FernUniversität in Hagen
Faculty of Mathematics and Computer Science

Artificial Intelligence Group

Investigating the Influence of Graph Properties on the Prediction Quality of Machine Learning Methods in the Context of Abstract Argumentation

Bachelor's Thesis

in partial fulfillment of the requirements for
the degree of Bachelor of Science (B.Sc.)
in Informatik

submitted by

Sandra Hoffmann

First examiner:	Prof. Dr. Matthias Thimm Artificial Intelligence Group
Advisor:	Isabelle Kuhlmann Artificial Intelligence Group

Statement

I hereby certify that this thesis has been composed by me and is based on my own work, that I did not use any further resources than specified – in particular no references unmentioned in the reference section – and that I did not submit this thesis to another examination before. The paper submission is identical to the submitted electronic version.

	Yes	No
I agree to have this thesis published in the library.	☒	☐
I agree to have this thesis published on the webpage of the artificial intelligence group.	☒	☐
The thesis text is available under a Creative Commons License (CC BY-SA 4.0).	☒	☐
The source code is available under a GNU General Public License (GPLv3).	☒	☐
The collected data is available under a Creative Commons License (CC BY-SA 4.0).	☒	☐

Stuttgart, 15.05.2023 S. Hoffmann

(Place, Date) (Signature)

Zusammenfassung

Aufbauend auf Ergebnissen vorheriger Forschung zur Nutzung maschineller Lernverfahren bei der Klassifizierung von Argumenten in abstrakten Argumentationsgraphen beschäftigt sich diese Arbeit mit dem Ziel, verschiedene Grapheneigenschaften und ihre Auswirkungen auf die Vorhersagequalität klassischer maschineller Lernklassifikatoren zu untersuchen. Dabei werden graphenbasierte und knotenbasierte Merkmale sowohl einzeln als auch in Kombination betrachtet. Wir untersuchen glaubwürdige und skeptische Akzeptanzaufgaben für bevorzugte, stabile, begründete und vollständige Semantik. Durch umfangreiche Experimente schlagen wir eine ideale Merkmalskombination vor, die die Vorhersagequalität für die von uns untersuchten Datensätze signifikant verbessert hat. Wir kommen weiterhin zu dem Ergebnis, dass die Anzahl der eingehenden und ausgehenden Angriffe eines Arguments den größten Einfluss auf die Klassifizierungsgenauigkeit hat und dass zufällige Wälder die höchste Klassifikationsleistung erzielen. Wir zeigen die Bedeutung der Verwendung sowohl von knotenbasierten als auch von graphenbasierten Merkmalen auf und unterstreichen die Notwendigkeit der Ähnlichkeit zwischen Trainings- und Testdaten für eine effektive Klassifizierung.

Abstract

This thesis builds upon previous research on utilizing classical machine learning methods for argument classification in abstract argumentation frameworks. The objective is to explore several graph properties and machine learning classifiers suitable for argumentation frameworks and evaluate their impact on prediction quality. We investigate both node-based and graph-based properties, testing them separately and in combination. Credulous and skeptical acceptance tasks are considered across preferred, stable, grounded, and complete semantics. Through extensive experiments, we propose and evaluate an ideal feature combination that significantly enhanced prediction quality for multiple datasets. Our results highlight degree centrality as the most informative feature and demonstrate that random forests consistently yield the highest classification performance. Our findings emphasize the importance of considering both node-based and graph-based features, as well as highlight the necessity of similarity between training and testing data for effective classification.

Contents

List of Figures

List of Tables

1. Introduction

This section introduces the topic of this thesis as well as the motivation behind our research. We also present related studies and state the problem we aim to solve.

1.1. Preface & Motivation

Abstract argumentation is a formal framework for representing and analyzing arguments. It is a subfield of artificial intelligence and has been applied in a wide range of domains, including philosophy [36], law [5] and multi-agent systems [8]. When dealing with argumentation frameworks (AFs) as introduced by Dung in [19] one is usually interested in finding sets of arguments that present a coherent view on the whole argumentation. Such a set would be an extension, and there exist several semantics that specify the conditions arguments have to satisfy in order to be part of an extension. One typical problem in abstract argumentation is deciding if an argument is part of an extension under a specific semantics.

To solve this problem, there are plenty of complete solvers [11]. However, depending on the semantics, finding a solution can be NP-complete, or even more difficult. Thus, traditional methods reach their limits when dealing with large, complex AFs. In recent years there has been some research in using neural networks [28, 14, 30] to reduce the time needed to solve an AF, even if the results are only approximations. While some of these approaches showed near perfect results, training a neural network is a resource- and time intensive process. In [25] Kuhlmann et al. showed that even classical machine learning (ML) methods, trained solely on the incoming and outgoing attacks of each argument, can perform the task of classifying whether an argument is part of an extension quite effectively.

This thesis aims to expand on this observation and explore the influence of additional graph properties. We consider graph-based properties, such as the number of strongly connected components or the average degree of each node, as well as additional node-based properties, such as various centrality-based properties. We examine credulous and skeptical acceptance for preferred, stable, complete, and grounded semantics. Although our primary objective is to identify properties that can improve the prediction quality, this work also provides a general overview of the ability of ML classifiers to predict argument acceptances for different semantics.

The remainder of this work is organized as follows. In Chapter 1, we provide an overview of previous research on approximate abstract argumentation, as well as our problem statement. In Chapter 2, we state our research questions and goals before presenting our research approach and design choices in Chapter 3. We then proceed to explain the theoretical concepts behind abstract argumentation, graph properties, and machine learning in Chapter 4, before detailing our experiments and results in Chapter 5. Finally, in Chapter 6, we discuss our findings and provide an outlook on potential future research.

Throughout this thesis, we use the terms AF and graph, feature and property as well as the terms node and argument interchangeably.

1.2. Literature Review

There has been some prior research regarding the use of machine learning in abstract argumentation. The authors in [28, 14] and [30] used Graph Neural Networks to determine the acceptability of an argument.

Kuhlmann and Thimm in [28] employed a graph convolutional network to solve AFs under the credulously preferred semantics and showed that this approach can solve an AF significantly faster than a complete solver while keeping the accuracy at about 80%. Malmqvist et al. improved this "with a randomised training regime, dynamic balancing of training data, and improved residual connections and achieved up to 97.15% accuracy" [30]. In [14] Craandijk and Bex proposed "an argumentation graph neural network (AGNN) that learns a message passing algorithm to predict the likelihood of an argument being accepted" [14]. Under several semantics, an AGNN was able to predict both credulous and sceptical acceptance almost perfectly.

Another approach in approximate abstract argumentation was taken by Cerutti et al. in [10], where the authors tested whether reasoning with grounded semantics provides a good enough approximation of other semantics to be used in pratice. They found that "in many graphs models, reasoning with grounded semantics actually approximates reasoning with other semantics almost perfectly."[10]

Building on the prior research in [28, 14, 30] Kuhlmann et al. suggested that ML classifiers should be trained on datasets that are challenging enough to justify the tradeoff between faster solving time and lower accuracy in using approximate methods. They proposed selecting a set of instances with a variety of properties (size, complexity, etc.) and varying degrees of difficulty [25]. To create these datasets, in particular, they designed the *KWT* graph generator. The authors also demonstrated that classical ML approaches learn graph-theoretic features fairly well, suggesting that complex deep learning approaches might also "just" learn these features. In their study they only considered the incoming and outgoing degree of each node and used several ML-classifiers to determine the skeptical acceptability status under preferred semantics with very promising results.

The degree of a node is also known as it's degree centrality. In general centrality measures where developed to analyse social network structures but have sinced been used to analyze other networks as well [32].

Geilen and Thimm implemented several centrality measures as heuristics to optimize the backtracking algorithm in their *Heureka* solver for abstract argumentation frameworks [23]. To the best of our knowlege, other than that, there has been little research in using centrality measures to determine the acceptability of arguments in abstract argumentation.

In terms of other graph properties, Vallati et al. [39] extracted 147 graph features in order to determine the fastest solver for an AF. Building on this work, Klein et al. [27] used supervised learning algorithms as well as graph neural networks, focussing on the best classifier to predict the fastest solver for a given AF. In order to train the ML models, they used the three most expressive features determined by

Vallati et al. Specifically, these are the number of vertices, the density, and the minimum degree value of the directed graph. They also used the query nodes' in-degree and out-degree in addition to these graph-based features. Further to the features most helpful in determining the fastest solver, the authors in [39] also tested, which of the extracted features are most informative in order to predict wether a given AF has one, several or no prefered extensions. They concluded that these are the number of strongly connected components, the average degree and the aperiodicity of the directed graph. Another study by Doumbouya et al. [17] combined "several graph properties to provide three main propositions; one for identifying accepted arguments under the admissible, preferred semantics and the other to easily identify stable extension."[17]

So far, these works have only considered classifying arguments under preferred or stable semantics and have used the graph features mainly to pick the fastest solver [39, 27] or to assess whether dominant graph features bias the training of neural networks [25]. The focus of this work will hence be directed towards trying to improve the accuracy of classical machine learning methods by investigating other graph features that are helpful when classifying arguments under several semantics.

1.3. Problem Statement

Despite the availability of complete solvers for determining whether an argument belongs to an extension, the problem can be computationally expensive and intractable for large, complex AFs. This might pose problems for real-world applications relying on argumentation [10].

To address this issue, some studies have explored the use of machine learning methods to approximate solutions with reduced computation time. Although some studies show promising results, training those networks is a resource-intensive process. However, previous research suggests that even traditional ML techniques show promising results in classifying arguments [25]. So far, however, only the incoming and outgoing attacks for each argument and its skeptical acceptability status under preferred semantics have been considered. Therefore, the aim of this thesis is to investigate the influence of additional graph and node properties, separately as well as in combination, on the prediction quality of machine learning methods in determining whether an argument belongs to an extension under different semantics. We will investigate credulous (DC) and skeptical (DS) acceptance for the four classic semantics as defined by Dung in [19], namely preferred (PR), stable (ST), complete (CO), and grounded (GR).

2. Research Questions and Goals

The goal of this thesis is to find out what influence graph properties have on the prediction quality of machine learning methods when classifying arguments of an AF.

The thesis aims to answer the following research questions:

RQ1. Which investigated graph property has the largest effect on prediction quality? Is this consistent across graph types and semantics?

RQ2. Which is the ideal set of graph properties to achieve the highest prediction accuracy for a given machine learning classifier?

Other questions of interest, that will be answered during the experimental analysis stage of this thesis include:

- Which effect do graph properties, that have been extracted for the whole graph, have on the classification of individual nodes?
- Given that credulously deciding if an argument is part of an extension is equally hard for all considered semantics, does this hold true for the classification accuracy as well, or are some semantics "easier" to classify, e.g., consistently achive a higher accuracy?

3. Approach & Research Design

The first step in our experimental setup is to define the graph properties that we will investigate. To choose these properties, we draw on previous research in abstract argumentation and examine the features proposed by Vallati et al. [39] and Doumbouya et al. [17] as informative for the entire graph. Specifically, we consider the total number of strongly connected components (SCCs), the average degree of each node, the aperiodicity of the graph, and whether the graph is strongly connected, symmetric, or irreflexive. As strong connectivity was found to be important in both studies, we also use the size of the SCC a node belongs to as a node-based property. Other investigated node-based properties are the degree centrality, as well as Katz centrality, closeness centrality, and betweenness centrality.

Our ideal scenario would be to use the *kwt-train* and *kwt-test* [1] datasets from [25] in order to directly compare our results. However, this dataset was specifically designed to contain graphs that are difficult to solve for skeptical acceptance under preferred semantics, and it is too unbalanced for different semantics to be used as our main dataset. In the case of credulous acceptance under stable semantics, for instance, practically all arguments are rejected. Therefore, we will create a *balanced-kwt-train* and *balanced-kwt-test* dataset by augmenting the original *kwt* dataset with graphs generated using the *Probo Benchmark Suite* [2] and the *AFBenchGen2* [3] graph generators. These datasets will be the starting point for the experimental approach in this thesis.

[1] `https://fernuni-hagen.sciebo.de/s/ZEmipULEN05FxxC`
[2] `https://sourceforge.net/projects/probo/`
[3] `https://sourceforge.net/projects/afbenchgen/files/AFBenchGen2/`

Since the task of classifying arguments in an AF is best suited for supervised learning methods, we will train decision trees, random forests, Naïve Bayes, K-nearest neighbors, and support vector machines. We will first test each selected feature separately and compare the results based on the Matthews correlation coefficient (MCC), as this metric takes all values of the confusion matrix into account and also works well for unbalanced datasets. We will then combine the features in order of their individual results to classify the data. Once we have identified the ideal feature combination, we will use these features on the datasets used in [25] to test whether our observations hold true for other datasets. In order to asses the influence of the individual graph properties on the different graph types we will split the *balanced-kwt-test* into six sets, each one only containing a single graph type. We will then test the individual properties on each of those graph types using the *balanced-kwt-train* dataset for training. For training and evaluating these models and extracting node features, we will use the Python modules *scikit-learn* [4] and *networkX* [5], respectively.

There are some limitation to our study. In the datsets the class distributions for several semantics coincide with each other. For instance if an argument is credulously accepted under preferred semantics it is by definition also credulously accepted under complete semantics, because every preferred extension is also a complete extension and credulous acceptance only requires an argument to be part of at least one extension. Similarly, the distributions for grounded semantics (both credulous and skeptical acceptance) and complete semantics (skeptical acceptance) coincide in several datasets. While it is not unusual for certain semantics to coincide [10], this convergence makes it difficult to draw meaningful distinctions between different semantics in terms of an ML classifier's ability to classify arguments.

To determine the order in which we combined features for testing, we assigned a rank to each feature for each acceptance task. The feature that achieved the best overall result for a given task was given rank 1. However, when setting the rank, we only considered the best result overall and did not consider the best result for each classifier. Nevertheless, in many cases, the individual rank of a feature for each classifier was very similar to the overall rank for that feature.

Regarding the training of the ML classifiers, we kept the default values whenever possible. This approach was taken to ensure comparability with the results in the reference study [25]. We also restricted ourselves to using linear Support Vector Machines (SVMs). Although non-linear SVMs might have yielded better results for certain features and feature combinations, the runtime of non-linear SVMs would have been too long for our study. Finally, we only included a small fraction of all existing graph properties. However, the graph-based properties we chose have been shown to be helpful in other studies, and the centralities we used are among the most popular ones, which justifies our selection.

[4] `https://scikit-learn.org/stable/`

[5] `https://networkx.org/documentation/stable/`

4. Background

This chapter provides the theoretical foundation for abstract argumentation, as well as the graph- and node-based features that we investigate in this thesis. Additionally, we describe the machine learning classifiers and evaluation metrics that we use in our experiments.

4.1. Abstract Argumentation

In 1995, Phan Minh Dung created a mathematical basis for the field of abstract argumentation with his paper *"On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games"* [19]. In this paper, Dung argued that "a statement is believable if it can be argued successfully against attacking arguments." By focusing exclusively on the argument attack relation and abstracting from the structure of the individual arguments, Dung was able to model this relation as a directed graph, called "argumentation framework" (AF), where arguments are represented as nodes and attacks as directed edges [40]. In mathematical terms, Dung defines an AF as follows:

Definition 4.1.1. An *argumentation framework* is a pair $AF = (AR, attacks)$, where AR is a set of arguments and $attacks$ is a binary relation on AR, i.e $attacks \subseteq AR \times AR$.

One of the main results of Dung's paper was defining when arguments in an AF can be accepted by way of argumentation semantics.

Definition 4.1.2. Let $F = (AR, attacks)$ be an AF.

- A set $S \subseteq AR$ is said to be *conflict-free* if there are no arguments $a, b \in S$ such that a attacks b.
- An argument $a \in AR$ is *defendable* with reference to (wrt) S if and only if (iff) for each argument $b \in AR$: if b attacks a then b is attacked by S.
- A conflict-free set of arguments S is *admissible* iff each argument in S is defendable wrt S.
- A *preferred extension* of F is a maximal (wrt set inclusion) admissible set of F.
- A conflict-free set of arguments S is called a *stable extension* iff S attacks each argument which does not belong to S
- An admissible set S is called a *complete extension* iff each argument which is acceptable wrt S, belongs to S.
- S is called a *grounded extension* if S is a minimal (wrt set inclusion) complete extension of F [2].

These original semantics where subsequently expanded to include further semantics, such as the ideal extension, which is defined as the maximal admissible set S that is contained in every preferred extension [18] as well as the naïve [4], semi-stable [41, 6], eager [7], stage [41] and prudent [13] semantics.

Definition 4.1.2 shows that different semantics are related to each other. In fact, many are actually "restricted cases of complete semantics" [42], because every stable extension is also a preferred extension [6], and both preferred and grounded extensions are also complete extensions [19]. These relationships are visualized in Figure 1.

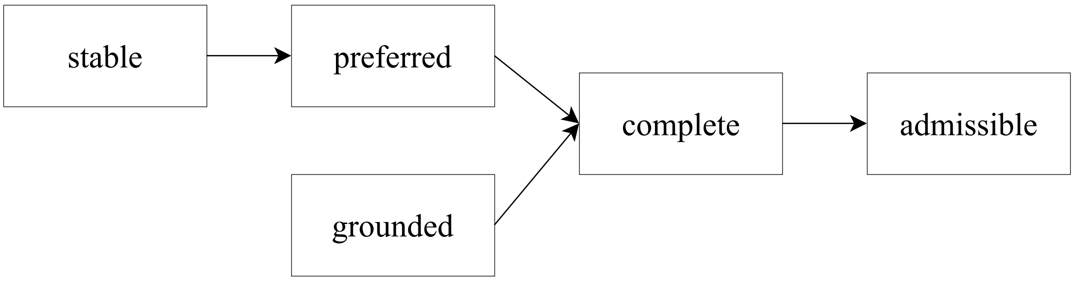

Figure 1: Relationships Between Different Semantics

It is possible that multiple extensions of the same semantics exist in a given AF. In order to determine the overall status of an argument, one therefore needs some form of justification measure. This measure is provided by the notion of skeptical and credulous justification. Skeptical justification requires that an argument is accepted in all extensions, while credulous justification requires that an argument is accepted in at least one extension [3].

An example of an AF is shown in Figure 2. The corresponding argument could be:

1. Tweety is a bird, and therefore it can fly (a)
2. Tweety is actually a penguin, so it cannot fly (b)
3. Tweety might look like a penguin but is in fact a puffin, so it can fly (c)
4. But Tweety has a broken wing preventing it from flying (d)

While this AF is pretty straightforward, it is also possible for an AF to have two arguments attacking each other, or attacks going in circles. One such example can be seen in Figure 3. A real-life argument that could be modeled by this AF could be about determining which one of three given movies can be considered the best one. Assuming movie a has grossed the most money, one might think movie a is the best movie. However, if movie b has won the most awards and movie c has the highest critic score, one might as well consider one of these movies to be the best one. This intuitive conflict is reflected in the extensions that can be calculated for the given AF.

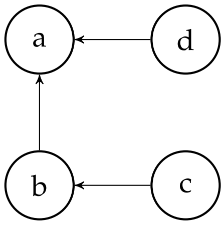

Figure 2: Example of an Abstract Argumentation Framework

Each argument attacks all other arguments but itself. Thus, each argument is only admissible and acceptable wrt a set that only includes itself. Each of these sets is a preferred, stable, and complete extension. However, none of the arguments hold up under skeptical justification. This again is in line with the intuitive view. While a credulous observer might consider any one of the movies to be the best, a skeptical observer will not deduce anything from this AF.

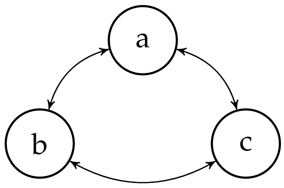

Figure 3: Example of an Abstract Argumentation Framework with Circular Conflicts

The complexity of deciding whether an argument is credulously or skeptically accepted varies greatly depending on the selected semantics, as can be seen in Table 1.

	admissible	grounded	complete	preferred	stable
Credulous	NP-complete	in P	NP-complete	NP-complete	NP-complete
Skeptical	trivial	in P	P-complete	Π_2^P-complete	co-NP-complete
	[22]	[20]	[22]	[16, 21]	[16, 21]

Table 1: Complexity for Different Decision Problems and Semantics

The relations between the different complexity classes are as follows [22]:

$$L \subseteq P \subseteq \begin{matrix} NP \\ coNP \end{matrix} \subseteq \Pi_2^P$$

We can see that deciding if an argument can be skeptically accepted under preferred or stable semantics is a much harder problem than deciding for grounded or complete semantics. We can further see that, apart from grounded semantics, deciding if an argument can be credulously accepted is equally hard for all semantics.

4.2. Graph Properties

Graphs can be represented as a set of nodes connected by either directed or undirected edges or be encoded in matrix form. This representation is called an adjacency matrix, which is of shape $n \times n$, where n is the number of nodes in the graph. In an adjacency matrix for a directed graph, a value of 1 in position (b, a) indicates that there is a directed edge from node b to node a, while a value of 0 indicates that there is no directed edge from node b to node a. Table 2 shows an example of an adjacency matrix for the graph presented in Figure 2.

	a	**b**	**c**	**d**
a	0	0	0	0
b	1	0	0	0
c	0	1	0	0
d	1	0	0	0

Table 2: Example of an Adjacency Matrix

To analyze a graph, one can examine graph-based properties, which take the whole graph into consideration, as well as node-based properties, which describe the characteristics of individual nodes in the graph. In this thesis, in order to investigate which graph features are helpful in training a ML classifier we will look at both graph- and node-based properties.

4.2.1. Graph-based Properties

In [39], the authors extracted 147 features from the undirected, directed, and matrix representation of an AF to predict whether this AF has one, several, or no preferred extensions. They found that the most informative features for this prediction were the number of strongly connected components (SCCs), the average degree, and the aperiodicity of the directed graph.

Definition 4.2.1. Let $F = (AR, attacks)$ be an AF.

- F is **strongly connected** if for every pair of arguments a and $b \in AR$, there is a path of attacks from a to b and a path of attacks from b to a. A **strongly connected component (SCC)** of F is a maximal subgraph of F that is strongly connected.
- The **average degree** of F is the sum of the incoming and outgoing attacks of all arguments divided by the number of arguments.
- F is said to be **aperiodic** if there is no integer $k > 1$ that divides the length of every cycle of F.

These features may also be useful in determining the acceptability status of individual arguments. For example, if a classifier can predict that a given AF has only one empty preferred extension using these features, then all arguments can be classified as neither credulously nor skeptically accepted under preferred semantics. Additionally, since each stable extension is a preferred extension, the arguments could also be classified as not credulously accepted under stable semantics. In such a scenario, there would be no stable extension which would lead to all arguments being (trivially) skeptically accepted under stable semantics.

In another noteworthy study, Doumbouya et al. [17] extracted several graph features from an AF and demonstrated their direct impact on the acceptability status of an argument. In their work, they considered preferred and stable semantics and found that if an AF is strongly connected, symmetric, and irreflexive, then all of its arguments can be credulously accepted under preferred semantics.

Definition 4.2.2. Let $F = (AR, attacks)$ be an AF.

- F is **symmetric** if for any pair of arguments $a, b \in AR$, $a \times b \in attacks$ and $b \times a \in attacks$. F is **asymmetric** if for any pair of arguments $a, b \in AR$, if $a \times b \in attacks$ then $b \times a \notin attacks$.

- F is said to be **irreflexive** if there is no argument $a \in AR$ with $a \times a \in attacks$. Thus, if F is asymmetric, F is also irreflexive.

An example of such a graph can be seen in Figure 3 in Section 4.1. This AF is strongly connected because there exists a path from each argument to all other arguments. Moreover, it is symmetric because each argument counters all attacks on itself. Also, none of its arguments attack themselves, making this AF irreflexive. As proposed in [17] and as we have already deduced in discussing this example in Section 4.1, all arguments can be credulously accepted under preferred semantics.

The authors also demonstrated that if an AF is asymmetric and if one of its arguments attacks all the others, then this argument forms a unique stable extension. To illustrate this proposition, consider the AF in Figure 4, which is an altered version of the AF depicted in Figure 2 where argument d attacks not only argument a but also b and c. Since no two arguments mutually attack each other, this AF is asymmetric. As it is therefore irreflexive, each argument forms a conflict-free set on its own, and as argument d attacks every other argument, it remains undefeated and becomes the only admissible set, forming a stable extension. This extension must be unique because if there were a second one, it would have to attack argument d, thereby violating the asymmetric property of the graph. As a result, all other arguments in this AF are not accepted under stable semantics. However, AFs like this can be considered to be edge cases. Furthermore, it would be easy to test these features for a given AF and assign each argument its acceptance class without the need for any form of classification or even a solver.

