# Real-Time Shill Bidding Fraud Detection Empowered With Fussed Machine Learning

## A PROJECT REPORT

**Submitted in partial fulfilment of requirements to**

**CS 452-PROJECT II**

**By**

**Batch No: 15**

**T. SAI CHARANA GANESH (Y19CS167)**

**U. RAMA KRISHNA (Y19CS177)**

**Y. YESWANTH SAI (Y19CS192)**

**DECEMBER, 2022**

**R.V.R & J.C COLLEGE OF ENGINEERING (AUTONOMOUS)**

**(NAAC- ‘ A+ ’ Grade)**

**(Approved by AICTE, Affiliated to Acharya Nagarjuna University)**

**Chandramoulipuram :: Chowdavaram,**

**GUNTUR- 522 01**

# R.V.R. & J.C. COLLEGE OF ENGINEERING

## (Autonomous)

## DEPARTMENT OF COMPUTER SCIENCE& ENGINEERING

### CERTIFICATE

This is to certify that this project work report titled **"Real-Time Shill Bidding Fraud Detection Empowered with Fussed Machine Learning "** is the work done by **T.Sai Charana Ganesh (Y19CS167), U.Rama Krishna (Y19CS177), and Y.Yeswanth Sai (Y19CS192),** under our supervision, and submitted in partial fulfillment of the requirements to CS452 – Project II, during the Academic Year **2022-2023.**

| | | |
|---|---|---|
| **Ms. Ch. V. Madhavi Lakshmi** | **Dr. R. Lakshmi Tulasi** | **Dr. M. Sreelatha** |
| **Project Work Guide** | **In-charge, Project Work** | **Professor & HOD, CSE** |

# ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without proper suggestions, guidance and environment. Combination of these three factors acts like backbone of our project "**Real Time Shill Bidding Empowered with Fussed Machine Learning**".

We are very glad to express our special thanks to **Ms. Ch. V. Madhavi Lakshmi**, Guide for the Project, who has inspired us to select this topic, and also for his valuable advice in preparing this project topic and support to carry out this project successfully.

We are very thankful to **Dr. R. Lakshmi Tulasi**, Project incharge who extended her encouragement and support to carry out this project successfully.

We express our sincere thanks to **Dr. M. Sreelatha**, Head of the Department of Computer Science and Engineering for her encouragement and support to carry out this project successfully.

We are very much thankful to **Dr. K. Ravindra**, principal of R.V.R& J.C COLLEGE OF ENGINEERING, Guntur for providing a supportive environment.

Finally, we submit our reserves thanks to lab staff in the Department of Computer Science and Engineering and to all our friends for their cooperation during the preparation.

**T. SAI CHARANA GANESH (Y19CS167)**
**U. RAMA KRISHNA (Y19CS177)**
**Y. YESWANTH SAI (Y19CS192)**

# ABSTRACT

Shill Bidding (SB) occurs when the fake bidders are introduced by the seller’s side to increase the final price. SB is a crime committed during the e-Auction, and it is pretty difficult to detect because of its normal bidding behavior. The bidder gets a lot of loss because he pays extra money, and the sellers benefit from shill bidding, so this article proposed a fusion base model.

This proposed model is split into two parts training and validation, into 70 and 30 percent. This model has been divided into three sub-modules; the first module, two machine learning algorithms named Support vector machine (SVM), and Artificial neural network (ANN) trained parallel on the same dataset and predicting the bidding fraud.

The prediction of these models becomes the input of the fuzzy-based fussed module, and fuzzy decide the actual output based on SVM and ANN predictions. On every bid, it predicts whether the fraud is committed or not. If the bidding behavior is normal, continue the bidding; otherwise, cancel the bid and block the user.

The prediction accuracy of the proposed fussed machine learning approach is 99.63%. Simulation results have shown that the proposed fussed machine learning approach gives more attractive results than state-of-the-art published methods.

# CONTENTS

## List of Tables

## List of Figures

# List of Abbreviations

| | | |
|---|---|---|
| SB | - | Shill Bidding |
| SVM | - | Support vector machine |
| ANN | - | Artificial Neural Network |
| MLP | - | Multi Layer Perceptron |
| DFM-SB | - | detect fraud in Shill Bidding |
| LC | - | Learning Criteria |
| PPV | - | Positive Predictive Value |
| NPV | - | Negative Positive Value |
| FPR | - | False Positive Rate |
| FDR | - | False Discovery Rate |
| FNR | - | False Negative Rate |

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

Virtual Marketplace hosted on the internet is known as the E-auction. It is the process of buying and selling items through online platforms. The bidder bids the item, and the highest bidder is the winner of the item. At the beginning of the auction, bidding starts from the lowest price to a higher price depending upon the buyer's interest.

The history of an auction is found in about 500 B.C when the women and the slaves were sold. In those ages, it was legal by law. In the United States, the auction was started to sell the estates, farms, and slaves, with the growth of technology, the auction was started from the computers, fax, smartphone, and many online platforms, e.g., eBay is the first online auction website started in 1995 in the United States. It is the largest auction site whose net value recorded in 2017 is 1.7 billion US dollars. As in auctions, there is the involvement of money, so it attracts some malicious persons. Shill Bidding (SB) is a prevalent method for auction fraud. In SB, bidding item prices are increased by fake bids. As these are real-time bids, so it's difficult to detect because of their normal resemblance behavior. These moneymakers used different types of SB techniques like as

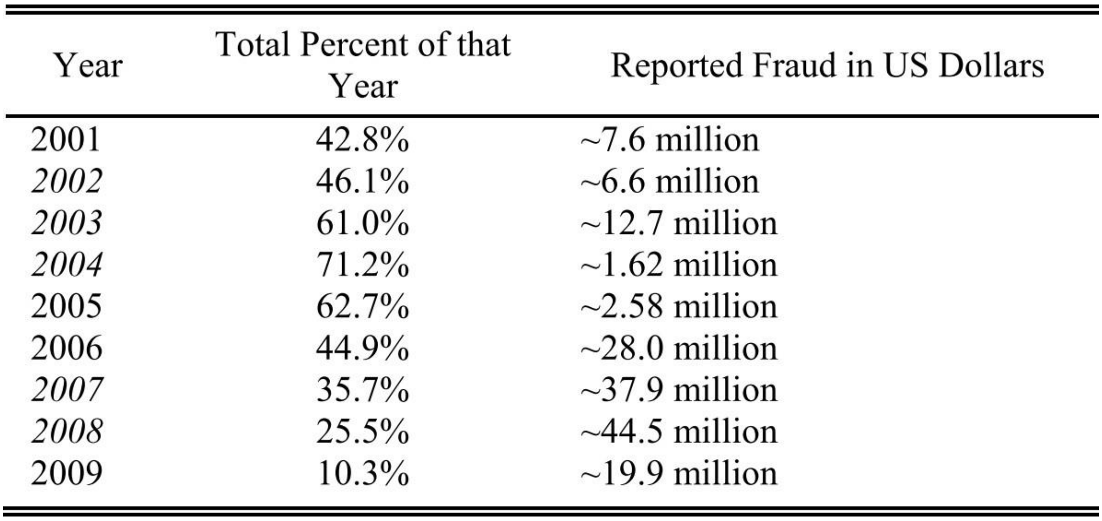
1) Pre-bidding

2) Post bidding

3) In bidding

SB is the cybercrime, and according to the Internet Crime Report (2001-2009), it is at the top of the list which people report through complaints recorded in IFCC.

Tab. 1. describes the online auction fraud reported from 2001-2009. According to IFCC reports, the 2001 to 2009 online auction fraud period was at the top of the list. For example, in 2001, total online fraud was reported as $17.8 million while 42.8% was online auction fraud, i.e., approximately $7.6 million. The next year 1 January to 31 December 2002, the total online auction fraud reported is $6.6 million, and further fraud is described in Tab.1.



| Year | Total Percent of that Year | Reported Fraud in US Dollars |
|---|---|---|
| 2001 | 42.8% | ~7.6 million |
| *2002* | 46.1% | ~6.6 million |
| *2003* | 61.0% | ~12.7 million |
| *2004* | 71.2% | ~1.62 million |
| 2005 | 62.7% | ~2.58 million |
| 2006 | 44.9% | ~28.0 million |
| *2007* | 35.7% | ~37.9 million |
| *2008* | 25.5% | ~44.5 million |
| 2009 | 10.3% | ~19.9 million |

**Table 1 :** Internet crime report (2001-2009).

## 1.2 Problem Statement

Shill Bidding occur when fake bidders make bids and increase prices. It is a crime in e-auction. Shill bidders are introduced by sellers to increase their money. Due to this behaviour buyers are losing their money and paying more price to the item and seller get more profit. The behaviour of shill bidders also same as real bidders. It is difficult to predict shill bidder.

This article is proposed a fusion base model to predict the shill bidder. The dataset is divided into 70 percent for training and 30 percent for testing. This model is divided into 3 modules; first module has two machine learning algorithms named Support vector machine (SVM), and Artificial neural network (ANN) trained parallel on the same dataset and predicting the shill bidder. Second module is the prediction of first module becomes the input of the fuzzy-based fussed module Fuzzy model decides whether the bidder is fraud or not. On every bid, If the bidder is fraud, cancel the bid and block the user otherwise, continue the bidding.

## 1.3 Objectives

To detect the fraud in real time e-auction the proposed method is fusion-based model.
Main objectives to obtain these are summarized as follows

1. To develop a project that uses the combination of two or more machine learning algorithms.
2. To get better accuracy, by using the machine learning algorithms SVM and ANN along with Fuzzy model.
3. Train the project with different types of data better results in different conditions.

## 1.4 Significance of the work

The main aim of the paper is to detect the fraud in real time e-auction such that the loss is reduced. Since, there is an involvement of money it attracts many people. There are many techniques followed by many researchers to reduce these loss due to the presence of shill bidders. There is necessarily to detect the shill bidder.

There is a need to develop a project which helps to detect fraud in real time e-auction. So, the fusion base model is proposed, which efficiently produce an accuracy of 99.63.

# CHAPTER 2

## LITERATURE REVIEW

**Anowar and Sadaoui [4]** detect auction fraud in commercial sites; their model has divided into two parts: offline classification and the other part is online classification. In offline classification, scarp data is collected and then preprocess the auction data. They are using Pattern measurement methods on preprocessing data. After the SB pattern measurement, SB is labeling based on data clustering. Some of the data is imbalanced during this clustering, so to handle the imbalanced data classification optimization tool is used. In the online part of their model, real-time data is collected from the site and preprocessed. After preprocessing, pattern measurement is used and classifies the data. Based on classifying data, the model has decided the bidder's activity is suspicious or not; if yes, then the model verify the fraud detection. In this paper, the authors are using different models in which three models show the best accuracy that is SVM (98.1%), Random Forest (97.1%), and ANN with MLP classifier (97.5%).

**Ganguly and Sadaoui [6]** devised an online base SVM system for SB fraud detection. To fulfill their purpose, authors apply clustering and labeling techniques and solve the misbalancing learning issues. Once bidding is done, data is collected and applied to the model, and the fraud activity is decided more accurately. The authors create an automatic system because of time handling issues, and the accuracy they achieve is 77.8%.

**Alzahrani and Sadaoui [7]** proposed the algorithm to optimize the dataset. The author used labeling and clustering techniques to optimize the imbalanced data and use the Hoeffding Tree algorithm on the overSampling and the oversampling algorithm. Their proposed algorithm's overall performance is good, which is 99.7%, 94% under-sampling, and over-sampling, respectively.

**Alzahrani and Sadaoui [1]** proposed the model in which data is collected through an online eBay site from iPhone 7 device. Collected data is the raw data, so data is preprocessed by using pattern measurement based on matrices. This highquality data is split date-wise, then data is divided into two parts, i.e., training and validation. The authors used 80% of data in training, and the remaining 20% of data used invalidation.

**Anowar et al. [8]** used hierarchy clustering techniques to split the same type of behavior of data, then applied a semi-automated approach to labeling the normal and suspicious data. In this paper, the authors use three oversampling sampling methods, under-sampling, and hybrid sampling. SVM is used to compare these methods' performance using the 5-K fold and 10-K fold. The best accuracy achieved by this research is 94.0%.

**Elshaar and Sadaoui [9]** focus the problem on multidimensional training data. For this purpose, the authors are using the SSC approach. SSC approach helps in fraud detection with the small amount of data, and skewed class distribution is used with the hierarchical clustering approach to detect anomalies in the dataset. In their statistical testing, the SSC model is separate from the regular and ambiguous bidders, and the overall achieved accuracy is 76%.

**Ganguly and Sadaoui [10]** focused on the dataset imbalance issue, and after preprocessing the data, the author implemented its dataset into three models that are Naïve Bayes, Neural network, and Decision tree. The author claims that Naïve Bayes is less sensitive than NN and Decision Tree in data quality. On the other hand, the decision tree is working better than other models on the rebalanced training dataset. The best accuracy achieved by the decision tree is 98%.

**Gupta and Mundra [11]** proposed a hybrid model, which is a combination of 2 methods one of them is the Prevention method (Authentication Phase) and the other method is the Detection method (Fraud Detection using HMM). The authors divided its model into 2 phases that are the training phase and the detection phase. In the training phase model, create the cluster, identify the bidding habit of bidders, choose the initial probability based on the bidder's habit, and construct a sequence of training data in the last step model. While in the detection phase, auctions are placed, models observe users' behaviors and generate the observation, then calculate the test sequences and decide if the behavior is normal or not. On abnormal behavior, models announce the winner or discard the bid. The problem in this model is that if the model found the abnormal behavior, then there is no method to decide the winner announcement or discard. Thus, there is an ambiguity to take the decision, which may fail the system.

**Elshaar and Sadaoui [12]** make two new patterns in the dataset. On these patterns, authors create a new high-quality dataset used in a semi-supervised machine learning-based model, which helps to label the multi-dimensional data. Afterward, the authors used oversampling and undersampling methods to use imbalanced class issues. The overall best accuracy achieved is 94% by the classifier named Yasti-J48.

**Dong et al. [13]** proposed an SVM-FDF model for detecting real-time fraud. They implement the spread prominence for a limited marketing scheme to update the credibility when an offer is applied, and fraud sampling is driven using the clustering algorithm. Finally, SVM is applied to each finding and specifies that the transaction is corrupted or fraud. The best accuracy achieved by the SVM-FDF model is 96.8%.

