#953998159

Masters
Thesis

Alkheliwi,
Turki M.

ENHANCING RISK PREDICTION IN FINANCIAL APPLICATIONS USING DATA MINING AND GAME THEORY PRINCIPLES

by

Turki M. Alkheliwi

B.S. (Michigan State University) 2011

THESIS

Submitted in partial satisfaction of the requirements

for the degree of

MASTER OF SCIENCE

in

INFORMATION TECHNOLOGY

in the

GRADUATE SCHOOL

of

HOOD COLLEGE

May 2016

Accepted:

Carol Jim, Ph.D.
Thesis Advisor

Khalid Lateef, Ph.D.
Thesis Co-Advisor

Stephen Penn, Ph.D.
Thesis Co-Advisor

Maria G. Cowles, Ph.D.
Dean of the Graduate School

Ahmed Salem, Ph.D.
Director, Information Technology Program
Committee Member

George Dimitoglou, Ph.D.
Committee Member

STATEMENT OF USE AND COPYRIGHT WAIVER

I authorize Hood College to lend this thesis, or reproductions of it, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

ABSTRACT

This thesis examines the potential of applying Game Theory to Data Mining mechanisms to enhance the accuracy of predicting risk in financial settings. There have been many attempts made in the past to enhance Data Mining results using different methods including Game Theory principles. Despite the promising results of previous work in integrating Game Theory and Data Mining, further research is needed to explore the potential of creating a combined model that can be applied to a range of datasets to successfully enhance risk prediction. We apply a variety of different tree data mining algorithms to the German Credit Dataset. Then, we propose a combined model to enhance the accuracy of the data mining results by using Game Theory principles. Our approach focuses on correcting the error from the incorrectly classified instances by our proposed enhanced game tree model. By using the payoff table derived from our enhanced game tree model and the binomial distribution, we can determine the percentage of enhancement to the tree-based data mining results. Our results show that applying Game Theory principles to Data Mining techniques in a combined model can improve overall accuracy and enhance decision support systems in financial applications.

ACKNOWLEDGEMENTS

I would like to thank my academic advisor, Dr. Ahmed Salem, who inspired me and ignited my interest in research. Dr. Salem led me through this journey until he placed me into the safe hands of Dr. Carol Jim who guided me from the beginning of my research journey until the very end. I could not have done it without their guidance, help, and encouragement. I also would like to thank my thesis co-advisors Dr. Khalid Lateef and Dr. Stephen Penn, I thank you for all of your advice, suggestions, and review of this manuscript. Thanks are also expressed to Dr. Maria Cowles and Dr. George Dimitoglou for their service on the reading committee.

Special thanks goes to my family for their unwavering support and encouragement throughout my education.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Game theory focuses on situations involving multiple parties with different competing interests. The agents or players are assumed to be rational and informed thus situations can be either cooperative or non-cooperative. In a non-cooperative situation, sets of possible actions available to an agent are called options; furthermore, the sets of options that can be taken by the agent are called strategies. Once an agent chooses a strategy, the result is called an outcome [1].

The concept of Game Theory is based on the idea that the two competing agents are aiming to maximize profit and minimize cost. This is why we can think of it as a cost function which is different for each agent. If there is only one agent, the agent will enter the field of optimization given that there is no competing agent [1].

The two forms usually applied are: the extensive form and the strategic or normal form. The strategic or normal form is commonly used where there are two agents or decision makers involved. It is represented as a payoff matrix and the agents choose their strategies simultaneously. The other form is the extensive form which is represented as a game tree and the agents make sequential decisions [1].

Game Theory has many applications in real life such as in economics, business, political science, biology, philosophy, computer science and many others. One particular application of Game Theory is its use to assist in decision making. There are many studies examining the effectiveness of applying Game Theory to data sets in order to come up with better decisions. Chen and Wang [2] analyzed the competitive decision making process between enterprises using the game model. They proposed a model based on Bayes rule that provides guidance for enterprises making competitive strategies in complex or uncertain conditions. In a situation where there is incomplete information and reduced rationality among decision makers, decision makers are required to adjust their strategy continuously as more information becomes available [2]. Another closely related application would be in finance where [3] and [4] both look at Game Theory from a financial angle exploring the possibility of yielding better results. There have been many attempts to use Data Mining and Machine Learning on financial datasets before for the same goal of arriving at better results that could help with financial decision-making [5]. Others explored the option of combining Game Theory with Data Mining techniques to further enhance the results [6]. Despite the promising results, further research is needed to better understand the effects of applying the combination of Game Theory and Data Mining on financial datasets. We intend to focus on a less explored yet promising path of combining game theory and machine learning to enhance decision support systems in financial applications.

Game theory rules and data mining algorithms can be combined to further enhance the accuracy of predictions in financial applications. We have begun to explore this possibility in theory in our previous paper titled 'Applying Game Theory Rules to Enhance Decision Support Systems in Credit and Financial Applications' [19]. We presented a model using game trees as a basis for making financial decisions involving the assessment of a potential customer's creditworthiness and the risk associated with each one. The data set that was used was the German Credit data set that was taken from the University of California, Irvine UCI Machine Learning Repository [10]. The extensive form of game theory was proposed due to the sequential nature of making decisions in the process of applying for credit at a bank or a financial institution.

The risk associated with credit applications is a major problem for financial institutions. Being able to reduce this risk and increase the accuracy of predicting an applicant's creditworthiness even by a small amount can save financial institutions millions of dollars or more. Thus, we propose a combined model integrating data mining and game theory principles to improve the accuracy of predicting an applicant's creditworthiness which will lead to a reduction in risk for financial institutions.

The remainder of this thesis is organized as follows. In Chapter 2, we provide a comprehensive literature review about previous scholarly research focusing on the German credit dataset and other attempts to combine game theory and data mining techniques. Chapter 3 presents a general overview of the model design combining data mining with game theory principles. It also introduces the German credit dataset and explains the experimentation process and results of applying different tree-based data mining algorithms to the dataset. Chapter 4 explains the application of game theory techniques to the data mining results to improve upon merely using a model based solely on a data mining technique. The results of this accuracy enhancement are also discussed. Chapter 5 summarizes the thesis and discusses the potential for future work.

CHAPTER 2: LITERATURE REVIEW

2.1 DECISION MAKING USING GAME THEORY

Many scholars studied Game Theory and the potential of combining Game Theory and Data Mining. There are a number of research papers exploring the potential of Game Theory and the combination of Game Theory and Data Mining and they are in different fields like finance, engineering, science and other fields. Dong-hong et al [7] proposed a price bidding model called 'bidding price game model' which was used to analyze the string bid phenomenon consequently leading to finding the root of this phenomenon. The paper discussed a way to improve the probability of winning a bid and how to strengthen the competitiveness of a bidder. The authors cite one shortcoming related to the need to change the model if the base price is classified which is an area of further research [7].

When it comes to making investment decisions, Liao et al [3] looked at investments in technology as an area of competition between enterprises. They proposed replacing traditional investing methods with one that is based on game theory and real options. It results in making incomplete information being matched leading the management to realize production cost and the entire competition status [3].

Decision makers in a conflict sometimes make their decisions under unclear fuzzy information. Li et al [1] used two fuzzy approaches to investigate the Game of Chicken -2x2 game model extensively studied in Game Theory literature-, the first one is a fuzzy multicriteria decision analysis method to optimize strategies for players taking into account subjective factors based on the player's objective and then aggregates those objectives using a weight vector. The second approach involves the theory of fuzzy moves (TFM) which is the opposite of the traditional theory of moves (TOM). Those fuzzy approaches are effective in dealing with subjective, uncertain or fuzzy information and they present valuable and realistic insights into the strategic aspects of the game of Chicken. The TFM findings depend of the transformation function mapping the local game to a global game in addition to the inference engine. They used a simple linear transformation and a simple inference engine. Further research can look deeper into possible improvement of the fuzzy move algorithm [1].

Sometimes the decision making process can get very complicated. Castillo and Dorao [8] looked at an example in the area of Liquefied Natural Gas projects. Due to the high price of investing in Liquified Natural Gas (LNG) projects and the complexity of the decision making (DM) process, there is a need for a systematic design framework that addresses the different phases and multiple actors involved in the process. This paper proposes a framework for LNG consisting of a multi-objective DM procedure based on game theory which takes into consideration the operational conditions of the equipment. This framework enabled the upper level UL to send information to the lower level LL which allowed the LL to solve its bargaining problem according to the preferences of the leader UL. The gaming outcome in the LL were beyond the normal interception space. Finally, threats were executed to get the final decision and lower the cost and the implementation of this framework which would facilitate communication between the different levels of the DM on LNG projects [8].

In complex systems, Li et al [9] explored the application of game theory on Fault Detection and Diagnosis (FDD), providing literature survey and a proposal for an alternative processing model and algorithm on the application of game theory for complex systems. The literature survey concludes that the introduction of game theory into the FDD decision problem is not popular. The proposal is tested on a machining center which resulted in elementary effect but it requires more validation and improvement to make it more practical [9].

2.2 APPROACHES USING THE GERMAN CREDIT DATASET

There are a number of papers that used the German Credit dataset to experiment with different mechanisms to better assess the risk associated with a potential customer. We plan to work with the same dataset. The approaches applied are the following: Hybrid Support Vector Machine (SVM), the combination of Genetic Algorithm (GA) and Hybrid Support Vector Machine, combining feature selection and Neural Network, Genetic Programming (GP), Backpropagation neural network (BP) and a variety of different combinations of different models. The following is a review of the approaches.

2.2.1 HYBRID SUPPORT VECTOR MACHINE

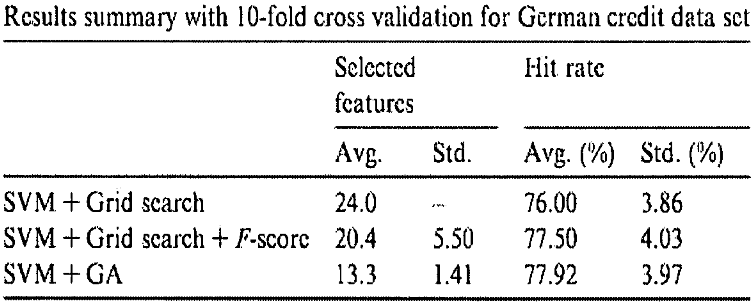
Huang et al [11] proposed a hybrid Support Vector Machine (SVM) based approach in analyzing the dataset and evaluating an applicant's creditworthiness. They compare their hybrid approach to neural networks, genetic programming and decision tree classifiers, and they found that the hybrid Support Vector Machine approach proposed in the paper achieves relatively the same level of accuracy. The Support Vector Machine when combined with genetic algorithm (GA-SVM) performs model parameters optimization and feature selection simultaneously. It is based on the decision function:

$$
\begin{aligned}
f(x) &= \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i^* \langle \Phi(x), \Phi(x_i) \rangle + b^*\right) \\
&= \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i^* \langle k(x, x_i) \rangle + b^*\right)
\end{aligned}
\qquad (1)
$$

α_i is Lagrange multipliers and α_i >=0

Φ is a mapping function for mapping training samples into higher-dimensional feature space

Table 1 [11] shows in summary the accuracy for each classification method represented by hit rate. Based on the Friedman test (with p = 0.32), the difference is not significant between the results. The models, SVM + Grid search, SVM + Grid search + F-score and SVM + GA, achieved 76%, 77.50% and 77.92%, with average selected features of 24, 20.4, and 13.3 respectively. We notice that the SVM + GA model used much less features compared to the SVM + Grid search and SVM + Grid search + F-score.



Results summary with 10-fold cross validation for German credit data set

	Selected features		Hit rate	
	Avg.	Std.	Avg. (%)	Std. (%)
SVM + Grid search	24.0	--	76.00	3.86
SVM + Grid search + *F*-score	20.4	5.50	77.50	4.03
SVM + GA	13.3	1.41	77.92	3.97

Table 1. Average hit rate and the number of features selected.