In the datasets used in this thesis, no graph satisfies the conditions of the stable extension proposition. Therefore, we disregard both properties used in this proposition and concentrate only on the symmetry, strong connectivity, and irreflexivity of the graph.

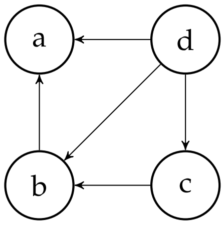

Figure 4: Example of an Abstract Argumentation Framework

While Vallati et al. showed that their graph properties are also informative when used separately, Doumbouya et al. have only used their extracted features in combination. Although the connection between these features and the acceptability status of individual arguments in AFs that satisfy the conditions is clear, and it seems reasonable to suspect that a machine learning classifier can learn these connections, it will be interesting to see if these features have any effect if used separately, as well as if they have any effect on graphs that do not satisfy the conditions as laid out in [17].

The features discussed in this section are extracted for the entire graph. However, when determining the acceptability of a particular argument, relying solely on these features may not be sufficient. While we have seen how these features can aid in assessing individual argument acceptability, they are most likely more useful in AFs where all arguments belong to the same acceptance class. Therefore, we need to consider node-based graph properties that can provide more information about the individual arguments.

4.2.2. Node-based Properties

In [25], the authors used the incoming and outgoing attacks, or degree, of each argument to train several ML classifiers to identify whether an argument can be skeptically accepted under preferred semantics. They showed that this feature provides a strong indication of the acceptance status of an argument. In graph theory, measures that quantify how important a single node is to a graph are called centrality measures. As there are different views on what makes a node important, there are likewise different centrality measures. The most commonly used are degree, eigenvector, betweenness, and closeness centrality [31], which we discuss in this chapter. To illustrate these measures, we revisit the Tweety example from Section 4.1. In this AF, arguments c and d form the only preferred, stable, grounded, and complete extension (see Figure 5).

Degree centrality measures the number of edges that are connected to a node in a graph. In directed graphs, the incoming and outgoing degrees of each node are measured separately. In the context of abstract argumentation, degree centrality

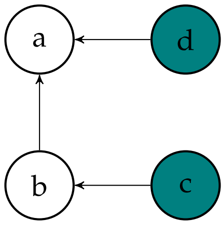

Figure 5: Tweety Example with Highlighted Extension

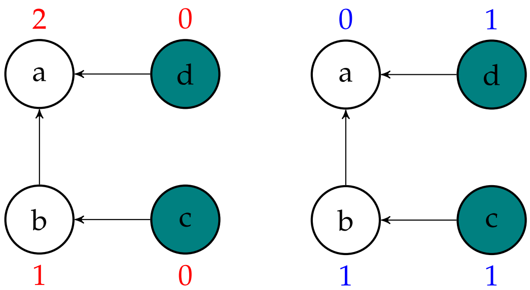

Figure 6: Incoming and Outgoing Degree Centrality

measures the incoming and outgoing attacks of every argument in the AF, making it a useful measure for gaining insights into the total connections an individual argument in an AF has [31]. In our example, arguments d and c are undefeated since they have no incoming attacks, but they each attack another argument whereas argument a is attacked twice but does not attack anyone in return. Thus, we could argue that arguments d and c are both very important while a is the least important argument in our example. However, degree centrality is a rather crude measure in that it only focuses on a single argument and does not take the influence of its neighbors into account at all [24]. For instance, b, c, and d have the same outgoing degree centrality. While for d and b this makes sense, as they are both attacking only argument a, argument c is actually defending a by attacking b. However, degree centrality does not capture this information.

An extension of the degree centrality that takes the neighbors of an argument into account as well is **eigenvector centrality**. Eigenvector centrality assigns a score to each argument based not only on how many neighbors it has but also on how important those neighbors are. Thus, an argument can have high centrality by being connected to many moderately important arguments or a few very important ones.

We can define the eigenvector centrality of an argument i using the adjacency matrix A of the AF i belongs to as:

$$C_E V(i) = \frac{1}{\lambda} \sum_{j=1}^{n} A_{ij} x_j$$

Or, in matrix notation:

$$\boldsymbol{Ax} = \lambda \boldsymbol{x}$$

Here, x is an eigenvector of A, and λ is the largest eigenvalue of A [32].

As adjacency matrices in directed graphs are most often asymmetric, there are two sets of eigenvectors in such graphs and therefore also two eigenvector centralities. "Right" eigenvector centrality measures the centrality of a node by taking its incoming edges into account while "left" eigenvector centrality looks at the outgoing edges a node has. However, if we calculate the eigenvector centralities for our example, they end up being zero for all arguments. This is because we have arguments that have no incoming edges (d and c) as well as no outgoing edges (a). Such arguments, however, have an eigenvector centrality of zero, which then propagates through the whole graph. Hence, in directed graphs, only nodes that are in an SCC of at least size 2, or nodes that are connected to such SCCs, can have an eigenvector centrality that is non-zero [32].

A variation of eigenvector centrality that solves this problem is **Katz centrality**. Katz centrality introduces a constant into the eigenvector centrality equation, thereby assigning every node a base centrality regardless of its connection to its neighbors. The equation for calculating the Katz centrality of argument i is calculated as follows:

$$C_K(i) = \alpha \sum_{j=1}^{n} A_{ij} x_j + \beta,$$

where α and β are constant, positive values and A is again the adjacency matrix of the given AF. The first part of the equation is the normal eigenvector centrality, but by adding β, Katz centrality ensures that even arguments with a degree equal to zero have an initial centrality value of β. In matrix notation, Katz centrality can be written as

$$\mathbf{x} = \alpha \mathbf{A} \mathbf{x} + \beta \mathbf{1},$$

with 1 being a vector where all elements are equal to 1. As we are typically more interested in seeing which nodes have high or low centralities than in the actual centrality values, the value that is assigned to β is unimportant. It is important to note, that the parameter α is used to balance the eigenvector and the constant part of the equation. If α is set to zero, all centralities would be equal to β. However, if α is too large, the centrality values diverge. Thus it is important to make sure that α is not larger than $\frac{1}{\lambda}$ [32]. In our experiments we achieved this by calculating $\frac{1}{\lambda}$ and then setting α to $\frac{1}{\lambda} * 0.9$. For the special case of $\lambda = 0$ we set α to 0.1, which is the default value for this parameter in *scikit-learn*. Figure 7 shows the Katz centralities with $\beta = 1$ and $\alpha = 0.1$ for our example. We can see that this measure is now able to assign more nuanced centralities, and in the case of outgoing centralities, it now also captures that argument c is more important than b and d.

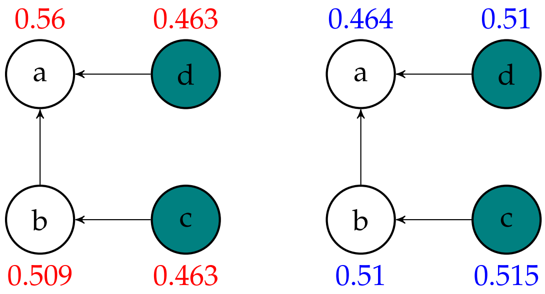

Figure 7: Incoming and Outgoing Katz Centrality

While degree and eigenvector based centralities rely on a node's degree, the other two measures that we will investigate in this thesis focus on the shortest paths between a node and its neighbors. The shortest path between two nodes is defined as the path with the minimum number of edges. The **betweenness centrality** of a node x is calculated by summing up the fractions of shortest paths for all node pairs in the graph that pass through x. This is done by first calculating the shortest paths between all node pairs in the graph. Then we can simply sum up the number of shortest paths that x is on and divide this number by the total number of shortest paths in the graph. This gives us the formula for calculating the betweenness centrality:

$$C_B(i) = \sum_{s \neq x \neq t} \frac{\sigma_{st}(x)}{\sigma_{st}}$$

where σ_{st} is the total number of shortest paths from node s to node t, and $\sigma_{st}(x)$ is the number of those paths that pass through node x. Betweenness centrality is often used as a measure of which nodes are important for controlling the flow of information or traffic through a network [32]. In Figure 8, we can see that only argument b has a betweenness centrality value other than zero because it is the only node that lies on a shortest path between two other arguments, namely c and a. A high betweenness centrality value can indicate that an argument acts as a bridge, connecting different components of the graph.

Closeness centrality, on the other hand, measures how well-connected a node is to the rest of the graph by calculating the average distance from a node to all other nodes in the graph. The central idea is that a node that is close to other nodes in the graph is more important than a node that is far away. The nodes that have the highest closeness centrality are the ones that can spread information or influence more efficiently than others because they are, on average, closer to all other nodes in the graph [24].

To calculate the closeness centrality, the distances between a node and all other nodes in the graph are summed up, and the reciprocal of that sum is taken. Nodes that are isolated or have limited connections have low closeness centrality, while

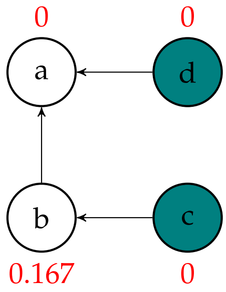

Figure 8: Betweenness Centrality

nodes that are well-connected and closer to other nodes in the graph have high closeness centrality. For directed graphs, closeness centrality can be calculated for incoming and outgoing edges separately.

The formula for calculating closeness centrality is:

$$C_C(i) = \frac{n-1}{\sum_{v \in V} d(u, v)}$$

where $C_c(v)$ represents the closeness centrality of node v, $d(u, v)$ is the shortest path distance between nodes u and v, and the summation is taken over all nodes u in the graph [32]. Figure 9 shows the closeness centralities for our example. We can see, that in our small example this measure relates closely to the degree centrality, however it can distinguish between nodes c, d and b in terms of outgoing centrality, whereas the outgoing degree centrality for these three nodes is the same.

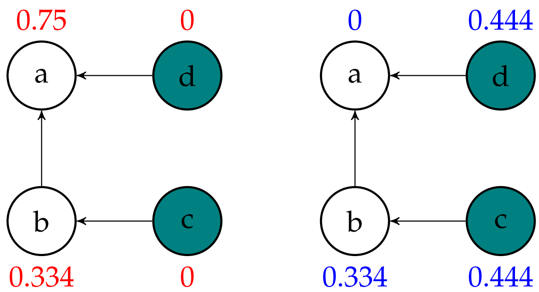

Figure 9: Incoming and Outgoing Closeness Centrality

This section has provided an overview of the theoretical concepts underlying the graph- and node-based features we examine in this thesis. In order to carry out our experiments, we employed a variety of machine learning classifiers, which we present in the subsequent section.

4.3. Machine Learning

Machine learning is often defined as a field of study that gives computers the ability to learn without being explicitly programmed. This definition is attributed to Arthur Samuel who stated in 1959, "Programming computers to learn from experience should eventually eliminate the need for much [...] programming effort" [37].The tasks that a computer can learn to perform with the help of machine learning are categorized into classification, anomaly detection, regression, clustering, and reinforcement learning problems [26]. The task of deciding which argument of an AF belongs to an extension is a classification problem because there is a fixed number of classes (yes or no), and those classes are known beforehand. Hence, in this section, we only discuss machine learning algorithms that are suitable for solving classification problems.

As seen in Figure 10, there are six stages that make up the creation of a machine learning model, independent of the algorithm that is being used.

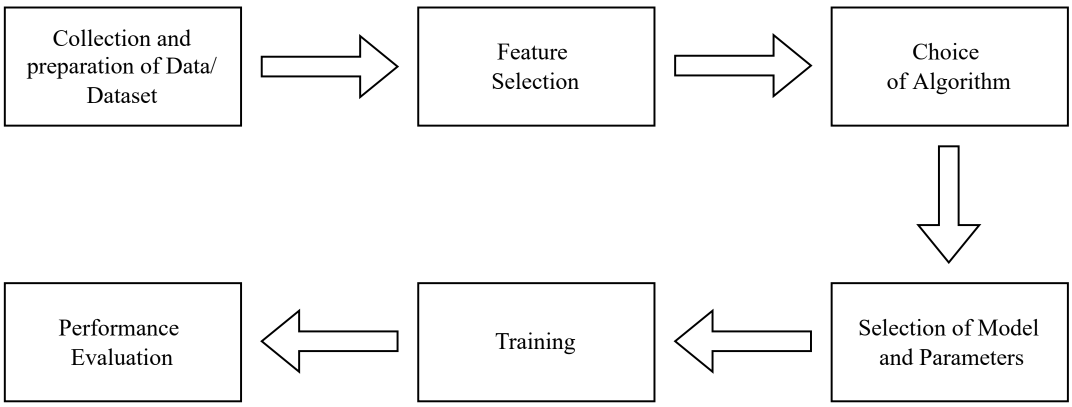

Figure 10: Stages of the machine learning model creation [26]

At the beginning of the process, relevant data needs to be collected, cleaned, and preprocessed to bring it into a structured format for further processing. That data may contain numerous features, of which some might be redundant or irrelevant to training the model. Once the set of relevant features is selected, one can choose which algorithm is best suited for the task. The selected algorithm might need some initial adjustment (e.g., setting the number of neighbors parameter for the KNN algorithm). Once these stages are completed, the model needs to be trained using the available data. The data set is thus split into training and testing data. The model is trained using the training data and then performs the specific task it was trained for on the unseen testing data. To evaluate the performance of the model on the testing data, there are various performance parameters like accuracy, precision, recall or the Matthews correlation coefficient (see section 4.3.5).

Machine learning algorithms can be classified by the different paradigms they adhere to, of which supervised learning is the most popular [26]. In supervised learning algorithms, the computer learns a function that maps an input to an output by looking at examples of input-output pairs [29]. These algorithms are supervised in that they need output labels to be provided for the training and testing data. Algorithms that can be used for supervised classification learning include Decision Trees and Random Forests, K Nearest Neighbors, Naïve Bayes, and Support Vector Machines.

4.3.1. Decision Trees and Random Forests

Decision trees (DT) are a type of classification method that uses a hierarchical set of decisions to model the classification process. These decisions are organized in a tree-like structure. At each node of the tree, a decision, also called split criterion, based on one (univariate) or more (multivariate) feature variables present in the training data is made. This divides the training data into two or more groups.

Ideally, we want to find a split criterion that best separates the class variables into each branch of the tree. There are multiple strategies for finding the best split criterion, like entropy or the Gini index. For the experiments in this thesis we used the Gini Index as split criterion.

The Gini index is defined as:

$$G(S) = 1 - \sum_{j=1}^{c} p_j^2$$

where c is the number of classes in the set, and p_j is the proportion of instances in class j.

The Gini index is 0 when all instances in the set belong to the same class. We find the best split criterion by calculating the overall Gini index for each possible split:

$$Gini - Split(S \Rightarrow S_1 ... S_r) = \sum_{i=1}^{r} \frac{|S_i|}{|S|} G(S_i)$$

and then selecting the criterion with the lowest overall Gini index [1].

The process of constructing decision trees involves recursively splitting the data until a stopping criterion is met, such as a predefined tree depth or the presence of homogeneous groups of data. However, when the tree is grown to the point where each leaf node contains only instances of a particular class, it is likely overfitted because it takes even random characteristic of the training data into account. This means that while it will be completely accurate in classifying the training data, it will probably perform poorly when classifying unseen data [1].

To reduce overfitting, pruning strategies are used to remove overfitting portions of the tree and convert internal nodes to leaf nodes. A simple pruning strategy is to only use part of the training data to construct the tree and then test the accuracy on the remaining data. If pruning a node would increase the accuracy, then it will be removed from the tree. This process will be done iteratively until the accuracy cannot be improved anymore [1].

Decision trees can be easily interpreted and explained. They are robust to outliers in the data and can also handle data that cannot be separated linearly. While the computational effort for classifying data with a decision tree is relatively small, building the tree is considerably time-intensive. Hence, they are unsuitable for high-dimensional data. Very large decision trees can have problems with data fragmentation (if there are too few records in the leaf nodes to make statistically significant decisions) and overfitting. In our experiments we used the base implementation of DTs provided by the framework *scikit-learn* and did not implement any pruning strategies. This is because we also used an ensemble model called **Random Forest (RF)** that can deal with these issues efficiently [38].

Random forests are an ensemble of decision trees, where each tree gets to vote on the most likely class of a given data point. The main idea is to create a forest of decision trees by randomly sampling (with replacement) subsets of the original dataset, of the same size as the original data, instead of building a single tree from the whole set of training data. This process is called bootstrapping. Additionally, the selection of split criteria is also randomized. A parameter q, smaller than the total number of features (d), determines the number of features used to build the trees. Before each split, q features are randomly selected and used to split the data. If q is small, the correlation between the trees is reduced and tree building is more efficient. However, the larger q is, the higher the accuracy of the resulting trees. If d is large, the best trade-off is achieved by selecting $q = log_2(d) + 1$ attributes [1]. However as the number of features used in this thesis is relativly small, we used $\sqrt{d}$ features to build the trees.

Trees in random forests do not need to be pruned. The variance of the model is reduced significantly compared to decision trees, and they are resistant to noise and outliers in the data. However, random forests might have a higher bias than decision trees because they might miss patterns in the data due to the reduced sample size [1].

4.3.2. K Nearest Neighbors

The K Nearest Neighbors (KNN) algorithm classifies instances based on the classes of its k nearest neighbors. The training examples used for classification are needed in memory at run-time, which is why the technique is sometimes called Memory-Based Classification [15]. Algorithms of this kind are often called "lazy learners" because they do not calculate a global target function but rather estimate it locally for every instance that needs to be classified. While this makes training fast, it slows down predictions [26].

Determining which class should be assigned to an instance a can be done in multiple ways. The most intuitive way would be to assign the class that most neighbors belong to. However, sometimes one might want to give more weight to the neighbors closest to a. One technique to achieve this would be to have the neighbors vote on the class "with votes weighted by the inverse of their distance to the query" [15]. Thus, the KNN classification happens in two stages: determining the nearest neighbors and then determining the class utilizing those neighbors. Which neighbors are the closest ones, however, can differ greatly based on what distance measure is used. Usually, this is some form of the Minkowski distance. The general Minkowski distance is defined as:

$$MD_p(q|x_i) = \left(\sum_{f \in F} |q_f - x_{if}|^p \right)^{\frac{1}{p}}$$

where p is a positive integer, q and x_i are two points in the p-dimensional space, and F is the set of features or dimensions in the space. q_f and x_{if} are the values for feature f in points q and x_i, respectively. When $p = 1$, this distance is called Manhattan distance, and when $p = 2$, it becomes the Euclidean distance. Using larger values of p gives greater weight to those attributes that are the most different between q and x. However, it is rather unusual to use values for p that are greater than two, with the exception of $p = \infty$, which is called the Chebyshev distance, also known as the chessboard distance. Here, the distance is that of the dimension in which q and x differ the most [15].

KNN can handle multimodal classes and objects with many labels. Its performance, however, depends on the value of k that is selected and on the size of the dataset [38]. For our experiments, we set $k = 5$ to ensure that our results are comparable to [25]. To calculate the nearest neighbors we used the Euclidean distance and gave equal weight to each neighbor.

4.3.3. Naïve Bayes

The Naïve Bayes classifier (NB) uses Bayes' Theorem of probability to calculate "the posterior probability of an event (A) given some prior probability of an event B represented by $P(A|B)$ as follows:" [26]

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

An example of a problem that can be solved using Bayes' Theorem is determining the probability that it will rain tomorrow, given that the forecast says there is a 60% chance of rain.

To solve this problem, we first need to calculate the prior probability $P(A)$, which would be the general probability of rain without knowing the forecast. For this example, we assume the prior probability of rain is 30%.

Next, we need to calculate the likelihood $P(B|A)$, which is the probability of the forecast given that it will rain. In this case, the likelihood is 100%, because if it rains, the forecast predicting rain will always be correct.

Lastly, we can calculate the posterior probability $P(A|B)$, which is the probability of rain given the forecast, using the formula:

$$P(\text{rain}|\text{ forecast}) = \frac{P(\text{forecast}|\text{ rain})P(\text{rain})}{P(\text{forecast})}$$

$$P(\text{rain}|\text{ forecast}) = \frac{100\% * 30\%}{60\%} = 50\%$$

This means that there is a 50% chance of rain tomorrow, given that the forecast says there is a 60% chance of rain.

In the context of machine learning, $P(A|B)$ would be the probability that instance B belongs to class A. More specifically, B is actually the n-dimensional feature vector $B = (x_1, \ldots, x_n)$ that describes the instance to be classified. As the denominator of Bayes' Theorem is class-independent, it can be disregarded when calculating the probabilities. Hence, the formula can be shortened as follows:

$$P(A|B) \approx P(B|A)P(A)$$

The prior probability of the class $P(A)$ can be estimated by the share of instances in the training data belonging to that class. However, trying to classify data with this equation has a disadvantage. For every instance we want to classify, we would need instances with the same feature values to predict the most probable class. Depending on the number of features and the size of the dataset, there might not be enough training instances that satisfy all feature conditions. Here the naïve part of the classifier comes into play. NB assumes that the values of different attributes are conditionally independent of each other, given the class [1]. This means that instead of calculating the probability of the whole feature vector belonging to one class, we can calculate the probability of each feature separately and multiply the results. So in order to classify our data, we can perform the following calculation for each class and select the class with the highest probability [1].

$$P(A|x_1...x_n) = P(A)\prod_{i=1}^{n} P(x_i|A)$$

The NB classifier is particularly useful for datasets with high dimensionality [26]. In our experiments, we utilize the *ComplementNB*[6] classifier, as it is particularly well-suited for handling unbalanced data.

4.3.4. Support Vector Machines

Support Vector Machines (SVMs) use a hyperplane to separate data into classes. As there are infinite ways to construct a hyperplane for linearly separable data, SVMs

[6] `https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.ComplementNB.html`

need a way to determine which hyperplane is best suited for classification. Therefore, their goal is to find the one that has the widest margin, as it is considered to be the most robust in terms of categorizing the data. The margin is defined as the total distance between the nearest training points on either side of the hyperplane. The vectors closest to the margin are called *support vectors*.

The hyperplane can be expressed as:

$$\mathbf{w}^T\mathbf{x} + b = 0,$$

where $\mathbf{w}^T$ is a vector normal to the hyperplane and b is the bias term. The goal of the training process is to learn the coefficients for $\mathbf{w}^T$ and b that maximize the margins between both classes. The hyperplane is exactly between both margins [1]. Once an SVM is trained, new instances are categorized by being mapped into the n-dimensional space and assigned a category based on where they land. In order to deal with outliers, SVMs use a *soft margin*, which allows some vectors to be on the "wrong side" of the hyperplane without affecting the result of the separation [34]. Should the data not be linearly separable, SVMs use the so-called *kernel trick*, mapping the data into a new and usually higher-dimensional space until linear separation is possible [35]. In our experiments, however, we only considered SVMs with a linear kernel.

4.3.5. Evaluation Measures

In order to determine which ML classifier performs the task of classifying the acceptability status of an argument best, we need some form of evaluation measure. Oftentimes, when dealing with a binary classification task, one class is labeled as positive and the other as negative. In this thesis, the class that contains the arguments that are accepted is the positive, and the class that contains the arguments that are rejected is the negative class. An argument a of the testing data that is classified can be grouped into one of the following four classes:

- a is **TP (true positive)** if a is positive and is correctly predicted positive.
- a is **FN (false negative)** if a is positive and is wrongly predicted negative.
- a is **TN (true negative)** if a is negative and is correctly predicted negative.
- a is **FP (false positive)** if a is negative and is wrongly predicted positive.

This relation can be visualized by a confusion matrix C as:

$$C = \begin{pmatrix} \text{TP} & \text{FN} \\ \text{FP} & \text{TN} \end{pmatrix}$$

The row sums of C are the actual positive $(TP + FN)$ and negative $(TN + FP)$ instances in the dataset. The goal of using ML classifiers is to achieve high values of true negatives and true positives. Thus, we can use the **TPR (true positive**

rate) $TPR = \frac{\text{TP}}{\text{TP + FN}}$, also called sensitivity, as well as the **TNR (true negative rate)** $TNR = \frac{\text{TN}}{\text{TN + FP}}$, or specificity, as a simple evaluation measure. A similar measure is the **precision** $= \frac{\text{TP}}{\text{TP + FP}}$, which is used to find the proportion of correctly classified positive arguments. If a model has high precision, we can have more confidence in its prediction when it labels an argument as accepted. However, each of these measures only uses two entries of the confusion matrix, so their information value is limited. To get a more accurate picture of the performance, we can look at the **accuracy** $= \frac{\text{TP + TN}}{\text{TP + TN + FP + FN}}$ of each model, which is calculated as the ratio of correctly predicted instances to the total number of instances in the dataset [12]. Accuracy uses all classes of C, however, if the dataset is unbalanced, it is biased in favor of the larger class. Therefore, the main evaluation measure used in this thesis will be the **MCC (Matthews correlation coefficient)**, which is unaffected by unbalanced datasets. The MCC is defined as

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$

and its values range from $[-1, 1]$. The higher the value of the MCC, the better the classifier was in predicting both the positive and negative instances correctly. An MCC of 1 means perfect classification, and an MCC of -1 means perfect misclassification, while an MCC of 0 is equivalent to random classification [12]. While the MCC will be the main evaluation measure, the values for TPR, TNR, accuracy and precision will also be calculated for each test run.