**Xiao et al. [14]** introduces the SSL group method for data handling and an ensemble learning technique to propose a GMDH based GCSSE model. This model involves two stages: first is the training of N base classifiers on the initial training set L with a class label. Then, in the second stage, construct a cost-sensitive GMDH neural network to achieve the selective ensemble classification output for the test set. This model is used on five datasets and gets the best accuracy, i.e., 93.20%.

**Limitations of the Existing System:**

Although there were many model came into existence there were many limitations while we take certain things in consideration.

1. Some models do not have good accuracy while measured.
2. Some models have good accuracy for only the given dataset.
3. Some models are good only for certain algorithms.
4. The dataset that all were working is imbalanced so the problem arises in the case of prediction .

# CHAPTER-3
# SYSTEM ANALYSIS

## SYSTEM REQUIREMENTS

### MINIMUM HARDWARE REQUIREMENTS:

- System: Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Monitor : 15 inch VGA Color.
- Mouse : Logitech Mouse.
- Ram : 512 MB
- Keyboard : Standard Keyboard

### MINIMUM SOFTWARE REQUIREMENTS:

- Operating System : Windows XP.
- Platform : PYTHON TECHNOLOGY
- Tool : Python 3.6

### Functional Requirements

A Functional Requirement is a description of the service that the software must offer. A function may be input to the system, it’s behaviour and outputs. It can be any functionality which defines what function a system is likely to perform.Functional requirements are also called as Functional Specifications.

1. The System must be able load the dataset from UPI repository.
2. The System must be able to read the dataset.
3. The System must support various machine learning algorithms like SVM,ANNetc.
4. The System must be able to predict precise results.
5. The System must be able to display Graphs and output of the project code.

### Non-functional requirement

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. **Non-functional requirements** add tremendous value to business analysis. It is commonly misunderstood by a lot of people. It is important for business stakeholders, and Clients to clearly explain the requirements and their expectations in measurable terms. If the non-functional requirements are not measurable then they should be revised or rewritten to gain better clarity. For example, User stories help in mitigating the gap between developers and the user community in Agile Methodology.

**Usability:**

Prioritize the important functions of the system based on usage patterns.

**Frequently used functions should be tested for usability**, as should complex and critical functions. Be sure to create a requirement for this.

**Reliability:**

Reliability defines the trust in the system that is developed after using it for a period of time. It defines the like ability of the software to work without failure for a given time period.

The number of bugs in the code, hardware failures, and problems can reduce the reliability of the software.

Your goal should be a long MTBF (mean time between failures). It is defined as the average period of time the system runs before failing.

Create a requirement that data created in the system will be retained for a number of years without the data being changed by the system.

It's a good idea to also include requirements that make it easier to monitor system performance.

**Performance:**

What should system response times be, as measured from any point, under what circumstances?

Are there specific peak times when the load on the system will be unusually high?

Think of stress periods, for example, at the end of the month or in conjunction with payroll disbursement.

**Supportability:**

The system needs to be **cost-effective to maintain**.

Maintainability requirements may cover diverse levels of documentation, such as system documentation, as well as test documentation, e.g. which test cases and test plans will accompany the system.

## 3.1 UML Diagrams for the project work

UML is an acronym that stands for Unified Modeling Language. Simply put, UML is a modern approach to modeling and documenting software. In fact, it’s one of the most popular business process modeling techniques.

It is based on diagrammatic representations of software components. As the old proverb says: “a picture is worth a thousand words”. By using visual representa- tions, we are able to better understand possible flaws or errors in software or business processes.

The elements are like components which can be associated in different ways to make a complete UML picture, which is known as diagram. Thus, it is very important to understand the different diagrams to implement the knowledge in real- life systems.

Any complex system is best understood by making some kind of diagrams or pictures. These diagrams have a better impact on our understanding. If we look around, we will realize that the diagrams are not a new concept but it is used widely in different forms in different industries.

Mainly, UML has been used as a general-purpose modeling language in the field of software engineering. However, it has now found its way into the documentation of several business processes or workflows. For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficiency. Use cases are best discovered by examining the actors an defining what the actor will be able to do with the system. Since all the needs of a system typically cannot be covered in one use case, it is usual to have a collection of use cases. Together this use case collection specifies all the ways the system. An association provides a pathway for communication. The communication can be between use cases, actors, classes or interfaces. Associations are the most general of all relationships and consequentially the most semantically weak. If two objects are usually considered independently, the relationship is an association. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficiency. Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system. Since all the needs of a system typically cannot be covered in one use case, it is usual to have a collection of use cases.

The various UML diagrams are:

1. Usecase diagram
2. Activity diagram
3. Sequence diagram
4. Colloboration diagram
5. Object diagram
6. State chart diagram
7. Class diagram
8. Component diagram
9. Deployment diagram

These diagrams provide the relationship between the members of the classes, objects and also the actors and the usecases.They are used to find the functional and non functional reqirements and to make relations between them.

### Usecase Diagram

A use case diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication (participation) associations between the actors and users and generalization among use cases. The use case model defines the outside (actors) and inside (use case) of the system’s behavior. Actors are not part of the system. Actors represent anyone or anything that interacts with (input to or receive output from) the system.Use-case diagrams can be used during analysis to capture the system requirements and to understand how the system should work. During the design phase, you can use use- case diagrams to specify the behavior of the system as implemented.Use case is a sequence of transactions performed by a system that yields a measurable result of values for a particular actor. The use cases are all the ways the system may be used.

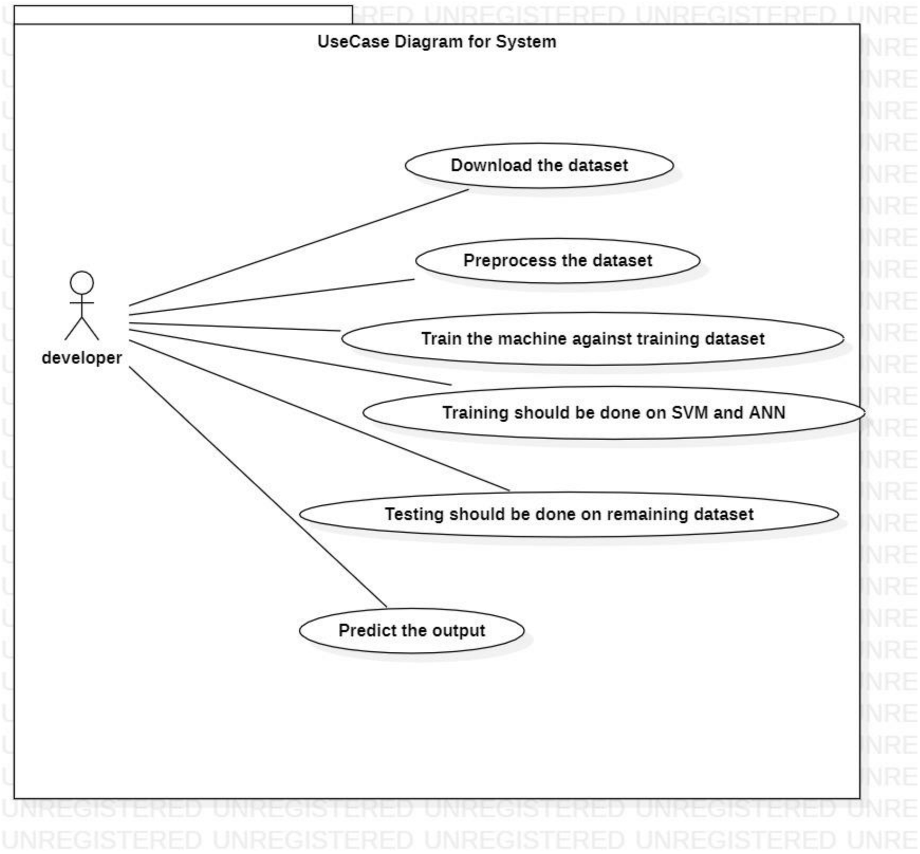

UseCase Diagram for System
developer
Download the dataset
Preprocess the dataset
Train the machine against training dataset
Training should be done on SVM and ANN
Testing should be done on remaining dataset
Predict the output

**Figure 3.1:Usecase Diagram for the System**

## Activity Diagram

An Activity diagram is a variation of a special case of a state machine, in which the states are activities representing the performance of operations and the transitions are triggered by the completion of the operations. The purpose of Activity diagram is to provide a view of flows and what is going on inside a use case or among several classes.Activity diagrams contain activities, transitions between the activities, decision points, and synchronization bars. An activity represents the performance of some behavior in the workflow. In the UML, activities are represented as rectangles with rounded edges, transitions are drawn as directed arrows, decision points are shown as diamonds, and synchronization bars are drawn as thick horizontal or vertical bars as shown in the following. The activity icon appears as a rectangle with rounded ends with a name and a component for actions.

Swim lanes may be used to partition an activity diagram. This typically is done to show what person or organization is responsible for the activities contained in the swim lane. Swim lanes are helpful when modeling a business workflow because they can represent organizational units or roles within a business model. Swim lanes are very similar to an object because they provide a way to tell who is performing a certain role. Swim lanes only appear on activity diagrams. When a swim lane is dragged onto an activity diagram, it becomes a swim lane view. Swim lanes appear as small icons in the browse while a swim lane views appear between the thin, vertical lines with a header that can be renamed and relocated. An activity represents the performance of some behavior in the work flow. In the UML, activities are represented as rectangles with rounded edges, transitions are drawn as directed arrows, decision points are shown as diamonds, and synchronization bars are drawn as thick horizontal or vertical bars as shown in the following. The activity icon appears as a rectangle with rounded ends with a name and a component for actions.

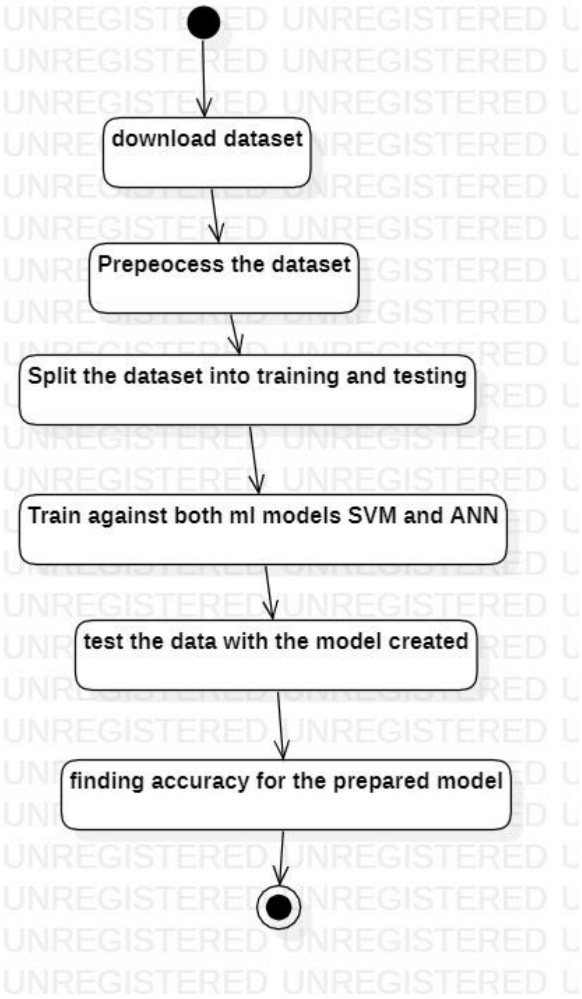

download dataset
Prepeocess the dataset
Split the dataset into training and testing
Train against both ml models SVM and ANN
test the data with the model created
finding accuracy for the prepared model

**Figure 3.2 : Activity Diagram**

**Sequence Diagram:**

A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called as event diagrams.

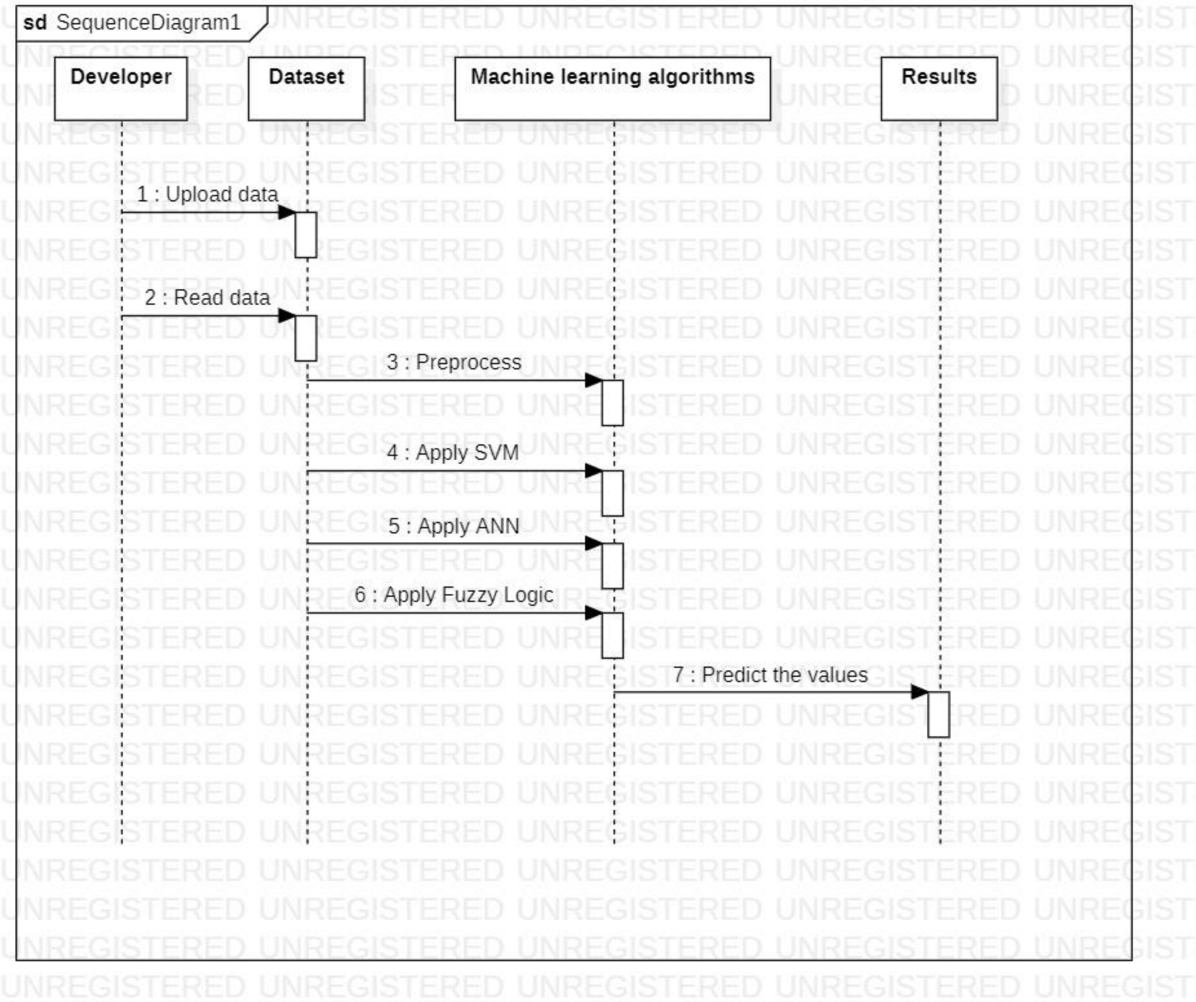


Figure 3.3 : Sequence Diagram

**Collaboration Diagram:**

A collaboration diagram shows that the order of messages that implement an operation or a transaction. Collaboration diagrams show objects, their links, and their messages. They can also contain simple class instances and class utility instances. Each collaboration diagram provides a view of the interactions or structural relationships that occur between objects and object like entities in the current model.Collaboration diagrams and sequence diagrams are called interaction diagrams. A collaboration diagram shows that the order of messages that implement an operation or a transaction. Collaboration diagrams show objects, their links, and their messages. They can also contain simple class instances and class utility instances. Each collaboration diagram provides a view of the interactions or structural relation- ships that occur between objects and object like entities in the current model.

