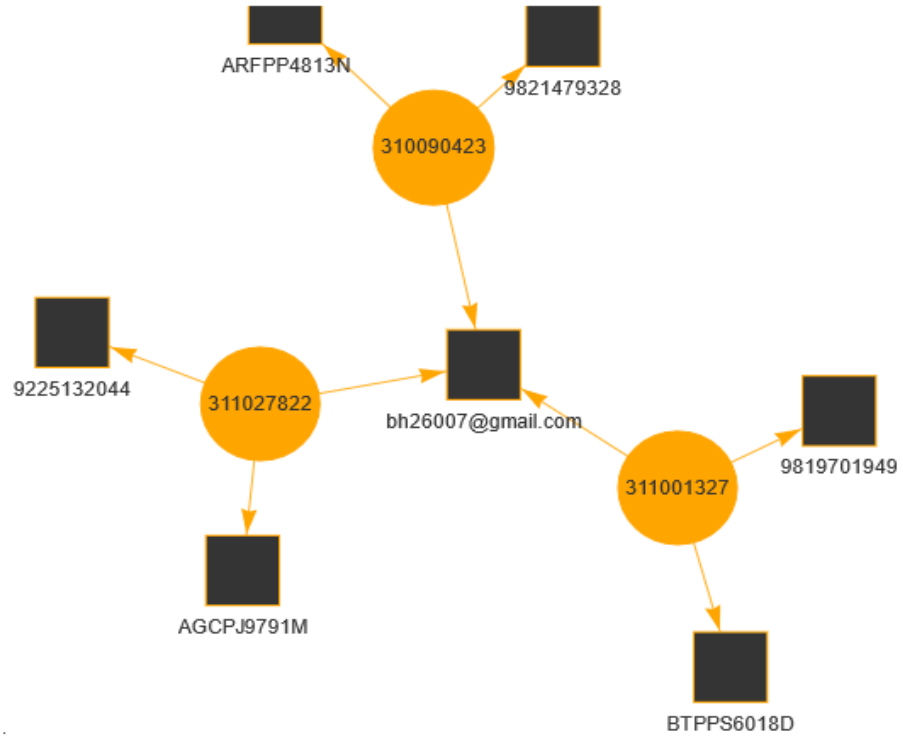

Fig 6.6 Zoomed image of Testing Data Set

6.3 PERFORMANCE ANALYSIS:

```
from sklearn.metrics import confusion_matrix
# create confusion matrix
confusion_matrix = pd.crosstab(data['Actual_outcome'], data['Analysis_outcome'],
rownames=['Actual'], colnames=['Predicted'])
# print confusion matrix
print(confusion_matrix)
print(type(confusion_matrix))
```

```
Predicted     0    1
Actual
0          2168  126
1             0   75
<class 'pandas.core.frame.DataFrame'>
```

Fig 6.7 Confusion Matrix

METRICS:

```
accuracy=(confusion_matrix[0][0]+confusion_matrix[1][1])/2369
print("Accuracy: ",accuracy)

#Sensitivity

sensitivity= confusion_matrix[0][0]/(confusion_matrix[0][0]+confusion_matrix[0][1])
print('Sensitivity : ', sensitivity)

#Specificity

specificity= confusion_matrix[1][1]/(confusion_matrix[1][0]+confusion_matrix[1][1])
print('Specificity : ', specificity)
```

```
Accuracy:  0.9468130012663571
Sensitivity :  1.0
Specificity :  0.373134328358209
```

Fig 6.8 Performance Metrics

```
from sklearn.metrics import cohen_kappa_score
print("Cohen-Kappa:
",cohen_kappa_score(data['Actual_outcome'],data['Analysis_outcome']))


from sklearn.metrics import f1_score
print("F1 Score: ",end=" ")
f1_score(data['Actual_outcome'],data['Analysis_outcome'],labels=None,
pos_label=1, average='binary', sample_weight=None, zero_division='warn')


print("Recall: ",end=" ")
sklearn.metrics.recall_score(data['Actual_outcome'],data['Analysis_outcome'],
labels=None, pos_label=1, average='binary', sample_weight=None,
zero_division='warn')


print("Precision Score: ",end=" ")
sklearn.metrics.precision_score(data['Actual_outcome'],data['Analysis_outcom
e'], labels=None, pos_label=1, average='binary', sample_weight=None,
zero_division='warn')
```

OUTPUT:

F1 Score: 0.5434782608695653
Recall: 1.0
Precision Score: 0.373134328358209

Cohen-Kappa: 0.521409537369286

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

This proposed system has attempted a small application in creating and plotting network type objects in R using the network, igraph, visigraph packages for static plots. visNetwork and networkD3 can be used for interactive plots. This proposed system demonstrates the effectiveness of R Programming Network analysis as forensic analysis tool. We have presented this information from the position of a non-specialist in network theory. We have only covered a very small percentage of the network analysis capabilities of R. There is a plethora of resources on network analysis in general and in R in particular. Network analysis is computation intensive. To demonstrative the application of network analysis connecting the nodes and detecting the duplicates claims, in this proposed system the network analysis was done with a sample data of 4,780 records from the population of 17.5 lakh records with only four variables were used for nodes.