5. Experimental Analysis

In this chapter, we provide comprehensive information about the datasets employed in our experiments and outline the experimental setup. Subsequently, we present the results derived from our analyses.

5.1. Setup

This section provides a comprehensive overview of the process we followed to create our dataset and describes in detail the scripts we developed to conduct our experiments.

5.1.1. Dataset

In order to compare our results with the work presented in [25], which forms the basis of this thesis, it would be ideal to use the same datasets. Kuhlmann et al. created a dataset consisting of 1000 argumentation frameworks (AFs) for training (*kwt-train*) and 1000 AFs for testing (*kwt-test*). Each AF poses a particular challenge for skeptical acceptance under preferred semantics and consists of 151 arguments. However, due to the specific goal of this dataset, the distribution of accepted and rejected arguments is highly imbalanced for certain semantics as can be seen in Table 3. Under DC-ST for instance, practically all arguments are rejected. This extreme unbalance mean that *kwt-train* and *kwt-test* were unsuitable for our experiments.

kwt-train				
	Yes	No	Share Yes	Share No
DC-PR	113539	37461	75%	25%
DC-CO	113539	37461	75%	25%
DC-ST	580	150420	0%	100%
DC-GR	16307	134693	11%	89%
DS-PR	69441	81559	46%	54%
DS-CO	16307	134693	11%	89%
DS-ST	142739	8261	95%	5%
DS-GR	16307	134693	11%	89%

kwt-test				
	Yes	No	Share Yes	Share No
DC-PR	112909	38091	75%	25%
DC-CO	112909	38091	75%	25%
DC-ST	596	150404	0%	100%
DC-GR	15898	135102	11%	89%
DS-PR	69115	81885	46%	54%
DS-CO	15898	135102	11%	89%
DS-ST	143013	7987	95%	5%
DS-GR	15898	135102	11%	89%

Table 3: Distribution of Accepted (Yes) and Rejected (No) Arguments for the *kwt-train* and *kwt-test* Datasets

Nevertheless, as pointed out by the authors in [25], it is advisable to ensure that the AFs used in a dataset are not too easy to solve; otherwise, there would be no benefit to using approximate methods, as the decrease in solving time would not outweigh the loss in accuracy. Therefore, we will retain the *kwt* datasets but augment them with additional graphs to achieve a better balance between accepted and rejected arguments.

To accomplish this, we utilized the *Probo Benchmark Suite* and the *AFBenchGen2* generators. The *Probo Benchmark Suite* provides three graph generators capable of generating AFs with different properties. The *GroundedGenerator* produces graphs

balanced-kwt-train				
	Yes	No	Share Yes	Share No
DC-PR	282430	170520	62%	38%
DC-CO	134703	318247	30%	70%
DC-ST	282430	170520	62%	38%
DC-GR	189277	263673	42%	58%
DS-PR	160128	292822	35%	65%
DS-CO	292926	160024	65%	35%
DS-ST	134703	318247	30%	70%
DS-GR	134703	318247	30%	70%

balanced-kwt-test				
	Yes	No	Share Yes	Share No
DC-PR	281388	171562	62%	38%
DC-CO	134047	318903	30%	70%
DC-ST	281388	171562	62%	38%
DC-GR	188665	264285	42%	58%
DS-PR	160061	292889	35%	65%
DS-CO	292744	160206	65%	35%
DS-ST	134047	318903	30%	70%
DS-GR	134047	318903	30%	70%

Table 4: Distribution of Accepted (Yes) and Rejected (No) Arguments for the *balanced-kwt-train* and *balanced-kwt-test* Datasets

with large grounded extensions, while the *SccGenerator* generates graphs with many strongly connected components. Lastly, the *StableGenerator* generates graphs with numerous stable, preferred, and complete extensions.

For both the training and testing datasets, we used the *GroundedGenerator* to generate 100 graphs, as they tend to exhibit a relatively balanced distribution of accepted and rejected arguments. We generated 50 additional graphs each using the *StableGenerator* and *SccGenerator*.

To match the number of arguments in the *kwt* datasets, we made slight modifications to the source code of the generators to ensure that all generated graphs contain 151 arguments.

In addition to the above generators, we also utilized the *AFBenchGen2* generator, which generates AFs using the *Barabási–Albert* (BA), *Erdős–Rényi* (ER), and *Watts–Strogatz* (WS) graph models. For the ER and WS models, we generated 50 graphs with 151 arguments each. Since BA graphs consistently exhibit a higher number of accepted arguments compared to rejected arguments among the six graph models, we generated 200 BA graphs to achieve a better balance in our dataset.

By combining these additional graphs with the kwt dataset, we obtained 2000 graphs for both the testing and training datasets. We refer to these datasets as *kwt-balanced-train* and *kwt-balanced-test*. In these datasets at least 30% of arguments are accepted for all semantics as can be seen in Table 4.

5.1.2. Experiments

In order to conduct our experiments, we utilized the Python libraries *scikit-learn* and *networkX*. We developed a script called *AF* that parses the *apx* files generated by the graph generators into *networkX* graphs and calculates the necessary graph properties for our experiments. Most of the required features are readily available in the *networkX* library. However, for certain properties such as symmetry, irreflexivity and the average node degree, we had to implement our own functions.

To determine symmetry we first check if the graph is weakly connected, ensuring the absence of isolated nodes. If this is true, we proceed to verify symmetry by checking if, for every edge from node a to node b, there is also an edge from b to

a. Irreflexivity is determined by examining whether any node in the graph has an edge to itself and in order to calculate the average degree of the nodes in the graph we sum up all the degrees and divide the total by the number of nodes.

Furthermore, the *AF* script is responsible for generating solutions for each graph under all examined semantics. To achieve this, we utilize the $\mu - toksia$[7] solver [33] as well as the $ArgSemSat$[8] solver [9] for grounded semantics.

A second script called *experiment*[9] handles the training and testing of the AFs. This script receives the paths to the training and testing datasets as well as the features to be used. It then assembles the relevant feature data and solution vectors for training and testing with the help of the *AF* script. All feature data is scaled using the *MinMaxScaler* module of *scikit-learn* to ensure that all feature values are in the range between 0 and 1.

For the training phase, we employed ML classifiers selected from the *scikit-learn* library. Specifically, we utilized the standard implementations for K-Nearest Neighbors (KNN), Complement Naive Bayes (NB) which is better suited for imbalanced datasets, Decision Trees (DTs), and Random Forests (RFs). For SVMs we used the linear implementation *LinearSVC*, which scales very well to large datasets.

Using these classifiers, we conducted extensive experiments on the *kwt-balanced-train* and *kwt-balanced-test* datasets to identify the most informative features for classifying AFs.

5.2. Results

To identify the most informative set of graph features, we started by first training and testing the classifiers on each individual feature. The results of these experiments, specifically the MCC values, are presented in Table 5 for credulous acceptance and in Table 6 for skeptical acceptance.

In both tables the highest MCC values per feature and semantics are marked in bold and the highest overall MCC value per semantics is marked red. Additionally each feature is ranked in its importance in classifying a given semantics. To obtain the rank we only focus on the overall MCCs values regardless of the classifier. Thus the feature that achieved the overall best MCC for a given semantics is assigned rank 1. While this approach is a simplification and there may be variations in the ranks for different classifiers (e.g., in Table 5, the closeness centrality feature received rank 2 for the DC-PR task, even though it seems to be uninformative for SVMs), in many cases, the maximal rank coincides with the individual ranks or is at least similar.

Our analysis reveals that degree centrality is the most informative feature across all semantics with closeness centrality following as the second most informative feature (see Table 8). Generally, the data suggests that centrality measures provide

[7] `https://bitbucket.org/andreasniskanen/mu-toksia/src/master/`

[8] `https://sourceforge.net/projects/argsemsat/`

[9] Both scripts as well as the datasets used can be found here: `https://github.com/sandra-hoffmann/ML-abstract-argumentation_thesis`

Classifier	Degree C.	Katz C.	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	Is SC	is Sym.	is Ir.
DC-PR											
KNN	0.775	**0.606**	**0.689**	**0.656**	0.246	0.409	0.391	0.000	**0.235**	0.000	0.000
NB	0.307	0.236	0.042	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.806	0.490	0.642	0.608	**0.339**	**0.424**	**0.433**	0.000	**0.235**	0.000	0.000
RF	**0.807**	0.597	0.678	0.618	**0.339**	**0.424**	0.429	0.000	**0.235**	0.000	0.000
SVM lin	0.243	0.080	0.000	0.305	0.314	0.000	0.000	0.000	**0.235**	0.000	0.000
Rank	1	4	2	3	7	6	5	10	8	10	10
DC-CO											
KNN	0.775	**0.606**	**0.689**	**0.656**	0.246	0.409	0.391	0.000	**0.235**	0.000	0.000
NB	0.307	0.236	0.042	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.807**	0.491	0.642	0.608	**0.339**	**0.424**	**0.433**	0.000	**0.235**	0.000	0.000
RF	**0.807**	0.597	0.678	0.618	**0.339**	**0.424**	0.432	0.000	**0.235**	0.000	0.000
SVM lin	0.243	0.080	0.000	0.305	0.314	0.000	0.000	0.000	**0.235**	0.000	0.000
Rank	1	4	2	3	7	6	5	10	8	10	10
DC-ST											
KNN	0.819	**0.613**	**0.709**	0.486	0.440	0.799	0.804	**0.513**	**0.106**	0.000	**0.733**
NB	0.104	0.225	-0.032	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.838**	0.501	0.641	0.503	**0.515**	**0.810**	**0.812**	0.000	0.000	0.000	**0.733**
RF	**0.838**	0.609	0.686	**0.504**	**0.515**	**0.810**	0.810	0.000	0.000	0.000	**0.733**
SVM lin	0.716	0.000	0.578	0.000	0.336	-0.160	0.707	0.000	0.000	0.000	**0.733**
Rank	1	6	5	9	7	3	2	8	10	11	4
DC-GR											
KNN	0.683	0.556	0.654	**0.520**	**0.443**	0.056	**0.260**	0.000	0.000	0.000	0.000
NB	0.154	0.135	-0.048	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.689	0.463	0.646	-0.027	0.000	**0.262**	**0.260**	0.000	0.000	0.000	0.000
RF	0.689	**0.562**	**0.656**	-0.027	0.000	**0.262**	0.259	0.000	0.000	0.000	0.000
SVM lin	**0.718**	0.000	0.509	0.000	0.000	0.103	0.000	0.000	0.000	0.000	0.000
Rank	1	3	2	4	5	6	7	10	10	10	10

Table 5: MCCs Values Achieved When Testing the Features Separately for Credulous Acceptance Under Several Semantics Using the *balanced-kwt-train* and *balanced-kwt-test* Datasets

more valuable information compared to graph-based features. With the exception of stable semantics, the top four ranks are consistently occupied by centrality measures. This result aligns with our expectation since graph-based features are identical for every node in a graph. Consequently, if used as the sole feature, a classifier would assign the same label to each node in the graph. However, the features proposed by Vallati et al. [39], particularly the number of SCCs and the average degree, perform reasonably well. In most cases, the number of SCCs is even more informative than the size of the SCC to which an argument belongs. Notably, the classification results DTs and RFs for DC-ST are impressive.

The features proposed by Doumbouya seem to provide limited informative value for our classification task, except for the irreflexivity feature in stable semantics. Our data clearly indicates that graph-based features perform exceptionally well in the context of stable semantics. This observation can be attributed to the phenomenon described in 4.2.1, where a graph without a stable extension results in all arguments being credulously rejected but skeptically accepted. As a consequence, all arguments in such a graph belong to the same acceptance class, thereby mitigating any disadvantage for graph-based features compared to node-based features. Hence,

Classifier	Degree C.	Katz C.	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	Is SC	is Sym.	is Ir.
DS-PR											
KNN	0.629	0.612	0.688	**0.544**	0.484	0.070	0.116	0.000	0.000	0.000	0.000
NB	0.302	0.240	0.082	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.716	0.520	0.692	0.523	**0.489**	**0.145**	0.137	0.000	0.000	0.000	0.000
RF	**0.717**	**0.616**	**0.708**	0.528	**0.489**	**0.145**	**0.138**	0.000	0.000	0.000	0.000
SVM lin	0.261	0.401	0.297	0.436	0.324	0.003	0.000	0.000	0.000	0.000	0.000
Rank	1	3	2	4	5	6	7	10	10	10	10
DS-CO											
KNN	0.683	0.556	0.654	**0.520**	**0.443**	0.056	**0.260**	0.000	0.000	0.000	0.000
NB	0.154	0.135	-0.048	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.689	0.461	0.646	-0.027	0.000	**0.262**	**0.260**	0.000	0.000	0.000	0.000
RF	0.689	**0.562**	**0.657**	-0.027	0.000	**0.262**	**0.260**	0.000	0.000	0.000	0.000
SVM lin	**0.718**	0.000	0.509	0.000	0.000	0.103	0.000	0.000	0.000	0.000	0.000
Rank	1	3	2	4	5	6	7	10	10	10	10
DS-ST											
KNN	0.326	0.418	0.464	**0.457**	0.308	0.347	0.461	**0.359**	0.000	0.000	**0.511**
NB	0.002	0.126	0.182	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.581**	0.325	0.523	0.411	**0.484**	**0.489**	0.468	0.000	0.000	0.000	**0.511**
RF	**0.581**	**0.427**	**0.542**	0.429	**0.484**	**0.489**	**0.469**	0.000	0.000	0.000	**0.511**
SVM lin	0.000	0.008	0.000	0.136	0.000	0.109	0.000	0.000	0.000	0.000	0.000
Rank	1	8	2	7	5	4	6	9	10	10	3
DS-GR											
KNN	0.683	0.556	0.654	**0.520**	**0.443**	0.056	**0.260**	0.000	0.000	0.000	0.000
NB	0.154	0.135	-0.048	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.689	0.462	0.646	-0.027	0.000	**0.262**	**0.260**	0.000	0.000	0.000	0.000
RF	0.689	**0.561**	**0.657**	-0.027	0.000	**0.262**	0.252	0.000	0.000	0.000	0.000
SVM lin	**0.718**	0.000	0.509	0.000	0.000	0.103	0.000	0.000	0.000	0.000	0.000
Rank	1	3	2	4	5	6	7	10	10	10	10

Table 6: MCCs Values Achieved When Testing the Features Separately for Skeptical Acceptance Under Several Semantics Using the *balanced-kwt-train* and *balanced-kwt-test* Datasets

we can infer that features related to the number of SCCs, average degree, and irreflexivity seem to aid in predicting such scenarios.

In terms of individual classifier performance, it is evident that the RF classifier outperformed the others. It achieved the best result in 36 instances across all semantics. On the other hand, the KNN and DT classifiers lagged behind, with the best results obtained in 26 and 25 instances, respectively. The NB classifier showed the poorest performance, while the SVM classifier, although achieving the overall best result for DC-GR, DS-CO, and DS-GR using the degree feature, exhibited inconsistent results across other features and semantics.

To determine whether degree centrality also emerges as the most informative feature for individual graph types, we conducted further analysis. We divided the *balanced-kwt-test* dataset into six separate test sets, each containing a single graph type. The *kwt* graph type was excluded from this analysis as we used the *kwt-test* dataset, which exclusively consists of *kwt* graphs, for evaluation purposes. We can infer from examining the results of these experiments (Table 57) that overall degree is the most informative feature for all semantics for *kwt* type graphs.

For each graph type, we tested each feature independently using the *balanced-kwt* dataset for training. The results of these tests are presented in Table 7.

Upon examination of the table, we observe that degree centrality often emerges as the most important feature. Across various graph types, the degree centrality feature consistently yielded high MCC scores, indicating its significance in classification or prediction tasks.

However, for stable graphs, neither DS-PR nor DS-ST show a clear feature that could be identified as the most important, as different features produced the best results for each ML classifier when classifying this graph type. Furthermore, in the case of WS and SCC graphs, no combination of features and classifiers was able to achieve an MCC value other than zero for stable semantics.

Overall however, these results show that degree centrality seems to be the most important feature not only for the whole dataset but also for the individual graph types.

	DC-PR	DS-PR	DC-CO	DS-CO	DC-ST	DS-ST	DC-GR	DS-GR
WS	Size SCC	Size SCC	Size SCC	Degree	-	-	Degree	Degree
SCC	Degree	Degree	Degree	Degree	-	-	Degree	Degree
Grounded	Degree	Closeness	Degree	Degree	Degree	Closeness	Degree	Degree
ERAF	Degree	Degree	Degree	Degree/ Closeness	Degree	Size SCC	Closeness/ Degree	Closeness/ Degree
Barabsi	Degree	Degree/ Betweenness	Degree	Degree	Degree	Betweenness	Degree	Degree
Stable	Degree	?	Degree	Degree	Degree	?	Degree	Degree

Table 7: MCCs Values Achieved When Testing the Features Separately Training with the *balanced-kwt-train* Dataset and Testing Only the WS-Graphs of the *balanced-kwt-test* Dataset

After analysing the individual performance of our selected features, the next step in our analysis was to assess how their combination can enhance classification accuracy. To determine the order in which to combine the features, we ranked them based on their helpfulness for each semantics. This ranking allowed us to calculate the average rank for each feature and thus determined the order of testing.

Starting with the two features that achieved the best results when used independently, namely degree centrality and closeness centrality, we subsequently added the remaining feature in the order of their importance.

The results of testing the feature combinations can be found in Table 9 for credulous acceptance tasks and in Table 10 for skeptical acceptance tasks.

Upon analyzing the data, we observe that combining the four tested centrality measures consistently improves the MCC values for all classifiers. In most cases, the prediction quality increased with each added centrality measure. Notably, the RF classifier demonstrated superior performance across all semantics, achieving the highest overall MCC values. The prediction quality for RF often improved with the addition of each feature, and even when it remained consistent, the added features did not introduce noise. Only Symmetry, the final feature added to our feature com-

Classifier	Degree C.	Katz C.	Closeness C.	Betweenness C.	Size SCCs	SCCs	AVG Deg.	is Aperiod	Is SC	is Sym.	is Ir.
DC-PR	1	4	2	3	7	6	5	10	8	10	10
DC-CO	1	4	2	3	7	6	5	10	8	10	10
DC-ST	1	6	5	9	7	3	2	8	10	11	4
DC-GR	1	3	2	4	5	6	7	10	10	10	10
DS-PR	1	3	2	4	5	6	7	10	10	10	10
DS-CO	1	3	2	4	5	6	7	10	10	10	10
DS-ST	1	8	2	7	5	4	6	9	10	10	3
DS-GR	1	3	2	4	5	6	7	10	10	10	10
AVG Rank	1,00	4,25	2,38	4,75	5,75	5,38	5,75	9,63	9,50	10,13	8,38
Order of Testing	1	3	2	4	7	5	6	10	9	11	8

Table 8: Rank Values for Classification with the *balanced-kwt-train* and *balanced-kwt-test* Datasets

bination, did not contribute to further improvement in prediction quality. However, considering that this feature was also uninformative when used individually and was included in the combination solely for the sake of completeness, these results are not surprising.

Based on these findings, we can identify the following feature combination as the most suitable for classifying AFs using the RF classifier: Degree centrality, closeness centrality, Katz centrality, betweenness centrality, number of SCCs, average degree, size of SCC, irreflexivity, strong connectivity, and aperiodicity. Similarly, for the DT and SVM classifiers, this feature combination either achieved the highest prediction quality or did at least not reduce the quality considerably. The results for the KNN and NB classifiers are more inconclusive. While they achieved the best results using only centrality measures for DC-PR, DC-CO, and DS-PR, they benefited from adding graph-based features for other semantics.

Given the considerably better prediction quality of the RF classifier compared to NB and KNN, further investigation into the factors influencing the prediction of these two classifiers seems unnecessary. Therefore, we will define the aforementioned feature combination that achieved the best result with the RF classifier as our ideal feature combination (IFC) over all semantics.

To gain a deeper understanding of the impact of the IFC on the classification quality, we compare the full set of evaluation measures obtained from our experiments. Table 11 holds these measures for classification using degree only, while Table 11 contains the results for classification using the IFC. We observe that in many cases IFC is able to improve both the TNR and the TPR resulting in increased precision and accuracy. Notably for DC-GR, DS-CO and DS-GR (which coincide in this dataset) classification with the IFC has a balancing effect on the prediction quality. While when using RF as classifier and only degree as feature, the classification tended to favor classifying arguments as rejected, as evident by the perfect TNR but low TNR rate, using the IFC the TPR rate could be improved by 30 points, while keeping the TNR rate reasonably high. Similar effects can be obsevered for other semantics and classifiers throught this dataset.

	Feature Combinations										
	1	2	3	4	5	6	7	8	9	10	11
Classifier	**Degree C.**	**1+ Closeness C.**	**2+ Katz C.**	**3+ Betweenness C.**	**4+ SCCs**	**5+ AVG Deg.**	**6+ Size SCCs**	**7+ is Ir.**	**8+ Is SC**	**9+ Is Aperiod**	**10 + is Sym.**
DC-PR											
KNN	0.775	0.849	0.846	**0.852**	0.851	0.851	0.851	0.849	0.848	0.849	0.849
NB	0.307	0.121	0.244	**0.544**	0.308	0.376	0.364	0.317	0.311	0.316	0.316
DT	0.806	0.814	0.830	0.845	0.851	0.853	0.863	0.863	**0.864**	0.863	0.861
RF	0.807	0.842	0.885	0.898	0.901	0.901	**0.909**	0.909	0.909	0.908	0.909
SVM lin	0.243	0.355	0.416	0.599	0.618	0.621	0.630	**0.637**	0.637	0.622	0.622
DC-CO											
KNN	0.775	0.849	0.846	**0.852**	0.851	0.851	0.851	0.849	0.848	0.849	0.849
NB	0.307	0.121	0.244	**0.544**	0.308	0.376	0.364	0.317	0.311	0.316	0.316
DT	0.807	0.813	0.830	0.845	0.851	0.852	**0.863**	0.863	0.863	0.863	0.863
RF	0.807	0.841	0.886	0.898	0.901	0.901	**0.909**	0.909	0.909	0.909	0.909
SVM lin	0.243	0.355	0.416	0.599	0.618	0.621	0.630	**0.637**	0.637	0.622	0.622
DC-ST											
KNN	0.819	0.855	0.858	0.863	0.862	0.863	**0.864**	0.864	0.864	0.863	0.863
NB	0.104	0.497	0.642	0.641	0.698	0.780	0.778	**0.786**	0.784	0.765	0.765
DT	0.838	0.817	0.837	0.851	0.856	0.859	0.869	**0.872**	0.872	0.869	0.869
RF	0.838	0.843	0.889	0.901	0.904	0.902	0.910	0.912	**0.913**	0.913	0.913
SVM lin	0.716	0.646	0.682	0.676	0.670	0.772	0.767	**0.815**	0.815	0.814	0.814
DC-GR											
KNN	0.683	0.752	0.761	0.764	0.764	0.764	**0.770**	0.768	0.768	0.769	0.769
NB	0.154	0.256	0.533	0.540	**0.584**	0.579	0.578	0.584	0.584	0.580	0.580
DT	0.689	0.739	0.764	0.768	0.774	0.777	0.790	0.791	0.790	**0.794**	0.792
RF	0.689	0.756	0.805	0.806	0.812	0.812	0.829	0.830	0.830	**0.831**	0.831
SVM lin	0.718	0.748	0.761	0.759	0.761	0.766	**0.768**	0.768	0.768	0.780	0.780

Table 9: MCCs Values Achieved when Testing the Features in Combination for Credulous Acceptance under Several Semantics using the *balanced-kwt-train* and *balanced-kwt-test* Datasets

Although we faced time constraints that prevented us from fully examining the runtime results of our experiments, a notable observation emerged from analyzing the detailed experiments. While classifying with the Random Forest (RF) algorithm leads to the most accurate results, it also requires the longest training time and the second longest testing time. Interestingly, using the Decision Tree (DT) algorithm significantly reduces the runtime while maintaining a high prediction quality.