The second interaction diagram is the collaboration diagram. It shows the object organization as seen in the following diagram. In the collaboration diagram, the method call sequence is indicated by some numbering technique. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. Method calls are similar to that of a sequence diagram. However, difference being the sequence diagram does not describe the object organization, whereas the collaboration diagram shows the object organization. To choose between these two diagrams, emphasis is placed on the type of requirement. If the time sequence is important, then the sequence diagram is used. If organization is required, then collaboration diagram is used. interaction diagrams are used to describe the dynamic nature of a system. Now, we will look into the practical scenarios where these diagrams are used. To understand the practical application, we need to understand the basic nature of sequence and collaboration diagram.

The main purpose of both the diagrams are similar as they are used to capture the dynamic behavior of a system. However, the specific purpose is more important to clarify and understand.

Sequence diagrams are used to capture the order of messages flowing from one object to another.

Collaboration diagrams are used to describe the structural organization of the structural organization of the objects taking part in the interaction. A single diagram is not sufficient to describe the dynamic aspect of the entire system,so as a set of diagrams are used to capture it as a whole

Interaction diagrams are used when we want to understand the message flow and the structural organization. Message flow means the sequence of control flow from one object to another. Structural organization means the visual organization of the elements in a system.

**Class Diagram:**

Class diagrams contain icons representing classes, interfaces, and their relationships. You can create one or more class diagrams to represent the classes at the top level of the current model; such class diagrams are themselves contained by the top level of the current model. You can also create one or more class diagrams to represent classes contained by each package in your model; such class diagrams are themselves contained by the package enclosing the classes they represent; the icons representing logical packages and classes in class diagrams.

1. Class diagrams are created to provide a picture or view of some or all of the classes in the model.
2. The main class diagram in the logical view of the model is typically a picture of the packages in the system. Each package also has its own main class diagram, which typically displays the public classes of the package.

A class diagram is a picture for describing generic descriptions of possible systems. Class diagrams and collaboration diagrams are alternate representations of object models.

A Class is a description of a group of objects with common properties (attributes) common behavior (operations), common relationships to other objects, and common semantics. Thus, a class is a template to create objects. Each object is an instance of some class objects cannot be instances of more than one class.In the UML, classes are represented as compartmentalized rectangles.

3.The top compartment contains the name of the class.

4.The middle compartment contains the structure of the class (attributes).

5.The bottom compartment contains the behavior of the class (operations).

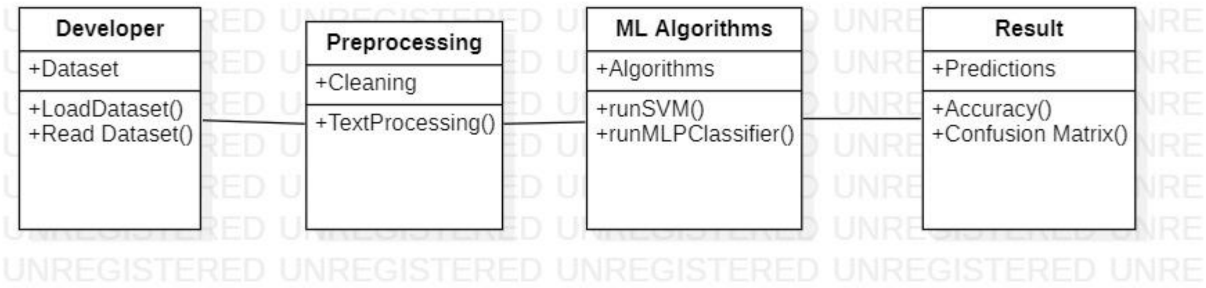


**Figure 3.4:Class Diagram for the System**

## State Transition

A state transition indicates that an object in the source state will perform certain specified actions and enter the destination state when a specified event occurs or when certain conditions are satisfied. A state transition is a relationship between two states, two activities, or between an activity and a state.

We can show one or more state transitions from a state as long as each transition is unique. Transitions originating from a state cannot have the same event, unless there are conditions on the event.

Provide a label for each state transition with the name of at least one event that causes the state transition. You do not have to use unique labels for state transitions because the same event can cause a transition to many different states or activities.

## State Chart Diagram

Use cases and scenarios provide a way to describe system behavior; in the form of interaction between objects in the system. Sometime it is necessary to consider inside behavior of an object.

A state chart diagram show s the states of a single objects, the events or messages that cause a transition from one state to another and the actions that result from a state change. As I activity diagram, state chart diagram also contains special symbols for start state and stop state.

State chart diagram cannot be created for every class in the system, it is only for those lass objects with significant behavior.

We can show one or more state transitions from a state as long as each transition is unique. Transitions originating from a state cannot have the same event, unless there are conditions on the event.

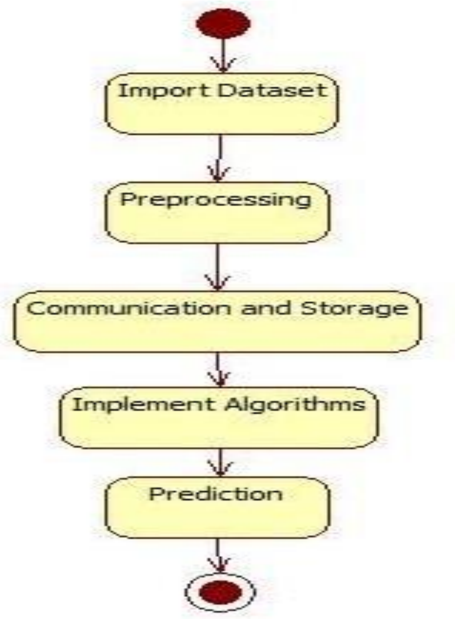


**Figure 3.5: State Chart diagram of a System**

## Detailed view diagrams

### Component Diagram:

Component Diagrams show the dependencies between software components in the system. The nature of these dependencies will depend on the language or languages used for the development and may exist at compile-time or at runtime.

In a large project there will be many files that make up the system. These files will have dependencies on one another. The nature of these dependencies will depend on the language or languages used for the development and may exist at compile-time, at link-time or at run-time. There are also dependencies between source code files and the executable files or byte code files that are derived from them by compilation.

Component diagrams are one of the two types of implementation diagram in UML. Component diagrams show these dependencies between software components in the system. Stereotypes can be used to show dependencies that are specific to particular languages also.

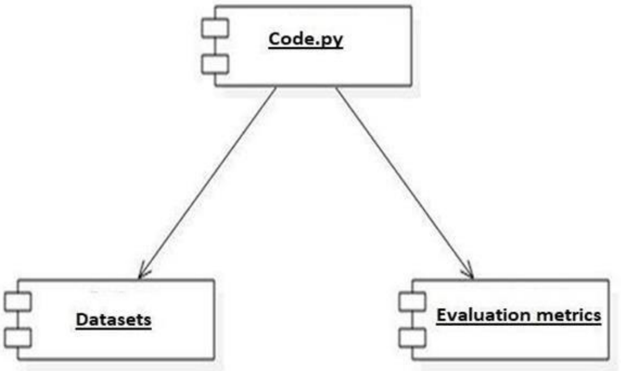


**Figure 3.6: Component Diagram for a System**

## Deployment Diagram

The second type of implementation diagram provided by UML is the deployment diagram. Deployment diagrams are used to show the configuration of run-time processing elements and the software components and processes that are located on them.

Deployment diagrams are made up of nodes and communication associations. Nodes are typically used to show computers and the communication associations show the network and protocols that are used to communicate between nodes. Nodes can be used to show other processing resources such as people or mechanical resources.

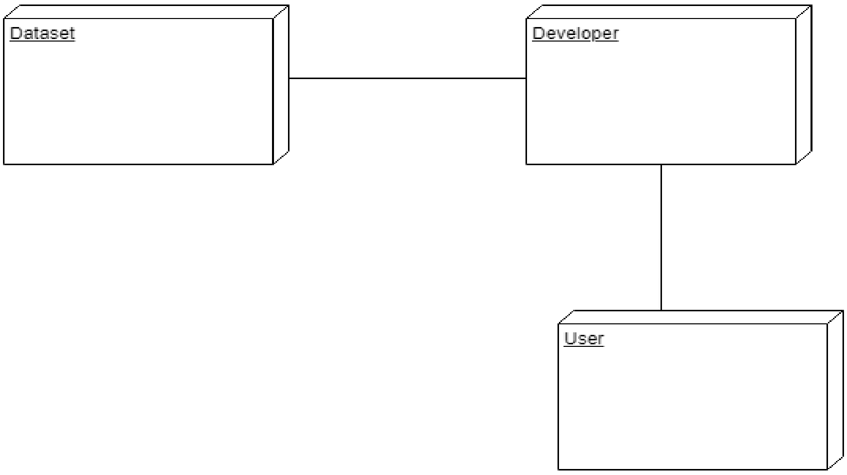


**Figure 3.7: Deployment Diagram for a System**

Nodes are drawn as 3D views of cubes or rectangular prisms, and the following figure shows a simplest deployment diagram where the nodes connected by communication associations .

# CHAPTER 4
# SYSTEM DESIGN

## 4.1 Architecture

This model takes data from commercial websites like e-bay. The preprocessing stage the scarp data was taken. Handling imbalancing data, Average imputation in missing data, clustering and labelling were done in preprocessing. The same data is sent into SVM model and ANN model in parallel. The output of those models were the input of fuzzy base model. This model is cloud based model. Whenever auction is going on , every bid is tested in this model and predict whether the bid is fraud or not. If the bidder is fraud, remove the bid and block the user otherwise, continue the bidding.

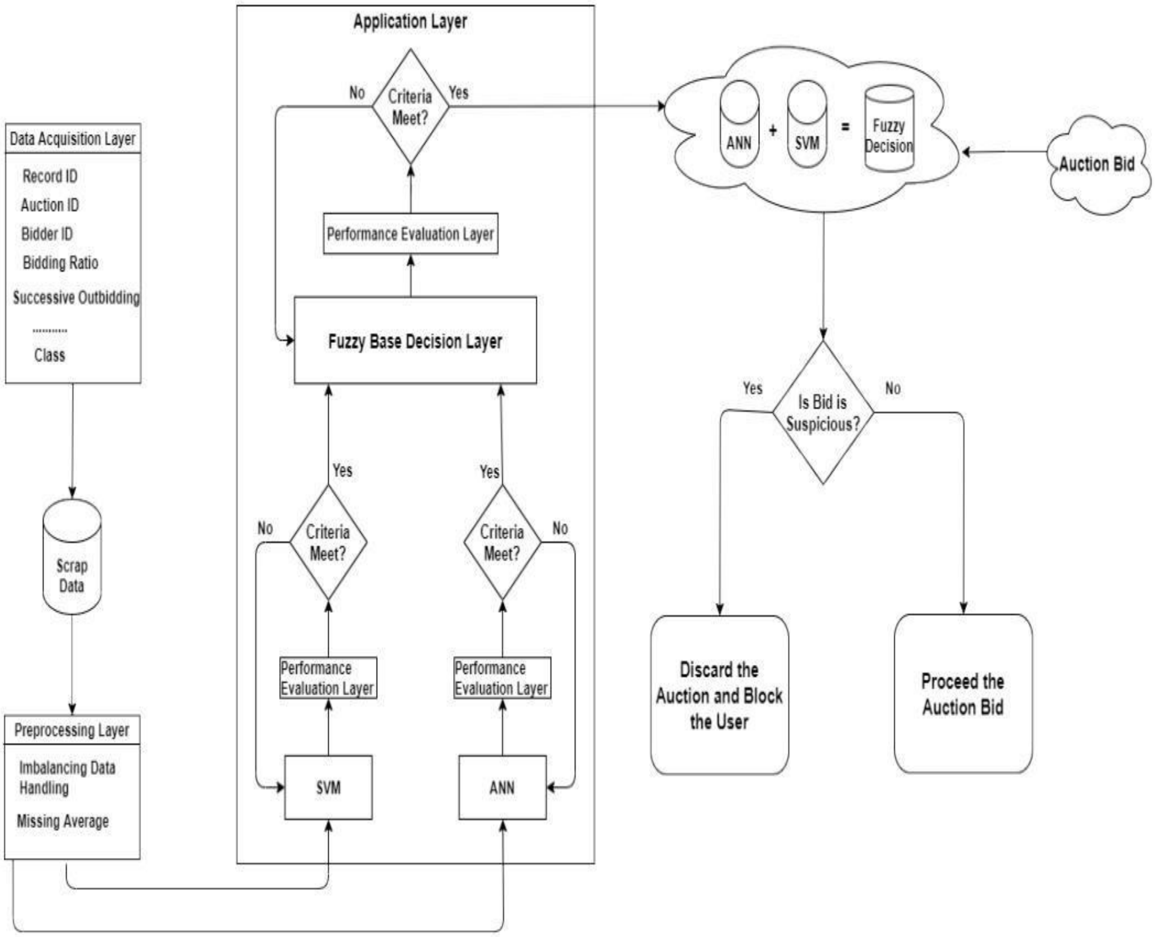


**Figure 4.1 :** Proposed DFM-SB model.

### 4.2 Proposed System

In this article, the proposed model is a real-time cloud and decision-based fusion model to detect fraud in Shill Bidding (DFM-SB). The proposed DFM-SB model is divided into 2 phases: the training phase and the detection phase on eBay auctions of popular brands. The collected data is in scrap form. First, data Labelling, Imbalance data handling, preprocesses data, and missing average. After preprocessing, this data will be used to train in the ANN and check the trained model achieved the Learning Criteria(LC). Then, on the parallel, train the SVM and check it achieved the LC or not. When both training algorithms met the LC, then decision level fusion is used with the help of fuzzy logic. If LC is not met, then retrain the model as described in Fig.3.1. Finally, fuzzy decides the actual output based on SVM and ANN results. This training model is stored in the cloud, and on every e-auction, this model will be used to detect fraud. If a user is found guilty, block the user and discard his auction; otherwise, proceed with the auction.