The paper concludes with the fact that statistical models are effective when accompanied with certain assumptions. However, artificial intelligence techniques like Support Vector Machine, Genetic Programming, Neural networks or decision tree do not require the same assumptions or field knowledge. Support Vector Machine can successfully categorize or classify loan applicants as high risk or low risk which is beneficial to the creditor or the financial institution lending the money reducing their risk and maximizing their savings. Also, Support Vector Machine can achieve almost identical results when compared to Neural Networks or Genetic Programming. One possible disadvantage of Support Vector Machine-Genetic Algorithm including Genetic Programming is that they require long training time.

2.2.2 COMBINING FEATURE SELECTION AND NEURAL NETWORK

Dea, Griffith and Riordan [12] propose a combined approach to solve classification problems. The proposal combines feature selection and neural networks. They use some techniques from the field of information theory to select and identify certain set of important attributes. A neural network will be used and trained with these attributes. The neural network then is used for classification. They use a feature selection algorithm, which selected 7 attributes out of the 20 attributes. This can be shown in Table 2 below [12] that shows the information gain *G* and normalized gain *G*'. That is using:

$$\text{accuracy} = \frac{\text{number of tuples correctly classified}}{\text{total number of tuples}} \qquad (2)$$

Based on the numbers the following attributes were selected: status, duration, credit history, credit amount, savings, housing and foreign worker.

No.	Attribute	$\mathcal{G}$	$\mathcal{G}'$
1.	status	0.08166752	0.04533132
2.	duration	0.01565728	0.01175013
3.	credit history	0.03506461	0.02039325
4.	purpose	0.02510743	0.00945620
5.	credit amount	0.01835606	0.02097592
6.	savings	0.04112237	0.02461317
7.	employment duration	0.01262678	0.00581796
8.	installment rate	0.00467093	0.00258875
9.	personal status	0.00621573	0.00404868
10.	debtors	0.00481950	0.00893288
11.	residence	0.00117720	0.00064023
12.	property	0.01892740	0.00971905
13.	age	0.01454505	0.00788522
14.	installment plans	0.00604603	0.00699498
15.	housing	0.01267492	0.01136562
16.	existing credits	0.00131140	0.00119170
17.	job	0.00468588	0.00326961
18.	liable people	0.00030049	0.00049862
19.	telephone	0.00002599	0.00002691
20.	foreign worker	0.00523591	0.02339419
AVERAGE		0.01551192	0.01094472

Table 2. Attribute gains of the data set.

They encountered some difficulties related to the quality of the data set. The degree of error is attributed to the level of noise in the dataset that includes irrelevant, missing, incorrect, and contradictory data, which generally reduces the accuracy of the prediction. The second possibility could be due to imbalance in the training set.

Twenty neural networks were used with 20 attributes and another twenty neural networks were used with 7 attributes selected by the algorithm. Tables 3 and 4 [12] show the difference.

Units	Links	Acc. on train set (%)		Acc. on test set (%)	
		Ave.	Std. Dev.	Ave.	Std. Dev.
1	27	77.83	0.23	75.85	0.35
2	54	77.58	1.02	74.45	0.46
3	81	78.88	1.36	74.45	1.65
4	108	80.38	1.09	73.15	0.46

Table 3. Results with 7 selected attributes used as input from the German Credit data set.

Units	Links	Acc. on train set (%)		Acc. on test set (%)	
		Ave.	Std. Dev.	Ave.	Std. Dev.
1	74	85.99	0.31	72.66	1.13
2	148	86.49	1.48	72.36	2.21
3	222	88.19	2.34	72.36	0.17
4	296	92.69	0.75	71.46	1.75

Table 4. Results with all 20 attributes used as input from the German Credit data set.

The goal behind the attempt to combine feature selection and neural networks is to achieve high accuracy. Advantages to this approach include the robustness of the approach to noise and it is easier to trace computationally with the reduction of the attributes. The shortcoming to this approach is that it is generally complicated and further simplification is needed. The approach is also in need of more experimentation to find out how it performs using different attributes. Future work can include examining a pruning algorithm to simplify the neural network and also trying out different numbers of attributes to see how the numbers affect the results.

2.2.3 GENETIC PROGRAMMING (GP), BACKPROPAGATION (BP), SUPPORT VECTOR MACHINE (SVM), COMBINATION OF MODELS

Zhang, Huang, Chen and Jiang [13] do a general comparison between different data mining techniques using the German Credit data set. They compare three credit scoring models which they think are powerful. They are genetic programming (GP), backpropagation neural networks (BP) and support vector machine (SVM). Then they proposed a combined model, which is compared to the three mentioned methods, and they claim that it produces good classification results.

The results showing the accuracy of Genetic Programming (GP), backpropagation neural networks (BP), Support Vector Machine (SVM) and the combined model (CM) proposed in the paper are shown below in Table 5 [13].

	G1	G2	G3	G4	G5	G6	G7	G8	average
BP	81.06	77.74	79.07	80.73	78.4	**80.4**	82.06	**78.73**	79.77
GP	80.73	78.74	78.4	81.06	78.4	79.07	82.06	77.74	79.53
SVM	81.06	78.07	77.74	81.73	77.07	77.07	80.39	78.4	78.94
CM	**81.72**	**79.73**	**80.73**	**82.06**	**78.73**	80.07	**82.39**	**78.73**	80.52
Best	**81.72**	**79.73**	**80.73**	**82.06**	**78.73**	**80.4**	**82.39**	**78.73**	80.56

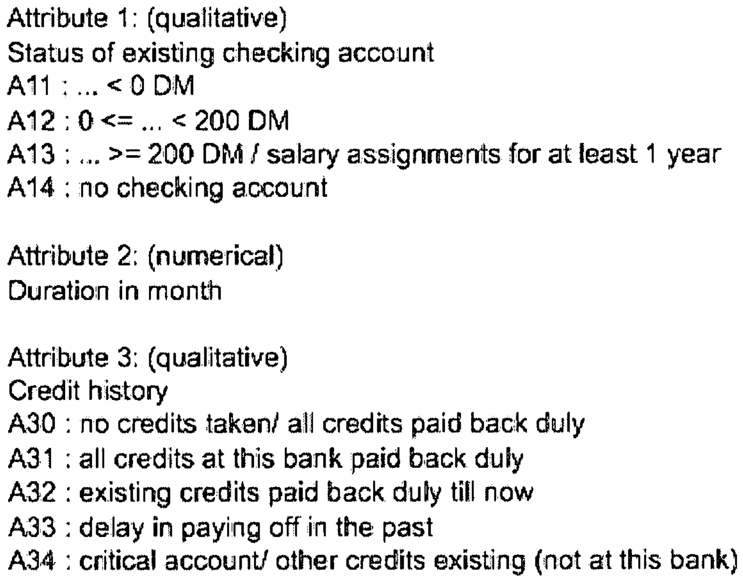
Table 5. Classification accuracy for BP, GP, SVM and CM

We can see that the combined model (CM) did better than other models 6 out of 8 times which is a good result overall, but the improvements in accuracy of the combined model are negligible compared to the BP, GP, and SVM models individually. In addition, the paper notes that due to the fact that the German Credit data set has more good credit customers than bad credit customers, that creates an imbalance that could affect the accuracy of each approach.

CHAPTER 3: MODEL, EXPERIMENTATION, AND RESULTS

3.1 DATA SET OVERVIEW

The German Credit data set is the primary focus of this research and here we explain the details of the data set. This data set is taken from the University of California, Irvine UCI Machine Learning Repository [10]. It was donated in 1994 by Dr. Hans Hoffman from Institut f"ur Statistik und"Okonometrie, Universit"at Hamburg. It contains financial data about individuals applying for credit or loans in Germany. It contains 1000 instances meaning information about 1000 customers or applicants. For each customer or applicant 20 attributes are collected. The data come in two data types: categorical and integer. Figure 1 is a snapshot of some attributes of the two data types:



Attribute 1: (qualitative)
Status of existing checking account
A11 : ... < 0 DM
A12 : 0 <= ... < 200 DM
A13 : ... >= 200 DM / salary assignments for at least 1 year
A14 : no checking account

Attribute 2: (numerical)
Duration in month

Attribute 3: (qualitative)
Credit history
A30 : no credits taken/ all credits paid back duly
A31 : all credits at this bank paid back duly
A32 : existing credits paid back duly till now
A33 : delay in paying off in the past
A34 : critical account/ other credits existing (not at this bank)

Figure 1. Snapshot of German Credit Data Set [1]

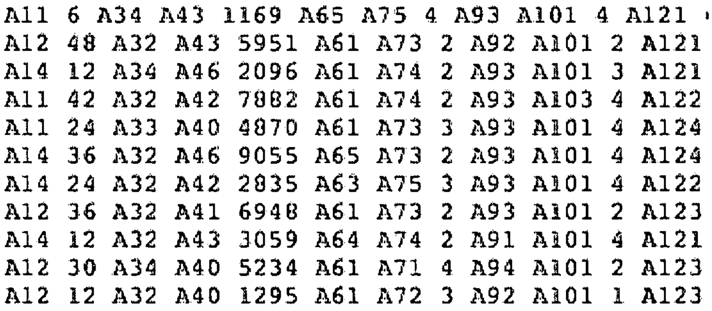
For Attribute 1, the status of existing checking account is a qualitative attribute instead on a numerical one that states the checking account balance. It is divided into different categories: having zero Deutsche Mark DM, having zero to 200 DM, equal or more than 200 DM or no checking account at all. When it comes to attribute 2, it describes the duration of the loan using the number of months as an integer value which is the second type of data in this data set. Attribute 3 is the same type as attribute 1 in terms of the data type but it describes the credit history of the customer in five categories. The data set attributes are:

1. Status of existing checking account (qualitative)
2. Duration in month (numerical)
3. Credit history (qualitative)
4. Purpose (qualitative)
5. Credit amount (numerical)
6. Savings account/bonds (qualitative)
7. Present employment since (qualitative)
8. Installment rate in percentage of disposable income (numerical)
9. Personal status and sex (qualitative)
10. Other debtors / guarantors (qualitative)
11. Present residence since (numerical)
12. Property (qualitative)
13. Age in years (numerical)
14. Other installment plans (qualitative)
15. Housing (qualitative)
16. Number of existing credits at this bank (numerical)
17. Job (qualitative)
18. Number of people being liable to provide maintenance for (numerical)

19. Telephone (qualitative)

20. Foreign worker (qualitative)

The raw data set as shown in Figure 1 is coded in a way that would associate a qualitative attribute to a certain symbol. For example, for attribute 1 if the customer has no checking account, the first column associated with that customer would be A14, which is the corresponding symbol associated with a customer who has no checking account. Figure 2 below shows a snapshot of part of the raw data. Each row of the raw data represents a different applicant or customer while each column represents an attribute from the 20 total attributes in this data set.

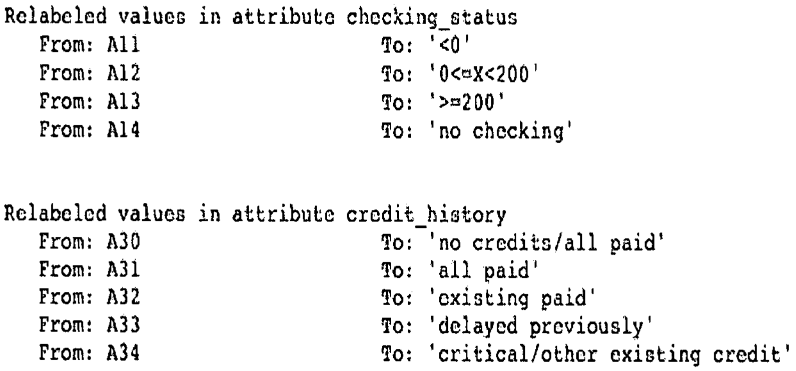


```
A11 6 A34 A43 1169 A65 A75 4 A93 A101 4 A121
A12 48 A32 A43 5951 A61 A73 2 A92 A101 2 A121
A14 12 A34 A46 2096 A61 A74 2 A93 A101 3 A121
A11 42 A32 A42 7882 A61 A74 2 A93 A103 4 A122
A11 24 A33 A40 4870 A61 A73 3 A93 A101 4 A124
A14 36 A32 A46 9055 A65 A73 2 A93 A101 4 A124
A14 24 A32 A42 2835 A63 A75 3 A93 A101 4 A122
A12 36 A32 A41 6948 A61 A73 2 A93 A101 2 A123
A14 12 A32 A43 3059 A64 A74 2 A91 A101 4 A121
A12 30 A34 A40 5234 A61 A71 4 A94 A101 2 A123
A12 12 A32 A40 1295 A61 A72 3 A92 A101 1 A123
```

Figure 2. A section of the raw data in the dataset.