In the next chapter we aim to evaluate our finding by testing the IFC on the datasets used in [25].

	Feature Combinations										
	1	2	3	4	5	6	7	8	9	10	11
Classifier	**Degree**	**1+ Close-ness**	**2+ Katz**	**3+ Between-ness**	**4+ SCCs**	**5+ AVG Deg.**	**6+ Size SCCs**	**7+ is Ir.**	**8+ is SC**	**9+ is Aperiod**	**10 + is Sym.**
DS-PR											
KNN	0.629	0.766	0.771	**0.774**	0.770	0.763	0.766	0.765	0.765	0.765	0.765
NB	0.302	0.242	0.380	**0.465**	0.334	0.449	0.407	0.359	0.357	0.353	0.353
DT	0.716	0.757	0.748	0.749	0.749	0.749	0.759	0.761	0.760	**0.764**	0.762
RF	0.717	0.776	0.799	0.801	0.801	0.801	0.814	0.814	**0.815**	0.815	0.816
SVM lin	0.261	0.240	0.466	0.619	0.620	0.638	0.647	0.658	0.656	**0.659**	0.659
DS-CO											
KNN	0.683	0.752	0.761	0.764	0.764	0.764	0.770	0.768	0.768	**0.769**	0.769
NB	0.154	0.256	0.533	0.540	0.584	0.579	0.578	**0.584**	0.584	0.580	0.580
DT	0.689	0.739	0.764	0.767	0.773	0.774	0.788	0.789	0.791	**0.792**	0.792
RF	0.689	0.756	0.805	0.806	0.812	0.812	0.830	0.829	0.829	0.830	**0.831**
SVM lin	0.718	0.748	0.761	0.759	0.761	0.766	0.768	0.768	0.768	**0.780**	0.780
DS-ST											
KNN	0.326	0.540	0.635	**0.648**	0.645	0.638	0.640	0.643	0.643	0.640	0.640
NB	0.002	0.200	0.395	0.418	0.413	0.435	0.430	0.503	0.532	**0.534**	0.534
DT	0.581	0.580	0.596	0.608	0.612	0.609	0.625	**0.629**	0.629	0.626	0.627
RF	0.581	0.614	0.596	0.691	0.691	0.685	0.700	0.705	0.705	**0.708**	0.705
SVM lin	0.000	0.000	0.163	0.233	0.260	0.338	0.368	0.486	**0.554**	0.554	0.554
DS-GR											
KNN	0.683	0.752	0.761	0.764	0.764	0.764	0.770	0.768	0.768	**0.769**	0.769
NB	0.154	0.256	0.533	0.540	0.584	0.579	0.578	**0.584**	0.584	0.580	0.580
DT	0.689	0.738	0.764	0.767	0.773	0.775	0.790	0.790	0.790	**0.792**	0.791
RF	0.689	0.756	0.806	0.806	0.812	0.812	0.829	0.829	0.829	**0.830**	0.830
SVM lin	0.718	0.748	0.761	0.759	0.761	0.766	0.768	0.768	0.768	**0.780**	0.780

Table 10: MCCs Values Achieved when Testing the Features in Combination for Skeptical Acceptance under Several Semantics using the *balanced-kwt-train* and *balanced-kwt-test* Datasets

Degree							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.775	0.906	0.937	0.835	0.931	0.255	13.561
NB	0.307	0.676	0.693	0.637	0.819	0.539	0.006
DT	0.806	0.920	0.949	0.850	0.937	0.295	0.016
RF	0.807	0.920	0.949	0.850	0.937	5.277	1.411
SVM lin	0.243	0.731	0.985	0.130	0.728	1.136	0.004
DC-CO							
KNN	0.775	0.906	0.937	0.835	0.931	0.264	13.537
NB	0.307	0.676	0.693	0.637	0.819	0.535	0.007
DT	0.807	0.920	0.949	0.850	0.938	0.276	0.017
RF	0.807	0.920	0.950	0.850	0.937	5.421	1.441
SVM lin	0.243	0.731	0.985	0.130	0.728	1.163	0.003
DC-ST							
KNN	0.819	0.920	0.894	0.932	0.864	0.249	13.560
NB	0.104	0.623	0.340	0.759	0.404	0.531	0.007
DT	0.838	0.929	0.899	0.943	0.884	0.288	0.015
RF	0.838	0.929	0.899	0.943	0.884	5.387	1.438
SVM lin	0.716	0.852	0.959	0.801	0.699	0.799	0.003
DC-GR							
KNN	0.683	0.878	0.832	0.892	0.699	0.255	13.771
NB	0.154	0.690	0.371	0.787	0.345	0.566	0.008
DT	0.689	0.893	0.540	1.000	1.000	0.335	0.011
RF	0.689	0.893	0.540	1.000	1.000	4.699	0.954
SVM lin	0.718	0.872	0.964	0.844	0.652	0.429	0.005
DS-PR							
KNN	0.629	0.816	0.828	0.807	0.753	0.397	15.732
NB	0.302	0.671	0.430	0.842	0.659	0.619	0.052
DT	0.716	0.861	0.724	0.957	0.923	0.301	0.017
RF	0.717	0.861	0.724	0.957	0.923	6.419	1.473
SVM lin	0.261	0.607	0.762	0.497	0.517	0.941	0.009
DS-CO							
KNN	0.683	0.878	0.832	0.892	0.699	0.268	15.696
NB	0.154	0.690	0.371	0.787	0.345	0.528	0.008
DT	0.689	0.893	0.540	1.000	1.000	0.278	0.011
RF	0.689	0.893	0.540	1.000	1.000	5.756	0.938
SVM lin	0.718	0.872	0.964	0.844	0.652	0.428	0.008
DS-ST							
KNN	0.326	0.736	0.825	0.498	0.815	0.257	14.097
NB	0.002	0.545	0.597	0.405	0.729	0.560	0.006
DT	0.581	0.813	0.814	0.812	0.921	0.281	0.019
RF	0.581	0.814	0.815	0.811	0.920	5.173	1.769
SVM lin	0.000	0.728	1.000	0.000	0.728	1.075	0.006
DS-GR							
KNN	0.683	0.878	0.832	0.892	0.699	0.259	13.703
NB	0.154	0.690	0.371	0.787	0.345	0.525	0.006
DT	0.689	0.893	0.540	1.000	1.000	0.266	0.011
RF	0.689	0.893	0.540	1.000	1.000	4.799	0.980
SVM lin	0.718	0.872	0.964	0.844	0.652	0.435	0.004

Table 11: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Degree

Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality, No of SCCs, AVG Degree, SCC Size, Irreflexitivity, Strong Connectivity, Aperiodicity							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.849	0.937	0.957	0.890	0.954	0.326	13.747
NB	0.316	0.669	0.665	0.678	0.830	0.534	0.011
DT	0.863	0.943	0.960	0.902	0.959	1.419	0.021
RF	0.908	0.961	0.968	0.945	0.977	21.948	1.870
SVM lin	0.622	0.849	0.955	0.599	0.849	5.097	0.006
DC-CO							
KNN	0.849	0.937	0.957	0.890	0.954	0.323	13.700
NB	0.316	0.669	0.665	0.678	0.830	0.535	0.007
DT	0.863	0.943	0.960	0.902	0.959	1.425	0.021
RF	0.909	0.962	0.968	0.946	0.977	21.553	1.885
SVM lin	0.622	0.849	0.955	0.599	0.849	5.122	0.006
DC-ST							
KNN	0.863	0.940	0.915	0.952	0.901	0.328	13.997
NB	0.765	0.877	0.985	0.825	0.730	0.514	0.006
DT	0.869	0.942	0.915	0.956	0.908	1.275	0.019
RF	0.913	0.962	0.936	0.974	0.946	20.830	1.790
SVM lin	0.814	0.912	0.949	0.895	0.812	3.228	0.005
DC-GR							
KNN	0.769	0.919	0.800	0.955	0.842	0.309	13.756
NB	0.580	0.766	0.985	0.700	0.498	0.523	0.006
DT	0.794	0.926	0.840	0.952	0.842	1.206	0.020
RF	0.831	0.941	0.841	0.971	0.898	18.847	1.635
SVM lin	0.780	0.923	0.795	0.962	0.864	1.170	0.005
DS-PR							
KNN	0.765	0.886	0.852	0.911	0.871	0.321	13.816
NB	0.353	0.662	0.769	0.586	0.568	0.559	0.007
DT	0.764	0.885	0.861	0.902	0.862	1.478	0.023
RF	0.815	0.910	0.879	0.933	0.902	22.077	2.177
SVM lin	0.659	0.835	0.713	0.921	0.865	3.425	0.005
DS-CO							
KNN	0.769	0.919	0.800	0.955	0.842	0.310	13.715
NB	0.580	0.766	0.985	0.700	0.498	0.520	0.006
DT	0.792	0.926	0.839	0.952	0.842	1.184	0.019
RF	0.830	0.940	0.840	0.971	0.897	18.831	1.624
SVM lin	0.780	0.923	0.795	0.962	0.864	1.150	0.005
DS-ST							
KNN	0.640	0.858	0.904	0.735	0.902	0.309	13.694
NB	0.534	0.759	0.711	0.886	0.944	0.529	0.006
DT	0.626	0.852	0.899	0.727	0.898	1.358	0.020
RF	0.708	0.884	0.921	0.786	0.920	25.394	2.404
SVM lin	0.554	0.810	0.830	0.755	0.901	6.036	0.008
DS-GR							
KNN	0.769	0.919	0.800	0.955	0.842	0.305	13.636
NB	0.580	0.766	0.985	0.700	0.498	0.518	0.006
DT	0.792	0.926	0.839	0.952	0.841	1.190	0.019
RF	0.830	0.941	0.840	0.971	0.897	18.385	1.669
SVM lin	0.780	0.923	0.795	0.962	0.864	1.134	0.006

Table 12: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality, No of SCCs, AVG Degree, SCC Size, Irreflexitivity, Strong Connectivity, Aperiodicity

5.3. Evaluation

In order to evaluate whether the IFC proposed in the previous section is able to improve the prediction quality of ML classifiers in general, we recreated our experiments using the *kwt-train* and *kwt-test* datasets. As explained in 5.1.1 these datasets are severely unbalanced for stable semantics, thus we disregard the results for these semantics. The results can be seen in Table 13 for credulous and in Table 14 for skeptical acceptance tasks.

		Feature Combinations							
		1	2	3	4	5	6	7	8
Classifier	**IFC**	**Degree C.**	**1+ Closeness C.**	**2+ Katz C.**	**3+ AVG Deg.**	**4+ SCCs**	**5+ Betweenness C.**	**6+ Size SCCs**	**7+ is Aperiod**
DC-PR									
KNN	**0.987**	0.975	0.984	**0.987**	0.987	0.987	0.987	0.987	0.987
NB	0.549	0.281	0.432	**0.657**	0.476	0.491	0.568	0.534	0.549
DT	0.974	0.977	**0.981**	0.972	0.975	0.974	0.974	0.973	0.974
RF	**0.987**	0.977	0.984	**0.987**	0.987	0.987	0.987	0.987	0.987
SVM lin	**0.975**	0.230	0.729	0.869	0.870	0.903	0.966	**0.975**	0.975
DC-CO									
KNN	**0.987**	0.975	0.984	**0.987**	0.987	0.987	0.987	0.987	0.987
NB	0.549	0.281	0.432	**0.657**	0.476	0.491	0.568	0.534	0.549
DT	0.975	0.977	**0.981**	0.973	0.974	0.974	0.974	0.975	0.974
RF	**0.987**	0.977	0.984	**0.987**	0.987	0.987	0.987	0.987	0.987
SVM lin	**0.975**	0.230	0.729	0.869	0.870	0.903	0.966	0.975	0.975
DC-ST									
KNN	-0.000	0.000	0.000	-0.000	-0.001	-0.001	-0.000	-0.000	-0.000
NB	0.042	0.034	0.019	0.031	0.031	0.025	0.038	0.038	0.041
DT	0.019	0.033	0.011	0.025	0.023	0.026	0.019	0.017	0.023
RF	-0.000	0.020	0.004	-0.001	-0.000	-0.000	-0.000	-0.001	-0.000
SVM lin	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DC-GR									
KNN	**1.000**	0.971	0.968	0.975	0.971	0.995	0.995	0.994	**1.000**
NB	**0.970**	-0.157	-0.142	0.855	0.728	**0.970**	0.970	0.970	0.970
DT	**1.000**	0.976	0.978	0.968	0.971	0.994	0.994	0.994	**1.000**
RF	**1.000**	0.976	0.978	0.973	0.980	0.996	0.995	0.996	**1.000**
SVM lin	**0.971**	0.938	0.970	0.970	0.970	0.970	0.970	0.970	**0.971**

Table 13: MCCs Values Achieved when Testing the Features in Combination for Credulous Acceptance under Several Semantics using the *kwt-train* and *kwt-test* Datasets

We can observe that, in most cases, classifying with the IFC achieves better results compared to classifying with degree centrality alone. Notably, for grounded semantics and skeptical complete semantics, the IFC classification was able to perfectly predict argument acceptance. However, for the DS-PR task (which was considered in [25]), the results with IFC were inferior to those obtained with degree centrality alone. When comparing our results to [25], we must acknowledge that the use of the *ComplementNB* classifier has yielded inferior results compared to the MCC of 0.868 reported in [25]. This discrepancy could be attributed to the relatively balanced nature of the *kwt* datasets for DS-PR, suggesting that a different NB classifier, such as

Feature Combinations									
		1	2	3	4	5	6	7	8
Classifier	**IFC**	**Degree C.**	**1+ Closeness C.**	**2+ Katz C.**	**3+ AVG Deg.**	**4+ SCCs**	**5+ Betweenness C.**	**6+ Size SCCs**	**7+ is Aperiod**
DS-PR									
KNN	0.898	0.848	**0.916**	0.909	0.898	0.898	0.898	0.898	0.898
NB	**0.727**	0.377	0.621	0.649	0.576	0.725	**0.727**	0.726	0.727
DT	0.872	**0.923**	0.922	0.871	0.868	0.870	0.870	0.870	0.872
RF	0.904	**0.924**	**0.924**	0.899	0.901	0.903	0.900	0.902	0.903
SVM lin	**0.912**	0.889	0.901	0.906	0.906	0.908	0.910	**0.912**	0.912
DS-CO									
KNN	**1.000**	0.971	0.968	0.975	0.971	0.995	0.995	0.994	**1.000**
NB	0.970	-0.157	-0.142	0.855	0.728	0.970	0.970	0.970	0.970
DT	**1.000**	0.976	0.978	0.968	0.971	0.994	0.994	0.994	**1.000**
RF	**1.000**	0.976	0.978	0.974	0.979	0.996	0.996	0.996	**1.000**
SVM lin	0.971	0.938	0.970	0.970	0.970	0.970	0.970	0.970	0.971
DS-ST									
KNN	0.072	0.014	0.023	0.052	0.071	0.071	0.072	0.072	0.072
NB	0.077	-0.010	-0.006	0.053	0.057	0.090	0.090	0.085	0.077
DT	0.081	0.007	0.006	0.046	0.068	0.079	0.079	0.080	0.083
RF	0.085	0.007	0.007	0.050	0.084	0.084	0.099	0.091	0.099
SVM lin	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DS-GR									
KNN	**1.000**	0.971	0.968	0.975	0.971	0.995	0.995	0.994	**1.000**
NB	0.970	-0.157	-0.142	0.855	0.728	0.970	0.970	0.970	0.970
DT	**1.000**	0.976	0.978	0.968	0.973	0.994	0.994	0.994	**1.000**
RF	**1.000**	0.976	0.978	0.974	0.979	0.996	0.996	0.996	**1.000**
SVM lin	0.971	0.938	0.970	0.970	0.970	0.970	0.970	0.970	0.971

Table 14: MCCs Values Achieved when Testing the Features in Combination for Skeptical Acceptance under Several Semantics using the *kwt-train* and *kwt-test* Datasets

GaussianNB, might be more suitable in this scenario.

In order to investigate whether another feature combination could improve the classification performance on this dataset, we conducted the same test chain as we did for our balanced dataset. We tested the features separately and then in combination, ranking them based on their individual usefulness (see Tables 55, 56, and 57).

Upon analyzing the results, we see that for this dataset, contrary to the *balanced-kwt* datasets, the average degree is more informative than betweenness centrality. Additionally, the strong connectivity and irreflexivity features of a graph appear to be uninformative. However, when classifying the *kwt-test* dataset with its ideal feature combination, the classification performance remained the same as when using the IFC. Therefore, we can conclude that the IFC is also helpful in classifying this dataset. It is worth noting that in this particular case, the degree centrality and closeness centrality seem to align so well with the class distribution for DS-PR that adding any additional feature seems to introduce noise to the classification.

We continued our evaluation by testing the *pbbg* datasets. The results can be seen in Table 15 for the credulous acceptance tasks and in Table 16 for the skeptical acceptance tasks. We observe that the IFC was able to classify the data better than degree centrality alone for all tasks. Once again, the RF classifier achieved the best performance, and with the exception of the DS-CO task, it classified the data as well as or even better than the feature combination obtained by ranking the features individually and combining them based on their average rank. For the DS-CO task, classifying with the ideal features derived from individual testing and combining based on average rank resulted in an MCC that was 0.04 points higher than the result obtained using the IFC.

Our investigation thus far has demonstrated that the IFC consistently enhanced the classification quality compared to using only the degree feature. Although not all features within the IFC are informative for every dataset, we observed that when employing the RF classifier, the inclusion of these noninformative features does not significantly diminish the prediction quality. We can thus conclude that our IFC is useful when classifying AFs using ML and that it archieves the best results using RF.

However, all our datasets were created using graph generators, resulting in training and testing graphs that shared similar properties. To thoroughly evaluate the IFC's performance on real-world data, in our next experiment we us the *ICCMA-450* dataset. This dataset consists of real-world examples with varying sizes and its graphs were utilized as benchmarks in the 2017 ICCMA competition[10]. We employed this dataset for testing while training the classifiers with the *pbbg-train* dataset as was done in [25]. The results are presented in Table 17 and Table 18.

This particular test proved to be the most challenging, as the training data significantly differed from the testing data. Notably, for the first time, the RF classifier did not deliver the best results. Surprisingly, the NB classifier emerged as the superior choice for this specific dataset for several semantics, showcasing the IFC's ability to substantially improve the accuracy of the NB classifier in constrast to using only the degree centrality as feature. The RF classifier however was not able to improve the classification accuracy.

These results suggest the importance of training and testing data sharing similar properties. To test this hypothesis further, we split the *ICCMA-450* dataset into a training set (*ICCMA-train*) and a testing set (*ICCMA-test*). Given the limited size of the dataset (450 graphs), we adopted a 90:10 split, similar to the *pbbg* dataset, randomly selecting 45 graphs to compose our testing set (*ICCMA-test*), while the remaining 405 graphs formed our training set (*ICCMA-train*). The result of this experiment can be found in Table 19 and Table 20.

Upon analyzing this dataset, we observe that the RF classifier once again produced the most accurate results and demonstrated an enhanced prediction quality across all semantics. These findings support our hypothesis of the importance of using training data that shares structural similarities with the testing data.

Our evaluation has provided compelling evidence that the proposed IFC played a

[10]`http://argumentationcompetition.org/2017/`

Feature Combinations								
		1	2	3	4	5	6	7
Classifier	**IFC**	**Degree C.**	**1+ Closeness C.**	**2+ Betweenness C.**	**3+ AVG Deg.**	**4+ Katz C.**	**5+ size SCC**	**6+ SCCs**
DC-PR								
KNN	0.696	0.554	0.645	0.682	**0.703**	0.697	0.702	0.699
NB	0.455	0.302	0.338	0.335	0.350	0.429	0.451	**0.465**
DT	0.657	0.607	0.640	0.625	0.643	0.646	0.652	**0.664**
RF	**0.761**	0.607	0.667	0.697	0.724	0.743	0.755	**0.761**
SVM lin	**0.624**	0.616	0.614	0.610	0.610	0.614	0.613	0.614
DC-CO								
KNN	0.696	0.554	0.645	0.682	**0.703**	0.697	0.702	0.699
NB	0.455	0.302	0.338	0.335	0.350	0.429	0.451	**0.465**
DT	0.659	0.607	0.641	0.627	0.645	0.648	0.651	**0.662**
RF	**0.761**	0.607	0.665	0.696	0.726	0.744	0.754	0.760
SVM lin	**0.624**	0.616	0.614	0.610	0.610	0.614	0.613	0.614
DC-ST								
KNN	**0.675**	0.537	0.599	0.640	0.670	0.664	0.671	0.672
NB	0.443	0.274	0.313	0.310	0.331	0.416	0.447	**0.450**
DT	**0.619**	0.586	0.608	0.596	0.615	0.590	0.605	0.605
RF	0.729	0.586	0.634	0.662	0.689	0.707	0.719	**0.730**
SVM lin	**0.615**	0.590	0.591	0.590	0.589	0.607	0.607	0.606
DC-GR								
KNN	0.791	0.665	0.767	0.777	0.794	0.771	0.779	**0.792**
NB	0.545	0.380	0.404	0.405	0.408	0.509	0.520	**0.560**
DT	0.741	0.702	**0.770**	0.761	0.755	0.744	0.749	0.749
RF	**0.845**	0.702	0.787	0.802	0.809	0.832	0.836	**0.845**
SVM lin	**0.755**	0.702	0.693	0.702	0.695	0.706	0.743	0.727

Table 15: MCCs Values Achieved when Testing the Features in Combination for Credulous Acceptance under Several Semantics using the *PBBG-Train* and *PBBG-Test* Datasets

crucial role in improving the prediction quality, not only in our original *balanced-kwt* datasets but also in various other datasets with different sizes and structures. While selecting only informative features specific to each dataset may yield slightly better results in some cases, employing the IFC alongside the RF classifier consistently achieved comparable MCC values.

These findings highlight the generalizability of our approach, as it remains robust and effective in different scenarios, without heavily relying on dataset-specific feature selection as long as the training data does not differ significantly from the testing data.