7.2 FUTURE WORK

The network analysis on a larger dataset can be done with distributed computing system like Hadoop with Map Reduce. Also, we are dependent on RDBMS which only stores the structured data. To solve the problem of such huge complex data, Hadoop provides the best solution. In the proposed system, we have to check the group code manually. In future enhancements, we can change this by creating functionalities for printing the group code with the number of times of its occurrence.

REFERENCES

[1] Au SK, Beck JL. Estimation of small failure probabilities in high dimensions by subset simulation. Probab Eng Mech 2016; 16(4): 263–277.

[2] Billinton R, Wenyuan L. Hybrid approach for reliability evaluation of composite generation and transmission systems using Monte-Carlo simulation and enumeration technique. IEEE Proc C 2019; 138(3): 233–241.

[3] Bompard E, Napoli R, Xue F. Analysis of structural vulnerabilities in power transmission grids. Int J Crit Infrastruct Prot 2019; 2(1–2): 5–12.

[4] Chaturvedi SK. Network reliability: measures and evaluation. New Jersey: John Wiley Sons, 2016.

[5] Chen G, Dong ZY, Hill DJ, et al. Attack structural vulnerability of power grids: a hybrid approach based on complex networks. Physica A 2012; 389(3): 595–603.

[6] Chui KT, Shen CW. Tolerance analysis in scale-free social networks with varying degree exponents. Libr Hi Tech 2019; 37(1): 57–71.

[7] Crucitti P, Latora V, Marchiori M, et al. Efficiency of scale-free networks: error and attack tolerance. Physica A 2003; 320: 622–642. Langseth H, Portinale L. Bayesian networks in reliability. Reliab Eng Syst Saf 2017; 92(1): 92–108.

[8] He L, Zhang X. Fuzzy reliability analysis using cellular automata for network systems. Inform Sci 2016; 348: 322–336.

[9] Hou K, Jia H, Yu X, et al. An impact increments-based state enumeration reliability assessment approach and its application in transmission systems. In 2016 IEEE power and energy society general meeting (PESGM), Boston, MA, 2016 July 17, pp. 1–5. New York: IEEE.

[10] Lin YK, Huang CF. Stochastic flow network reliability with tolerable error rate. Qual Techno Quant M 2013; 10(1): 57–73.

[11] Liu H, Luo S, Gan J, et al. Research on node importance of power communication network based on multi-attribute analysis. In: 2020 IEEE 4th information technology, networking, electronic and automation control conference, Chongqing, China, 12 June 2020, vol. 1, pp. 2683–2687. New York: IEEE.

[12] Mahadevan S, Zhang R, Smith N. Bayesian networks for system reliability reassessment. Struct Saf 2001; 23(3): 231–251

[13] Miao F, Ghosn M. Modified subset simulation method for reliability analysis of structural systems. Struct Saf 2011; 33(4–5): 251–260.

[14] Moreno JA. Network reliability assessment using a cellular automata approach. Reliab Eng Syst Saf 2012; 78(3): 289–295.

[15] Murray L, Cancela H, Rubino G. A splitting algorithm for network reliability estimation. Trans. 2013; 45(2): 177–189.

[16] Ramirez-Marquez JE, Coit DW. A Monte-Carlo simulation approach for approximating multi-state two-terminal reliability. Reliab Eng Syst Saf 2015; 87(2): 253–264.

[17] Rebaiaia ML, Ait-Kadi D. A new technique for generating minimal cut sets in nontrivial network. Procedia, Elsevier 2013; 5: 67–76.

[18] Shahraki AF, Yadav OP, Liao H. A review on degradation modelling and its engineering applications. Int J Performability Eng 2017; 13(3): 299–314.

[19] Wang J, Zuo W, Rhode-Barbarigos L, et al. Literature review on modeling and simulation of energy infrastructures from a resilience perspective. Reliab Eng Syst Saf 2018; 183: 360–373.

[20] Yeh WC, Lin YC, Chung YY. Performance analysis of cellular automata Monte Carlo Simulation for estimating network reliability. Expert Syst Appl 2010; 37(5): 3537–3544.

[21] Yeh WC. An improved sum-of-disjoint-products technique for the symbolic network reliability analysis with known minimal paths. Reliab Eng Syst Saf 2017; 92(2): 260–268.

[22] Younes A, Girgis MR. A tool for computing computer network reliability. Int J Comput Math 2015; 82(12): 1455–1465.

[23] Zaheri MM, Ghanbari R, Afshar MH. A two-phase simulation–optimization cellular automata method for sewer network design optimization. Eng Optm 2020; 52(4): 620–636.

[24] Zhai Q, Peng R, Xing L, et al. Binary decision diagram-based reliability evaluation of k-out-of-(n+k) warm standby systems subject to fault-level coverage. *P I Mech Eng O-J Ris* 2013; 227(5): 540–548.

[25] Zuev KM, Wu S, Beck JL. General network reliability problem and its efficient solution by subset simulation. Probab Eng Mech 2015; 40: 25-35.