For example, the first row starts with: A11 6 A34 A43. The first attribute is A11 and it means that this customer has a checking account with less than 0 DM. The second attribute is 6 which means that the customer is applying for a loan and the duration of that loan is 6 months. The third attribute is A34, which means that the customer's credit history is "critical/other credits existing". The rest of the attributes follow the same logic.

Since this data set is being used with the data mining tool WEKA, it has to be in a certain format that is compatible with the tool. The format is called Attribute-Relation File Format (ARFF). Thus, the raw data set on the UCI Machine Learning Repository has to be converted to the ARFF format. The Software Environment for the Advancement of Scholarly Research known as (SEASR) has already done the conversion of the data set to the ARFF format [15]. During the conversion process from the raw data to the ARFF format, some values had to be relabeled as shown in Figure 3.

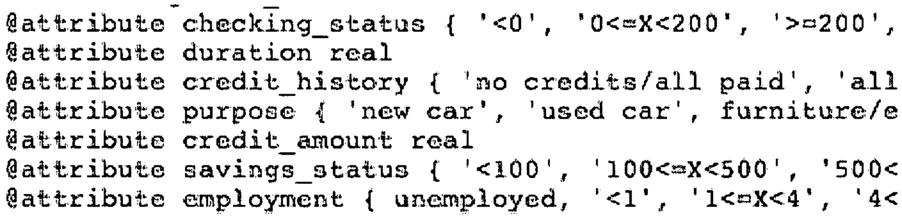


```
Relabeled values in attribute checking_status
   From: A11                       To: '<0'
   From: A12                       To: '0<=X<200'
   From: A13                       To: '>=200'
   From: A14                       To: 'no checking'


Relabeled values in attribute credit_history
   From: A30                       To: 'no credits/all paid'
   From: A31                       To: 'all paid'
   From: A32                       To: 'existing paid'
   From: A33                       To: 'delayed previously'
   From: A34                       To: 'critical/other existing credit'
```

Figure 3. Relabeling some values in the process of conversion.

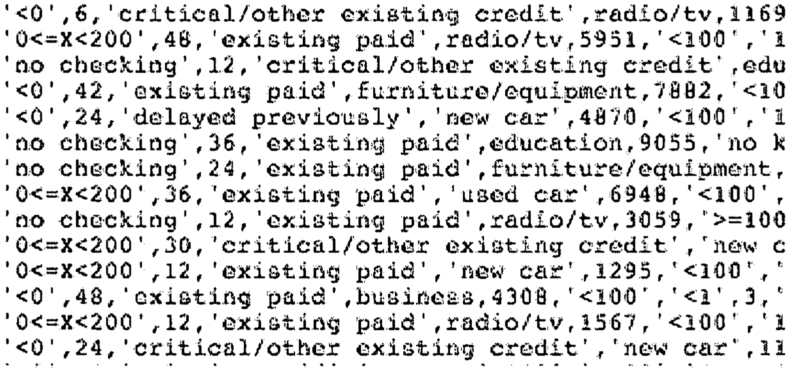
The raw data uses symbols as explained before but here the data is converted to strings. For the first attribute, instead of using A11 symbol to signify that the customer has less than 0 DM in his/her checking account, the ARFF format uses a string '<0' to indicate the same meaning. The same applies to the second attribute; instead of using A31 to indicate that the customer has paid all debt on time, the string 'all paid' is used. In order to do that, the attributes and its possible values has to be defined first as shown in Figure 4.



```
@attribute checking_status { '<0', '0<=X<200', '>=200',
@attribute duration real
@attribute credit_history { 'no credits/all paid', 'all
@attribute purpose { 'new car', 'used car', furniture/e
@attribute credit_amount real
@attribute savings_status { '<100', '100<=X<500', '500<
@attribute employment { unemployed, '<1', '1<=X<4', '4<
```

Figure 4. Defining attributes before converting to strings.

Some values are not shown due to their big size. However, we can see how the values of checking_status are a group of string values separated by commas. The second attribute 'duration' has a real number value. The third attribute is the same as the first one; it has a group of string values and so on. Here is how the raw data in ARFF format looks like in Figure 5. The data is a mixture of string values and real numbers separated by commas. This is the final look of the data set after conversion to ARFF format.

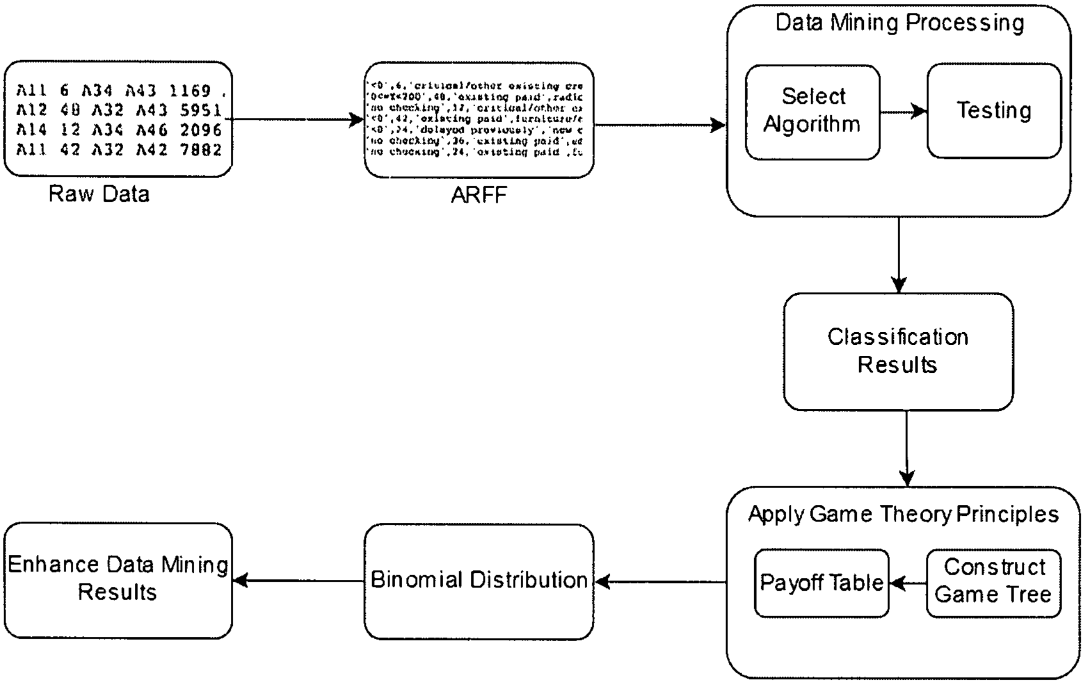


```
'<0',6,'critical/other existing credit',radio/tv,1169
'0<=X<200',48,'existing paid',radio/tv,5951,'<100','1
'no checking',12,'critical/other existing credit',edu
'<0',42,'existing paid',furniture/equipment,7882,'<10
'<0',24,'delayed previously','new car',4870,'<100','1
'no checking',36,'existing paid',education,9055,'no k
'no checking',24,'existing paid',furniture/equipment,
'0<=X<200',36,'existing paid','used car',6948,'<100',
'no checking',12,'existing paid',radio/tv,3059,'>=100
'0<=X<200',30,'critical/other existing credit','new c
'0<=X<200',12,'existing paid','new car',1295,'<100','
'<0',48,'existing paid',business,4308,'<100','<1',3,'
'0<=X<200',12,'existing paid',radio/tv,1567,'<100','1
'<0',24,'critical/other existing credit','new car',11
```

Figure 5. A snapshot of the raw data in ARFF format.

3.2 MODEL DESIGN

Our proposed approach consists of several steps. The first step involves running the data mining tree algorithms. Second, we look at the results to analyze and investigate the findings. Third, we attempt to enhance the results we obtained from the first step by applying game theory principles. Figure 6 below shows a detailed overview of the model design.



A11 6 A34 A43 1169 .
A12 48 A32 A43 5951
A14 12 A34 A46 2096
A11 42 A32 A42 7882
Raw Data
'<0',6,'critical/other existing cre
'0<=X<200',48,'existing paid',radio
'no checking',12,'critical/other ex
'<0',42,'existing paid',furniture/e
'<0',24,'delayed previously','new c
'no checking',36,'existing paid',ed
'no checking',24,'existing paid',fu
ARFF
Data Mining Processing
Select Algorithm
Testing
Classification Results
Apply Game Theory Principles
Payoff Table
Construct Game Tree
Binomial Distribution
Enhance Data Mining Results

Figure 6. A flowchart detailing the proposed model design.

The first step as shown in Figure 6 is to get the data from the dataset in its raw format. After we get the data, we need to prepare it for processing and that involves changing its format to Attribute Relation File Format (ARFF) before loading it into WEKA for processing. After converting the data to the right format and loading it into WEKA, the process of data mining begins at this point. The data mining phase consists of two steps. The first step in the data mining phase is selecting the algorithm. In our model, we selected tree algorithms. The second step in the data mining process is to select the testing condition and we selected cross-validation with 10-folds. After selecting the algorithm and the testing condition, we process the data and record the results. After processing every tree algorithm selected under 10-fold cross-validation, we collect the classification data all together and at this point the data mining phase is concluded. Next, we move to the phase of applying game theory principles which also consists of two steps. The first step in applying the game theory phase is to construct game trees that will capture the potential action for each player at each node. Second, based on the game trees we create the payoff tables to evaluate each event and its corresponding value to each player assigning a weight for each event. At this point, the phase of applying game theory ends and we move to the next step. The next step is to use the information we got from creating the payoff tables in the previous step and apply binomial distribution for further analysis to make sense of the payoff tables. Finally, we get the data from game theory and binomial distribution and apply it to the result of the data mining or the classification results. We notice at the end that applying game theory principles to data mining algorithms could theoretically improve the accuracy of predicting risk associated with extending credit to applicants in this dataset.

3.3 EXPERIMENTATION

The conducted research involves the German Credit data set in its ARFF format appropriate and compatible with WEKA, the data mining tool utilized in this project. The purpose of this research is to analyze the data set and perform non-trivial extraction of valuable information that could assist with predicting a high or low risk customer. Based on this information, financial institutions can make appropriate business decisions to market their financial products or services to the audience with the lowest risk or to assign the interest rate according to the risk level predicted by the model. This way more savings can be achieved as less risk is ensured.

When someone applies for a loan or credit, some information about that individual is collected like: status of existing checking account, duration, credit history, purpose, credit amount, saving account status, etc. We want to know based on the information collected if the applicant is creditworthy and relatively less risky compared to the average risk in that market. By applying some data mining techniques, we hope to be able to extract information that can help us determine some precursors to risk. A number of algorithms will be applied, when applicable, to determine and compare the accuracy of classification results across the various algorithms. We later apply Game Theory principles to further enhance the accuracy of the results by focusing on the incorrectly classified instances.

Here we plan to apply a group of algorithms available on WEKA to make accurate predictions based on the German Credit data set. We start by loading the data in its ARFF format to WEKA. We can have a general visual representation of each attribute in the data set that is easy to understand. We take a look at some of the visual representations. Figure 7 shows the distribution of good and bad credit customers across the four possible categories of the attribute 'checking status'. Good credit customers are represented in blue whereas bad credit customers are represented in red.