### WORKFLOW OF THE PROPOSED SYSTEM

The dataset which was downloaded from the UCI Repository should be preprocessed first .Later,The data was divided into the training and testing dataset which is of 70% and 30%.The data which was used for training should be trained parallely on the both machine learning algorithms SVM and ANN. Then the outputs obtained by the Machine learning algorithms are stored.Based on the Predictions of SVM and ANN the Fuzzy decides the output Whether the bidder is legitimate or not.By this workflow the above procedure happens.

The heapmap refers to correlation between of all the attributes. The correlation between early_bidding and last_bidding is almost equal to 1 which refers that these two attributes have strong positive correlation. Anyone of them are removed because one of these are enough to represent the data. Bidding ratio and successive outbiding are also greater than 0.5 which shows some common relation between them. Bidding ratio and winning ratio are also possess same relation. Auction bids and starting price average are also shows the similar data features with good correlation.

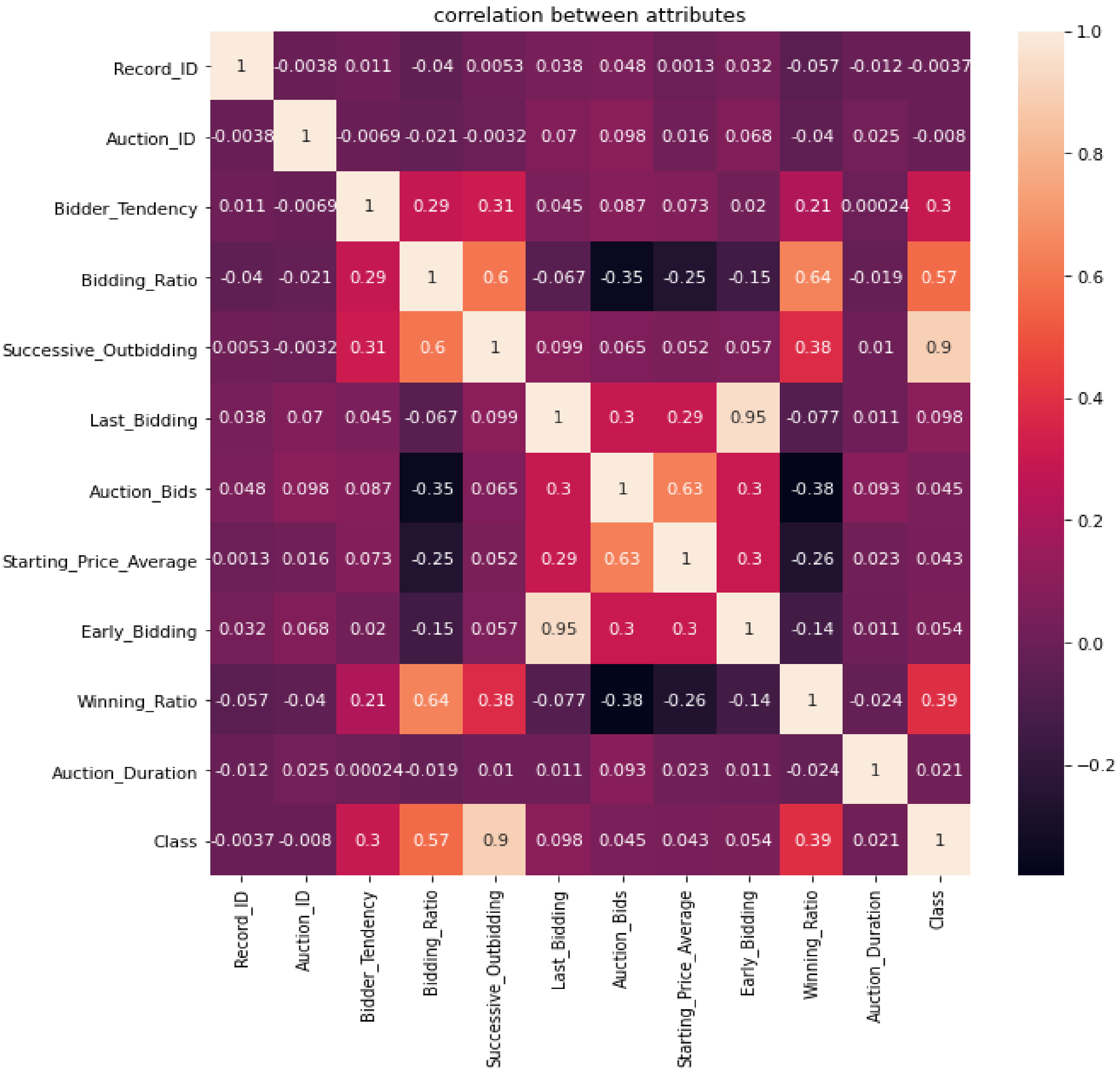


**Figure 4.2: Heatmap diagram of Attributes**

### 4.2.1 ARTIFICIAL NEURAL NETWORK

In ANN, preprocessed data is divided into two parts: training data and validation data. 70:30 of total data are used in the training and validating Phase. This data is running on the 15 hidden layers of neurons and trains the model. In ANN, there are 11 input neurons and one output neuron, which have two classes that are normal bidding or SB. Between input and output neurons, 15 hidden layers exist. The mathematical model of ANN model shell bidding is given below: Criteria are met, then proceed the model to the next step; otherwise, retrain the model. In the first layer (input layer), there are 11 input neurons represented as 1, 2, 3, . . . , 11. In the second layer (hidden layer), there are 15 neurons represented as ζ 1, ζ 2, ζ 3, . . . , ζ 15 and output is represented as ‘‘ out ’’ as describe in Fig. 1. The biases are represented as 1 and 2 respectively. To calculate the outð, net and out ’’, which can be calculated from the following Eq’s 1, 2, 3, and 4.

$$\text{net}\eth = \text{Б}1 \sum\nolimits_{\gamma=1}^{m} (u_{\gamma\eth} * \breve{\upsilon}) \tag{1}$$

$$\text{out}\eth = \frac{1}{1+e^{-net\eth}} \quad \text{where } \eth = 1, 2, \ldots, \text{n} \tag{2}$$

$$\text{net}\varrho = \text{Б}2 \sum\nolimits_{\eth=1}^{n} (\text{p}_{\eth\varrho} * \text{out}\eth) \tag{3}$$

$$\text{out}\varrho = \frac{1}{1+e^{-net\varrho}} \quad \text{where } \varrho = 1, 2, \ldots, \text{r} \tag{4}$$

Between input and output neurons, 15 hidden layers exist. The mathematical model of ANN model shell bidding is given below: Criteria are met, then proceed the model to the next step; otherwise, retrain the model.

The data was divided into the training and testing dataset which is of 70% and 30%.The data which was used for training should be trained parallely on the both machine learning algorithms SVM and ANN. Then the outputs obtained by the Machine learning algorithms are stored.Based on the Predictions of SVM and ANN the Fuzzy decides the output Whether the bidder is legitimate or not.

After preprocessing, this data will be used to train in the ANN and check the trained model achieved the Learning Criteria(LC). Then, on the parallel, train the SVM and check it achieved the LC or not. When both training algorithms met the LC, then decision level fusion is used with the help of fuzzy logic. If LC is not met, then retrain the model as described in Fig.3.1. Finally, fuzzy decides the actual output based on SVM and ANN results.

The total error ‘‘E’’ can be calculated by using Eq. 5.

$$E = \frac{1}{2}\sum\nolimits_{\varrho}\left(\tau_{\varrho} - out_{\varrho}\right)^{2} \qquad (5)$$

Weights need to be changed concerning errors that can be changed by using Eq. 6.

$$\Delta\omega \propto -\frac{\text{ɤ}E}{\text{ɤ}\omega} \qquad (6)$$

The weights between the hidden layer and the output layer are updating by using Eq. 7.

$$\Delta \text{ƿ}_{\tilde{\eth},\varrho} = -\varepsilon\frac{\text{ɤ}E}{\text{ɤ}V_{\tilde{\eth},\varrho}} \qquad (7)$$

As $V_{\tilde{\eth},\varrho}$ cannot be calculated directly so, calculated it using the Eq. 8 formula.

$$\Delta \text{ƿ}_{\tilde{\eth},\varrho} = -\varepsilon\frac{\text{ɤ}E}{\text{ɤ}out_{\varrho}} \times \frac{\text{ɤ}out_{\varrho}}{\text{ɤ}net_{\varrho}} \times \frac{\text{ɤ}net_{\varrho}}{\text{ɤ}\text{ƿ}_{\tilde{\eth},\varrho}} \qquad (8)$$

where $\tau_{\varrho}$ is the actual weight of describe in Eq. 9

$$\Delta \text{ƿ}_{\tilde{\eth},\varrho} = \varepsilon\left(\tau_{\varrho} - out_{\varrho}\right) \times out_{\varrho}\left(1 - out_{\varrho}\right)(out_{\tilde{\eth}}) \qquad (9)$$

Eq. 9 is simplified in Eq. 10.

$$\Delta \text{ƿ}_{\tilde{\eth},\varrho} = \varepsilon\, \text{ȝ}_{\varrho}\, out_{\tilde{\eth}} \qquad (10)$$

where value of is $\text{ȝ}_{\varrho}$ described in Eq. 11

$$\text{ȝ}_{\varrho} = \left(\tau_{\varrho} - out_{\varrho}\right) \times out_{\varrho}\left(1 - out_{\varrho}\right) \qquad (11)$$

Eq’s 12 to 16 are used to update the weights between hidden layer neurons and input layer neurons.

$$\Delta\mu_{i,\eth} \propto -\left(\sum_{\varrho} \frac{ɤE}{ɤout_{\varrho}} \times \frac{ɤout_{\varrho}}{ɤnet_{\varrho}} \times \frac{ɤnet_{\varrho}}{ɤout_{\eth}}\right) \times \frac{ɤout_{\eth}}{ɤnet_{\eth}} \times \frac{ɤnet_{\eth}}{ɤ\mu_{i,\eth}} \tag{12}$$

$$\Delta\mu_{i,\eth} = -\varepsilon\left(\sum_{\varrho} \frac{ɤE}{ɤout_{\varrho}} \times \frac{ɤout_{\varrho}}{ɤnet_{\varrho}} \times \frac{ɤnet_{\varrho}}{ɤout_{\eth}}\right) \times \frac{ɤout_{\eth}}{ɤnet_{\eth}} \times \frac{ɤnet_{\eth}}{ɤ\mu_{i,\eth}} \tag{13}$$

$$\Delta\mu_{i,\eth} = \varepsilon\left(\sum_{\varrho} \left(\tau_{\varrho} - out_{\varrho}\right) \times out_{\varrho} \left(1 - out_{\varrho}\right) \times \mathrm{p}_{i,\eth}\right) \times out_{\varrho} \left(1 - out_{\varrho}\right) \times \breve{\upsilon}_{i} \tag{14}$$

$$\Delta\mu_{i,\eth} = \varepsilon\left(\sum_{\varrho} \left(\tau_{\varrho} - out_{\varrho}\right) \times out_{\varrho} \left(1 - out_{\varrho}\right) \times \mathrm{p}_{i,\eth}\right) \times out_{\eth} (1 - out_{\eth}) \times \breve{\upsilon}_{i} \tag{15}$$

$$\Delta\mu_{i,\eth} = \varepsilon\left[\sum_{\varrho} \text{Ʒ}(\mathrm{p}_{\eth,\varrho})\right] \times out_{\eth} (1 - out_{\eth}) \times \breve{\upsilon}_{i} \tag{16}$$

Eq. 16 can be written in simplified form, as shown in Eq.17.

$$\Delta\mu_{i,\eth} = \varepsilon\, \text{Ʒ}_{\eth}\breve{\upsilon}_{i} \tag{17}$$

where value of ð, is described in Eq. 18

$$\text{Ʒ}_{\eth} = \sum_{\varrho} \text{Ʒ}(\mathrm{p}_{\eth,\varrho})] \times out_{\eth} (1 - out_{\eth}) \tag{18}$$

Weights updating formula describe in Eq. 19.

$$\Delta\mu_{i,\eth} = \varepsilon\, \text{Ʒ}_{i}\, \breve{\upsilon}_{i} + \lambda\Delta\mathrm{p}_{\eth,\varrho} \tag{19}$$

Updating weight and hidden layer can be written as in Eq. 20.

$$\Delta\mu_{i,\eth}(t+1) = \mu_{i,\eth}(t) + \lambda\Delta\mu_{i,\eth} \tag{20}$$

After the model is trained, save the training model and validate the model with 30% remaining dataset. Validating data is to enter data into the model and save its results. After saving the results validating data, the output is compared with the actual output, and it achieved 99% prediction accuracy.