Feature Combinations								
		1	2	3	4	5	6	7
Classifier	**IFC**	**Degree C.**	**1+ Close-ness C.**	**2+ Between-ness C.**	**3+ AVG Deg.**	**4+ Katz C.**	**5+ size SCC**	**6+ SCCs**
DS-PR								
KNN	0.716	0.609	0.695	0.707	**0.721**	0.707	0.711	**0.722**
NB	0.457	0.365	0.378	0.377	0.382	0.443	0.438	**0.472**
DT	**0.677**	0.630	0.691	0.666	0.667	0.660	0.662	**0.676**
RF	0.777	0.630	0.711	0.732	0.745	0.759	0.764	**0.775**
SVM lin	**0.636**	0.619	0.624	0.621	0.632	0.635	0.636	0.631
DS-CO								
KNN	0.800	0.665	0.766	0.777	**0.797**	0.786	0.793	**0.803**
NB	0.543	0.380	0.402	0.403	0.406	0.507	0.519	**0.560**
DT	**0.790**	0.702	0.775	0.766	0.772	0.766	0.783	**0.788**
RF	**0.847**	0.702	0.796	0.804	0.821	0.833	0.843	**0.851**
SVM lin	**0.714**	0.679	0.691	0.683	0.682	0.689	0.707	0.710
DS-ST								
KNN	**0.516**	0.251	0.439	0.454	0.491	0.489	0.490	0.506
NB	0.283	0.193	0.198	0.243	**0.298**	0.289	0.280	**0.158**
DT	**0.468**	0.344	0.447	0.407	0.429	0.424	0.442	0.459
RF	**0.589**	0.344	0.467	0.477	0.533	0.574	0.571	**0.589**
SVM lin	**0.397**	0.144	0.196	0.272	0.346	0.344	0.340	0.347
DS-GR								
KNN	0.791	0.665	0.767	0.777	**0.794**	0.771	0.779	**0.792**
NB	0.545	0.380	0.404	0.405	0.408	0.509	0.520	**0.560**
DT	0.741	0.702	**0.769**	0.761	0.756	0.743	0.750	0.747
RF	**0.847**	0.702	0.791	0.802	0.811	0.833	0.836	**0.848**
SVM lin	**0.755**	0.702	0.693	0.702	0.695	0.706	0.743	0.727

Table 16: MCCs Values Achieved when Testing the Features in Combination for Skeptical Acceptance under Several Semantics using the *PBBG-Train* and *PBBG-Test* Datasets

Feature Combinations									
		1	2	3	4	5	6	7	8
Classifier	**IBF**	**Degree**	**1+ Closeness C.**	**2+ AVG Deg.**	**3+ Size SCC**	**4+ SCCs**	**5+ Betweenness C.**	**6+ Katz**	**7+ is Ir.**
DC-PR									
KNN	0.248	0.173	0.313	0.254	**0.322**	0.251	0.249	0.256	0.251
NB	**0.626**	-0.008	0.156	0.163	0.142	0.619	0.619	0.622	0.619
DT	0.420	0.397	0.298	0.283	0.394	0.248	0.391	**0.439**	0.356
RF	0.379	**0.397**	0.324	0.287	0.344	0.360	0.361	0.370	0.391
SVM lin	0.361	0.416	0.406	0.395	0.370	**0.548**	0.548	0.411	0.415
DC-CO									
KNN	0.248	0.173	0.313	0.254	**0.322**	0.251	0.249	0.256	0.251
NB	**0.626**	-0.008	0.156	0.163	0.142	0.619	0.619	0.622	0.619
DT	0.337	**0.397**	0.298	0.277	0.394	0.248	0.355	0.379	0.334
RF	0.382	**0.397**	0.324	0.296	0.344	0.360	0.358	0.370	0.380
SVM lin	0.361	0.416	0.406	0.395	0.370	**0.548**	0.548	0.411	0.415
DC-ST									
KNN	0.262	0.018	0.303	0.212	0.167	0.194	0.208	0.194	**0.265**
NB	0.436	-0.092	0.153	0.139	0.109	0.430	0.429	0.432	**0.445**
DT	0.161	0.351	0.220	0.095	0.238	0.218	**0.249**	0.070	0.172
RF	0.351	0.351	0.314	0.237	0.298	0.355	**0.365**	0.334	0.382
SVM lin	0.322	0.354	0.355	0.366	0.323	**0.430**	0.429	0.327	0.375
DC-GR									
KNN	0.182	0.196	0.305	0.338	0.340	0.358	**0.372**	0.355	0.183
NB	**0.369**	0.054	0.123	0.155	0.122	0.367	0.367	0.367	**0.369**
DT	0.235	**0.431**	0.290	0.225	0.287	0.254	0.153	0.284	0.240
RF	0.374	**0.432**	0.341	0.372	0.394	0.332	0.334	0.352	0.372
SVM lin	0.403	**0.422**	0.195	0.183	0.174	0.400	0.400	0.400	0.400

Table 17: MCCs Values Achieved when Testing the Features in Combination for Credulous Acceptance under Several Semantics using the *PBBG-Train* and *ICCMA-450* Datasets

Feature Combinations									
		1	2	3	4	5	6	7	8
Classifier	**IBF**	**Degree C.**	**1+ Closeness C.**	**2+ AVG Deg.**	**3+ Size SCC**	**4+ SCCs**	**5+ Betweenness C.**	**6+ Katz C.**	**7+ is Ir.**
DS-PR									
KNN	0.168	0.194	0.278	0.308	0.315	0.330	**0.341**	0.323	0.167
NB	0.359	0.064	0.141	0.174	0.094	0.356	0.356	0.357	**0.361**
DT	0.138	**0.399**	0.306	0.127	0.063	0.047	0.177	0.136	0.138
RF	0.311	**0.399**	0.318	0.316	0.295	0.301	0.303	0.309	0.309
SVM lin	0.318	0.207	0.210	0.197	0.185	0.390	**0.392**	0.388	0.390
DS-CO									
KNN	0.185	0.190	0.296	0.329	0.334	0.355	**0.369**	0.345	0.186
NB	**0.364**	0.067	0.140	0.173	0.125	0.361	0.362	0.362	**0.364**
DT	0.037	**0.439**	0.296	0.260	0.264	0.176	0.101	0.268	0.240
RF	0.379	**0.439**	0.344	0.383	0.391	0.336	0.335	0.350	0.333
SVM lin	**0.398**	0.171	0.208	0.199	0.187	0.395	0.396	0.396	0.396
DS-ST									
KNN	-0.088	-0.024	-0.060	-0.120	-0.010	-0.103	-0.099	-0.024	-0.120
NB	**0.090**	-0.061	-0.128	0.022	-0.114	0.071	0.078	0.084	0.085
DT	**0.525**	0.027	0.093	0.129	0.169	0.137	0.097	0.145	0.245
RF	**0.570**	0.119	0.092	0.111	0.172	0.260	0.252	0.152	0.441
SVM lin	-0.203	0.031	-0.060	**0.070**	-0.157	-0.033	-0.164	-0.122	-0.161
DS-GR									
KNN	0.182	0.196	0.305	0.338	0.340	0.358	**0.372**	0.355	0.183
NB	**0.369**	0.054	0.123	0.155	0.122	0.367	0.367	0.367	**0.369**
DT	0.251	**0.431**	0.289	0.245	0.231	0.275	0.152	0.260	0.257
RF	0.374	**0.432**	0.340	0.372	0.393	0.328	0.345	0.366	0.332
SVM lin	0.403	**0.422**	0.195	0.183	0.174	0.400	0.400	0.400	0.400

Table 18: MCCs Values Achieved when Testing the Features in Combination for Skeptical Acceptance under Several Semantics using the *PBBG-Train* and *ICCMA-450* Datasets

Feature Combinations											
		1	2	3	4	5	6	7	8	9	10
Classifier	**IFC**	**Close-ness C.**	**1+ Degree C.**	**2+ SCCs**	**3 + Between-ness C.**	**4 + AVG Deg.**	**5 + Size SCC**	**6 + is Aperiod**	**7 + Katz C.**	**8 + is Ir.**	**9 + is SC**
DC-PR											
KNN	0.566	0.500	0.340	0.085	0.193	0.150	0.492	0.478	**0.594**	0.566	0.566
NB	0.588	0.225	0.206	0.551	0.551	0.554	0.565	0.594	0.636	**0.642**	0.588
DT	**0.521**	0.429	0.304	0.168	0.449	0.563	0.524	0.618	0.447	0.494	0.476
RF	0.761	0.542	0.292	0.185	0.579	0.709	0.734	0.729	0.756	**0.766**	0.766
SVM lin	0.645	0.596	0.617	0.599	0.596	0.601	0.611	0.640	**0.646**	0.645	0.645
DC-CO											
KNN	0.566	0.500	0.340	0.085	0.193	0.150	0.492	0.478	**0.594**	0.566	0.566
NB	0.588	0.225	0.206	0.551	0.551	0.554	0.565	0.594	0.636	**0.642**	0.588
DT	**0.534**	0.451	0.300	0.168	0.264	0.519	0.538	0.623	0.513	0.416	0.463
RF	**0.767**	0.535	0.303	0.181	0.572	0.712	0.735	0.746	0.755	0.761	0.763
SVM lin	0.645	0.596	0.617	0.599	0.596	0.601	0.611	0.640	**0.646**	0.645	0.645
DC-ST											
KNN	0.588	0.473	**0.589**	0.321	0.442	0.445	0.470	0.457	0.582	0.588	0.588
NB	0.554	0.213	0.435	0.549	0.549	0.552	0.562	0.606	**0.641**	0.567	0.554
DT	0.577	0.404	0.572	0.444	0.428	0.566	0.496	0.565	0.367	**0.583**	0.572
RF	0.700	0.523	0.615	0.516	0.496	0.635	0.674	0.662	0.679	**0.725**	0.721
SVM lin	**0.303**	0.000	0.000	0.000	0.003	0.011	0.019	0.025	0.118	0.294	0.303
DC-GR											
KNN	0.193	0.399	**0.505**	0.469	0.322	0.321	0.329	0.332	0.209	0.193	0.193
NB	**0.259**	0.142	0.149	-0.008	-0.011	-0.011	-0.006	0.027	0.102	0.256	0.259
DT	0.514	0.377	0.471	0.537	0.241	0.484	0.438	**0.542**	0.532	0.501	0.499
RF	0.666	0.530	0.573	0.573	0.651	0.658	**0.677**	0.673	0.664	0.666	0.663
SVM lin	-0.004	0.000	0.000	0.000	-0.002	-0.002	-0.002	-0.002	-0.004	-0.004	-0.004

Table 19: MCCs Values Achieved when Testing the Features in Combination for Credulous Acceptance under Several Semantics using the *ICCMA-450-Train* and *ICCMA-450-Test* Datasets

					Feature Combinations						
		1	2	3	4	5	6	7	8	9	10
Classifier	**IFC**	**Close-ness C.**	**1+ Degree C.**	**2+ SCCs**	**3 + Between-ness C.**	**4 + AVG Deg.**	**5 + Size SCC**	**6 + is Aperiod**	**7 + Katz C.**	**8 + is Ir.**	**9 + is SC**
					DS-PR						
KNN	0.165	0.353	**0.401**	0.386	0.274	0.273	0.284	0.286	0.173	0.165	0.165
NB	**0.247**	0.123	0.201	-0.019	-0.021	-0.017	-0.014	0.044	0.160	0.247	0.247
DT	0.345	0.295	0.290	**0.383**	0.378	0.381	0.290	0.288	0.346	0.344	0.353
RF	0.579	0.433	0.438	0.461	0.545	0.583	**0.585**	0.583	0.577	0.581	0.578
SVM lin	-0.005	0.000	0.000	0.000	-0.003	-0.003	-0.003	-0.003	-0.005	-0.005	-0.005
					DS-CO						
KNN	0.189	0.413	**0.468**	0.440	0.313	0.312	0.319	0.322	0.204	0.189	0.189
NB	0.205	0.166	0.152	-0.017	-0.018	-0.018	-0.013	0.018	0.096	**0.225**	0.205
DT	0.473	0.344	0.470	**0.505**	0.375	0.450	0.462	0.454	0.464	0.456	0.461
RF	0.654	0.529	0.565	0.568	0.630	0.648	**0.657**	0.656	0.657	0.652	0.661
SVM lin	0.002	0.000	0.000	0.000	-0.003	-0.003	-0.003	**0.013**	0.006	0.002	0.002
					DS-ST						
KNN	0.490	0.434	0.687	**0.727**	0.656	0.637	0.661	0.640	0.540	0.490	0.490
NB	**0.756**	0.029	-0.044	-0.074	-0.091	-0.078	0.507	0.755	0.755	0.755	0.756
DT	0.681	0.379	0.642	**0.736**	0.733	0.694	0.552	0.682	0.672	0.721	0.673
RF	**0.815**	0.440	0.723	0.742	0.748	0.781	0.736	0.755	0.776	0.807	0.792
SVM lin	**0.707**	0.007	-0.084	-0.168	-0.116	0.002	0.534	0.485	0.485	0.704	0.707
					DS-GR						
KNN	0.193	0.399	**0.505**	0.469	0.322	0.321	0.329	0.332	0.209	0.193	0.193
NB	**0.259**	0.142	0.149	-0.008	-0.011	-0.011	-0.006	0.027	0.102	0.256	0.259
DT	0.491	0.377	0.472	0.512	0.247	0.478	0.533	0.528	**0.551**	0.495	0.484
RF	0.662	0.532	0.576	0.585	0.656	0.664	0.672	0.667	**0.669**	0.654	0.652
SVM lin	-0.004	0.000	0.000	0.000	-0.002	-0.002	-0.002	-0.002	-0.004	-0.004	-0.004

Table 20: MCCs Values Achieved when Testing the Features in Combination for Skeptical Acceptance under Several Semantics using the *ICCMA-450-Train* and *ICCMA-450-Test* Datasets

6. Discussion

Based on our experiments and evaluation of the findings, we can now provide answers to the research questions posed in this thesis and give an outlook on further resarch that could be valuable.

6.1. Conclusions

Regarding the first research question, we aimed to identify the graph property that has the largest effect on prediction quality and determine if this effect is consistent across graph types and semantics. Our results indicate that the most informative graph property is degree centrality. This finding held true across all semantics as well as most graph types we investigated. We further observed that node-based features, particularly centrality measures, generally provided more information compared to graph-based features. However, graph-based features showed promise in determining the presence of stable extensions, exhibiting good qualification values for stable semantics.

Moving on to the second research question, we sought to determine the ideal set of graph properties that would yield the highest prediction accuracy for a given machine learning classifier. Our experiments revealed that the RF classifier consistently achieved the best overall results. Based on our tests, the ideal combination of graph properties for classification included the degree, closeness, Katz and betweenness centrality as well as the number of strongly connected components of the graph, the average degree of each node in the graph, the size of the SCC a node is part of and the irreflexivity, strong connectivity, and aperiodicity of the graph. Although there were slight variations in the ideal feature combination for each dataset we considered, our proposed combination generally produced similar classification results. It is important to note that for this approach to be effective, the training data should be similar to the testing data. We found that the classifiers struggled to generalize the learned properties across different datasets. Furthermore, in cases where a single feature was already able to effectively separate the classes, the addition of more features did not improve the classification performance.

We also found that graph-based properties, while generally inferior to node-based properties in terms of prediction importance, can help improve prediction quality, especially for stable semantics.

When comparing the experiments to determine if certain semantics were easier to classify than others, the results were inconclusive. For the main datasets, *balanced-kwt-train* and *balanced-kwt-train*, the credulous decision task generally appeared easier to classify than the skeptical task. However, for the evaluation datasets, there were cases where the skeptical classification achieved higher results than the credulous task for the same semantics. Grounded semantics, known to be efficiently solvable, did not necessarily translate into easy classification for ML classifiers.

Additionally, when comparing our results with those obtained by the AGNN in [25], it is evident that neural networks outperformed our approach, particularly in skeptical reasoning under preferred semantics and when the training and testing data differed significantly. This suggests that neural networks are better at generalizing their learning between different datasets and do not solely rely on dominant graph features.

In summary, our research provides insights into the influential graph properties for prediction accuracy and the optimal feature combination for classification tasks. Degree centrality emerged as the most informative graph property, while random forest classification consistently yielded the best results. While node-based features have been more informative, we have also shown that adding certain graph-based features, specifically the features proposed in [39], can further improve the prediction quality. These findings highlight the importance of considering both node-based and graph-based features in approximate argumentation. However, it is crucial to consider the similarity between training and testing data when applying these findings.

6.2. Outlook

We have demonstrated a significant improvement in prediction quality using the IFC and ML-classifiers. Future research could focus on testing additional graph-based features, such as the features used in [27], or exploring other node-based features and centralities. Furthermore, in this thesis, we did not extensively optimize the ML classifiers. While RF emerged as the best classifier, we observed that its training time was significantly higher. Thus, optimizing the other classifiers could be beneficial. Additionally, due to time constraints, we could not fully investigate the runtime advantages of using the considered ML classifiers compared to complete solvers.

It is important to acknowledge that, like any approximate method, when applying this approach to real-world applications, the correctness of the results cannot be guaranteed. Therefore, further research could involve utilizing the IFC as heuristics in a solver, such as *Heureka* [23], to reduce the backtracking steps required for calculating a sound solution for an AF. Compared to using neural networks, our approach is relatively lightweight, offering the potential for decreased solution time while still guaranteeing correct results.

References

[1] Charu C Aggarwal et al. *Data mining: the textbook*. Vol. 1. Springer, 2015.

[2] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. "An introduction to argumentation semantics". In: *The Knowledge Engineering Review* 26.4 (2011), pp. 365–410. ISSN: 0269-8889. DOI: 10.1017/s0269888911000166.

[3] Pietro Baroni and Massimiliano Giacomin. "Semantics of Abstract Argument Systems". In: *Argumentation in Artificial Intelligence*. Ed. by Guillermo Simari and Iyad Rahwan. Boston, MA: Springer US, 2009, pp. 25–44. ISBN: 978-0-387-98197-0. DOI: 10.1007/978-0-387-98197-0_2. URL: https://doi.org/10.1007/978-0-387-98197-0_2.

[4] Pietro Baroni et al. *Handbook of Formal Argumentation*. London, England: College Publications, 2018.

[5] T.J.M. Bench-Capon. "Before and after Dung: Argumentation in AI and Law". In: *Argument & Computation* 11.1-2 (2020), pp. 221–238. ISSN: 19462166. DOI: 10.3233/AAC-190477.

[6] M. W. A. Caminada, W. A. Carnielli, and P. E. Dunne. "Semi-stable semantics". In: *Journal of Logic and Computation* 22.5 (2011), pp. 1207–1254. ISSN: 0955-792X. DOI: 10.1093/logcom/exr033.

[7] Martin Caminada. *Comparing two unique extension semantics for formal argumentation: ideal and eager*. 2007.

[8] Álvaro Carrera and Carlos A Iglesias. "A systematic review of argumentation techniques for multi-agent systems research". In: *Artificial Intelligence Review* 44.4 (2015), pp. 509–535.

[9] F. Cerutti, Massimiliano Giacomin, and Mauro Vallati. "ArgSemSAT: Solving argumentation problems using SAT". In: *Proceedings of the 5th International Conference on Computational Models of Argument* 266 (Jan. 2014), pp. 455–456.

[10] Federico Cerutti, Matthias Thimm, and Mauro Vallati. "An experimental analysis on the similarity of argumentation semantics". In: *Argument & Computation* 11.3 (2020), pp. 269–304.

[11] Günther Charwat et al. "Methods for solving reasoning problems in abstract argumentation – A survey". In: *Artificial Intelligence* 220 (2015), pp. 28–63. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2014.11.008. URL: https://www.sciencedirect.com/science/article/pii/S0004370214001404.

[12] Davide Chicco and Giuseppe Jurman. "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation". In: *BMC genomics* 21 (2020), pp. 1–13.

[13] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. "Prudent semantics for argumentation frameworks". In: *17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)*. IEEE. 2005, 5–pp.

[14] Dennis Craandijk and Floris Bex. "Deep learning for abstract argumentation semantics". In: *arXiv preprint arXiv:2007.07629* (2020).

[15] Padraig Cunningham and Sarah Jane Delany. "k-Nearest Neighbour Classifiers: 2nd Edition (with Python examples)". In: *CoRR* abs/2004.04523 (2020). arXiv: 2004.04523. URL: https://arxiv.org/abs/2004.04523.

[16] Yannis Dimopoulos and Alberto Torres. "Graph theoretical structures in logic programs and default theories". In: *Theoretical Computer Science* 170.1-2 (1996), pp. 209–244.

[17] Mamadou Bilo Doumbouya, Bernard Kamsu-Foguem, and Hugues Kenfack. "Argumentation and graph properties". In: *Information Processing & Management* 52.2 (2016), pp. 319–325.

[18] P. M. Dung, P. Mancarella, and F. Toni. "Computing ideal sceptical argumentation". In: *Artificial Intelligence* 171.10-15 (2007), pp. 642–674. ISSN: 0004-3702. DOI: 10.1016/j.artint.2007.05.003.

[19] Phan Minh Dung. "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games". In: *Artificial Intelligence* 77.2 (1995), pp. 321–357. ISSN: 0004-3702. DOI: 10.1016/0004-3702(94)00041-X.

[20] Paul E Dunne and Michael Wooldridge. "Complexity of abstract argumentation". In: *Argumentation in artificial intelligence* (2009), pp. 85–104.

[21] Paul E. Dunne and T.J.M. Bench-Capon. "Coherence in finite argument systems". In: *Artificial Intelligence* 141.1 (2002), pp. 187–203. ISSN: 0004-3702. DOI: 10.1016/S0004-3702(02)00261-8. URL: https://www.sciencedirect.com/science/article/pii/S0004370202002618.

[22] Wolfgang Dvořák. "On the complexity of computing the justification status of an argument". In: *International Workshop on Theorie and Applications of Formal Argumentation*. Springer. 2011, pp. 32–49.

[23] Nils Geilen and Matthias Thimm. "Heureka: A General Heuristic Backtracking Solver for Abstract Argumentation". In: *Theory and Applications of Formal Argumentation*. Ed. by Elizabeth Black, Sanjay Modgil, and Nir Oren. Cham: Springer International Publishing, 2018, pp. 143–149. ISBN: 978-3-319-75553-3.

[24] Jennifer Golbeck. "Chapter 3 - Network Structure and Measures". In: *Analyzing the Social Web*. Ed. by Jennifer Golbeck. Boston: Morgan Kaufmann, 2013, pp. 25–44. ISBN: 978-0-12-405531-5. DOI: https://doi.org/10.1016/B978-0-12-405531-5.00003-1. URL: https://www.sciencedirect.com/science/article/pii/B9780124055315000031.

[25] Isabelle Kuhlmann, Thorsten Wujek, and Matthias Thimm. "On the Impact of Data Selection when Applying Machine Learning in Abstract Argumentation". In: *Proceedings of the 9th International Conference on Computational Models of Argument (COMMA'22)*. 2022.

[26] Jafar Alzubi, Anand Nayyar, and Akshi Kumar. "Machine Learning from Theory to Algorithms: An Overview". In: *Journal of Physics: Conference Series* 1142.1 (2018), p. 012012. ISSN: 1742-6596. DOI: 10.1088/1742-6596/1142/1/012012. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1142/1/012012/meta.

[27] Jonas Klein, Isabelle Kuhlmann, and Matthias Thimm. "Graph Neural Networks for Algorithm Selection in Abstract Argumentation". In: *Proceedings of the First International Workshop on Argumentation & Machine Learning (ArgML'22)*. Sept. 2022.

[28] Isabelle Kuhlmann and Matthias Thimm. "Using graph convolutional networks for approximate reasoning with abstract argumentation frameworks: A feasibility study". In: *International Conference on Scalable Uncertainty Management*. Springer. 2019, pp. 24–37.

[29] Batta Mahesh. "Machine learning algorithms-a review". In: *International Journal of Science and Research (IJSR).[Internet]* 9 (2020), pp. 381–386.

[30] Lars Malmqvist et al. "Determining the Acceptability of Abstract Arguments with Graph Convolutional Networks." In: *SAFA@ COMMA*. 2020, pp. 47–56.

[31] Natarajan Meghanathan. *Centrality Metrics for Complex Network Analysis: Emerging Research and Opportunities*. Information Science Reference, 2018.

[32] Mark Newman. *Networks*. Oxford university press, 2018.

[33] Andreas Niskanen and Matti Järvisalo. *μ-toksia (version 2019-10-31): SAT-based Solver for Static and Dynamic Argumentation Frameworks*. English. Dec. 2019.

[34] William S Noble. "What is a support vector machine?" In: *Nature biotechnology* 24.12 (2006), pp. 1565–1567.

[35] Ashis Pradhan. "Support vector machine-a survey". In: *International Journal of Emerging Technology and Advanced Engineering* 2.8 (2012), pp. 82–85.

[36] Henry Prakken. "An overview of formal models of argumentation and their application in philosophy". In: *Studies in logic* 4.1 (2011), pp. 65–86.

[37] A. L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229. DOI: 10.1147/rd.33.0210.

[38] Amanpreet Singh, Narina Thakur, and Aakanksha Sharma. "A review of supervised machine learning algorithms". In: *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. 2016, pp. 1310–1315.

[39] Mauro Vallati, Federico Cerutti, and Massimiliano Giacomin. "Predictive models and abstract argumentation: the case of high-complexity semantics". In: *The Knowledge Engineering Review* 34 (2019).

[40] Frans H. van Eemeren. *Handbook of Argumentation Theory*. Springer eBook Collection Humanities, Social Sciences and Law. Dordrecht: Springer, 2014. ISBN: 9789048194735. DOI: 10.1007/978-90-481-9473-5.

[41] Bart Verheij. "Two approaches to dialectical argumentation: admissible sets and argumentation stages". In: *Proc. NAIC* 96 (1996), pp. 357–368. URL: https://www.ai.rug.nl/~verheij/publications/pdf/cd96.pdf.

[42] Yining Wu, Martin Caminada, and Dov M. Gabbay. "Complete Extensions in Argumentation Coincide with 3-Valued Stable Models in Logic Programming". In: *Studia Logica: An International Journal for Symbolic Logic* 93.2/3 (2009), pp. 383–403. ISSN: 00393215. URL: http://www.jstor.org/stable/40587173.