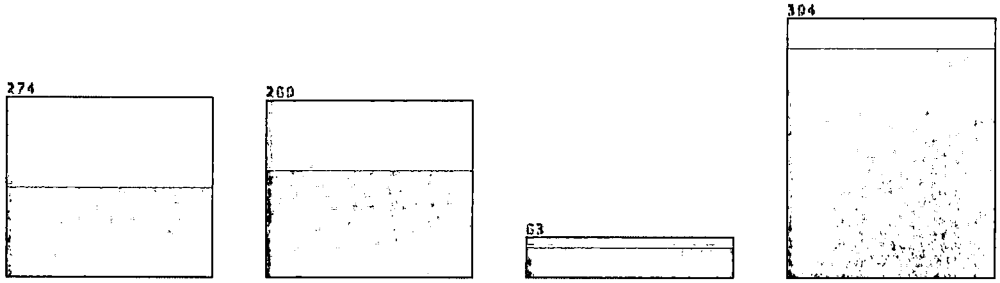


Figure 7. Visualization of 'checking_status' showing the 4 possible categories and the distribution of good (in blue) and bad credit (in red) customers.

Next in Figure 8, we see a visual representation of the second attribute 'duration' which is a real value in the data set and how good credit customers (represented in blue) and bad credit customers (represented in red) are distributed across the recorded loans' duration in the data set. The same visual representation is found on WEKA for each attribute. It helps with visualizing and simplifying the interpretation of a data set, especially a large one.

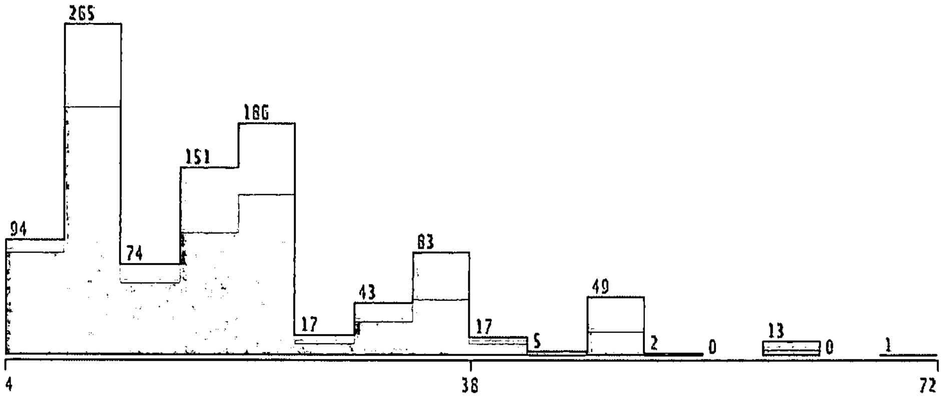


Figure 8. The distribution of good (blue) & bad (red) credit customers across loan's duration in the German Credit data set.

The 'classify' tab at the top lets the user choose the desired specification of classifying a dataset. We plan to use all the applicable classification algorithms available under Trees. We solely focus on using Tree data mining algorithms as opposed to other categories of algorithms such as Bayes and Functions for two main reasons. First, decision trees simplify the decision making process by having binary decisions at every node in the tree. Second, the sequential nature of the decision making process can be easily represented by a tree, and it easily lends itself to game trees, which we will use to enhance accuracy in our combined model. The Trees category has many algorithms but some of them will not be available for use if they are not compatible with the data set or if they cannot be applied to predict a certain data type like class which is qualitative or categorical. Figure 9 is a picture of the Trees list of algorithms where some appear grayed out meaning they cannot be used.

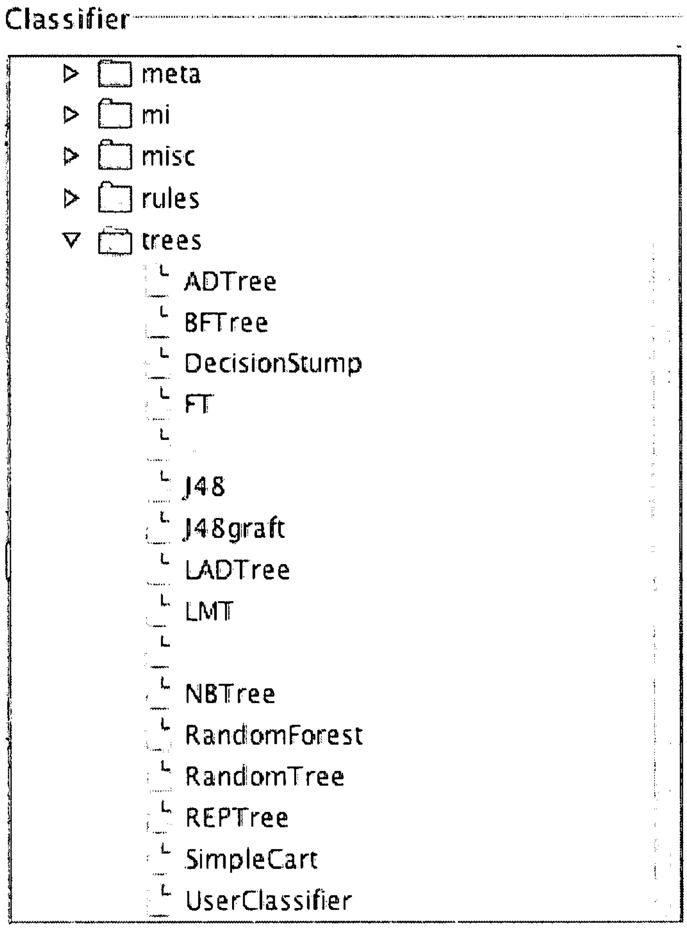


Figure 9. The list of Tree algorithms on WEKA.

The Trees algorithms available are:

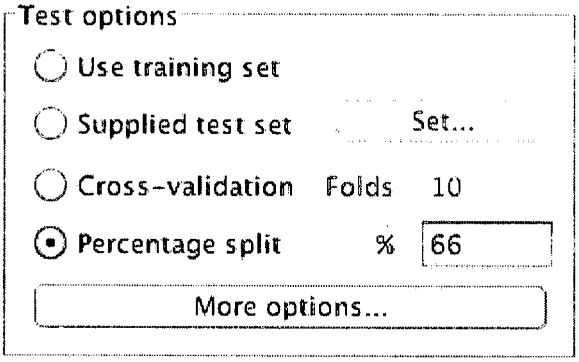
1. J48
2. LADTree
3. J48graft
4. ADTree
5. BFTree
6. DecisionStump
7. FT

8. LMT
9. NBTree
10. RandomForest
11. RandomTree
12. REPTree
13. SimpleCart
14. UserClassifier

The rest cannot be used with this dataset or the class attribute prediction.

For all data mining algorithms available on WEKA, there are a few testing options that users can choose from. Figure 10 shows the default testing options [14] available for these data mining algorithms which include:

1. Training Set
2. Cross-Validation with 10 Folds
3. Percentage Split at 66%



Test options

○ Use training set

○ Supplied test set Set...

○ Cross-validation Folds 10

● Percentage split % 66

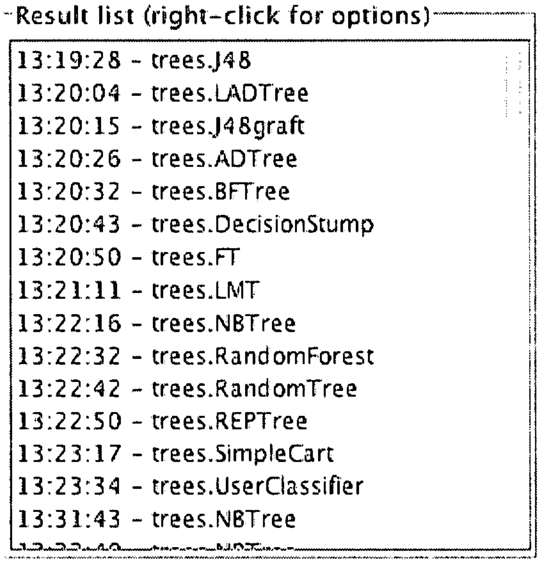
More options...

Figure 10. Testing options as they appear on WEKA.

Although there are quite a few different testing options, we focus on one testing option in our work which is 'cross-validation' with ten folds for comparison sake across all compatible tree algorithms. This means that the dataset is divided into ten equal subsets and one subset is used for testing while the rest of the nine remaining subsets are used for training. This is done ten times with the subset for testing changing every time this is conducted.

3.4 EXPERIMENTATION RESULTS

The application of all previewed tree algorithms under cross-validation with ten folds testing option yielded different results. We review the results in this section. WEKA has a way of saving testing results in a buffer that looks like Figure 11 below.

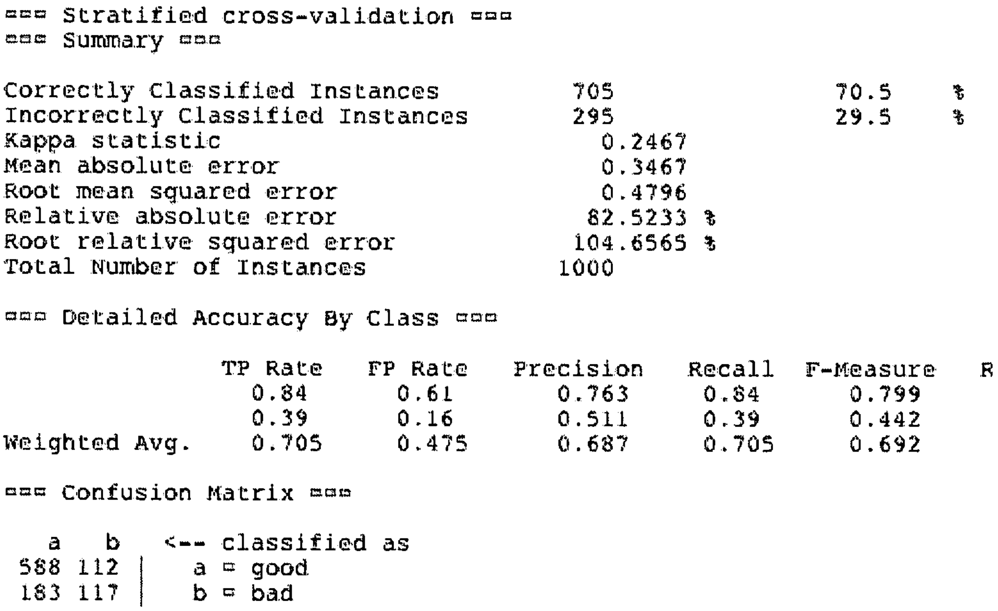


Result list (right-click for options)

```
13:19:28 - trees.J48
13:20:04 - trees.LADTree
13:20:15 - trees.J48graft
13:20:26 - trees.ADTree
13:20:32 - trees.BFTree
13:20:43 - trees.DecisionStump
13:20:50 - trees.FT
13:21:11 - trees.LMT
13:22:16 - trees.NBTree
13:22:32 - trees.RandomForest
13:22:42 - trees.RandomTree
13:22:50 - trees.REPTree
13:23:17 - trees.SimpleCart
13:23:34 - trees.UserClassifier
13:31:43 - trees.NBTree
```

Figure 11. The results buffer in WEKA where a log of testing results is saved.

We begin with the J48 algorithm under the testing option cross-validation with ten folds. Figure 12 below shows a sample of the result. Here we can see that the number of correctly classified instances is 705 out of 1000 and that translates into 70.5%. The number of incorrectly classified instances is 295 out of 1000, which translates into 29.5%. The results also include other measures and a confusion matrix. This sample is a representation of what we get with each unique test, meaning an algorithm and a specific testing option.



```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         705               70.5    %
Incorrectly Classified Instances       295               29.5    %
Kappa statistic                          0.2467
Mean absolute error                      0.3467
Root mean squared error                  0.4796
Relative absolute error                 82.5233 %
Root relative squared error            104.6565 %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   R
                 0.84      0.61       0.763     0.84      0.799
                 0.39      0.16       0.511     0.39      0.442
Weighted Avg.    0.705     0.475      0.687     0.705     0.692

=== Confusion Matrix ===

   a   b   <-- classified as
 588 112 |   a = good
 183 117 |   b = bad
```

Figure 12. A sample of the results after running J48 under cross-validation.

The tree algorithms under the testing condition cross-validation with ten folds using all attributes of the data set all ran as planned except 'UserClassifer' which took too long and the test was cancelled. Table 6 shows the percentage of both correctly and incorrectly classified instances for all tree algorithms under the cross-validation with ten folds testing option. The results show that the tree algorithm with highest classification accuracy is LMT with 75.9% whereas the tree algorithm that performed the worst is LADTree since it has the lowest classification accuracy of 66.7%.