### 4.2.2 SUPPORT VECTOR MACHINE

SVM is supervised machine learning and is used in the smaller dataset. The idea behind the SVM is to draw the hyperplane that separates it into different classes. SVM separates the Shill

Bidding and normal bidding. To separate the classes in a hyperplane, first, we draw the line. As the equation of a line is described in Eq. 21

$$\chi_2 = a\chi_1 + b \qquad (21)$$

where a is the slope of the line and b is the intersect point so that it can be written as

$$a\chi_1 - \chi_2 + b = 0$$

Let suppose $\bar{x} = (\chi 1, \chi 2)^{T}$ & $\bar{\omega} = (a - 1)$ then above equation can be written as Eq. 22.

$$\vec{\omega}.\bar{x} + b = 0 \qquad (22)$$

This equation is called the equation of hyperplane and is useful for multi-dimensional vectors. Eq. 23 describe the vector of $\bar{x} = (\chi 1, \chi 2)$ is written as $\bar{\omega}$.

$$\omega = \frac{x_1}{\|x\|} + \frac{x_2}{\|x\|} \qquad (23)$$

Where ||x|| is defined as

$$\|x\| = \sqrt{x_i^2 + x_2^2 + x_3^2 + \cdots + x_n^2}$$

As we know that the value of cos(ς) is

$$\cos(\varsigma) = \frac{x_1}{\|x\|}$$

And the value of $\cos(ϐ)$ is

$$\cos(ϐ) = \frac{x_2}{\|x\|}$$

Now, Eq. 23 can be written the value of ω as

$$\begin{aligned} \omega &= (\cos(\varsigma), \cos(ϐ)) \\ \vec{\omega} \cdot \vec{x} &= \|\omega\| \, \|x\| \cos(\varsigma) \qquad (24) \end{aligned}$$

As $\varsigma = \vartheta - ϐ,$ then

$$\begin{aligned} \cos(\varsigma) &= \cos(\vartheta) - \cos(ϐ) \\ \cos(\varsigma) &= \cos(\vartheta)\cos(ϐ) - \sin(\vartheta)\sin(ϐ) \end{aligned}$$

cos(ς) can also be written as

$$\cos(\varsigma) = \frac{\omega_1}{\|\omega\|}\frac{x_1}{\|x\|} + \frac{\omega_2}{\|\omega\|}\frac{x_2}{\|x\|}$$

By simplifying the above Eq.

$$\cos(\varsigma) = \frac{\omega_1 x_1 + \omega_2 x_2}{\|\omega\|\,\|x\|}$$

Put the value of cos(ς) is Eq. 24.

$$\vec{\omega} \cdot \vec{x} = \|\omega\|\,\|x\| \frac{\omega_1 x_1 + \omega_2 x_2}{\|\omega\|\,\|x\|}$$

As the above Eq. explain the two dimensions vector, for the n-dimensions vector, it can be written as shown in Eq. 25

$$\vec{\omega} \cdot \vec{x} = \sum\nolimits_{i=1}^{n} \omega_i x_i \quad \text{where i} = 1, 2, \ldots, \text{n} \qquad (25)$$

Eq. 25 is used to validate the correctly classifying the data

D = ÿ(w.x + b)

Given data is correctly classified if the value of D is greater than 0; if not, it is not correctly classified. For our SB data set, compute the dataset onto D for ith times which can be mathematically represented as

Di = ¨yi (ω.x + b)

d is called the functional margin of the dataset and is written as

$$d = \min_{i=1\ldots m} D_i$$

The hyperplane is selected as favorable, which has the most significant value. Where do is called the geometric margin of the dataset, we find out the optimal hyperplane in this article. To find out the optimal hyperplane, use the Lagrangian function i.e.

$$\text{Ɣ}(\omega, b, \beta) = \frac{1}{2}\omega \cdot \omega - \sum_{i=1}^{m} \beta_i \,[y : (\omega.x + b) - 1]$$

$$\nabla_\omega \text{Ɣ}(\omega, b, \beta) = \omega - \sum\nolimits_{i=1}^{m} \beta_i y_i x_i = 0 \qquad (26)$$

$$\nabla_b \text{Ɣ}(\omega, b, \beta) = -\sum\nolimits_{i=1}^{m} \beta_i y_i = 0 \qquad (27)$$

Get from Eq. 26 and 27, we can write as Eq. 28.

$$\omega = \sum\nolimits_{i=1}^{m} \beta_i y_i x_i \quad and \ \sum\nolimits_{i=1}^{m} \beta_i y_i = 0 \qquad (28)$$

By substituting the Lagrangian function

$$\omega\ (\beta, b) = \sum_{i=1}^{m} \beta_i - \frac{1}{2} \sum_{i=1}^{m} \cdot \sum_{j=1}^{m} \beta_i \beta_j y_i y_j x_i x_j$$

where i = 1, 2, 3, . . . , m

Due to constraint inequalities, extend the Lagrangian multipliers method to Karush-Kuhn Tucker (KKT) condition, which state that

$$\beta_i \left[ y_i \left( \omega_i \cdot x^* + b \right) - 1 \right] = 0 \qquad (30)$$

Eq. 30 $x^*$ is the optimal point and is the positive value, and for other points, its values are nearly equal to zero. So, we can write as in Eq. 31

$$y_i \left( \omega_i \cdot x^* + b \right) - 1 = 0 \qquad (31)$$

These are the closest points to the hyperplane is also known as support vectors. According to Eq. 31,

$$\omega - \sum_{i=1}^{m} \beta_i y_i x_i = 0$$

This can also be written as

$$\omega = \sum\nolimits_{i=1}^{m} \beta_i y_i x_i \qquad (32)$$

Eq. 33 gets when we compute the value of b

$$y_i(\left( \omega_i \cdot x^* + b \right) - 1) = 0 \qquad (33)$$

Multiply both sides with yi

$$y_i^2(\left( \omega_i \cdot x^* + b \right) - 1) = 0$$

As we know $yi^2$ is equal to 1

$$b = y_i - \omega_i \cdot x^* \tag{34}$$

$$b = \frac{1}{S} \sum\nolimits_{i=1}^{S} (y_i - \omega . x) \tag{35}$$

In Eq. 35, S is the number of support vectors, and on the hyperplane, we make the predictions. The hypothesis function is described in Eq. 36

$$U_{SVM} = \mathrm{H}\Gamma(\omega_i) = \begin{cases} +1 & \mathit{if}\ \omega . x + b \geq 0 \\ -1 & \mathit{if}\ \omega . x + b < 0 \end{cases} \tag{36}$$

The above points of the hyperplane, i.e., +1 is Shill bidding, and the below point of the hyperplane, i.e., −1, is the no shill bidding or the normal bid of the bidder. The same dataset was used in the SVM, which was used in the ANN. SVM data is tarin to linear SVM, Quadratic SVM, Cubic SVM, Fine Gaussian SVM, Medium Gaussian SVM, and Coarse Gaussian SVM with 5-fold cross-validation. For prediction, define the input parameters and output parameters with the k-fold cross-validation, then run this data on all the SVM models. In this article, run the dataset into 5-fold cross-validation. In 5-fold cross-validation, it divides the data into five numbers or chunks and validates with the next five numbers of chunks. After that, chunks are incremented by 5 to the next. In this method, the data is used for input and as well as output. Finally, SVM separated the classes with the hyperplane and predicted the result.

### 4.2.3 FUSSED ML ALGORITHM EMPOWER WITH FUZZY

Membership functions are used in the fuzzy. Output variables of SVM and ANN are used in the fuzzy as input variables. After defining the membership functions, we define the set of rules based on the membership functions of input and output variables. Based on ANN and SVM, out detection on SB fuzzy will decide whether the bidding is normal or shill bided. Blockage of users, discard the bidding is dependent on the fuzzy decision. The mathematically fuzzy-based decision can be written as

$$U_{ANN} \cap U_{SVM} (ANN, SVM) = \min [U_{ANN} (ANN), U_{SVM} (SVM)]$$

where UANN and USVM represent the membership function of ANN & SVM, respectively. These statements are relating the core ground for the structure of fuzzy rules.

IF (ANN is SB) and (SVM is SB)
Then (Bidding is SB).
IF (ANN is SB) and (SVM is Normal)
Then (Bidding is SB).
IF (ANN is Normal) and (SVM is SB)
Then (Bidding is SB).
IF (ANN is Normal) and (SVM is Normal)
Then (Shill Bidding is Normal).

According to the output parameters of ANN and SVM, possible outcome parameters are either normal or SB on both models. So, concerning the fuzzy logic, 4 rule sets are described in the Tab. 2. The proposed DFM-SB model uses the fuzzy set theory to map input feathers. A fuzzy inference engine is represented as a $\ddot{Ru}$ which is described as

$$\ddot{Ru}^e = \zeta^e \times \varsigma^e \tag{37}$$

$$U_{ANN\cap svm} = U_{ANN(\zeta)} \cap U_{svm(\varsigma)} \tag{38}$$

The rules are then deduced as a fuzzy relation Q4 as:

$$Q_4 = \bigcup_{e=1}^{4} \ddot{Ru}^e \tag{39}$$

$$U_{\ddot{R}}\,(Decision\ Base) = max_{1<x<4}\left[\prod_{\gamma=1}^{4}\left(U_{ANN_y}, N_{SVM_y}\right)\right] \tag{40}$$

There are many methods available for defuzzification. De-fuzzifier can be implemented through a centroid method, weighted average, mean-max, and max membership principle. But in proposed model uses the centroid type of defuzzifier. It describes the transformation of fuzzy output generated by the interface engine to frangible using similar functionalities in distinction to those used by the fuzzifier.

Eq. (41) describes the crisp point ξ .

$$\xi = \frac{\int \ddot{R} U_{\ddot{R}}\left(\ddot{R}\right) d\ddot{R}}{\int U_{\ddot{R}}\left(\ddot{R}\right) d\ddot{R}} \qquad (41)$$

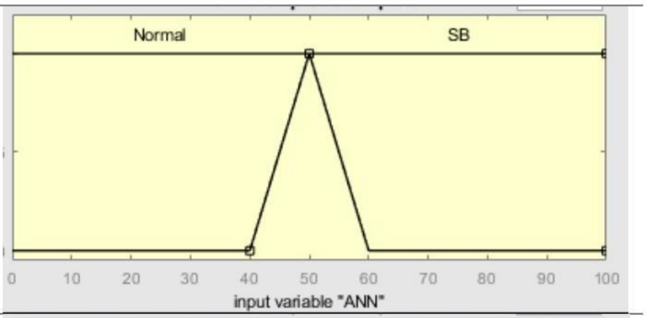
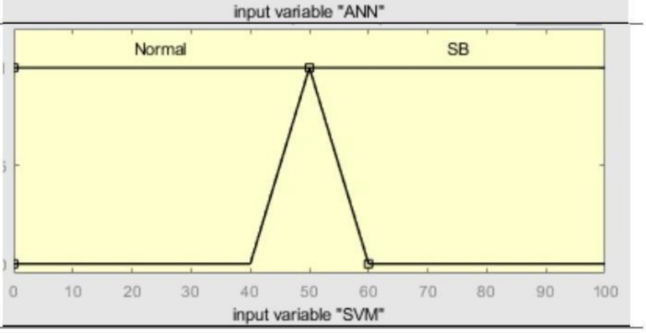
| | | |
|---|---|---|
| $ANN_{(U_{ANN(\zeta)})}$ | $U_{(Normal)}(\zeta)$<br>$= \left\{\max\left(\min\left(1, \frac{60-\zeta}{20}\right), 0\right)\right\}$<br>$U_{(SB)}(\zeta)$<br>$= \left\{\max\left(\min\left(\frac{\zeta-40}{20}, 1\right), 0\right)\right\}$ |  |
| $SVM_{(U_{SVM(\varsigma)})}$ | $U_{(Normal)}(\varsigma)$<br>$= \left\{\max\left(\min\left(1, \frac{60-\varsigma}{20}\right), 0\right)\right\}$<br>$U_{(SB)}(\varsigma)$<br>$= \left\{\max\left(\min\left(\frac{\varsigma-40}{20}, 1\right), 0\right)\right\}$ |  |

**Table 2:** Membership function of proposed DFM-SB system empower with fuzzy.

Fig. 2 describes that SVM and ANN are on the x and y-axis of the graph while SB-Detection is on the z-axis. Color-wise, yellow is define the SB detection, while the dark blue area is defined that the normal bid,