A. Distribution of Arguments for each Acceptance Class

pbbg-train				
	Yes	No	Share Yes	Share No
DC-PR	707697	1190630	37%	63%
DC-CO	707697	1190630	37%	63%
DC-ST	662222	1236105	35%	65%
DC-GR	495651	1402676	26%	74%
DS-PR	596092	1302235	31%	69%
DS-CO	532626	1365701	28%	72%
DS-ST	1062390	835937	56%	44%
DS-GR	495651	1402676	26%	74%

pbbg-test				
	Yes	No	Share Yes	Share No
DC-PR	8928	16072	36%	64%
DC-CO	8928	16072	36%	64%
DC-ST	8402	16598	34%	66%
DC-GR	6135	18865	25%	75%
DS-PR	7067	17933	28%	72%
DS-CO	6135	18865	25%	75%
DS-ST	14962	10038	60%	40%
DS-GR	6135	18865	25%	75%

Table 21: Distribution of Accepted (Yes) and Rejected (No) Arguments for the *pbbg-train* and *pbbg-test* Datasets

ICCMA-450				
	Yes	No	Share Yes	Share No
DC-PR	77271	214950	26%	74%
DC-CO	77271	214950	26%	74%
DC-ST	51110	241111	17%	83%
DC-GR	25634	266587	9%	91%
DS-PR	31605	260616	11%	89%
DS-CO	26587	265634	9%	91%
DS-ST	166530	125691	57%	43%
DS-GR	25634	266587	9%	91%

Table 22: Distribution of Accepted (Yes) and Rejected (No) Arguments for the *ICCMA-450* Dataset

ICCMA-train				
	Yes	No	Share Yes	Share No
DC-PR	69649	180290	28%	72%
DC-CO	69649	180290	28%	72%
DC-ST	43962	205977	18%	82%
DC-GR	23860	226079	10%	90%
DS-PR	29038	220901	12%	88%
DS-CO	24537	225402	10%	90%
DS-ST	151489	98450	61%	39%
DS-GR	23860	226079	10%	90%

ICCMA-test				
	Yes	No	Share Yes	Share No
DC-PR	7622	34660	18%	82%
DC-CO	7622	34660	18%	82%
DC-ST	7148	35134	17%	83%
DC-GR	1774	40508	4%	96%
DS-PR	2567	39715	6%	94%
DS-CO	2050	40232	5%	95%
DS-ST	15041	27241	36%	64%
DS-GR	1774	40508	4%	96%

Table 23: Distribution of Accepted (Yes) and Rejected (No) Arguments for the *ICCMA-train* and *ICCMA-test* Datasets

B. Detailed Test Results for *balanced-kwt-train* and *balanced-kwt-test*

B.1. Test Results for Individual Features

Katz Centrality							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.606	0.838	0.894	0.703	0.877	0.268	3.108
NB	0.236	0.627	0.625	0.632	0.801	0.527	0.005
DT	0.490	0.773	0.798	0.715	0.869	1.023	0.042
RF	0.597	0.829	0.870	0.735	0.886	30.841	3.788
SVM lin	0.080	0.703	0.978	0.052	0.710	0.935	0.004
DC-CO							
KNN	0.606	0.838	0.894	0.703	0.877	0.263	3.051
NB	0.236	0.627	0.625	0.632	0.801	0.525	0.008
DT	0.491	0.774	0.798	0.716	0.869	1.030	0.043
RF	0.597	0.830	0.870	0.735	0.886	30.113	3.788
SVM lin	0.080	0.703	0.978	0.052	0.710	0.953	0.003
DC-ST							
KNN	0.613	0.831	0.736	0.876	0.740	0.257	3.036
NB	0.225	0.651	0.512	0.718	0.466	0.512	0.013
DT	0.501	0.786	0.622	0.865	0.689	0.979	0.040
RF	0.609	0.831	0.708	0.891	0.757	30.123	4.136
SVM lin	0.000	0.676	0.000	1.000	1.000	1.000	0.009
DC-GR							
KNN	0.556	0.848	0.610	0.920	0.697	0.281	3.189
NB	0.135	0.647	0.441	0.709	0.314	0.533	0.010
DT	0.463	0.819	0.522	0.909	0.634	0.970	0.037
RF	0.562	0.856	0.541	0.951	0.769	29.980	3.419
SVM lin	0.000	0.768	0.000	1.000	1.000	0.933	0.017
DS-PR							
KNN	0.612	0.812	0.765	0.846	0.778	0.288	3.115
NB	0.240	0.639	0.477	0.754	0.578	0.541	0.006
DT	0.520	0.770	0.665	0.844	0.751	1.028	0.040
RF	0.616	0.815	0.714	0.887	0.817	29.871	3.818
SVM lin	0.401	0.703	0.704	0.703	0.626	0.820	0.015
DS-CO							
KNN	0.556	0.848	0.610	0.920	0.697	0.262	3.058
NB	0.135	0.647	0.441	0.709	0.314	0.520	0.005
DT	0.461	0.818	0.523	0.908	0.631	1.027	0.039
RF	0.562	0.856	0.542	0.951	0.768	30.810	3.389
SVM lin	0.000	0.768	0.000	1.000	1.000	0.787	0.006
DS-ST							
KNN	0.418	0.775	0.859	0.550	0.837	0.264	3.068
NB	0.126	0.582	0.595	0.545	0.778	0.527	0.008
DT	0.325	0.723	0.789	0.545	0.823	1.004	0.044
RF	0.427	0.772	0.841	0.588	0.846	31.021	4.080
SVM lin	0.008	0.729	1.000	0.000	0.729	0.975	0.004
DS-GR							
KNN	0.556	0.848	0.610	0.920	0.697	0.260	3.054
NB	0.135	0.647	0.441	0.709	0.314	0.519	0.006
DT	0.462	0.819	0.524	0.908	0.632	0.969	0.036
RF	0.561	0.855	0.540	0.951	0.768	30.411	3.422
SVM lin	0.000	0.768	0.000	1.000	1.000	0.769	0.006

Table 24: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Katz Centrality

Closeness Centrality							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.689	0.871	0.913	0.773	0.905	0.268	4.661
NB	0.042	0.474	0.404	0.641	0.727	0.530	0.013
DT	0.642	0.846	0.873	0.783	0.905	0.679	0.037
RF	0.678	0.864	0.895	0.791	0.910	19.542	3.146
SVM lin	0.000	0.703	1.000	0.000	0.703	1.225	0.005
DC-CO							
KNN	0.689	0.871	0.913	0.773	0.905	0.267	4.555
NB	0.042	0.474	0.404	0.641	0.727	0.548	0.042
DT	0.642	0.847	0.873	0.783	0.905	0.647	0.033
RF	0.678	0.864	0.895	0.791	0.910	19.967	3.347
SVM lin	0.000	0.703	1.000	0.000	0.703	1.226	0.007
DC-ST							
KNN	0.709	0.872	0.814	0.899	0.795	0.260	4.512
NB	-0.032	0.467	0.528	0.438	0.311	0.539	0.029
DT	0.641	0.845	0.721	0.905	0.784	0.611	0.035
RF	0.686	0.863	0.772	0.907	0.800	19.972	3.184
SVM lin	0.578	0.800	0.813	0.793	0.653	0.971	0.009
DC-GR							
KNN	0.654	0.881	0.681	0.942	0.780	0.262	4.521
NB	-0.048	0.475	0.465	0.478	0.212	0.521	0.005
DT	0.646	0.879	0.670	0.942	0.777	0.568	0.025
RF	0.656	0.884	0.651	0.954	0.812	17.325	2.241
SVM lin	0.509	0.820	0.655	0.869	0.602	0.565	0.003
DS-PR							
KNN	0.688	0.849	0.789	0.892	0.838	0.265	4.521
NB	0.082	0.567	0.375	0.703	0.472	0.544	0.027
DT	0.692	0.851	0.808	0.881	0.828	0.697	0.032
RF	0.708	0.859	0.795	0.905	0.855	20.459	2.906
SVM lin	0.297	0.656	0.606	0.692	0.582	1.258	0.005
DS-CO							
KNN	0.654	0.881	0.681	0.942	0.780	0.274	4.641
NB	-0.048	0.475	0.465	0.478	0.212	0.550	0.009
DT	0.646	0.879	0.671	0.942	0.777	0.570	0.025
RF	0.657	0.884	0.652	0.954	0.811	17.316	2.287
SVM lin	0.509	0.820	0.655	0.869	0.602	0.584	0.019
DS-ST							
KNN	0.464	0.776	0.815	0.670	0.869	0.261	4.643
NB	0.182	0.629	0.663	0.537	0.794	0.545	0.045
DT	0.523	0.812	0.873	0.649	0.870	0.616	0.038
RF	0.542	0.818	0.873	0.672	0.877	19.610	3.523
SVM lin	0.000	0.728	1.000	0.000	0.728	2.035	0.008
DS-GR							
KNN	0.654	0.881	0.681	0.942	0.780	0.272	4.700
NB	-0.048	0.475	0.465	0.478	0.212	0.539	0.005
DT	0.646	0.879	0.671	0.942	0.777	0.613	0.025
RF	0.657	0.884	0.652	0.954	0.811	17.505	2.293
SVM lin	0.509	0.820	0.655	0.869	0.602	0.532	0.014

Table 25: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Closeness Centrality

Betweenness Centrality							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.656	0.856	0.896	0.760	0.899	0.240	51.436
NB	0.000	0.297	0.000	1.000	1.000	0.575	0.005
DT	0.608	0.839	0.896	0.702	0.877	0.535	0.029
RF	0.618	0.844	0.909	0.692	0.875	22.957	2.857
SVM lin	0.305	0.746	0.975	0.205	0.744	0.694	0.003
DC-CO							
KNN	0.656	0.856	0.896	0.760	0.899	0.255	53.506
NB	0.000	0.297	0.000	1.000	1.000	0.569	0.005
DT	0.608	0.839	0.896	0.702	0.877	0.577	0.032
RF	0.618	0.844	0.908	0.694	0.875	22.883	2.849
SVM lin	0.305	0.746	0.975	0.205	0.744	0.713	0.004
DC-ST							
KNN	0.486	0.723	0.864	0.654	0.546	0.234	51.328
NB	0.000	0.676	0.000	1.000	1.000	0.517	0.005
DT	0.503	0.719	0.906	0.629	0.540	0.802	0.034
RF	0.504	0.719	0.908	0.628	0.539	32.294	2.753
SVM lin	0.000	0.676	0.000	1.000	1.000	0.639	0.003
DC-GR							
KNN	0.520	0.746	0.919	0.693	0.475	0.239	51.577
NB	0.000	0.768	0.000	1.000	1.000	0.547	0.005
DT	-0.027	0.765	0.001	0.996	0.045	0.559	0.024
RF	-0.027	0.765	0.001	0.995	0.048	22.916	2.181
SVM lin	0.000	0.768	0.000	1.000	1.000	0.642	0.003
DS-PR							
KNN	0.544	0.749	0.905	0.638	0.639	0.262	53.686
NB	0.000	0.586	0.000	1.000	1.000	0.605	0.005
DT	0.523	0.741	0.886	0.639	0.634	0.524	0.028
RF	0.528	0.741	0.897	0.631	0.632	20.471	2.613
SVM lin	0.436	0.701	0.826	0.612	0.601	0.705	0.004
DS-CO							
KNN	0.520	0.746	0.919	0.693	0.475	0.240	52.773
NB	0.000	0.768	0.000	1.000	1.000	0.539	0.004
DT	-0.027	0.765	0.001	0.996	0.045	0.543	0.022
RF	-0.027	0.765	0.001	0.995	0.047	23.405	2.130
SVM lin	0.000	0.768	0.000	1.000	1.000	0.672	0.003
DS-ST							
KNN	0.457	0.805	0.947	0.423	0.815	0.240	52.919
NB	0.000	0.272	0.000	1.000	1.000	0.558	0.004
DT	0.411	0.784	0.905	0.460	0.818	0.870	0.042
RF	0.429	0.793	0.919	0.454	0.819	39.083	3.389
SVM lin	0.136	0.736	0.988	0.059	0.738	0.633	0.005
DS-GR							
KNN	0.520	0.746	0.919	0.693	0.475	0.244	51.687
NB	0.000	0.768	0.000	1.000	1.000	0.522	0.004
DT	-0.027	0.765	0.001	0.996	0.045	0.524	0.022
RF	-0.027	0.765	0.001	0.995	0.047	22.480	2.082
SVM lin	0.000	0.768	0.000	1.000	1.000	0.632	0.003

Table 26: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Betweenness Centrality

SCC Size							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.246	0.716	0.881	0.327	0.756	0.250	39.890
NB	0.000	0.297	0.000	1.000	1.000	0.554	0.004
DT	0.339	0.754	0.987	0.201	0.745	0.250	0.009
RF	0.339	0.754	0.987	0.201	0.745	4.165	0.775
SVM lin	0.314	0.747	0.990	0.169	0.739	0.801	0.003
DC-CO							
KNN	0.246	0.716	0.881	0.327	0.756	0.237	39.739
NB	0.000	0.297	0.000	1.000	1.000	0.536	0.005
DT	0.339	0.754	0.987	0.201	0.745	0.245	0.008
RF	0.339	0.754	0.987	0.201	0.745	4.178	0.784
SVM lin	0.314	0.747	0.990	0.169	0.739	0.785	0.004
DC-ST							
KNN	0.440	0.697	0.841	0.628	0.520	0.230	39.783
NB	0.000	0.676	0.000	1.000	1.000	0.527	0.004
DT	0.515	0.797	0.562	0.910	0.750	0.233	0.010
RF	0.515	0.797	0.562	0.910	0.750	3.512	0.743
SVM lin	0.336	0.670	0.704	0.654	0.494	0.790	0.003
DC-GR							
KNN	0.443	0.760	0.732	0.768	0.488	0.235	40.207
NB	0.000	0.768	0.000	1.000	1.000	0.530	0.005
DT	0.000	0.768	0.000	1.000	1.000	0.230	0.008
RF	0.000	0.768	0.000	1.000	1.000	3.441	0.771
SVM lin	0.000	0.768	0.000	1.000	1.000	0.590	0.004
DS-PR							
KNN	0.484	0.753	0.622	0.845	0.740	0.241	39.859
NB	0.000	0.586	0.000	1.000	1.000	0.563	0.004
DT	0.489	0.756	0.610	0.859	0.754	0.261	0.009
RF	0.489	0.756	0.610	0.859	0.754	4.851	0.748
SVM lin	0.324	0.657	0.706	0.623	0.570	0.810	0.003
DS-CO							
KNN	0.443	0.760	0.732	0.768	0.488	0.244	39.811
NB	0.000	0.768	0.000	1.000	1.000	0.522	0.004
DT	0.000	0.768	0.000	1.000	1.000	0.236	0.009
RF	0.000	0.768	0.000	1.000	1.000	3.463	0.726
SVM lin	0.000	0.768	0.000	1.000	1.000	0.587	0.003
DS-ST							
KNN	0.308	0.683	0.701	0.637	0.838	0.239	39.934
NB	0.000	0.272	0.000	1.000	1.000	0.529	0.008
DT	0.484	0.799	0.872	0.604	0.855	0.241	0.010
RF	0.484	0.799	0.872	0.604	0.855	3.780	0.764
SVM lin	0.000	0.728	1.000	0.000	0.728	0.800	0.003
DS-GR							
KNN	0.443	0.760	0.732	0.768	0.488	0.235	39.964
NB	0.000	0.768	0.000	1.000	1.000	0.522	0.009
DT	0.000	0.768	0.000	1.000	1.000	0.230	0.009
RF	0.000	0.768	0.000	1.000	1.000	3.616	0.759
SVM lin	0.000	0.768	0.000	1.000	1.000	0.592	0.004

Table 27: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: SCC Size

No of SCCs							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.409	0.776	0.962	0.334	0.774	0.248	49.931
NB	0.000	0.297	0.000	1.000	1.000	0.551	0.004
DT	0.424	0.781	0.952	0.376	0.783	0.236	0.006
RF	0.424	0.781	0.952	0.376	0.783	3.318	0.527
SVM lin	0.000	0.703	1.000	0.000	0.703	0.797	0.003
DC-CO							
KNN	0.409	0.776	0.962	0.334	0.774	0.236	48.801
NB	0.000	0.297	0.000	1.000	1.000	0.536	0.005
DT	0.424	0.781	0.952	0.376	0.783	0.228	0.006
RF	0.424	0.781	0.952	0.376	0.783	3.288	0.510
SVM lin	0.000	0.703	1.000	0.000	0.703	0.785	0.003
DC-ST							
KNN	0.799	0.907	0.922	0.901	0.817	0.225	49.337
NB	0.000	0.676	0.000	1.000	1.000	0.525	0.004
DT	0.810	0.914	0.909	0.917	0.840	0.222	0.006
RF	0.810	0.914	0.909	0.917	0.840	3.096	0.495
SVM lin	-0.160	0.577	0.056	0.828	0.134	0.772	0.004
DC-GR							
KNN	0.056	0.766	0.027	0.989	0.422	0.236	48.673
NB	0.000	0.768	0.000	1.000	1.000	0.526	0.005
DT	0.262	0.784	0.217	0.956	0.596	0.226	0.006
RF	0.262	0.784	0.217	0.956	0.596	2.932	0.490
SVM lin	0.103	0.760	0.091	0.962	0.422	0.742	0.003
DS-PR							
KNN	0.070	0.589	0.126	0.917	0.518	0.237	48.605
NB	0.000	0.586	0.000	1.000	1.000	0.549	0.004
DT	0.145	0.610	0.230	0.880	0.575	0.240	0.006
RF	0.145	0.610	0.230	0.880	0.575	3.629	0.533
SVM lin	0.003	0.578	0.051	0.951	0.422	0.781	0.004
DS-CO							
KNN	0.056	0.766	0.027	0.989	0.422	0.227	48.700
NB	0.000	0.768	0.000	1.000	1.000	0.536	0.004
DT	0.262	0.784	0.217	0.956	0.596	0.221	0.005
RF	0.262	0.784	0.217	0.956	0.596	2.935	0.496
SVM lin	0.103	0.760	0.091	0.962	0.422	0.763	0.003
DS-ST							
KNN	0.347	0.617	0.526	0.859	0.909	0.234	48.494
NB	0.000	0.272	0.000	1.000	1.000	0.538	0.005
DT	0.489	0.763	0.756	0.780	0.902	0.254	0.007
RF	0.489	0.763	0.756	0.780	0.902	3.092	0.481
SVM lin	0.109	0.728	0.969	0.082	0.739	0.794	0.003
DS-GR							
KNN	0.056	0.766	0.027	0.989	0.422	0.224	48.578
NB	0.000	0.768	0.000	1.000	1.000	0.543	0.005
DT	0.262	0.784	0.217	0.956	0.596	0.218	0.006
RF	0.262	0.784	0.217	0.956	0.596	2.992	0.477
SVM lin	0.103	0.760	0.091	0.962	0.422	0.736	0.003

Table 28: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: No of SCCs

AVG Degree							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.391	0.763	0.889	0.462	0.797	0.250	34.035
NB	0.000	0.297	0.000	1.000	1.000	0.545	0.005
DT	0.433	0.777	0.892	0.504	0.810	0.311	0.009
RF	0.429	0.777	0.901	0.484	0.805	8.572	0.708
SVM lin	0.000	0.703	1.000	0.000	0.703	0.769	0.003
DC-CO							
KNN	0.391	0.763	0.889	0.462	0.797	0.240	35.575
NB	0.000	0.297	0.000	1.000	1.000	0.575	0.005
DT	0.433	0.777	0.892	0.504	0.810	0.319	0.009
RF	0.432	0.778	0.903	0.484	0.806	8.652	0.741
SVM lin	0.000	0.703	1.000	0.000	0.703	0.916	0.003
DC-ST							
KNN	0.804	0.910	0.925	0.902	0.820	0.233	35.867
NB	0.000	0.676	0.000	1.000	1.000	0.530	0.005
DT	0.812	0.915	0.909	0.918	0.842	0.338	0.007
RF	0.810	0.915	0.909	0.917	0.840	9.642	0.668
SVM lin	0.707	0.839	0.987	0.768	0.671	0.528	0.003
DC-GR							
KNN	0.260	0.779	0.247	0.940	0.555	0.254	34.780
NB	0.000	0.768	0.000	1.000	1.000	0.534	0.004
DT	0.260	0.784	0.217	0.955	0.593	0.268	0.006
RF	0.259	0.783	0.217	0.954	0.589	5.557	0.544
SVM lin	0.000	0.768	0.000	1.000	1.000	0.636	0.003
DS-PR							
KNN	0.116	0.595	0.275	0.822	0.522	0.318	36.455
NB	0.000	0.586	0.000	1.000	1.000	0.579	0.005
DT	0.137	0.608	0.221	0.881	0.568	0.330	0.009
RF	0.138	0.606	0.255	0.855	0.554	9.340	0.823
SVM lin	0.000	0.586	0.000	1.000	1.000	0.796	0.003
DS-CO							
KNN	0.260	0.779	0.247	0.940	0.555	0.236	35.224
NB	0.000	0.768	0.000	1.000	1.000	0.544	0.004
DT	0.260	0.784	0.217	0.955	0.593	0.292	0.007
RF	0.260	0.784	0.217	0.955	0.593	5.364	0.543
SVM lin	0.000	0.768	0.000	1.000	1.000	0.635	0.003
DS-ST							
KNN	0.461	0.731	0.701	0.814	0.910	0.234	34.385
NB	0.000	0.272	0.000	1.000	1.000	0.557	0.005
DT	0.468	0.738	0.710	0.811	0.910	0.323	0.007
RF	0.469	0.739	0.712	0.811	0.910	9.428	0.697
SVM lin	0.000	0.728	1.000	0.000	0.728	0.693	0.003
DS-GR							
KNN	0.260	0.779	0.247	0.940	0.555	0.237	35.153
NB	0.000	0.768	0.000	1.000	1.000	0.530	0.005
DT	0.260	0.784	0.217	0.955	0.593	0.272	0.006
RF	0.252	0.781	0.217	0.952	0.576	5.363	0.578
SVM lin	0.000	0.768	0.000	1.000	1.000	0.685	0.003

Table 29: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: AVG Degree

Aperiodicity							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.000	0.703	1.000	0.000	0.703	0.254	144.824
NB	0.000	0.297	0.000	1.000	1.000	0.585	0.006
DT	0.000	0.703	1.000	0.000	0.703	0.212	0.005
RF	0.000	0.703	1.000	0.000	0.703	2.310	0.418
SVM lin	0.000	0.703	1.000	0.000	0.703	0.799	0.003
DC-CO							
KNN	0.000	0.703	1.000	0.000	0.703	0.228	135.886
NB	0.000	0.297	0.000	1.000	1.000	0.543	0.006
DT	0.000	0.703	1.000	0.000	0.703	0.212	0.005
RF	0.000	0.703	1.000	0.000	0.703	1.968	0.424
SVM lin	0.000	0.703	1.000	0.000	0.703	0.767	0.003
DC-ST							
KNN	0.513	0.682	0.995	0.531	0.505	0.221	137.106
NB	0.000	0.676	0.000	1.000	1.000	0.592	0.005
DT	0.000	0.676	0.000	1.000	1.000	0.221	0.005
RF	0.000	0.676	0.000	1.000	1.000	2.052	0.420
SVM lin	0.000	0.676	0.000	1.000	1.000	0.790	0.004
DC-GR							
KNN	0.000	0.768	0.000	1.000	1.000	0.234	138.514
NB	0.000	0.768	0.000	1.000	1.000	0.570	0.005
DT	0.000	0.768	0.000	1.000	1.000	0.212	0.005
RF	0.000	0.768	0.000	1.000	1.000	2.077	0.409
SVM lin	0.000	0.768	0.000	1.000	1.000	0.648	0.003
DS-PR							
KNN	0.000	0.586	0.000	1.000	1.000	0.238	135.733
NB	0.000	0.586	0.000	1.000	1.000	0.545	0.005
DT	0.000	0.586	0.000	1.000	1.000	0.215	0.004
RF	0.000	0.586	0.000	1.000	1.000	2.063	0.391
SVM lin	0.000	0.586	0.000	1.000	1.000	0.767	0.003
DS-CO							
KNN	0.000	0.768	0.000	1.000	1.000	0.223	134.543
NB	0.000	0.768	0.000	1.000	1.000	0.524	0.004
DT	0.000	0.768	0.000	1.000	1.000	0.205	0.004
RF	0.000	0.768	0.000	1.000	1.000	1.980	0.381
SVM lin	0.000	0.768	0.000	1.000	1.000	0.642	0.003
DS-ST							
KNN	0.359	0.589	0.465	0.922	0.941	0.235	134.439
NB	0.000	0.272	0.000	1.000	1.000	0.531	0.004
DT	0.000	0.728	1.000	0.000	0.728	0.211	0.005
RF	0.000	0.728	1.000	0.000	0.728	1.992	0.390
SVM lin	0.000	0.728	1.000	0.000	0.728	0.746	0.003
DS-GR							
KNN	0.000	0.768	0.000	1.000	1.000	0.226	134.288
NB	0.000	0.768	0.000	1.000	1.000	0.534	0.005
DT	0.000	0.768	0.000	1.000	1.000	0.206	0.005
RF	0.000	0.768	0.000	1.000	1.000	1.994	0.387
SVM lin	0.000	0.768	0.000	1.000	1.000	0.647	0.003