Tree Algorithm	Correctly Classified %	Incorrectly Classified %
LMT	75.9	24.1
NBTree	75.3	24.7
SimpleCart	73.9	26.1
RandomForest	73.6	26.4
BFTree	73	27
ADTree	72.4	27.6
REPTree	71.8	28.2
J48graft	70.7	29.3
J48	70.5	29.5
DecisionStump	70	30
FT	68.3	31.7
RandomTree	67.1	32.9
LADTree	66.7	33.3

Table 6. The results of Tree algorithms under 10-fold cross-validation.

Although we achieve a fairly high classification accuracy by solely using data mining, increasing the accuracy of correctly predicting an applicant's creditworthiness even by a little will help financial institutions reduce risk. Hence, the key question becomes whether or not we can reduce the amount of incorrectly classified applicants. Our approach to answering this question involves applying game theory principles to the incorrectly classified percentage of applicants post data mining.

CHAPTER 4: APPLICATION OF GAME THEORY TO THE CASE STUDY

The results obtained alone by applying Data Mining mechanisms are limited in their ability to predict risk. For example, we can see in Table 6 how the LMT algorithm under cross-validation with ten folds results in 75.9% accuracy in its ability to correctly predict the creditworthiness of a loan applicant. Being able to raise this percentage even by a small number will help financial institutions reduce cost associated with loan risk and raise their revenues. Tree algorithms build decision trees in order to make predictions. Predicting whether a loan applicant should be approved for a loan or not is just a means for automating the bank's strategy for making a decision. Thus, decision trees are very appropriate for combining with game theory.

We plan to integrate Game Theory rules to improve the accuracy of the results. We have seen examples where applying Game Theory rules did in fact result in positive results leading many researchers to believe that there is potential in combining Game Theory and Data Mining mechanisms. Most of the research conducted in this area indicated an interest in further exploring the field and Game Theory's integration with Data Mining applications. Bruce [16] looked at integrating Game Theory and Data Mining. The author proposes a game theory model that is strategic and competitive to group spectral bands when exploiting hyperspectral imagery. The proposed model utilizes conflict data filtering and the Nash equilibrium in a conflict situation to maximize payoff and reach a steady state solution to the band grouping problem. This proposed model is used as a part of a multi-classifier decision fusion (MCDF). The paper presents experimental results demonstrating how the application of the game theoretic approach provides better results and is superior to the comparison models [16].

Another study applying Game Theory investigated large-scale decision models with many invested individuals. They proposed a model using a Bayesian belief aggregation to deal with decision problems. This proposal is unique in a way that ensures divergence in beliefs instead of the traditional method that averages beliefs to form a general consensus. This approach makes it possible to apply game theory by enabling the competitive aspect to emerge in a cooperative situation. By using this model, the application of game theory is more realistic since it takes into consideration parties with opposing goals compared to the assumption implied by the traditional method that focuses on creating consensus. This model is applied to data in stem cell research. It has potential for defining and analyzing policy and how it affects individuals. Future work looks to apply this approach to strategic games involving multiple decisions made by multiple parties [15].

4.1 GAME TREE MODELS

Game theory focuses on situations involving multiple parties with competing interests. It is the mathematical or logical analysis of conflict and cooperation. A game in game theory is defined to consist of at least two players, a set of strategies available for each player, and a preference relation over possible outcomes. The strategies are the particular actions available to each player. The outcome is determined by the mutual choice of strategies, and each outcome has an associated payoff for each player for that particular combination of actions.

When a customer applies for credit, we can treat the situation as a competition between two players. These two players are the customer and the bank. Likewise, this also applies to the process of appealing a denial of credit decision. This situation can be presented using the simplest tree design which is based on the concept of a decision tree. Here we focus solely on the incorrectly classified applicants resulting from running the tree-based data mining algorithms since we already know the creditworthiness of the correctly classified applicants. The following tree shows how the two players will interact in one simple scenario when a customer appeals for reconsideration. The applicant or the customer is first presented with two options. The first option is to appeal for reconsideration and the second is to not appeal. If the applicant chooses to not appeal, then the process ends there. When the customer appeals for reconsideration, the bank then has two options. The bank can either approve the applicant's request for reconsideration and extend credit or decline the appeal. If the bank decides to reconsider, then the bank faces two more possibilities. The first possibility is reconsidering an applicant who is low risk and the second possibility is reconsidering an applicant who is high risk. The low risk applicant will be someone who was incorrectly classified as having bad credit or misinterpreted as being high risk whereas the high risk applicant will be someone who was incorrectly classified as having good credit or misinterpreted as being low risk. In this simple scenario, the process ends at this point as shown in Figure 13.

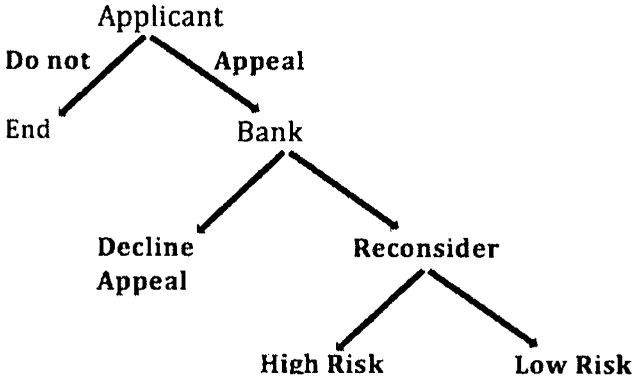


Figure 13. Simple game tree model.

This process is too simplistic and it should be expanded to emulate a real life process. After the applicant submits an application appealing for reconsideration and if the bank decides to accept the appeal, the bank has two options for each possibility. For a low risk applicant, to reconsider that constitutes a success and denial constitutes a failure for the bank because a low risk applicant will most likely pay back the loan. In other words, the bank gains or succeeds when a low risk applicant is reconsidered and extended credit while the bank loses or fails when a low risk applicant is denied reconsideration and therefore, not extended credit. For a high risk applicant, to reconsider that constitutes a failure and denial constitutes a success for the bank because a high risk applicant has a greater chance of defaulting on a loan. In other words, the bank loses or fails when a high risk applicant is reconsidered and extended credit while the bank gains or succeeds when a high risk applicant is denied reconsideration and not extended credit. The following enhanced tree in Figure 14 shows the detailed process as a game tree.

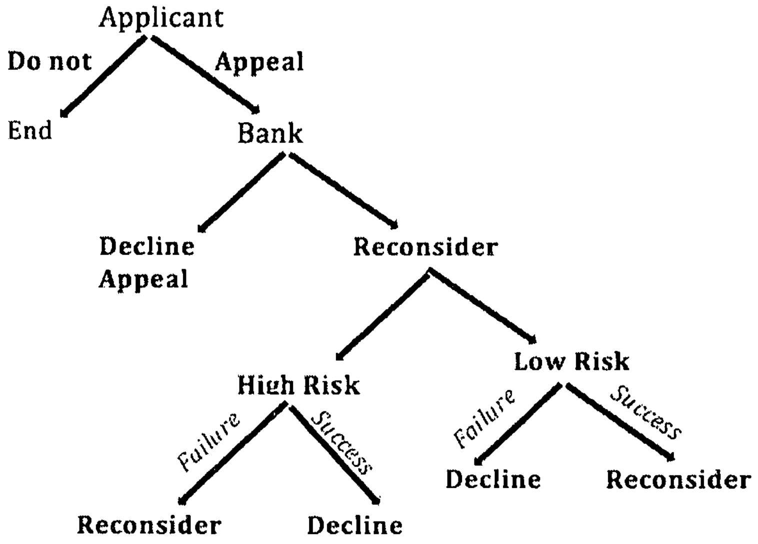


Figure 14. Enhanced Game Tree.

4.2 USE CASES

To better understand each model and capture potential shortcomings, creating a use case can be helpful. The following are use cases for each model.

4.2.1 USE CASE 1

Use Case Element	Description
Use Case Number	1
Use Case Name	Customer appealing for reconsideration
Use Case Description	This use case explains the process of appealing for

	reconsideration after being denied credit in its simplest form. This specifically describes the first game tree introduced earlier. The customer appeals for reconsideration and fills out an application. The bank then processes the application and either approves to reconsider or declines to reconsider the application. The bank then deals with two possibilities: the customer being low risk or high risk.
Primary Actor	Customer
Precondition	None
Trigger	When the customer submits an application appealing for reconsideration.
Basic Flow	1- Customer appeals for reconsideration and fills out an application. 2- The bank then reviews the application. 3- The bank approves the application to reconsider extending credit to the customer. 4- Customer accepts the credit, signs an agreement and receives credit. 5- Customer makes periodic payment until the full loan is repaid
Alternate Flows	3a- The bank denies the application of the customer.

	4a- The customer does not like the offer and decides to withdraw the application.

Table 7. Use Case 1.

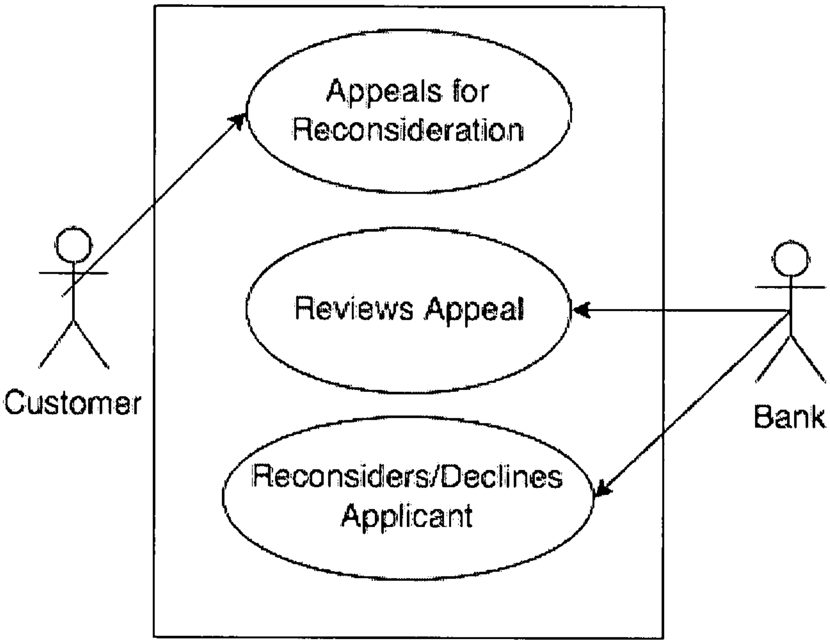


Figure 15. Use Case 1 Diagram.

In Use Case 1 Figure 15, the simple form is represented and we can see that the process is missing some steps specifically at step number three where the bank makes a decision to approve or deny an application for reconsideration but the rest of possibilities are not addressed. This is where the need to expand the model arises and more options for the bank are introduced to deal with the range of risk involved in a typical credit application.

4.2.2 USE CASE 2

Use Case Element	Description
Use Case Number	2
Use Case Name	Customer applying for credit (Enhanced Model)
Use Case Description	This use case explains the process of appealing for reconsideration after being denied credit in a more advanced form. This use case describes the enhanced game tree.. The customer appeals for reconsideration and fills out an application. The bank then processes the application. After the bank processes the application, the bank approves or denies the application to reconsider the applicant. Once approved for reconsideration, the bank can either end up lending money to a low risk applicant or a high risk applicant. Reconsidering a high risk applicant constitutes a failure and declining to reconsider a high risk applicant constitutes a success. Also, reconsidering a low risk applicant constitutes a success and declining to reconsider a low risk applicant constitutes a failure.
Primary Actor	Customer
Precondition	None
Trigger	When the customer submits an application appealing for reconsideration.

Basic Flow	1- Customer appeals for reconsideration and fills out an application. 2- The bank then reviews the application. 3- The bank approves the application to reconsider the applicant. 4- The bank then approves the low risk applicant or declines to reconsider a high risk applicant. 5- Customer makes periodic payment until the full loan is repaid.
Alternate Flows	3a- The bank denies the application for reconsideration of the customer. 4a- The bank approves the high risk applicant or denies the low risk applicant. 5a- The customer does not make periodic payments and eventually defaults on the loan.