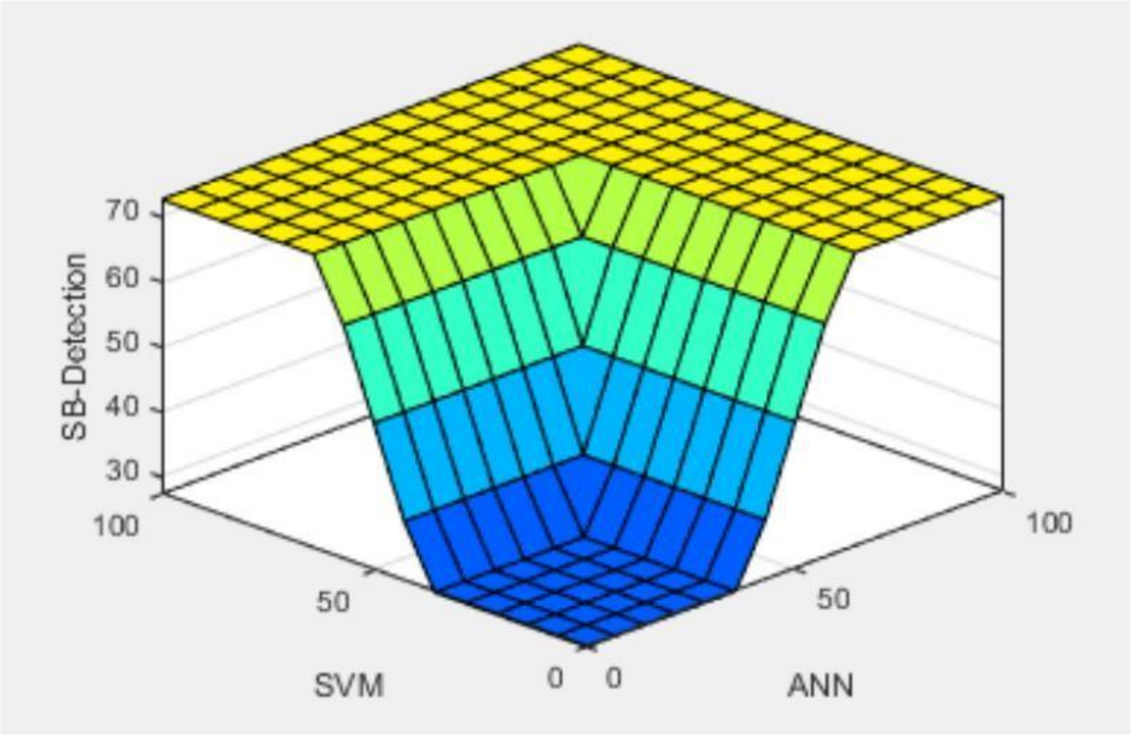


**Figure 4.3 :** Proposed fuzzy decision surface diagram

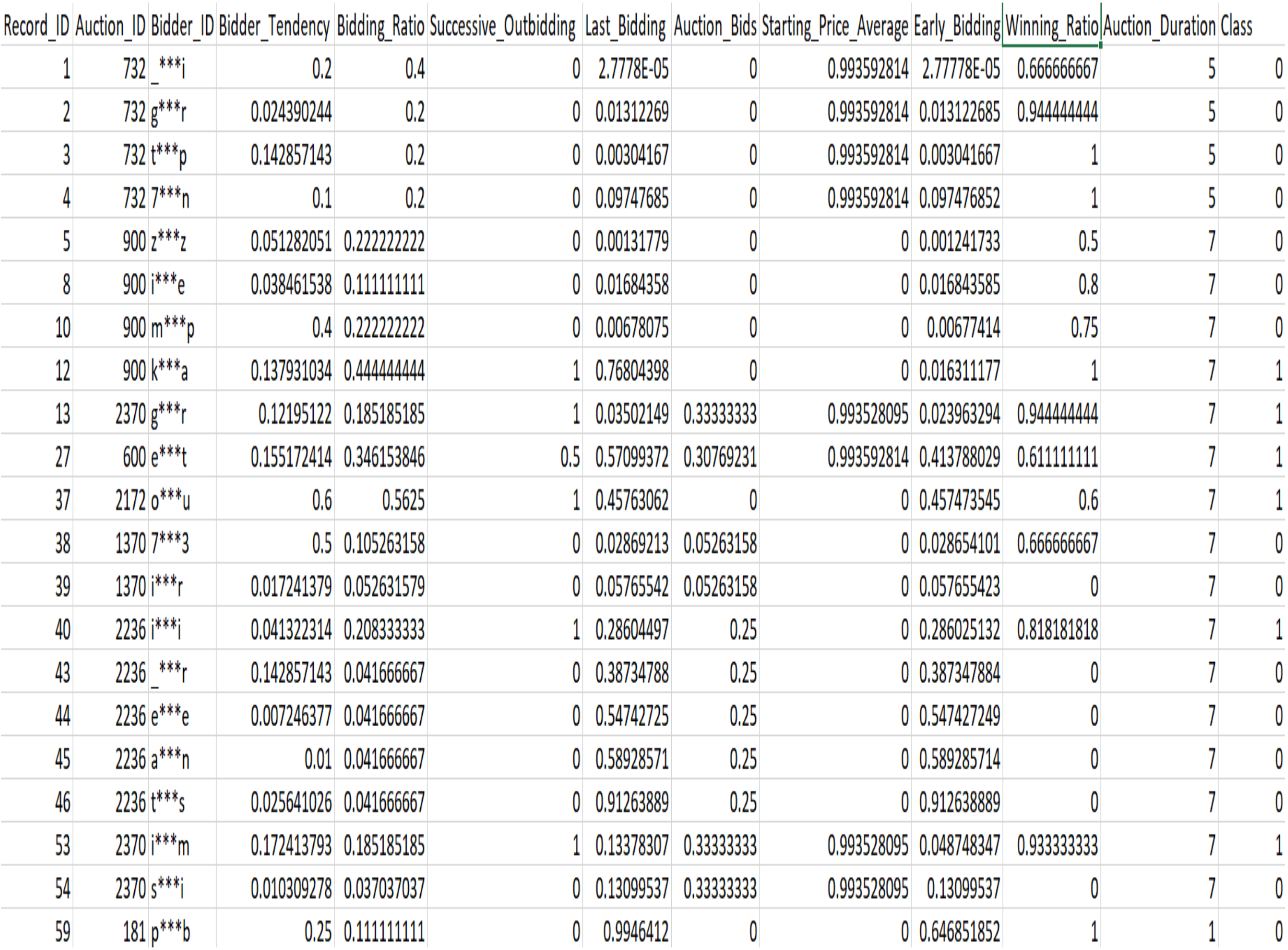
while the area between the yellow and blue is may or may not. SB depends upon the rule which we describe in the membership functions.

If the value of SVM and ANN is 0 to 40, then the SB-detection is Normal bid as the value of both models is increased by 40 to 60 then fuzzy decision is in slope between 0 to 1 may be normal bid or SB. When SVM and ANN value is greater than 60, then the SB is detected.

### 4.3 Dataset

The dataset is collected from the eBay auction record, in which popular brands' e-auction data is collected from the UCI data repository. The total auction record is 6321, which is used to predict real-time fraud detection.

## Shill Bidding Dataset



| Record_ID | Auction_ID | Bidder_ID | Bidder_Tendency | Bidding_Ratio | Successive_Outbidding | Last_Bidding | Auction_Bids | Starting_Price_Average | Early_Bidding | Winning_Ratio | Auction_Duration | Class |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 732 | _***i | 0.2 | 0.4 | 0 | 2.7778E-05 | 0 | 0.993592814 | 2.77778E-05 | 0.666666667 | 5 | 0 |
| 2 | 732 | g***r | 0.024390244 | 0.2 | 0 | 0.01312269 | 0 | 0.993592814 | 0.013122685 | 0.944444444 | 5 | 0 |
| 3 | 732 | t***p | 0.142857143 | 0.2 | 0 | 0.00304167 | 0 | 0.993592814 | 0.003041667 | 1 | 5 | 0 |
| 4 | 732 | 7***n | 0.1 | 0.2 | 0 | 0.09747685 | 0 | 0.993592814 | 0.097476852 | 1 | 5 | 0 |
| 5 | 900 | z***z | 0.051282051 | 0.222222222 | 0 | 0.00131779 | 0 | 0 | 0.001241733 | 0.5 | 7 | 0 |
| 8 | 900 | i***e | 0.038461538 | 0.111111111 | 0 | 0.01684358 | 0 | 0 | 0.016843585 | 0.8 | 7 | 0 |
| 10 | 900 | m***p | 0.4 | 0.222222222 | 0 | 0.00678075 | 0 | 0 | 0.00677414 | 0.75 | 7 | 0 |
| 12 | 900 | k***a | 0.137931034 | 0.444444444 | 1 | 0.76804398 | 0 | 0 | 0.016311177 | 1 | 7 | 1 |
| 13 | 2370 | g***r | 0.12195122 | 0.185185185 | 1 | 0.03502149 | 0.33333333 | 0.993528095 | 0.023963294 | 0.944444444 | 7 | 1 |
| 27 | 600 | e***t | 0.155172414 | 0.346153846 | 0.5 | 0.57099372 | 0.30769231 | 0.993592814 | 0.413788029 | 0.611111111 | 7 | 1 |
| 37 | 2172 | o***u | 0.6 | 0.5625 | 1 | 0.45763062 | 0 | 0 | 0.457473545 | 0.6 | 7 | 1 |
| 38 | 1370 | 7***3 | 0.5 | 0.105263158 | 0 | 0.02869213 | 0.05263158 | 0 | 0.028654101 | 0.666666667 | 7 | 0 |
| 39 | 1370 | i***r | 0.017241379 | 0.052631579 | 0 | 0.05765542 | 0.05263158 | 0 | 0.057655423 | 0 | 7 | 0 |
| 40 | 2236 | i***i | 0.041322314 | 0.208333333 | 1 | 0.28604497 | 0.25 | 0 | 0.286025132 | 0.818181818 | 7 | 1 |
| 43 | 2236 | _***r | 0.142857143 | 0.041666667 | 0 | 0.38734788 | 0.25 | 0 | 0.387347884 | 0 | 7 | 0 |
| 44 | 2236 | e***e | 0.007246377 | 0.041666667 | 0 | 0.54742725 | 0.25 | 0 | 0.547427249 | 0 | 7 | 0 |
| 45 | 2236 | a***n | 0.01 | 0.041666667 | 0 | 0.58928571 | 0.25 | 0 | 0.589285714 | 0 | 7 | 0 |
| 46 | 2236 | t***s | 0.025641026 | 0.041666667 | 0 | 0.91263889 | 0.25 | 0 | 0.912638889 | 0 | 7 | 0 |
| 53 | 2370 | i***m | 0.172413793 | 0.185185185 | 1 | 0.13378307 | 0.33333333 | 0.993528095 | 0.048748347 | 0.933333333 | 7 | 1 |
| 54 | 2370 | s***i | 0.010309278 | 0.037037037 | 0 | 0.13099537 | 0.33333333 | 0.993528095 | 0.13099537 | 0 | 7 | 0 |
| 59 | 181 | p***b | 0.25 | 0.111111111 | 0 | 0.9946412 | 0 | 0 | 0.646851852 | 1 | 1 | 0 |

Record ID: Unique identifier of a record in the dataset.

Auction ID: Unique identifier of an auction.

Bidder ID: Unique identifier of a bidder.

Bidder Tendency: A shill bidder participates exclusively in auctions of few sellers rather than a diversified lot. This is a collusive act involving the fraudulent seller and an accomplice.

Bidding Ratio: A shill bidder participates more frequently to raise the auction price and attract higher bids from legitimate participants.

Successive Outbidding: A shill bidder successively outbids himself even though he is the current winner to increase the price gradually with small consecutive increments.

Last Bidding: A shill bidder becomes inactive at the last stage of the auction (more than 90\% of the auction duration) to avoid winning the auction.

Auction Bids: Auctions with SB activities tend to have a much higher number of bids than the average of bids in concurrent auctions.

Auction Starting Price: a shill bidder usually offers a small starting price to attract legitimate bidders into the auction.

Early Bidding: A shill bidder tends to bid pretty early in the auction (less than 25\% of the auction duration) to get the attention of auction users.

Winning Ratio: A shill bidder competes in many auctions but hardly wins any auctions.

Auction Duration: How long an auction lasted.

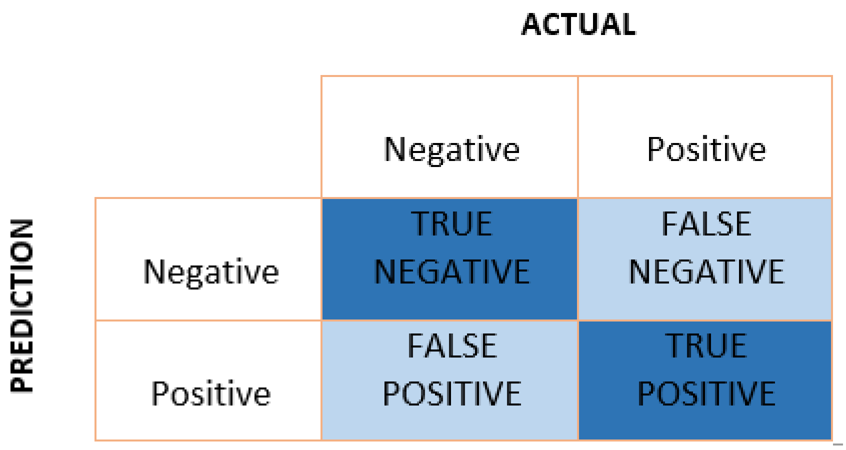
Class: 0 for normal behaviour bidding; 1 for otherwise.

## 4.4 METRICS CALCULATED

### Confusion Matrix

Simply, it is a matrix of 2×2 size for binary classification with one axis consisting of actual values and the other axis with predicted values. The size of the matrix can increase depending on the number of classes being predicted.

Otherwise known as the ‘error matrix’, it is a tabular visual representation of the predictions of the model against the ground truth labels.



| | | ACTUAL | |
| --- | --- | --- | --- |
| | | Negative | Positive |
| PREDICTION | Negative | TRUE NEGATIVE | FALSE NEGATIVE |
| | Positive | FALSE POSITIVE | TRUE POSITIVE |

**True Positive** is the correct positive prediction by the model.

**True Negative** is the correct negative prediction by the model.

**False Positive** is the wrong prediction of the positive by the model.

**False Negative** is the wrong prediction of the negative by the model.

With these values we can calculate the rate of each prediction category by a simple equation.

$$TPR = \frac{TP}{Actual\ Positive} = \frac{TP}{TP + FN}$$

$$FNR = \frac{FN}{Actual\ Positive} = \frac{FN}{TP + FN}$$

$$TNR = \frac{TN}{Actual\ Negative} = \frac{TN}{TN + FP}$$

$$FPR = \frac{FP}{Actual\ Negative} = \frac{FP}{TN + FP}$$

### Classification Accuracy

The simplest metric, it is calculated by **dividing the number of correct predictions by the total number of predictions, multiplied by 100**.

$$Accuracy = (TP+TN)/(TP+TN+FP+FN)$$

### Precision/Specificity

If the class distribution is imbalanced, classification accuracy isn't the best indicator for the performance of the model. To tackle a class-specific problem, we need a precision metric which is calculated by **True Positives divided by the sum of True Positives and False Positives.**

$$Precision = TP/(TP + FP)$$

### Recall/Sensitivity

Recall is the fraction of samples from one class which are predicted correctly by the model. It is calculated by **True Positives divided by the sum of True Positives and False Negatives.**

$$Recall = TP/(TP + FN)$$

### F1 Score

Now that we know what precision and recall are for classification problems, to calculate both simultaneously—F1, the harmonic mean of both, which also performs well on an imbalance dataset.

$$F1\ score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 * (Precision * Recall)}{(Precision + Recall)}$$

As shown in the above equation, F1 score gives the same importance to both—recall and precision. If we want to give more weight to one of them, F1 score can be calculated by attaching a value to either recall or precision depending on how many times the value is important. In the equation below, β is the weightage.