Table 30: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Aperiodicity

Strong Connectivity							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.235	0.726	1.000	0.077	0.720	0.257	252.792
NB	0.000	0.297	0.000	1.000	1.000	0.657	0.006
DT	0.235	0.726	1.000	0.077	0.720	0.276	0.014
RF	0.235	0.726	1.000	0.077	0.720	2.629	0.502
SVM lin	0.235	0.726	1.000	0.077	0.720	0.789	0.004
DC-CO							
KNN	0.235	0.726	1.000	0.077	0.720	0.246	253.510
NB	0.000	0.297	0.000	1.000	1.000	0.609	0.005
DT	0.235	0.726	1.000	0.077	0.720	0.207	0.004
RF	0.235	0.726	1.000	0.077	0.720	1.958	0.407
SVM lin	0.235	0.726	1.000	0.077	0.720	0.629	0.003
DC-ST							
KNN	0.106	0.347	1.000	0.034	0.332	0.224	251.622
NB	0.000	0.676	0.000	1.000	1.000	0.538	0.004
DT	0.000	0.676	0.000	1.000	1.000	0.206	0.004
RF	0.000	0.676	0.000	1.000	1.000	1.948	0.387
SVM lin	0.000	0.676	0.000	1.000	1.000	0.583	0.004
DC-GR							
KNN	0.000	0.768	0.000	1.000	1.000	0.226	248.375
NB	0.000	0.768	0.000	1.000	1.000	0.580	0.005
DT	0.000	0.768	0.000	1.000	1.000	0.223	0.004
RF	0.000	0.768	0.000	1.000	1.000	2.087	0.417
SVM lin	0.000	0.768	0.000	1.000	1.000	0.545	0.003
DS-PR							
KNN	0.000	0.586	0.000	1.000	1.000	0.238	256.666
NB	0.000	0.586	0.000	1.000	1.000	0.703	0.009
DT	0.000	0.586	0.000	1.000	1.000	0.296	0.006
RF	0.000	0.586	0.000	1.000	1.000	2.295	0.466
SVM lin	0.000	0.586	0.000	1.000	1.000	0.644	0.003
DS-CO							
KNN	0.000	0.768	0.000	1.000	1.000	0.245	263.124
NB	0.000	0.768	0.000	1.000	1.000	0.617	0.006
DT	0.000	0.768	0.000	1.000	1.000	0.222	0.006
RF	0.000	0.768	0.000	1.000	1.000	2.105	0.424
SVM lin	0.000	0.768	0.000	1.000	1.000	0.559	0.003
DS-ST							
KNN	0.093	0.294	0.031	1.000	1.000	0.239	254.411
NB	0.000	0.272	0.000	1.000	1.000	0.588	0.005
DT	0.000	0.728	1.000	0.000	0.728	0.223	0.005
RF	0.000	0.728	1.000	0.000	0.728	2.136	0.420
SVM lin	0.000	0.728	1.000	0.000	0.728	0.603	0.003
DS-GR							
KNN	0.000	0.768	0.000	1.000	1.000	0.247	262.969
NB	0.000	0.768	0.000	1.000	1.000	0.612	0.007
DT	0.000	0.768	0.000	1.000	1.000	0.218	0.005
RF	0.000	0.768	0.000	1.000	1.000	2.196	0.405
SVM lin	0.000	0.768	0.000	1.000	1.000	0.571	0.004

Table 31: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Strong Connectivity

Symmetry							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.000	0.703	1.000	0.000	0.703	0.255	282.863
NB	0.000	0.297	0.000	1.000	1.000	0.569	0.005
DT	0.000	0.703	1.000	0.000	0.703	0.217	0.005
RF	0.000	0.703	1.000	0.000	0.703	1.931	0.398
SVM lin	0.000	0.703	1.000	0.000	0.703	0.528	0.003
DC-CO							
KNN	0.000	0.703	1.000	0.000	0.703	0.231	264.244
NB	0.000	0.297	0.000	1.000	1.000	0.657	0.007
DT	0.000	0.703	1.000	0.000	0.703	0.252	0.005
RF	0.000	0.703	1.000	0.000	0.703	2.131	0.467
SVM lin	0.000	0.703	1.000	0.000	0.703	0.699	0.006
DC-ST							
KNN	0.000	0.324	1.000	0.000	0.324	0.239	276.987
NB	0.000	0.676	0.000	1.000	1.000	0.570	0.007
DT	0.000	0.676	0.000	1.000	1.000	0.206	0.005
RF	0.000	0.676	0.000	1.000	1.000	1.833	0.400
SVM lin	0.000	0.676	0.000	1.000	1.000	0.559	0.004
DC-GR							
KNN	0.000	0.768	0.000	1.000	1.000	0.226	271.534
NB	0.000	0.768	0.000	1.000	1.000	0.600	0.006
DT	0.000	0.768	0.000	1.000	1.000	0.258	0.006
RF	0.000	0.768	0.000	1.000	1.000	1.913	0.373
SVM lin	0.000	0.768	0.000	1.000	1.000	0.528	0.003
DS-PR							
KNN	0.000	0.586	0.000	1.000	1.000	0.238	283.178
NB	0.000	0.586	0.000	1.000	1.000	0.662	0.007
DT	0.000	0.586	0.000	1.000	1.000	0.219	0.004
RF	0.000	0.586	0.000	1.000	1.000	2.005	0.397
SVM lin	0.000	0.586	0.000	1.000	1.000	0.541	0.004
DS-CO							
KNN	0.000	0.768	0.000	1.000	1.000	0.224	254.399
NB	0.000	0.768	0.000	1.000	1.000	0.608	0.005
DT	0.000	0.768	0.000	1.000	1.000	0.241	0.005
RF	0.000	0.768	0.000	1.000	1.000	2.096	0.406
SVM lin	0.000	0.768	0.000	1.000	1.000	0.623	0.003
DS-ST							
KNN	0.000	0.272	0.000	1.000	1.000	0.252	267.104
NB	0.000	0.272	0.000	1.000	1.000	0.572	0.006
DT	0.000	0.728	1.000	0.000	0.728	0.215	0.005
RF	0.000	0.728	1.000	0.000	0.728	2.091	0.418
SVM lin	0.000	0.728	1.000	0.000	0.728	0.543	0.004
DS-GR							
KNN	0.000	0.768	0.000	1.000	1.000	0.233	262.414
NB	0.000	0.768	0.000	1.000	1.000	0.559	0.004
DT	0.000	0.768	0.000	1.000	1.000	0.213	0.005
RF	0.000	0.768	0.000	1.000	1.000	2.001	0.409
SVM lin	0.000	0.768	0.000	1.000	1.000	0.554	0.003

Table 32: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Symmetry

Irreflexitivity							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.000	0.703	1.000	0.000	0.703	0.232	134.056
NB	0.000	0.297	0.000	1.000	1.000	0.552	0.005
DT	0.000	0.703	1.000	0.000	0.703	0.212	0.005
RF	0.000	0.703	1.000	0.000	0.703	2.034	0.396
SVM lin	0.000	0.703	1.000	0.000	0.703	0.836	0.003
DC-CO							
KNN	0.000	0.703	1.000	0.000	0.703	0.230	130.454
NB	0.000	0.297	0.000	1.000	1.000	0.554	0.006
DT	0.000	0.703	1.000	0.000	0.703	0.213	0.005
RF	0.000	0.703	1.000	0.000	0.703	2.135	0.400
SVM lin	0.000	0.703	1.000	0.000	0.703	0.897	0.003
DC-ST							
KNN	0.733	0.858	0.980	0.799	0.700	0.225	141.506
NB	0.000	0.676	0.000	1.000	1.000	0.535	0.005
DT	0.733	0.858	0.980	0.799	0.700	0.210	0.005
RF	0.733	0.858	0.980	0.799	0.700	2.006	0.389
SVM lin	0.733	0.858	0.980	0.799	0.700	0.848	0.003
DC-GR							
KNN	0.000	0.768	0.000	1.000	1.000	0.239	136.520
NB	0.000	0.768	0.000	1.000	1.000	0.538	0.005
DT	0.000	0.768	0.000	1.000	1.000	0.207	0.004
RF	0.000	0.768	0.000	1.000	1.000	2.122	0.395
SVM lin	0.000	0.768	0.000	1.000	1.000	1.638	0.027
DS-PR							
KNN	0.000	0.586	0.000	1.000	1.000	0.717	141.652
NB	0.000	0.586	0.000	1.000	1.000	0.579	0.005
DT	0.000	0.586	0.000	1.000	1.000	0.235	0.007
RF	0.000	0.586	0.000	1.000	1.000	2.178	0.471
SVM lin	0.000	0.586	0.000	1.000	1.000	0.880	0.003
DS-CO							
KNN	0.000	0.768	0.000	1.000	1.000	0.238	139.090
NB	0.000	0.768	0.000	1.000	1.000	0.569	0.005
DT	0.000	0.768	0.000	1.000	1.000	0.221	0.005
RF	0.000	0.768	0.000	1.000	1.000	2.137	0.401
SVM lin	0.000	0.768	0.000	1.000	1.000	0.903	0.003
DS-ST							
KNN	0.511	0.747	0.701	0.871	0.936	0.242	131.930
NB	0.000	0.272	0.000	1.000	1.000	0.541	0.005
DT	0.511	0.747	0.701	0.871	0.936	0.209	0.005
RF	0.511	0.747	0.701	0.871	0.936	2.069	0.404
SVM lin	0.511	0.747	0.701	0.871	0.936	0.923	0.005
DS-GR							
KNN	0.000	0.768	0.000	1.000	1.000	0.225	133.770
NB	0.000	0.768	0.000	1.000	1.000	0.585	0.006
DT	0.000	0.768	0.000	1.000	1.000	0.210	0.004
RF	0.000	0.768	0.000	1.000	1.000	2.245	0.444
SVM lin	0.000	0.768	0.000	1.000	1.000	0.832	0.003

Table 33: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Irreflexitivity

B.2. Test Results for Feature Combinations

Degree, Closeness Centrality							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.849	0.937	0.954	0.897	0.957	0.269	5.274
NB	0.121	0.562	0.557	0.574	0.756	0.530	0.008
DT	0.813	0.922	0.943	0.872	0.946	0.597	0.027
RF	0.842	0.933	0.948	0.899	0.957	16.581	2.382
SVM lin	0.355	0.760	0.951	0.307	0.765	2.149	0.006
DC-CO							
KNN	0.849	0.937	0.954	0.897	0.957	0.264	5.299
NB	0.121	0.562	0.557	0.574	0.756	0.530	0.007
DT	0.813	0.922	0.943	0.871	0.946	0.614	0.044
RF	0.841	0.933	0.948	0.899	0.957	16.855	2.363
SVM lin	0.355	0.760	0.951	0.307	0.765	2.139	0.005
DC-ST							
KNN	0.855	0.937	0.903	0.953	0.902	0.258	5.279
NB	0.497	0.745	0.821	0.709	0.575	0.516	0.006
DT	0.817	0.920	0.874	0.942	0.878	0.598	0.029
RF	0.843	0.932	0.886	0.954	0.902	17.078	2.526
SVM lin	0.646	0.825	0.885	0.797	0.676	1.333	0.013
DC-GR							
KNN	0.752	0.912	0.795	0.948	0.822	0.263	5.320
NB	0.256	0.658	0.631	0.667	0.364	0.539	0.005
DT	0.738	0.908	0.781	0.946	0.815	0.528	0.020
RF	0.756	0.914	0.791	0.951	0.831	14.710	1.838
SVM lin	0.748	0.900	0.900	0.900	0.732	0.518	0.004
DS-PR							
KNN	0.766	0.886	0.868	0.899	0.859	0.274	5.366
NB	0.242	0.610	0.693	0.551	0.522	0.556	0.005
DT	0.757	0.882	0.870	0.890	0.848	0.617	0.027
RF	0.776	0.891	0.875	0.903	0.864	17.101	2.455
SVM lin	0.240	0.632	0.551	0.689	0.556	1.477	0.006
DS-CO							
KNN	0.752	0.912	0.795	0.948	0.822	0.262	5.841
NB	0.256	0.658	0.631	0.667	0.364	0.531	0.006
DT	0.738	0.908	0.781	0.946	0.814	0.544	0.021
RF	0.757	0.914	0.793	0.951	0.831	14.633	1.861
SVM lin	0.748	0.900	0.900	0.900	0.732	0.546	0.004
DS-ST							
KNN	0.540	0.811	0.849	0.708	0.886	0.268	5.390
NB	0.200	0.530	0.437	0.781	0.843	0.538	0.008
DT	0.580	0.836	0.895	0.676	0.881	0.658	0.035
RF	0.614	0.849	0.904	0.702	0.891	18.866	3.184
SVM lin	0.000	0.728	1.000	0.000	0.728	2.223	0.011
DS-GR							
KNN	0.752	0.912	0.795	0.948	0.822	0.263	5.352
NB	0.256	0.658	0.631	0.667	0.364	0.531	0.009
DT	0.739	0.908	0.782	0.946	0.815	0.556	0.022
RF	0.756	0.914	0.792	0.951	0.830	14.820	1.926
SVM lin	0.748	0.900	0.900	0.900	0.732	0.545	0.004

Table 34: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Degree, Closeness Centrality

Degree, Closeness Centrality, Katz Centrality							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.846	0.936	0.958	0.885	0.952	0.283	4.996
NB	0.244	0.680	0.763	0.484	0.778	0.536	0.006
DT	0.830	0.929	0.949	0.881	0.950	1.154	0.024
RF	0.885	0.952	0.963	0.925	0.968	23.989	2.195
SVM lin	0.416	0.778	0.950	0.372	0.782	2.735	0.005
DC-CO							
KNN	0.846	0.936	0.958	0.885	0.952	0.275	4.987
NB	0.244	0.680	0.763	0.484	0.778	0.532	0.010
DT	0.830	0.929	0.949	0.881	0.950	1.138	0.024
RF	0.886	0.952	0.963	0.926	0.969	24.228	2.197
SVM lin	0.416	0.778	0.950	0.372	0.782	2.643	0.004
DC-ST							
KNN	0.858	0.937	0.914	0.949	0.895	0.270	5.009
NB	0.642	0.793	0.983	0.701	0.612	0.516	0.008
DT	0.837	0.929	0.889	0.948	0.891	1.202	0.025
RF	0.889	0.951	0.923	0.965	0.926	25.094	2.277
SVM lin	0.682	0.834	0.943	0.782	0.675	1.717	0.004
DC-GR							
KNN	0.761	0.916	0.792	0.954	0.838	0.279	5.032
NB	0.533	0.731	0.973	0.659	0.463	0.528	0.015
DT	0.764	0.916	0.819	0.945	0.819	1.049	0.020
RF	0.805	0.932	0.816	0.967	0.882	20.419	1.775
SVM lin	0.761	0.909	0.880	0.918	0.764	0.666	0.004
DS-PR							
KNN	0.771	0.889	0.861	0.909	0.870	0.284	5.059
NB	0.380	0.648	0.879	0.485	0.547	0.548	0.007
DT	0.748	0.878	0.853	0.895	0.852	1.154	0.025
RF	0.799	0.903	0.868	0.927	0.894	24.475	2.419
SVM lin	0.466	0.734	0.751	0.722	0.656	2.089	0.007
DS-CO							
KNN	0.761	0.916	0.792	0.954	0.838	0.276	4.990
NB	0.533	0.731	0.973	0.659	0.463	0.536	0.006
DT	0.764	0.916	0.818	0.945	0.819	1.038	0.020
RF	0.805	0.932	0.816	0.967	0.881	20.869	1.792
SVM lin	0.761	0.909	0.880	0.918	0.764	0.646	0.007
DS-ST							
KNN	0.635	0.856	0.902	0.731	0.900	0.279	5.024
NB	0.395	0.638	0.541	0.898	0.934	0.536	0.006
DT	0.596	0.840	0.888	0.709	0.891	1.388	0.030
RF	0.680	0.873	0.910	0.772	0.915	28.609	2.985
SVM lin	0.163	0.734	0.958	0.134	0.748	2.660	0.006
DS-GR							
KNN	0.761	0.916	0.792	0.954	0.838	0.277	5.012
NB	0.533	0.731	0.973	0.659	0.463	0.523	0.007
DT	0.764	0.916	0.818	0.945	0.819	1.030	0.020
RF	0.806	0.932	0.817	0.967	0.882	20.794	1.835
SVM lin	0.761	0.909	0.880	0.918	0.764	0.643	0.011

Table 35: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Degree, Closeness Centrality, Katz Centrality

Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.852	0.938	0.958	0.891	0.954	0.287	5.599
NB	0.544	0.807	0.856	0.693	0.869	0.535	0.007
DT	0.845	0.935	0.955	0.889	0.953	1.337	0.023
RF	0.898	0.957	0.967	0.935	0.972	22.954	2.090
SVM lin	0.599	0.840	0.971	0.529	0.830	3.005	0.004
DC-CO							
KNN	0.852	0.938	0.958	0.891	0.954	0.283	5.620
NB	0.544	0.807	0.856	0.693	0.869	0.531	0.006
DT	0.845	0.935	0.954	0.889	0.953	1.347	0.024
RF	0.898	0.957	0.967	0.935	0.972	24.005	2.069
SVM lin	0.599	0.840	0.971	0.529	0.830	3.081	0.004
DC-ST							
KNN	0.863	0.940	0.915	0.952	0.901	0.277	5.650
NB	0.641	0.792	0.984	0.700	0.612	0.515	0.005
DT	0.851	0.935	0.899	0.952	0.900	1.302	0.023
RF	0.901	0.956	0.931	0.969	0.934	23.662	2.114
SVM lin	0.676	0.836	0.920	0.796	0.684	2.097	0.005
DC-GR							
KNN	0.764	0.918	0.792	0.956	0.843	0.281	5.598
NB	0.540	0.737	0.972	0.666	0.468	0.528	0.008
DT	0.768	0.917	0.821	0.947	0.823	1.162	0.020
RF	0.806	0.932	0.816	0.967	0.883	20.648	1.799
SVM lin	0.759	0.908	0.884	0.915	0.758	0.689	0.005
DS-PR							
KNN	0.774	0.890	0.860	0.911	0.873	0.287	5.586
NB	0.465	0.703	0.886	0.573	0.595	0.544	0.006
DT	0.749	0.878	0.852	0.896	0.853	1.285	0.024
RF	0.801	0.904	0.870	0.928	0.895	24.799	2.438
SVM lin	0.619	0.798	0.903	0.724	0.698	2.079	0.011
DS-CO							
KNN	0.764	0.918	0.792	0.956	0.843	0.301	5.657
NB	0.540	0.737	0.972	0.666	0.468	0.521	0.006
DT	0.767	0.917	0.821	0.946	0.822	1.168	0.020
RF	0.806	0.932	0.816	0.968	0.884	20.701	1.794
SVM lin	0.759	0.908	0.884	0.915	0.758	0.709	0.004
DS-ST							
KNN	0.648	0.861	0.907	0.739	0.903	0.284	5.684
NB	0.418	0.661	0.574	0.893	0.935	0.540	0.007
DT	0.608	0.844	0.891	0.718	0.895	1.568	0.030
RF	0.691	0.878	0.916	0.776	0.916	28.894	2.988
SVM lin	0.233	0.750	0.971	0.156	0.755	3.396	0.013
DS-GR							
KNN	0.764	0.918	0.792	0.956	0.843	0.279	5.661
NB	0.540	0.737	0.972	0.666	0.468	0.523	0.013
DT	0.767	0.917	0.820	0.946	0.822	1.174	0.020
RF	0.806	0.932	0.817	0.967	0.883	20.657	1.802
SVM lin	0.759	0.908	0.884	0.915	0.758	0.691	0.004

Table 36: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality

Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality, No of SCCs							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.851	0.938	0.957	0.893	0.955	0.289	6.695
NB	0.308	0.661	0.650	0.686	0.831	0.539	0.006
DT	0.851	0.938	0.956	0.894	0.955	1.326	0.021
RF	0.901	0.959	0.968	0.937	0.973	21.715	1.999
SVM lin	0.618	0.847	0.973	0.549	0.836	3.396	0.012
DC-CO							
KNN	0.851	0.938	0.957	0.893	0.955	0.286	6.833
NB	0.308	0.661	0.650	0.686	0.831	0.544	0.006
DT	0.851	0.938	0.956	0.894	0.955	1.341	0.021
RF	0.901	0.959	0.968	0.937	0.973	21.550	2.004
SVM lin	0.618	0.847	0.973	0.549	0.836	3.501	0.005
DC-ST							
KNN	0.862	0.939	0.912	0.953	0.902	0.282	6.825
NB	0.698	0.838	0.967	0.777	0.675	0.515	0.007
DT	0.856	0.937	0.905	0.952	0.900	1.329	0.023
RF	0.904	0.958	0.932	0.970	0.938	21.058	2.006
SVM lin	0.670	0.834	0.916	0.794	0.681	2.352	0.006
DC-GR							
KNN	0.764	0.917	0.794	0.955	0.841	0.282	6.894
NB	0.584	0.772	0.980	0.709	0.504	0.527	0.007
DT	0.774	0.919	0.829	0.947	0.824	1.086	0.018
RF	0.812	0.935	0.822	0.968	0.887	18.667	1.692
SVM lin	0.761	0.909	0.876	0.919	0.766	0.911	0.005
DS-PR							
KNN	0.770	0.889	0.858	0.910	0.871	0.289	6.696
NB	0.334	0.655	0.747	0.590	0.563	0.548	0.006
DT	0.749	0.878	0.853	0.896	0.853	1.381	0.024
RF	0.801	0.904	0.870	0.928	0.895	22.832	2.399
SVM lin	0.620	0.804	0.875	0.754	0.716	2.218	0.007
DS-CO							
KNN	0.764	0.917	0.794	0.955	0.841	0.288	6.812
NB	0.584	0.772	0.980	0.709	0.504	0.525	0.007
DT	0.773	0.919	0.829	0.946	0.823	1.094	0.019
RF	0.812	0.934	0.822	0.968	0.886	18.646	1.683
SVM lin	0.761	0.909	0.876	0.919	0.766	0.893	0.009
DS-ST							
KNN	0.645	0.861	0.908	0.733	0.901	0.284	6.793
NB	0.413	0.691	0.642	0.822	0.906	0.538	0.007
DT	0.612	0.846	0.893	0.720	0.895	1.559	0.028
RF	0.691	0.878	0.918	0.771	0.915	25.937	2.864
SVM lin	0.260	0.755	0.970	0.180	0.760	3.575	0.009
DS-GR							
KNN	0.764	0.917	0.794	0.955	0.841	0.280	6.748
NB	0.584	0.772	0.980	0.709	0.504	0.521	0.006
DT	0.773	0.919	0.829	0.946	0.823	1.098	0.018
RF	0.812	0.934	0.822	0.968	0.886	19.049	1.708
SVM lin	0.761	0.909	0.876	0.919	0.766	0.929	0.004

Table 37: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality, No of SCCs

Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality, No of SCCs, AVG Degree							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.851	0.938	0.957	0.894	0.955	0.306	6.786
NB	0.376	0.705	0.709	0.695	0.847	0.538	0.010
DT	0.853	0.939	0.957	0.897	0.956	1.347	0.021
RF	0.901	0.958	0.968	0.936	0.973	28.290	1.874
SVM lin	0.621	0.848	0.972	0.555	0.838	3.777	0.008
DC-CO							
KNN	0.851	0.938	0.957	0.894	0.955	0.298	6.763
NB	0.376	0.705	0.709	0.695	0.847	0.534	0.006
DT	0.852	0.938	0.956	0.895	0.956	1.357	0.021
RF	0.901	0.958	0.968	0.937	0.973	27.941	1.884
SVM lin	0.621	0.848	0.972	0.555	0.838	3.811	0.005
DC-ST							
KNN	0.863	0.940	0.912	0.953	0.903	0.284	6.655
NB	0.780	0.890	0.968	0.852	0.759	0.514	0.011
DT	0.859	0.938	0.903	0.955	0.907	1.277	0.019
RF	0.902	0.957	0.932	0.969	0.935	26.106	1.776
SVM lin	0.772	0.888	0.955	0.856	0.760	2.414	0.005
DC-GR							
KNN	0.764	0.918	0.794	0.955	0.842	0.290	6.720
NB	0.579	0.767	0.980	0.703	0.499	0.528	0.007
DT	0.777	0.921	0.829	0.948	0.829	1.137	0.018
RF	0.812	0.934	0.825	0.967	0.884	24.354	1.637
SVM lin	0.766	0.919	0.777	0.962	0.860	0.893	0.004
DS-PR							
KNN	0.763	0.885	0.848	0.912	0.872	0.297	6.669
NB	0.449	0.704	0.847	0.603	0.601	0.544	0.006
DT	0.749	0.878	0.852	0.897	0.854	1.411	0.023
RF	0.801	0.904	0.868	0.929	0.897	28.496	2.143
SVM lin	0.638	0.825	0.694	0.918	0.856	2.450	0.005
DS-CO							
KNN	0.764	0.918	0.794	0.955	0.842	0.289	6.646
NB	0.579	0.767	0.980	0.703	0.499	0.522	0.009
DT	0.774	0.919	0.827	0.947	0.826	1.137	0.019
RF	0.812	0.934	0.825	0.968	0.885	24.147	1.635
SVM lin	0.766	0.919	0.777	0.962	0.860	0.925	0.005
DS-ST							
KNN	0.638	0.857	0.903	0.735	0.901	0.291	6.702
NB	0.435	0.698	0.643	0.845	0.918	0.539	0.006
DT	0.609	0.846	0.894	0.714	0.894	1.324	0.019
RF	0.685	0.875	0.912	0.774	0.915	32.913	2.319
SVM lin	0.338	0.767	0.920	0.354	0.793	4.062	0.007
DS-GR							
KNN	0.764	0.918	0.794	0.955	0.842	0.291	6.705
NB	0.579	0.767	0.980	0.703	0.499	0.526	0.007
DT	0.775	0.920	0.828	0.947	0.826	1.144	0.019
RF	0.812	0.934	0.824	0.967	0.884	24.579	2.155
SVM lin	0.766	0.919	0.777	0.962	0.860	0.953	0.005