Table 8. Use Case 2.

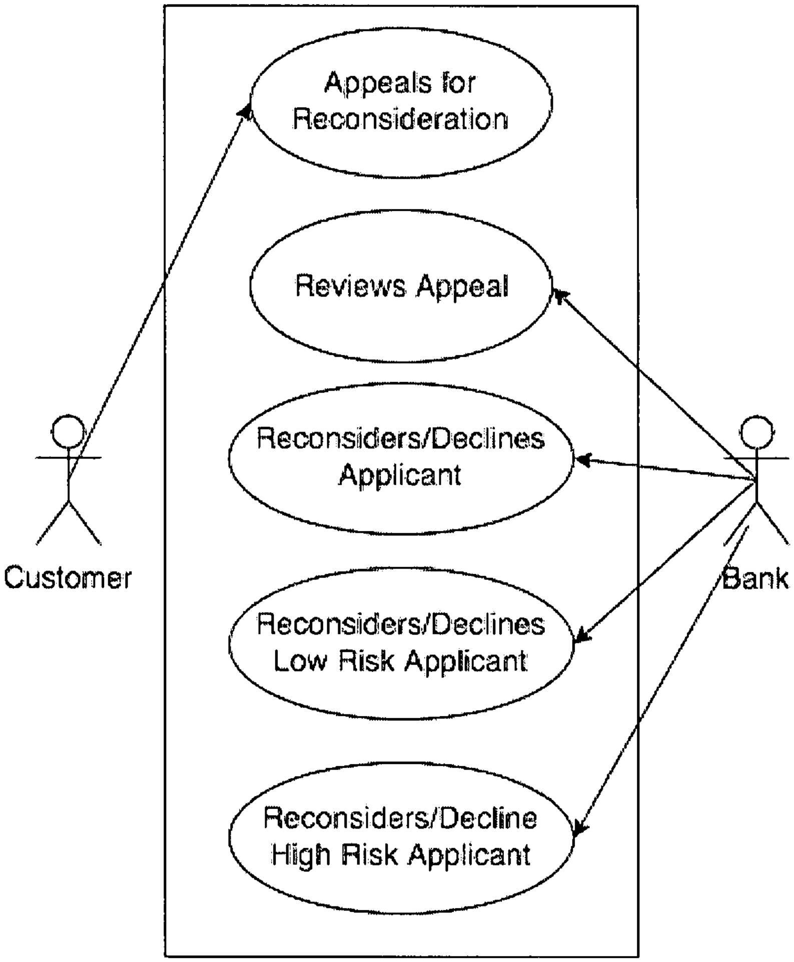

Appeals for Reconsideration
Reviews Appeal
Reconsiders/Declines Applicant
Reconsiders/Declines Low Risk Applicant
Reconsiders/Decline High Risk Applicant
Customer
Bank

Figure 16. Use Case 2 Diagram.

In Use Case 2 Figure 16, more options are introduced to deal with the potential of extending credit to the wrong applicant. The bank is faced with two scenarios for each option. If the bank decides to reconsider a low risk applicant, then that would be a success and if the bank decides to decline reconsidering a low risk applicant, that would be regarded as a failure. Alternatively, if the bank decides to reconsider a high risk applicant, that would be a failure and if the bank decides to decline reconsidering a high risk applicant, that would be considered a success.

4.3 DEFINE FORMULA

After proposing the game tree models, we need to create the appropriate corresponding game theory formula. Since we are using the strategic form of game theory, its basic form is shown in Equation (3) [18]:

$$\Gamma = (N, C_i, u_i), \forall i \in N \tag{3}$$

Where:

N is the finite set of players

C_i is the ensemble of strategies available to player i

$\mathbf{C} = \mathbf{X}_{j \in N}$ C_i: the set of possible strategy profiles

and u_i: $\mathbf{C} \rightarrow \mathbb{R}$ is the utility of player i

In each of the two models, the set of players N should remain the same where $N =$ {Applicant, Bank} since that does not change. C_i, which is the ensemble of strategies available to each player, will be different for each model. The utility, or payoff, for each player is defined in Section 4.4.1.

4.3.1 MODEL 1

For the first simple game tree, the set of strategies C_i for the applicant is $C_{Applicant} =$ {Appeal, Do not} and C_i for the bank is $C_{Bank} =$ {Decline Appeal, Reconsider}. Thus, there are two strategies available to the applicant: to appeal for reconsideration and to choose not to appeal. For the bank, the strategies available are either to decline the appeal or to reconsider the applicant.

4.3.2 MODEL 2

Model 2 expands upon and is more enhanced than Model 1. Therefore, the strategies available to each player are more complex. The set of strategies C_i for the applicant is $C_{Applicant}$= {Appeal, Do not} and C_i for the bank is C_{Bank}= {Decline Appeal, Reconsider, Reconsider High Risk, Decline High Risk, Reconsider Low Risk, Decline Low Risk}. This means that the applicant has two strategies: appeal and do not appeal. For the bank, it means that the bank has six strategies: decline appeal to reconsider, reconsider application, reconsider a high risk applicant, decline to reconsider a high risk applicant, reconsider a low risk applicant, and decline to reconsider a low risk applicant. Model 2 is more complex as compared to Model 1. The players have more strategies to choose from and the decisions needed to reach the end results increase as a result.

4.4 ENHANCEMENT TO CLASSIFICATION RESULTS

The percentage of incorrectly classified applicants from the tree-based data mining results can be reduced by applying game theory principles. We can derive the associated payoff table from the enhanced game tree of Model 2. From this payoff table, we can apply the binomial distribution calculation to determine the probability of success. This probability becomes the percentage of enhancement to the tree-based data mining algorithms, thus increasing the accuracy of classification and correcting some of the error from the incorrectly classified applicants.

4.4.1 PAYOFF TABLES

Payoff tables are used to assist with decision analysis. They help a decision maker evaluate each course of action but first the decision maker has to assign a certain value for each possible event [21]. By assigning a certain value for each possible event, the decision maker can make a more informed decision as to what would be the best course of action to take. A payoff is associated with each player for every possible combination of actions. We focus on the more realistic scenario of the enhanced game tree (Model 2) to enhance the data mining classification results using game theory principles. The corresponding payoff table displayed in Table 9 can be derived from the enhanced game tree. The payoff table shows the generalization of whether each decision made under different circumstances constitutes a success or a failure for the bank. For our purposes the conditions are either high risk or low risk. Low risk means that the applicant is more likely to pay off the loan and high risk means that the applicant is more likely to default on paying the entire loan or a significant portion of the loan. This generalization of payoffs derived from the enhanced game tree model justifies the binomial distribution probability explained in Section 4.4.2.

	0: High Risk	1: Low Risk
Decline	Success	Failure
Reconsider	Failure	Success

Table 9. Generalization of payoffs derived from Model 2.

The payoff table for both the bank and the applicant based on the enhanced game tree of Model 2 is illustrated in Table 10. The payoffs are represented in the format of X,Y where X represents the payoff for the bank and Y represents the payoff for the applicant for a particular combination of actions. For example, the payoff of 2,0 means that the bank has a payoff of 2 when it declines a high risk applicant while the high risk applicant has a payoff of 0 if he/she is declined by the bank. The payoffs from Table 10 verify the generalization depicted in Table 9. The bank receives the highest payoff when declining a high risk applicant and when reconsidering a low risk applicant, both of which can be considered successes for the bank. On the other hand, the bank receives the lowest payoff when reconsidering a high risk applicant and when declining a low risk applicant, both of which are considered failures for the bank. The combination of actions that is most beneficial to both players is when the bank reconsiders and the applicant is low risk.

		Applicant	
		High Risk	Low Risk
Bank	Decline	2,0	0,1
	Reconsider	0,1	2,1

Table 10. Model 2 Payoff Table.

4.4.2 BINOMIAL DISTRIBUTION

The binomial distribution is appropriate to use in situations where there are two mutually exclusive outcomes and both outcomes are either a success or a failure. It is also common to use the binomial distribution when information is minimal as in our case since we only know the percentage of incorrectly classified applicants but no other information associated with these instances. The binomial distribution is used to get the probability of observing a number of successes in a number of independent trials. It calculates a probability of success for each trial and that probability does not change with each trial [22]. The formula for binomial distribution is the following:

$$P(r) = \frac{n!}{r!\,(n-r)!} p^r (1-p)^{n-r} \qquad (4)$$

Where:

n = number of events

r = number of successful events

p = probability of success

Based on the payoff table from Table 9, we see that there are a total of four possible events: decline a high risk applicant, decline a low risk applicant, reconsider a high risk applicant, and reconsider a low risk applicant. Out of the four possible events, two of them are successful events, i.e. decline a high risk applicant and reconsider a low risk applicant. So, the probability of success is $p = r / n = 2 / 4 = 0.5$. Thus, the values for the variables of the binomial distribution based on Table 9 are n = 4 events, r = 2 successful events, and p = 0.5.

We substitute the values into Equation (4) to determine the probability of observing successes and find that $P(r)$ = 0.375. This means that all tree-based data mining algorithms can be enhanced by 37.5%. For example, the 24.1%, 28.2%, and 33.3% of misclassified applicants using LMT, REP Tree, and LAD Tree can be enhanced by 37.5% of the 24.1%, 28.2%, 33.3%, respectively. Therefore, regardless of the error percentage of the tree algorithm, the enhancement will always be constant at 37.5%. We can determine the percentage of improvement after applying game theory techniques and the overall improvement to the accuracy from our proposed combined model integrating data mining and game theory principles as seen in Table 11. We can see the potential improvement obtained by applying game theory principles to tree algorithms. Combining game theory principles and tree data mining algorithms can considerably increase and improve the accuracy results compared to just using tree data mining algorithms alone.

Tree Algorithm	Correctly Classified %	Incorrectly Classified %	Game Theory % of Improvement	Tree Algorithm & Game Theory Improvement (%)
LMT	75.9	24.1	15.0625	90.9625
NBTree	75.3	24.7	15.4375	90.7375
SimpleCart	73.9	26.1	16.3125	90.2125
RandomForest	73.6	26.4	16.5	90.1
BFTree	73	27	16.875	89.875
ADTree	72.4	27.6	17.25	89.65
REPTree	71.8	28.2	17.625	89.425
J48graft	70.7	29.3	18.3125	89.0125
J48	70.5	29.5	18.4375	88.9375
DecisionStump	70	30	18.75	88.75
FT	68.3	31.7	19.8125	88.1125
RandomTree	67.1	32.9	20.5625	87.6625
LADTree	66.7	33.3	20.8125	87.5125

Table 11. Improvement of Results after applying Game Theory.

Visually, we can see the comparison of the correctly classified applicants (%) from data mining alone compared to the accuracy improvement after combining data mining and game theory for each tree-based algorithm in Figure 17. Because of the constant nature of the enhancement of 37.5% from the binomial distribution, we can see that the overall accuracy enhancement for even the worst performing tree algorithm of LAD Tree is now almost comparable to the best performing tree algorithm of LMT. Therefore, accuracy in predicting an applicant's creditworthiness can be increased and risk can be greatly reduced for financial institutions using our proposed combined model integrating both data mining and game theory.