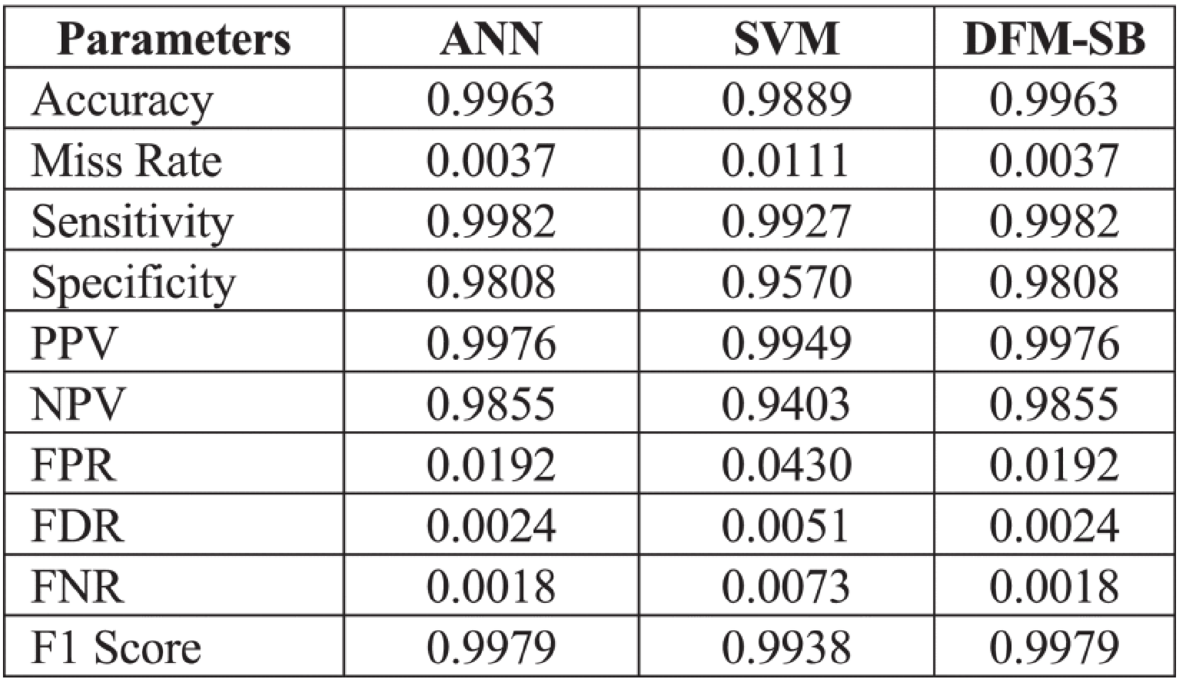
$$F_\beta = (1 + \beta^2) * \frac{(Precision * Recall)}{(\beta^2 * Precision) + Recall}$$

**Support**

Support is the number of actual occurrences of the class in the specified dataset. Imbalanced support in the datasetmay indicate structural weaknesses in the reportedscores of the classifier and could indicate the need for stratified sampling or rebalancing.

Support doesn’t change between models but instead diagnoses the evaluation process.

**METHODS COMPARED**



| Parameters | ANN | SVM | DFM-SB |
|---|---|---|---|
| Accuracy | 0.9963 | 0.9889 | 0.9963 |
| Miss Rate | 0.0037 | 0.0111 | 0.0037 |
| Sensitivity | 0.9982 | 0.9927 | 0.9982 |
| Specificity | 0.9808 | 0.9570 | 0.9808 |
| PPV | 0.9976 | 0.9949 | 0.9976 |
| NPV | 0.9855 | 0.9403 | 0.9855 |
| FPR | 0.0192 | 0.0430 | 0.0192 |
| FDR | 0.0024 | 0.0051 | 0.0024 |
| FNR | 0.0018 | 0.0073 | 0.0018 |
| F1 Score | 0.9979 | 0.9938 | 0.9979 |

The proposed model compares the proposed model in this article and the previous research work described in the literature review section. The proposed model performs a better approach and accuracy than the previous models. Model 3 shows the best accuracy is 99.7, i.e., under-sampling data, and the oversampling results are 94% on average. Our model accuracy is better, and its overall accuracy is 99.6 that is described below.

The data was divided into the training and testing dataset which is of 70% and 30%.The data which was used for training should be trained parallely on the both machine learning algorithms SVM and ANN. Then the outputs obtained by the Machine learning algorithms are stored.Based on the Predictions of SVM and ANN the Fuzzy decides the output Whether the bidder is legitimate or not.

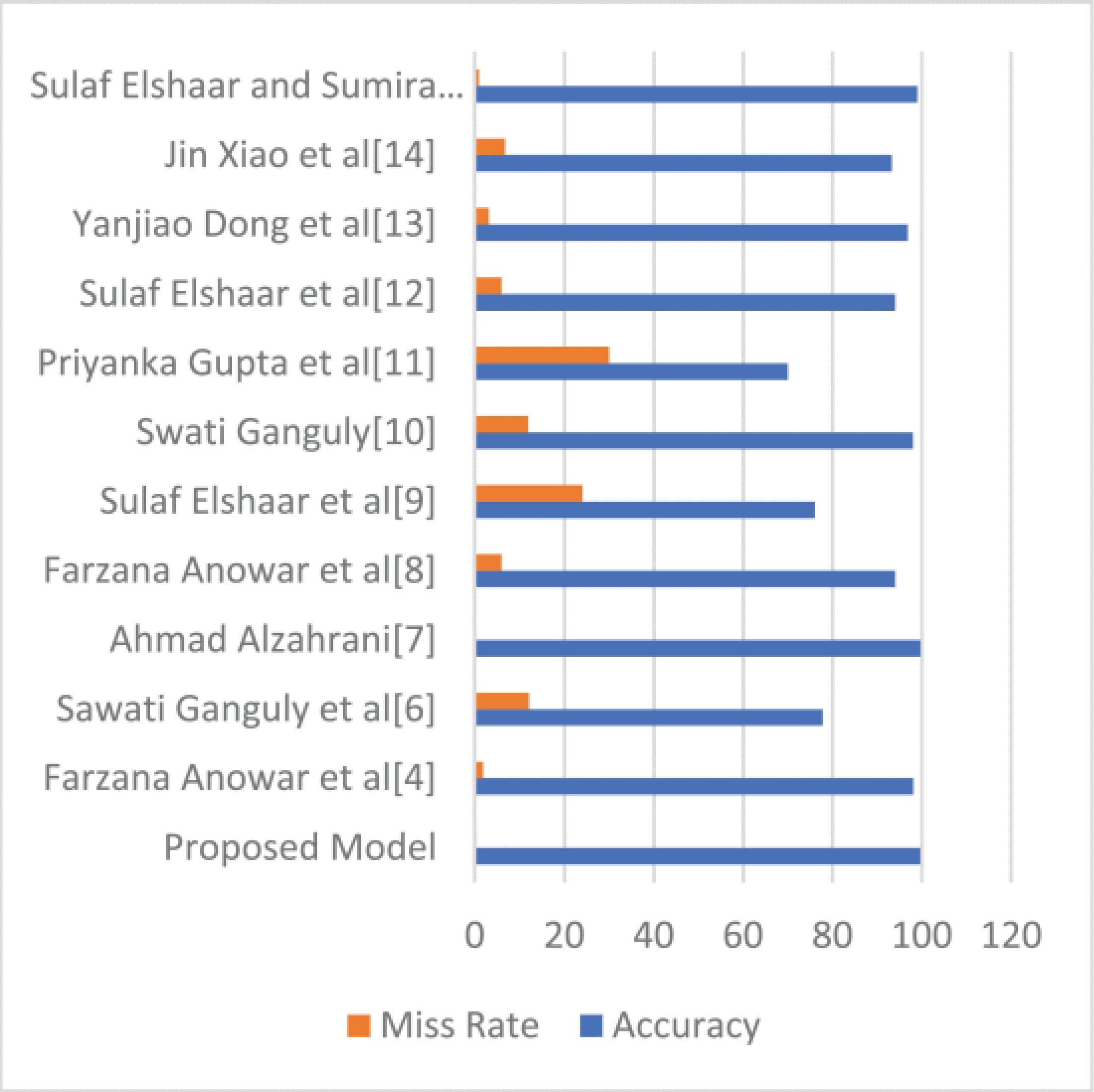
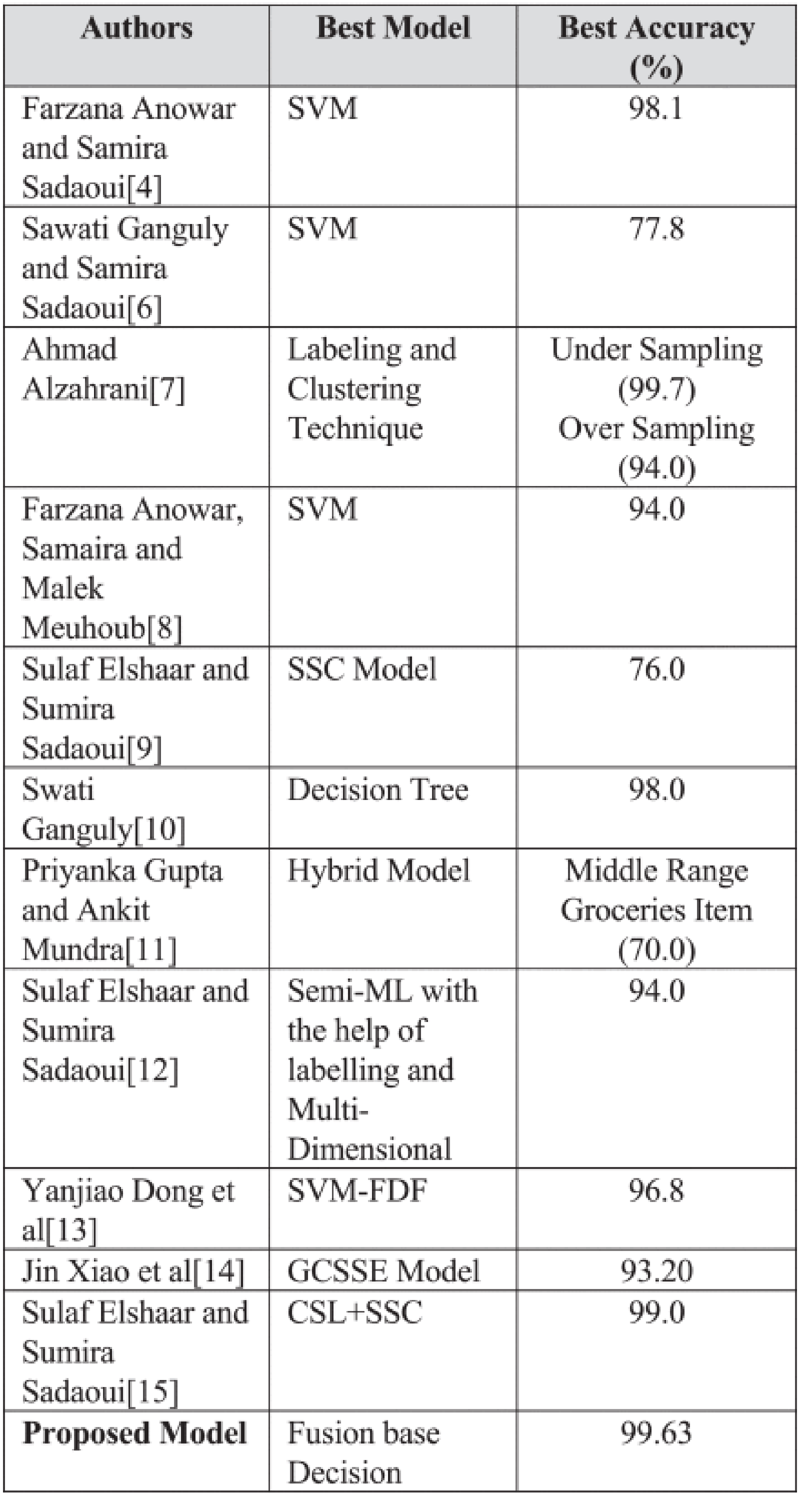


Table described the comparison between the proposed model with the previous models. The authors used different models to achieve the best accuracy. Different data preprocessing techniques are used on SVM and achieve the best 98.1% accuracy by using under-sampling. Oversampling attains 99.7% and 94% accuracy. With the help of the SSC, the model accomplishes 76% accuracy. By using a decision tree successfully achieve 98% accuracy.

### Comparison of Proposed DFM-SB Model With the Previous Models



| Authors | Best Model | Best Accuracy (%) |
|---|---|---|
| Farzana Anowar and Samira Sadaoui[4] | SVM | 98.1 |
| Sawati Ganguly and Samira Sadaoui[6] | SVM | 77.8 |
| Ahmad Alzahrani[7] | Labeling and Clustering Technique | Under Sampling (99.7)<br>Over Sampling (94.0) |
| Farzana Anowar, Samaira and Malek Meuhoub[8] | SVM | 94.0 |
| Sulaf Elshaar and Sumira Sadaoui[9] | SSC Model | 76.0 |
| Swati Ganguly[10] | Decision Tree | 98.0 |
| Priyanka Gupta and Ankit Mundra[11] | Hybrid Model | Middle Range Groceries Item (70.0) |
| Sulaf Elshaar and Sumira Sadaoui[12] | Semi-ML with the help of labelling and Multi-Dimensional | 94.0 |
| Yanjiao Dong et al[13] | SVM-FDF | 96.8 |
| Jin Xiao et al[14] | GCSSE Model | 93.20 |
| Sulaf Elshaar and Sumira Sadaoui[15] | CSL+SSC | 99.0 |
| **Proposed Model** | Fusion base Decision | 99.63 |

# CHAPTER-5
# TESTING

## 5.1 Objective of Testing

Testing is a fault detection technique that tries to create failure and erroneous states in a planned way. This allows the developer to detect failures in the system before it is released to the customer.

Note that this definition of testing implies that a successful test is test that identifies faults. We will use this definition throughout the definition phase. Another often used definition of testing is that it demonstrates that faults are not present.

Testing can be done in two ways:

- Top down approach.
- Bottom up approach

**Top-down approach –**

This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written.

**Bottom-up approach –**

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded within the larger system. In this project, bottom-up approach is used where the lower level modules are tested first and the next ones having much data in them.

## 5.2 Testing Methodologies

1. Unit testing
2. Integration testing
3. User Acceptance testing
4. Output testing
5. Validation testing

**Unit testing:** Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module’s control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

**Integration testing:** Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

**User Acceptance testing:** User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

**Output testing:** After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways - one is on screen and another in printed format.