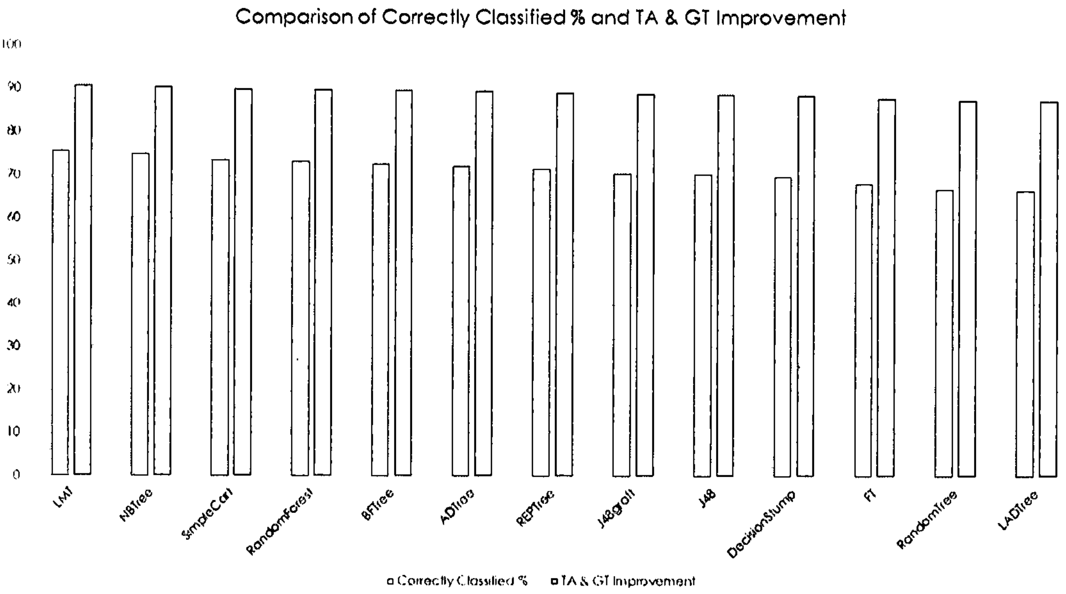


Figure 17. Comparison of Correctly Classified % and TA & GT Improvement.

CHAPTER 5: CONCLUSION AND FUTURE WORK

Applying Game Theory principles to the traditional Data Mining mechanisms shows a promising potential in enhancing risk prediction accuracy. The results of the data mining algorithms applied to the German Credit Dataset can be improved by using Game Theory which is suitable for the scenario and the dataset. The situation is similar to traditional scenarios of Game Theory where the strategic form is applied to predict the outcome. Combining the principles of Game Theory in its strategic form with tree Data Mining algorithms in practice show that it can have a positive effect on decision making algorithms to minimize risk that can further be expanded to other applications or other datasets with different nature of risk.

There is a need to further expand on the application of Game Theory to construct a more practical model. The intention of this research will be to find possible models for rule-based machine learning algorithms, using game theory to enhance the learning process or decision-making. The model will be combined with the classifier itself automatically improving the performance and eliminating the need of the user to manually select or deselect attributes. Such practical combined models based on the principles of game theory combined with decision making mechanisms will be applicable in real life business decisions.

Enhancing the game trees went through many phases and each phase reveals a more complex process. While the basic game tree was functioning, it did not address all of the issues that may arise. The enhanced model addresses all potential courses of action that can happen when a customer appeals for credit. We see that enhancing the game tree model presents us with a more practical model that we can use to enhance the accuracy of predicting risk in credit applications when combined with decision trees. The feedback from running the decision trees on the dataset and the output of the enhanced game tree presents a potential increased level of accuracy for the decision support system when they are combined in this proposed model. Related previous work with similar approaches attempted to combine game theory with a fault detection algorithm showing some elementary improvements in practice [9]. Looking at another similar example, the authors encourage further exploration and suggest making changes to the dataset to make it more game dependent [6]. Generally, every approach was combined with a specific algorithm but none of them was a decision tree which is the focus of our approach. Future work can look into expanding the model to be applied for different datasets of the same nature that are current and larger. There is potential for improvements in the area of expanding the application of the combined model to more current datasets used today in financial institutions. Also, further experimentation and testing of our theoretical improvement is an area of potential future work.

REFERENCES

[1] Li, K.W.; Karray, F.; Hipel, K.W.; Kilgour, D.M., "Fuzzy approaches to the game of Chicken," Fuzzy Systems, IEEE Transactions on , vol.9, no.4, pp.608,623, Aug 2001 doi: 10.1109/91.940972

[2] Lixin Chen; Xin Wang, "The Game Model in Competitive Decision-making," E-Business and E-Government (ICEE), 2010 International Conference on , vol., no., pp.5300,5303, 7-9 May 2010 doi: 10.1109/ICEE.2010.1327

[3] Wei-Cheng Liao; Chi-Yen Yin; Chiang, J.K., "Decision making model on strategic technology investment using game theory," Industrial Engineering and Engineering Management, 2009. IEEM 2009. IEEE International Conference on , vol., no., pp.813,817, 8-11 Dec. 2009

[4] Yu Haidong; Tian Qihua; Zou Ying, "Game analysis with incomplete information and selection of service providers in financial competitive intelligence activity," Computer Design and Applications (ICCDA), 2010 International Conference on , vol.2, no., pp.V2-461,V2-465, 25-27 June 2010

[5] G. Boetticher, " Teaching Financial Data Mining using Stocks and Futures Contracts " , Journal of Systemic, Cybernetics and Informatics , Vol 3, no 3, p.26-32, 2006.

[6] Ganesh Ganesh, M.; Sunke, A.; Ganesh, S.; Avinesh, D., "KDGT: Knowledge Discovery in Game Theory," Knowledge Discovery and Data Mining, 2009. WKDD 2009. Second International Workshop on , vol., no., pp.261,264, 23-25 Jan. 2009

[7] Cui Dong-hong; Zhang Xi-yan, "Application of game theory on bidding price decision," Industrial Engineering and Engineering Management, 2009. IE&EM '09. 16th

International Conference on , vol., no., pp.58,61, 21-23 Oct. 2009

[8] Castillo, L.; Dorao, C.A., "Decision-Making on Liquefied Natural Gas (LNG) projects using game theory," Computational Intelligence in Multicriteria Decision-Making (MDCM), 2011 IEEE Symposium on , vol., no., pp.60,66, 11-15 April 2011

[9] Pan-Jing Li; Xian-Sheng Qin; Adjallah, K.H.; Eynard, Benoit; Jay Lee, "Cooperative Decision Making for Diagnosis of Complex System based on Game Theory: Survey and an Alternative Scheme," Industrial Informatics, 2006 IEEE International Conference on , vol., no., pp.725,730, 16-18 Aug. 2006 doi: 10.1109/INDIN.2006.275651

[10] H. Hofmann. (1994, November 11). Statlog (German Credit Data) Data Set [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29

[11] Huang L., Chen C., and Wang J., "Credit Scoring with a Data Mining Approach Based on Support Vector Machines," Computer Journal of Expert Systems with Applications , vol. 33, no. 4, pp. 847-856, 2007.

[12] Paul O' Dea and Josephine Griffith and Colm O' Riordan. "Combining Feature Selection and Neural Networks for Solving Classification Problems". Information Technology Department, National University of Ireland.

[13] Defu Zhang; Hongyi Huang; Qingshan Chen; Yi Jiang, "A Comparison Study of Credit Scoring Models," Natural Computation, 2007. ICNC 2007. Third International Conference on , vol.1, no., pp.15,18, 24-27 Aug. 2007

[14] I. H. Witten and E. Frank, Data Mining: Practical machine learning tools and techniques, 2nd Edition . San Francisco: Morgan Kaufmann, 2005

[15] The Software Environment for the Advancement of Scholarly Research known as (SEASR) [Online]. Available: http://repository.seasr.org/Datasets/UCI/arff/credit-g.arff

[16] Bruce, L.M., "Game theory applied to big data analytics in geosciences and remote sensing," Geoscience and Remote Sensing Symposium (IGARSS), 2013 IEEE International , vol., no., pp.4094,4097, 21-26 July 2013 doi: 10.1109/IGARSS.2013.6723733

[17] Greene, K.A.; Kniss, J.M.; Luger, G.F.; Stern, C.R., "Satisficing the Masses: Applying Game Theory to Large-Scale, Democratic Decision Problems," Computational Science and Engineering, 2009. CSE '09. International Conference on , vol.4, no., pp.1156,1162, 29-31 Aug. 2009

[18] Tomassini, M , Introduction to Evolutionary Game Theory. Retrieved December , 2013 Available: http://dl.acm.org/citation.cfm%3Fid=2480808%26dl=ACM%26coll=DL%26CFID=324928133%26CFTOKEN=95390852

[19] Alkheliwi, T.; Jim, C.; Lateef, K.; Penn, S.; Salem, A., "Applying game theory rules to enhance decision support systems in credit and financial applications," Computer Games: AI, Animation, Mobile, Multimedia, Educational and Serious Games (CGAMES), 2014 , vol., no., pp.1-10, 28-30 July 2014

[20] Consumers Report (2014, October), How to get the best auto loan [Online]. Available: http://www.consumerreports.org/cro/2012/12/how-to-get-the-best-car-loan/index.htm

[21] D. Levine, D. Stephan, T. Krehbiel, M. Berenson, A. De, D. Mukherjee, S. Yagan and D. Levine, *Statistics for managers using Microsoft Excel.*

[22] "Binomial Distribution", *NIST*, 2016. [Online]. Available: http://www.itl.nist.gov/div898/handbook/eda/section3/eda366i.htm. [Accessed: 21- Feb-2016].

APPENDIX 1: ORIGINAL RAW DATA

This is a sample from the original raw data before being converted to the appropriate format required to perform the analysis.

```
A11 6 A34 A43 1169 A65 A75 4 A93 A101 4 A121 67 A143 A152 2 A173 1 A192 A201 1 A12 48 A32
A43 5951 A61 A73 2 A92 A101 2 A121 22 A143 A152 1 A173 1 A191 A201 2 A14 12 A34 A46 2096
A61 A74 2 A93 A101 3 A121 49 A143 A152 1 A172 2 A191 A201 1 A11 42 A32 A42 7882 A61 A74 2
A93 A103 4 A122 45 A143 A153 1 A173 2 A191 A201 1 A11 24 A33 A40 4870 A61 A73 3 A93 A101 4
A124 53 A143 A153 2 A173 2 A191 A201 2 A14 36 A32 A46 9055 A65 A73 2 A93 A101 4 A124 35
A143 A153 1 A172 2 A192 A201 1 A14 24 A32 A42 2835 A63 A75 3 A93 A101 4 A122 53 A143 A152 1
A173 1 A191 A201 1 A12 36 A32 A41 6948 A61 A73 2 A93 A101 2 A123 35 A143 A151 1 A174 1 A192
A201 1 A14 12 A32 A43 3059 A64 A74 2 A91 A101 4 A121 61 A143 A152 1 A172 1 A191 A201 1 A12
30 A34 A40 5234 A61 A71 4 A94 A101 2 A123 28 A143 A152 2 A174 1 A191 A201 2 A12 12 A32 A40
1295 A61 A72 3 A92 A101 1 A123 25 A143 A151 1 A173 1 A191 A201 2 A11 48 A32 A49 4308 A61
A72 3 A92 A101 4 A122 24 A143 A151 1 A173 1 A191 A201 2 A12 12 A32 A43 1567 A61 A73 1 A92
A101 1 A123 22 A143 A152 1 A173 1 A192 A201 1 A11 24 A34 A40 1199 A61 A75 4 A93 A101 4 A123
60 A143 A152 2 A172 1 A191 A201 2 A11 15 A32 A40 1403 A61 A73 2 A92 A101 4 A123 28 A143
A151 1 A173 1 A191 A201 1 A11 24 A32 A43 1282 A62 A73 4 A92 A101 2 A123 32 A143 A152 1 A172
1 A191 A201 2 A14 24 A34 A43 2424 A65 A75 4 A93 A101 4 A122 53 A143 A152 2 A173 1 A191 A201
1 A11 30 A30 A49 8072 A65 A72 2 A93 A101 3 A123 25 A141 A152 3 A173 1 A191 A201 1 A12 24
A32 A41 12579 A61 A75 4 A92 A101 2 A124 44 A143 A153 1 A174 1 A192 A201 2 A14 24 A32 A43
3430 A63 A75 3 A93 A101 2 A123 31 A143 A152 1 A173 2 A192 A201 1 A14 9 A34 A40 2134 A61 A73
4 A93 A101 4 A123 48 A143 A152 3 A173 1 A192 A201 1 A11 6 A32 A43 2647 A63 A73 2 A93 A101 3
A121 44 A143 A151 1 A173 2 A191 A201 1 A11 10 A34 A40 2241 A61 A72 1 A93 A101 3 A121 48
A143 A151 2 A172 2 A191 A202 1 A12 12 A34 A41 1804 A62 A72 3 A93 A101 4 A122 44 A143 A152 1
A173 1 A191 A201 1 A14 10 A34 A42 2069 A65 A73 2 A94 A101 1 A123 26 A143 A152 2 A173 1 A191
A202 1 A11 6 A32 A42 1374 A61 A73 1 A93 A101 2 A121 36 A141 A152 1 A172 1 A192 A201 1 A14 6
A30 A43 426 A61 A75 4 A94 A101 4 A123 39 A143 A152 1 A172 1 A191 A201 1 A13 12 A31 A43 409
```