**Validation testing:** Validation checks are performed on the following fields:

**Text field:** The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always ashes and error message.

**Numeric field:** The numeric field can contain only numbers from 0 to 9. An entry of any character ashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

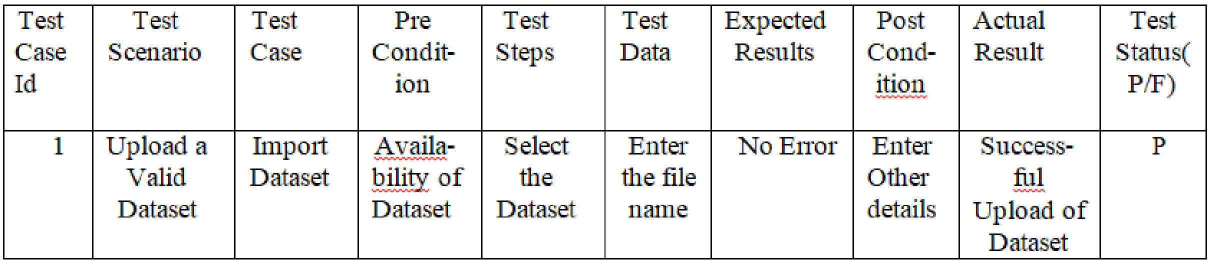
**Preparation of test data:** Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and correctionsare also noted for future use.

**Using Live test data:** Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves. It is difficult to obtain live data in sufficient amounts to conductive extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systemstest and in fact ignores the cases most likely to cause system failure.

**Using artificial test data:** Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program. The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an in dependent team of testers formulates a testing plan, using the systems specifications. The package" Virtual Private Network" has satisfied all the requirements specified as per software requirement specification and was accepted.

## 5.3 Test Cases

**Table 5.1: Test Case to Import Dataset**



| Test Case Id | Test Scenario | Test Case | Pre Condit-ion | Test Steps | Test Data | Expected Results | Post Cond-ition | Actual Result | Test Status( P/F) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Upload a Valid Dataset | Import Dataset | Availa-bility of Dataset | Select the Dataset | Enter the file name | No Error | Enter Other details | Success-ful Upload of Dataset | P |

**Table 5.2 :Test Case for Estimating Missing Values**

| Test Case Id | Test Scenaric | Test Case | Pre Conditio n | Test Steps | Test Data | Expected | Post Condition | Actual Result | Test Status (P/F) |
|---|---|---|---|---|---|---|---|---|---|
| 2 | Missing values | Preproce ssing | Avail-ability of Dataset | Import the features with missing values | dataset | No Error | Missing Values Estimate d | No missing values found | P |

**Table 5.3 :Test Case for Feature selection**

| Test Case Id | Test Scenario | Test Case | Pre Conditio n | Test Steps | Test Data | Expected | Post Condition | Actual Result | Test Status (P/F) |
|---|---|---|---|---|---|---|---|---|---|
| 3 | Feature Selection | Preproce ssing | Avail-ability of Dataset | Import the data Implemen SVM and ANN algorithm | data set | Features extracted | Important features are extracted | Features extracte d | P |

**Table 5.4 :Test Case for fraud detection**

| Test Case Id | Test Scenario | Test Case | Pre Condition | Test Steps | Test Data | Expected | Post Condition | Actual Result | Test Status (P/F) |
|---|---|---|---|---|---|---|---|---|---|
| 4 | Predictio | Predictio | Data | Import data and implement SVM and ANN | Dataset | Shill bidder is predicted | Predicts Shill bidder with most accuracy | Shill bidder found | P |

## CODING

```
import numpy as np
import pandas as pd
import tensorflow as tf
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

df=pd.read_csv("C:/Users/Shill Bidding Dataset.csv")
df.head(30)

df1 = df.drop(columns=['Record_ID','Auction_ID','Bidder_ID'],axis=0)
df1.head(15)

X=df1.iloc[:,:-1]
Y=df1.iloc[:,-1]

from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3,random_state=0)

from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.neural_network import MLPClassifier

MLPC=MLPClassifier(hidden_layer_sizes=(15,),max_iter=10000)
MLPC.fit(X_train,Y_train)
y_pred=MLPC.predict(X_test)
acc=accuracy_score(Y_test,y_pred)
print(acc)
```

```
cnf_matrix = confusion_matrix(y_pred,Y_test)
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu", fmt='g')
plt.ylabel('Actual Label')
plt.xlabel('Predicted Label')

svm = SVC()
svm.fit(X_train, Y_train)
y_pred_svm=svm.predict(X_test)
acc=accuracy_score(y_pred_svm,Y_test)
Acc

cnf_matrix = confusion_matrix(y_pred_svm,Y_test)
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu", fmt='g')
plt.ylabel('Actual Label')
plt.xlabel('Predicted Label')

l=[]
for i in range(len(y_pred)):
    l.append(y_pred[i] and y_pred_svm[i])
l1=np.array(l)
l1

ac1=accuracy_score(l1,Y_test)
ac1
```

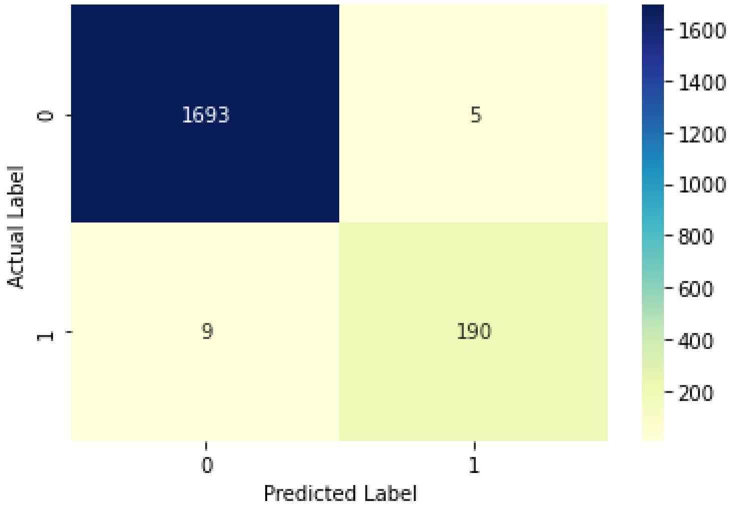
# CHAPTER 6

# RESULTS

## 6.1:Actual Results Obtained

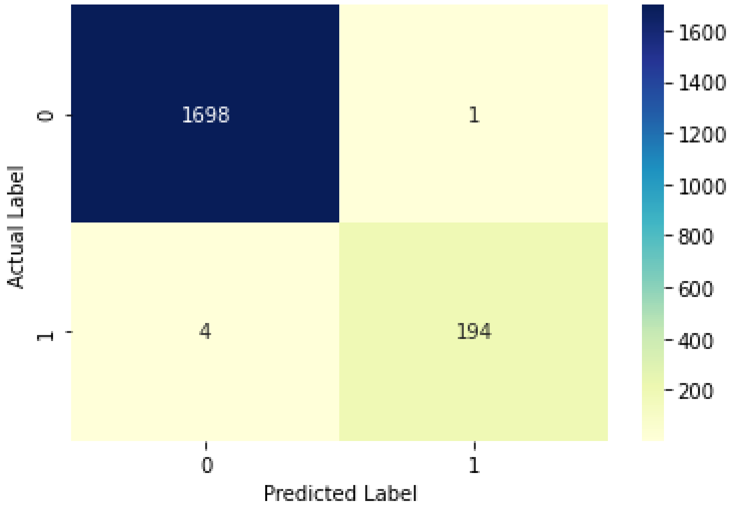
The accuracy obtained when trained on the artificial neural Networks is 0.992619926199262

The Confusion matrix obtained when trained on the artificial neural networks is



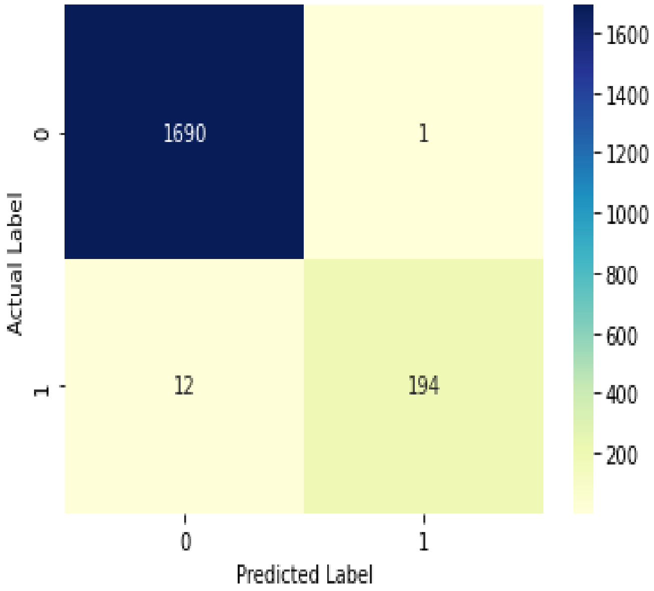
The accuracy obtained when trained on the SVM is 0.9973642593568793.

The Confusion matrix obtained when trained on the SVM is



The accuracy based on the both artificial neural networks and the svm is 0.993147074.

The confusion matrix for fuzzy based on both the artificial neural networks and svm is



## 6.2: Analysis on the results obtained

In the model ,firstly 70 % of the data is set for training. The remaining 30% is for testing the data.So the model is trained with SVM and ANN by the training data that is available.After training the model, then the data which is for testing will be applied on the SVM then the results obtained for it are accuracy with 99.7 approx.Then on applying the same testing data the results that are obtained are accuracy with 99.2 approx.After testing against the both the algorithms the fuzzy decides whether the bidder is shill bidder or not.After that based on the predictions of the SVM and ANN the Fuzzy decides the output by the training data.Now on comparing with the test data on the Fuzzy the results that are obtained are accuracy with 99.3.The above fusion works even for larger dataset since it has both the comparisions of the algorithms.So it produces the greater results.

# CHAPTER 7

# CONCLUSION

Online SB detection is a prevalent crime and very difficult to detect because of its very similar behavior during real-time bidding. Due to SB, the other bidder gets a lot of money loss, and the seller receives the extra money. As defined earlier, the previous researcher has a lot of work, but there is always a gap in research. So, this article developed a decision base fusion model used to find the SB in the real-time auction in which SVM and ANN are used for prediction, and Fuzzy is used to decide whether the SB is committed or the bid is normal.

When the bid is made the bidding, the behavior is judge by both SVM and ANN at the same time. Based on their prediction, fuzzy decide the bid is normal or SB.

The proposed model predicts the best results compared to the previous research models, which help detect real-time auction fraud, which helps to block the user and discard fake bids. Therefore, this research will be helpful to both bidders and the e-Auction companies that face the yearly millions of dollars loss and fraud reports.

# CHAPTER 8

## REFERENCES

[1] A. Alzahrani and S. Sadaoui, ‘‘Scraping and preprocessing commercial auction data for fraud classification,’’ Dept. Comput. Sci., Univ. Regina, Regina, SK, Canada, Tech. Rep. CS 2018-05, 2018, p. 17, doi: 10.6084/m9.figshare.6272342.

[2] J. Trevathan, ‘‘Getting into the mind of an ‘in-auction’ fraud perpetrator,’’ Comput. Sci. Rev., vol. 27, pp. 1–15, Feb. 2018, doi: 10.1016/j. cosrev.2017.10.001.

[3] J. Trevathan, C. Aitkenhead, N. Majadi, and W. Read, ‘‘Detecting multiple seller collusive shill bidding,’’ Aug. 2018, arXiv:1812.10868. [Online]. Available: http://arxiv.org/abs/1812.10868.

[4] F. Anowar and S. Sadaoui, ‘‘Detection of auction fraud in commercial sites,’’ J. Theor. Appl. Electron. Commer. Res., vol. 15, no. 1, pp. 81–98, 2020, doi: 10.4067/S0718-18762020000100107.

[5] B. C. McCannon and E. Minuci, ‘‘Shill bidding and trust,’’ J. Behav. Exp. Finance, vol. 26, Jun. 2020, Art. no. 100279, doi: 10.1016/j.jbef. 2020.100279.

[6] S. Ganguly and S. Sadaoui, ‘‘Online detection of shill bidding fraud based on machine learning techniques,’’ in Recent Trends and Future Technology in Applied Intelligence (Lecture Notes in Artificial Intelligence), vol. 10868. Montreal, QC, Canada: Springer, 2018.

[7] A. Alzahrani and S. Sadaoui, ‘‘Instance-incremental classification of imbalanced bidding fraud data,’’ in Proc. 11th Int. Conf. Agents Artif. Intell. (ICAART), vol. 2, 2019, pp. 92–102.

[8] F. Anowar, S. Sadaoui, and M. Mouhoub, ‘‘Auction fraud classification based on clustering and sampling techniques,’’ in Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA), Dec. 2018, pp. 366–371, doi: 10.1109/ICMLA.2018.00061.

[9] S. Elshaar and S. Sadaoui, ‘‘Detecting bidding fraud using a few labeled data,’’ in Proc. 12th Int. Conf. Agents Artif. Intell. (ICAART), vol. 2, 2020, pp. 17–25, doi: 10.5220/0008894100170025.

[10] S. Ganguly and S. Sadaoui, ‘‘Classification of imbalanced auction fraud data,’’ in Proc. Can. Conf. Artif. Intell., in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, 2017, vol. 10233, no. 2, pp. 84–89, doi: 10.1007/978-3-319-57351-9_11.

[11] P. Gupta and A. Mundra, ‘‘Online in-auction fraud detection using online hybrid model,’’ in Proc. Int. Conf. Comput., Commun. Autom. (ICCCA), May 2015, pp. 901–907, doi: 10.1109/CCAA.2015.7148504.

[12] S. Elshaar and S. Sadaoui, ‘‘Semi-supervised classification of fraud data in commercial auctions,’’ Appl. Artif. Intell., vol. 34, no. 1, pp. 47–63, Jan. 2020, doi: 10.1080/08839514.2019.1691341.

[13] Y. Dong, Z. Jiang, M. Alazab, and P. KUMAR, ‘‘Real-time fraud detection in e-market using machine learning algorithms,’’ J. Multiple-Valued Logic Soft Comput., vol. 36, nos. 1–3, pp. 191–209, 2021.

[14] J. Xiao, X. Zhou, Y. Zhong, L. Xie, X. Gu, and D. Liu, ‘‘Costsensitive semi-supervised selective ensemble model for customer credit scoring,’’ Knowl.-Based Syst., vol. 189, Feb. 2020, Art. no. 105118, doi: 10.1016/j.knosys.2019.105118