A64 A73 3 A92 A101 3 A121 42 A143 A151 2 A173 1 A191 A201 1 A12 7 A32 A43 2415 A61 A73 3 A93 A103 2 A121 34 A143 A152 1 A173 1 A191 A201 1 A11 60 A33 A49 6836 A61 A75 3 A93 A101 4 A124 63 A143 A152 2 A173 1 A192 A201 2 A12 18 A32 A49 1913 A64 A72 3 A94 A101 3 A121 36 A141 A152 1 A173 1 A192 A201 1 A11 24 A32 A42 4020 A61 A73 2 A93 A101 2 A123 27 A142 A152 1 A173 1 A191 A201 1 A12 18 A32 A40 5866 A62 A73 2 A93 A101 2 A123 30 A143 A152 2 A173 1 A192 A201 1 A14 12 A34 A49 1264 A65 A75 4 A93 A101 4 A124 57 A143 A151 1 A172 1 A191 A201 1 A13 12 A32 A42 1474 A61 A72 4 A92 A101 1 A122 33 A141 A152 1 A174 1 A192 A201 1 A12 45 A34 A43 4746 A61 A72 4 A93 A101 2 A122 25 A143 A152 2 A172 1 A191 A201 2 A14 48 A34 A46 6110 A61 A73 1 A93 A101 3 A124 31 A141 A153 1 A173 1 A192 A201 1 A13 18 A32 A43 2100 A61 A73 4 A93 A102 2 A121 37 A142 A152 1 A173 1 A191 A201 2 A13 10 A32 A44 1225 A61 A73 2 A93 A101 2 A123 37 A143 A152 1 A173 1 A192 A201 1 A12 9 A32 A43 458 A61 A73 4 A93 A101 3 A121 24 A143 A152 1 A173 1 A191 A201 1 A14 30 A32 A43 2333 A63 A75 4 A93 A101 2 A123 30 A141 A152 1 A174 1 A191 A201 1 A12 12 A32 A43 1158 A63 A73 3 A91 A101 1 A123 26 A143 A152 1 A173 1 A192 A201 1 A12 18 A33 A45 6204 A61 A73 2 A93 A101 4 A121 44 A143 A152 1 A172 2 A192 A201 1 A11 30 A34 A41 6187 A62 A74 1 A94 A101 4 A123 24 A143 A151 2 A173 1 A191 A201 1 A11 48 A34 A41 6143 A61 A75 4 A92 A101 4 A124 58 A142 A153

Attribute Information:

Attribute 1: (qualitative)

Status of existing checking account

A11 : ... < 0 DM

A12 : 0 <= ... < 200 DM

A13 : ... >= 200 DM / salary assignments for at least 1 year

A14 : no checking account

Attribute 2: (numerical)

Duration in month

Attribute 3: (qualitative)

Credit history

A30 : no credits taken/ all credits paid back duly

A31 : all credits at this bank paid back duly

A32 : existing credits paid back duly till now

A33 : delay in paying off in the past

A34 : critical account/ other credits existing (not at this bank)

Attribute 4: (qualitative)

Purpose

A40 : car (new)

A41 : car (used)

A42 : furniture/equipment

A43 : radio/television

A44 : domestic appliances

A45 : repairs

A46 : education

A47 : (vacation - does not exist?)

A48 : retraining

A49 : business

A410 : others

Attribute 5: (numerical)

Credit amount

Attibute 6: (qualitative)

Savings account/bonds

A61 : ... < 100 DM

A62 : 100 <= ... < 500 DM

A63 : 500 <= ... < 1000 DM

A64 : .. >= 1000 DM

A65 : unknown/ no savings account

Attribute 7: (qualitative)

Present employment since

A71 : unemployed

A72 : ... < 1 year

A73 : 1 <= ... < 4 years

A74 : 4 <= ... < 7 years

A75 : .. >= 7 years

Attribute 8: (numerical)

Installment rate in percentage of disposable income

Attribute 9: (qualitative)

Personal status and sex

A91 : male : divorced/separated

A92 : female : divorced/separated/married

A93 : male : single

A94 : male : married/widowed

A95 : female : single

Attribute 10: (qualitative)

Other debtors / guarantors

A101 : none

A102 : co-applicant

A103 : guarantor

Attribute 11: (numerical)

Present residence since

Attribute 12: (qualitative)

Property

A121 : real estate

A122 : if not A121 : building society savings agreement/ life insurance

A123 : if not A121/A122 : car or other, not in attribute 6

A124 : unknown / no property

Attribute 13: (numerical)

Age in years

Attribute 14: (qualitative)

Other installment plans

A141 : bank

A142 : stores

A143 : none

Attribute 15: (qualitative)

Housing

A151 : rent

A152 : own

A153 : for free

Attribute 16: (numerical)

Number of existing credits at this bank

Attribute 17: (qualitative)

Job

A171 : unemployed/ unskilled - non-resident

A172 : unskilled - resident

A173 : skilled employee / official

A174 : management/ self-employed/

highly qualified employee/ officer

Attribute 18: (numerical)

Number of people being liable to provide maintenance for

Attribute 19: (qualitative)

Telephone

A191 : none

A192 : yes, registered under the customers name

Attribute 20: (qualitative)

foreign worker

A201 : yes

A202 : no

APPENDIX 2: ATTRIBUTE RELATION FILE FORMAT (ARFF)

After converting the raw data to ARFF format, the dataset looks like the following:

@relation german_credit

@attribute checking_status { '<0', '0<=X<200', '>=200', 'no checking'} @attribute duration real @attribute credit_history { 'no credits/all paid', 'all paid', 'existing paid', 'delayed previously', 'critical/other existing credit'}

@attribute purpose { 'new car', 'used car', furniture/equipment, radio/tv, 'domestic appliance', repairs, education, vacation, retraining, business, other}

@attribute credit_amount real @attribute savings_status { '<100', '100<=X<500', '500<=X<1000', '>=1000', 'no known savings'}

@attribute employment { unemployed, '<1', '1<=X<4', '4<=X<7', '>=7'} @attribute installment_commitment real

@attribute personal_status { 'male div/sep', 'female div/dep/mar', 'male single', 'male mar/wid', 'female single'}

@attribute other_parties { none, 'co applicant', guarantor}

@attribute residence_since real

@attribute property_magnitude { 'real estate', 'life insurance', car, 'no known property'}

@attribute age real

@attribute other_payment_plans { bank, stores, none} @attribute housing { rent, own, 'for free'}

@attribute existing_credits real

@attribute job { 'unemp/unskilled non res', 'unskilled resident', skilled, 'high qualif/self emp/mgmt'}

@attribute num_dependents real

@attribute own_telephone { none, yes}

@attribute foreign_worker { yes, no}

@attribute class { good, bad}

@data

```
'<0',6,'critical/other existing credit',radio/tv,1169,'no known savings','>=7',4,'male single',none,4,'real
estate',67,none,own,2,skilled,1,yes,yes,good '0<=X<200',48,'existing
paid',radio/tv,5951,'<100','1<=X<4',2,'female div/dep/mar',none,2,'real
estate',22,none,own,1,skilled,1,none,yes,bad 'no checking',12,'critical/other existing
credit',education,2096,'<100','4<=X<7',2,'male single',none,3,'real estate',49,none,own,1,'unskilled
resident',2,none,yes,good '<0',42,'existing paid',furniture/equipment,7882,'<100','4<=X<7',2,'male
single',guarantor,4,'life insurance',45,none,'for free',1,skilled,2,none,yes,good '<0',24,'delayed
previously','new car',4870,'<100','1<=X<4',3,'male single',none,4,'no known property',53,none,'for
free',2,skilled,2,none,yes,bad 'no checking',36,'existing paid',education,9055,'no known
savings','1<=X<4',2,'male single',none,4,'no known property',35,none,'for free',1,'unskilled
resident',2,yes,yes,good 'no checking',24,'existing
paid',furniture/equipment,2835,'500<=X<1000','>=7',3,'male single',none,4,'life
insurance',53,none,own,1,skilled,1,none,yes,good '0<=X<200',36,'existing paid','used
car',6948,'<100','1<=X<4',2,'male single',none,2,car,35,none,rent,1,'high qualif/self
emp/mgmt',1,yes,yes,good 'no checking',12,'existing paid',radio/tv,3059,'>=1000','4<=X<7',2,'male
div/sep',none,4,'real estate',61,none,own,1,'unskilled resident',1,none,yes,good '0<=X<200',30,'critical/other
existing credit','new car',5234,'<100',unemployed,4,'male mar/wid',none,2,car,28,none,own,2,'high
qualif/self emp/mgmt',1,none,yes,bad '0<=X<200',12,'existing paid','new car',1295,'<100','<1',3,'female
div/dep/mar',none,1,car,25,none,rent,1,skilled,1,none,yes,bad '<0',48,'existing
paid',business,4308,'<100','<1',3,'female div/dep/mar',none,4,'life
insurance',24,none,rent,1,skilled,1,none,yes,bad '0<=X<200',12,'existing
paid',radio/tv,1567,'<100','1<=X<4',1,'female div/dep/mar',none,1,car,22,none,own,1,skilled,1,yes,yes,good
'<0',24,'critical/other existing credit','new car',1199,'<100','>=7',4,'male
single',none,4,car,60,none,own,2,'unskilled resident',1,none,yes,bad '<0',15,'existing paid','new
car',1403,'<100','1<=X<4',2,'female div/dep/mar',none,4,car,28,none,rent,1,skilled,1,none,yes,good
'<0',24,'existing paid',radio/tv,1282,'100<=X<500','1<=X<4',4,'female
div/dep/mar',none,2,car,32,none,own,1,'unskilled resident',1,none,yes,bad 'no checking',24,'critical/other
existing credit',radio/tv,2424,'no known savings','>=7',4,'male single',none,4,'life
```

```
insurance',53,none,own,2,skilled,1,none,yes,good '<0',30,'no credits/all paid',business,8072,'no known
savings','<1',2,'male single',none,3,car,25,bank,own,3,skilled,1,none,yes,good '0<=X<200',24,'existing
paid','used car',12579,'<100','>=7',4,'female div/dep/mar',none,2,'no known property',44,none,'for
free',1,'high qualif/self emp/mgmt',1,yes,yes,bad 'no checking',24,'existing
paid',radio/tv,3430,'500<=X<1000','>=7',3,'male single',none,2,car,31,none,own,1,skilled,2,yes,yes,good
'no checking',9,'critical/other existing credit','new car',2134,'<100','1<=X<4',4,'male
single',none,4,car,48,none,own,3,skilled,1,yes,yes,good '<0',6,'existing
paid',radio/tv,2647,'500<=X<1000','1<=X<4',2,'male single',none,3,'real
estate',44,none,rent,1,skilled,2,none,yes,good '<0',10,'critical/other existing credit','new
car',2241,'<100','<1',1,'male single',none,3,'real estate',48,none,rent,2,'unskilled resident',2,none,no,good
'0<=X<200',12,'critical/other existing credit','used car',1804,'100<=X<500','<1',3,'male single',none,4,'life
insurance',44,none,own,1,skilled,1,none,yes,good 'no checking',10,'critical/other existing
credit',furniture/equipment,2069,'no known savings','1<=X<4',2,'male
mar/wid',none,1,car,26,none,own,2,skilled,1,none,no,good '<0',6,'existing
paid',furniture/equipment,1374,'<100','1<=X<4',1,'male single',none,2,'real estate',36,bank,own,1,'unskilled
resident',1,yes,yes,good 'no checking',6,'no credits/all paid',radio/tv,426,'<100','>=7',4,'male
mar/wid',none,4,car,39,none,own,1,'unskilled resident',1,none,yes,good
```