Table 38: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality, No of SCCs, AVG Degree

Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality, No of SCCs, AVG Degree, SCC Size							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.851	0.938	0.957	0.891	0.954	0.299	7.595
NB	0.364	0.700	0.706	0.685	0.842	0.550	0.006
DT	0.863	0.943	0.960	0.902	0.959	1.393	0.020
RF	0.909	0.962	0.969	0.946	0.977	26.464	1.840
SVM lin	0.630	0.851	0.974	0.560	0.840	4.692	0.005
DC-CO							
KNN	0.851	0.938	0.957	0.891	0.954	0.307	7.605
NB	0.364	0.700	0.706	0.685	0.842	0.536	0.006
DT	0.863	0.943	0.960	0.902	0.959	1.388	0.020
RF	0.909	0.962	0.969	0.945	0.977	26.647	1.837
SVM lin	0.630	0.851	0.974	0.560	0.840	4.638	0.006
DC-ST							
KNN	0.864	0.940	0.914	0.953	0.903	0.299	7.656
NB	0.778	0.889	0.967	0.851	0.757	0.523	0.009
DT	0.869	0.943	0.911	0.958	0.912	1.321	0.019
RF	0.910	0.960	0.935	0.973	0.943	24.845	1.745
SVM lin	0.767	0.885	0.955	0.851	0.754	2.803	0.006
DC-GR							
KNN	0.770	0.919	0.801	0.955	0.843	0.307	7.623
NB	0.578	0.767	0.979	0.703	0.499	0.527	0.007
DT	0.790	0.925	0.838	0.952	0.840	1.184	0.018
RF	0.829	0.940	0.840	0.971	0.896	23.065	1.612
SVM lin	0.768	0.919	0.791	0.958	0.850	1.011	0.005
DS-PR							
KNN	0.766	0.887	0.852	0.912	0.872	0.303	7.520
NB	0.407	0.687	0.805	0.604	0.590	0.549	0.007
DT	0.759	0.883	0.858	0.901	0.859	1.476	0.024
RF	0.814	0.910	0.878	0.933	0.902	26.410	2.124
SVM lin	0.647	0.830	0.704	0.918	0.859	2.741	0.007
DS-CO							
KNN	0.770	0.919	0.801	0.955	0.843	0.294	7.451
NB	0.578	0.767	0.979	0.703	0.499	0.523	0.006
DT	0.788	0.925	0.838	0.951	0.837	1.174	0.018
RF	0.830	0.941	0.841	0.971	0.896	22.756	1.629
SVM lin	0.768	0.919	0.791	0.958	0.850	1.052	0.004
DS-ST							
KNN	0.640	0.858	0.903	0.736	0.902	0.297	7.564
NB	0.430	0.701	0.653	0.831	0.912	0.532	0.006
DT	0.625	0.851	0.897	0.730	0.899	1.339	0.019
RF	0.700	0.881	0.918	0.783	0.919	30.683	2.327
SVM lin	0.368	0.774	0.917	0.392	0.802	4.641	0.005
DS-GR							
KNN	0.770	0.919	0.801	0.955	0.843	0.296	7.563
NB	0.578	0.767	0.979	0.703	0.499	0.526	0.035
DT	0.790	0.925	0.838	0.952	0.840	1.182	0.018
RF	0.829	0.940	0.841	0.970	0.895	23.062	1.601
SVM lin	0.768	0.919	0.791	0.958	0.850	1.028	0.005

Table 39: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality, No of SCCs, AVG Degree, SCC Size

Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality, No of SCCs, AVG Degree, SCC Size, Irreflexitivity							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.849	0.937	0.957	0.889	0.953	0.311	11.135
NB	0.317	0.669	0.664	0.681	0.831	0.537	0.006
DT	0.863	0.943	0.960	0.903	0.959	1.408	0.020
RF	0.909	0.962	0.969	0.945	0.977	24.366	1.839
SVM lin	0.637	0.854	0.973	0.573	0.844	4.557	0.005
DC-CO							
KNN	0.849	0.937	0.957	0.889	0.953	0.305	11.155
NB	0.317	0.669	0.664	0.681	0.831	0.543	0.030
DT	0.863	0.943	0.960	0.903	0.959	1.401	0.021
RF	0.909	0.962	0.968	0.946	0.977	24.194	1.827
SVM lin	0.637	0.854	0.973	0.573	0.844	4.485	0.004
DC-ST							
KNN	0.864	0.940	0.915	0.952	0.902	0.301	11.239
NB	0.786	0.892	0.978	0.850	0.758	0.519	0.007
DT	0.872	0.944	0.914	0.959	0.914	1.261	0.019
RF	0.912	0.961	0.935	0.974	0.945	23.107	1.748
SVM lin	0.815	0.912	0.949	0.895	0.812	2.802	0.005
DC-GR							
KNN	0.768	0.919	0.800	0.954	0.841	0.303	11.280
NB	0.584	0.770	0.984	0.705	0.502	0.526	0.009
DT	0.791	0.926	0.838	0.952	0.841	1.184	0.019
RF	0.830	0.941	0.841	0.971	0.896	20.998	1.613
SVM lin	0.768	0.919	0.791	0.958	0.849	1.244	0.005
DS-PR							
KNN	0.765	0.886	0.851	0.911	0.871	0.309	11.162
NB	0.359	0.666	0.768	0.595	0.572	0.551	0.008
DT	0.761	0.884	0.860	0.902	0.861	1.473	0.023
RF	0.814	0.910	0.877	0.933	0.902	24.847	2.152
SVM lin	0.658	0.835	0.722	0.915	0.857	2.885	0.006
DS-CO							
KNN	0.768	0.919	0.800	0.954	0.841	0.302	11.149
NB	0.584	0.770	0.984	0.705	0.502	0.525	0.009
DT	0.789	0.925	0.838	0.951	0.837	1.181	0.018
RF	0.829	0.940	0.839	0.971	0.896	21.159	1.610
SVM lin	0.768	0.919	0.791	0.958	0.849	1.218	0.008
DS-ST							
KNN	0.643	0.859	0.904	0.739	0.903	0.303	11.221
NB	0.503	0.737	0.684	0.882	0.939	0.539	0.006
DT	0.629	0.853	0.899	0.730	0.899	1.342	0.019
RF	0.705	0.883	0.919	0.788	0.921	28.611	2.351
SVM lin	0.486	0.797	0.860	0.625	0.860	5.268	0.005
DS-GR							
KNN	0.768	0.919	0.800	0.954	0.841	0.305	11.222
NB	0.584	0.770	0.984	0.705	0.502	0.533	0.006
DT	0.790	0.925	0.837	0.952	0.841	1.194	0.019
RF	0.829	0.940	0.840	0.971	0.896	20.863	1.604
SVM lin	0.768	0.919	0.791	0.958	0.849	1.268	0.012

Table 40: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality, No of SCCs, AVG Degree, SCC Size, Irreflexitivity

Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality, No of SCCs, AVG Degree, SCC Size, Irreflexitivity, Strong Connectivity							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.848	0.937	0.957	0.888	0.953	0.319	11.811
NB	0.311	0.667	0.664	0.673	0.828	0.537	0.012
DT	0.864	0.943	0.960	0.904	0.960	1.423	0.021
RF	0.909	0.962	0.969	0.946	0.977	22.954	1.868
SVM lin	0.637	0.854	0.972	0.577	0.845	4.381	0.004
DC-CO							
KNN	0.848	0.937	0.957	0.888	0.953	0.308	11.806
NB	0.311	0.667	0.664	0.673	0.828	0.531	0.010
DT	0.863	0.943	0.959	0.903	0.959	1.415	0.021
RF	0.909	0.962	0.968	0.946	0.977	22.915	1.855
SVM lin	0.637	0.854	0.972	0.577	0.845	5.521	0.006
DC-ST							
KNN	0.864	0.940	0.915	0.952	0.902	0.300	11.918
NB	0.784	0.890	0.980	0.847	0.755	0.521	0.011
DT	0.872	0.944	0.915	0.958	0.913	1.280	0.019
RF	0.913	0.962	0.936	0.974	0.946	21.842	1.778
SVM lin	0.815	0.912	0.949	0.895	0.812	2.781	0.010
DC-GR							
KNN	0.768	0.919	0.800	0.954	0.841	0.305	11.812
NB	0.584	0.769	0.984	0.704	0.502	0.526	0.006
DT	0.790	0.925	0.839	0.951	0.839	1.213	0.019
RF	0.830	0.940	0.840	0.971	0.897	19.906	1.633
SVM lin	0.768	0.919	0.791	0.958	0.849	1.271	0.009
DS-PR							
KNN	0.765	0.886	0.851	0.911	0.871	0.321	11.915
NB	0.357	0.665	0.768	0.592	0.571	0.553	0.008
DT	0.760	0.884	0.858	0.901	0.860	1.471	0.023
RF	0.815	0.910	0.878	0.933	0.903	23.696	2.181
SVM lin	0.656	0.834	0.718	0.916	0.858	2.889	0.008
DS-CO							
KNN	0.768	0.919	0.800	0.954	0.841	0.306	11.792
NB	0.584	0.769	0.984	0.704	0.502	0.524	0.008
DT	0.791	0.926	0.838	0.952	0.841	1.190	0.019
RF	0.829	0.940	0.840	0.970	0.896	19.813	1.613
SVM lin	0.768	0.919	0.791	0.958	0.849	1.204	0.006
DS-ST							
KNN	0.643	0.859	0.904	0.739	0.903	0.304	11.803
NB	0.532	0.759	0.713	0.882	0.942	0.533	0.006
DT	0.629	0.853	0.898	0.732	0.900	1.337	0.020
RF	0.705	0.883	0.919	0.787	0.920	27.366	2.400
SVM lin	0.554	0.810	0.831	0.755	0.901	5.138	0.008
DS-GR							
KNN	0.768	0.919	0.800	0.954	0.841	0.303	11.955
NB	0.584	0.769	0.984	0.704	0.502	0.538	0.008
DT	0.790	0.925	0.837	0.952	0.840	1.200	0.019
RF	0.829	0.940	0.841	0.970	0.894	20.068	1.642
SVM lin	0.768	0.919	0.791	0.958	0.849	1.261	0.006

Table 41: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality, No of SCCs, AVG Degree, SCC Size, Irreflexitivity, Strong Connectivity

Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality, No of SCCs, AVG Degree, SCC Size, Irreflexitivity, Strong Connectivity, Aperiodicity, Symmetry							
Classifier	**MCC**	**Accuracy**	**TPR**	**TNR**	**Precision**	**Time Train**	**Time Predict**
DC-PR							
KNN	0.849	0.937	0.957	0.890	0.954	0.350	14.509
NB	0.316	0.669	0.665	0.678	0.830	0.538	0.008
DT	0.861	0.942	0.959	0.902	0.959	1.421	0.021
RF	0.909	0.962	0.968	0.946	0.977	20.470	1.902
SVM lin	0.622	0.849	0.955	0.599	0.849	5.050	0.005
DC-CO							
KNN	0.849	0.937	0.957	0.890	0.954	0.321	14.460
NB	0.316	0.669	0.665	0.678	0.830	0.535	0.007
DT	0.863	0.943	0.960	0.903	0.959	1.426	0.021
RF	0.909	0.962	0.968	0.946	0.977	21.148	1.885
SVM lin	0.622	0.849	0.955	0.599	0.849	5.189	0.005
DC-ST							
KNN	0.863	0.940	0.915	0.952	0.901	0.315	14.523
NB	0.765	0.877	0.985	0.825	0.730	0.519	0.007
DT	0.869	0.943	0.914	0.956	0.909	1.278	0.019
RF	0.913	0.962	0.935	0.975	0.947	19.946	1.821
SVM lin	0.814	0.912	0.949	0.895	0.812	3.261	0.006
DC-GR							
KNN	0.769	0.919	0.800	0.955	0.842	0.319	14.517
NB	0.580	0.766	0.985	0.700	0.498	0.527	0.007
DT	0.792	0.926	0.839	0.952	0.841	1.207	0.019
RF	0.831	0.941	0.841	0.971	0.898	17.905	1.631
SVM lin	0.780	0.923	0.795	0.962	0.864	1.149	0.008
DS-PR							
KNN	0.765	0.886	0.852	0.911	0.871	0.328	14.472
NB	0.353	0.662	0.769	0.586	0.568	0.550	0.007
DT	0.762	0.885	0.860	0.902	0.862	1.489	0.023
RF	0.816	0.911	0.879	0.933	0.903	21.696	2.216
SVM lin	0.659	0.835	0.713	0.921	0.865	3.358	0.006
DS-CO							
KNN	0.769	0.919	0.800	0.955	0.842	0.317	14.501
NB	0.580	0.766	0.985	0.700	0.498	0.527	0.007
DT	0.792	0.926	0.839	0.952	0.842	1.206	0.020
RF	0.831	0.941	0.841	0.971	0.898	18.215	1.634
SVM lin	0.780	0.923	0.795	0.962	0.864	1.175	0.005
DS-ST							
KNN	0.640	0.858	0.904	0.735	0.902	0.326	14.496
NB	0.534	0.759	0.711	0.886	0.944	0.535	0.006
DT	0.627	0.852	0.899	0.728	0.899	1.368	0.020
RF	0.705	0.883	0.920	0.785	0.920	24.082	2.436
SVM lin	0.554	0.810	0.830	0.755	0.901	6.091	0.006
DS-GR							
KNN	0.769	0.919	0.800	0.955	0.842	0.319	14.513
NB	0.580	0.766	0.985	0.700	0.498	0.524	0.006
DT	0.791	0.926	0.838	0.952	0.841	1.197	0.019
RF	0.830	0.940	0.840	0.971	0.897	18.092	1.636
SVM lin	0.780	0.923	0.795	0.962	0.864	1.155	0.005

Table 42: Results for Test Set *balanced-kwt-test*. Classifiers Trained with Training Set *balanced-kwt-train*, Features Used: Degree, Closeness Centrality, Katz Centrality, Betweenness Centrality, No of SCCs, AVG Degree, SCC Size, Irreflexitivity, Strong Connectivity, Aperiodicity, Symmetry

B.3. Test Results for *balanced-kwt-train* and *balanced-kwt-test* split up in graph types

Classifier	Degree	Katz	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	Is SC	is Sym.	is Ir.
DC-PR											
KNN	0.128	0.000	0.069	0.052	**0.404**	0.000	0.000	0.000	0.022	0.000	0.000
NB	**0.033**	0.030	-0.039	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.260	-0.001	0.048	0.042	**0.404**	0.000	0.000	0.000	0.022	0.000	0.000
RF	0.260	0.014	0.108	0.051	**0.404**	0.000	0.000	0.000	0.022	0.000	0.000
SVM lin	0.000	0.002	0.000	0.024	**0.404**	0.000	0.000	0.000	0.022	0.000	0.000
DC-CO											
KNN	0.128	0.000	0.069	0.052	**0.404**	0.000	0.000	0.000	0.022	0.000	0.000
NB	0.033	0.030	-0.039	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.260	-0.001	0.033	0.042	**0.404**	0.000	0.000	0.000	0.022	0.000	0.000
RF	0.260	0.014	0.115	0.041	**0.404**	0.000	0.000	0.000	0.022	0.000	0.000
SVM lin	0.000	0.002	0.000	0.024	**0.404**	0.000	0.000	0.000	0.022	0.000	0.000
DC-ST											
KNN	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
NB	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DC-GR											
KNN	**1.000**	0.078	0.463	0.480	0.774	0.000	0.000	0.000	0.000	0.000	0.000
NB	0.027	0.022	**0.033**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**1.000**	0.054	0.252	-0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**1.000**	0.162	0.612	-0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.866**	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 43: MCCs Values Achieved When Testing the Features Separately for Credulous Acceptance Under Several Semantics: Training with the *balanced-kwt-train* Dataset and Testing Only the WS-Graphs of the *balanced-kwt-test* Dataset

Classifier	Degree	Katz	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	Is SC	is Sym.	is Ir.
DS-PR											
KNN	0.319	0.013	0.109	0.117	**0.404**	0.000	0.000	0.000	0.000	0.000	0.000
NB	**0.044**	0.033	-0.025	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.260	0.022	0.064	0.052	**0.404**	0.000	0.000	0.000	0.000	0.000	0.000
RF	0.260	0.021	0.172	0.051	**0.404**	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.017	0.097	-0.001	0.369	**0.404**	0.000	0.000	0.000	0.000	0.000	0.000
DS-CO											
KNN	**1.000**	0.078	0.463	0.480	0.774	0.000	0.000	0.000	0.000	0.000	0.000
NB	0.027	0.022	**0.033**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**1.000**	0.054	0.260	-0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**1.000**	0.165	0.866	-0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.866	0.000	1000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DS-ST											
KNN	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
NB	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DS-GR											
KNN	**1.000**	0.078	0.463	0.480	0.774	0.000	0.000	0.000	0.000	0.000	0.000
NB	0.027	0.022	0.033	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**1.000**	0.055	0.255	-0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**1.000**	0.160	0.654	-0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.866	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 44: MCCs Values Achieved When Testing the Features Separately for Skeptical Acceptance Under Several Semantics: Training with the *balanced-kwt-train* Dataset and Testing Only the WS-Graphs of the *balanced-kwt-test* Dataset

Classifier	**Degree**	**Katz**	**Close-ness C.**	**Between-ness C.**	**Size Sccs**	**SCCs**	**AVG Deg.**	**is Aperiod**	**Is SC**	**is Sym.**	**is Ir.**
DS-PR											
KNN	**0.181**	0.062	0.073	0.050	**0.087**	0.005	-0.065	0.000	0.036	0.000	0.000
NB	**0.178**	0.168	-0.155	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.201**	0.027	0.071	0.061	**0.059**	0.045	-0.074	0.000	0.036	0.000	0.000
RF	**0.202**	0.047	0.074	0.057	**0.059**	0.045	-0.074	0.000	0.036	0.000	0.000
SVM lin	0.016	**0.061**	0.000	0.030	**0.036**	0.000	0.000	0.000	0.036	0.000	0.000
DC-CO											
KNN	**0.181**	0.062	0.073	0.050	0.087	0.005	-0.065	0.000	0.036	0.000	0.000
NB	**0.178**	0.168	**-0.155**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.201**	0.038	0.074	0.061	0.059	0.045	-0.074	0.000	0.036	0.000	0.000
RF	**0.199**	0.049	0.065	0.057	0.059	0.045	-0.074	0.000	0.036	0.000	0.000
SVM lin	0.016	**0.061**	0.000	0.030	0.036	0.000	0.000	0.000	0.036	0.000	0.000
DC-ST											
KNN	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
NB	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DC-GR											
KNN	**0.608**	0.111	0.195	0.385	0.221	0.000	-0.015	0.000	0.000	0.000	0.000
NB	**0.121**	0.092	0.091	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.847**	0.063	0.196	-0.008	0.000	0.000	-0.015	0.000	0.000	0.000	0.000
RF	**0.847**	0.091	0.272	-0.009	0.000	0.000	-0.015	0.000	0.000	0.000	0.000
SVM lin	**0.473**	0.000	0.261	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 45: MCCs Values Achieved When Testing the Features Separately for Credulous Acceptance Under Several Semantics: Training with the *balanced-kwt-train* Dataset and Testing Only the SCC-Graphs of the *balanced-kwt-test* Dataset

Classifier	Degree	Katz	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	Is SC	is Sym.	is Ir.
DS-PR											
KNN	**0.173**	0.100	0.095	0.081	**0.087**	-0.026	-0.053	0.000	0.000	0.000	0.000
NB	**0.187**	0.164	-0.132	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.177**	0.060	0.100	0.091	**0.094**	0.000	-0.053	0.000	0.000	0.000	0.000
RF	**0.176**	0.088	0.122	0.087	**0.094**	0.000	-0.053	0.000	0.000	0.000	0.000
SVM lin	0.129	**0.146**	-0.072	**0.164**	**0.059**	0.000	0.000	0.000	0.000	0.000	0.000
DS-CO											
KNN	**0.608**	0.111	0.195	0.385	0.221	0.000	-0.015	0.000	0.000	0.000	0.000
NB	**0.121**	0.092	**0.091**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.847**	0.069	0.196	-0.008	0.000	0.000	-0.015	0.000	0.000	0.000	0.000
RF	**0.847**	0.095	0.266	-0.009	0.000	0.000	-0.015	0.000	0.000	0.000	0.000
SVM lin	**0.473**	**0.000**	0.261	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DS-ST											
KNN	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
NB	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DS-GR											
KNN	**0.608**	0.111	0.195	0.385	0.221	0.000	-0.015	0.000	0.000	0.000	0.000
NB	**0.121**	0.092	0.091	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.847**	0.069	0.196	-0.008	0.000	0.000	-0.015	0.000	0.000	0.000	0.000
RF	**0.847**	0.107	0.272	-0.009	0.000	0.000	-0.015	0.000	0.000	0.000	0.000
SVM lin	**0.473**	0.000	0.261	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 46: MCCs Values Achieved When Testing the Features Separately for Skeptical Acceptance Under Several Semantics: Training with the *balanced-kwt-train* Dataset and Testing Only the SCC-Graphs of the *balanced-kwt-test* Dataset

Classifier	Degree	Katz	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	Is SC	is Sym.	is Ir.
DS-PR											
KNN	**0.391**	0.266	0.315	0.318	**0.054**	-0.000	0.006	0.000	0.000	0.000	0.000
NB	**0.466**	0.350	-0.379	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.389**	0.150	0.343	0.260	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.389**	0.233	0.333	0.258	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.000	**0.131**	0.000	**0.087**	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
DC-CO											
KNN	**0.391**	0.266	0.315	0.318	0.054	-0.000	0.006	0.000	0.000	0.000	0.000
NB	**0.466**	0.350	**-0.379**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.389**	0.155	0.343	0.260	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.389**	0.235	0.333	0.259	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.000**	**0.131**	0.000	0.087	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DC-ST											
KNN	**0.398**	0.199	0.285	0.290	0.057	-0.057	0.021	0.000	0.000	0.000	0.170
NB	**0.449**	0.313	0.332	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.428**	0.114	0.283	0.257	-0.077	0.000	0.000	0.000	0.000	0.000	0.170
RF	**0.428**	0.194	0.274	0.254	-0.077	0.000	0.000	0.000	0.000	0.000	0.170
SVM lin	**0.290**	0.000	0.040	0.000	0.073	0.000	0.000	0.000	0.000	0.000	0.170
DC-GR											
KNN	**0.591**	0.322	0.582	0.435	0.087	0.001	0.007	0.000	0.000	0.000	0.000
NB	**0.516**	0.369	0.389	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.672**	0.154	0.567	-0.039	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.672**	0.289	0.572	-0.041	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.574**	0.000	0.213	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 47: MCCs Values Achieved When Testing the Features Separately for Credulous Acceptance Under Several Semantics: Training with the *balanced-kwt-train* Dataset and Testing Only the Grounded-Graphs of the *balanced-kwt-test* Dataset

Classifier	Degree	Katz	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	Is SC	is Sym.	is Ir.
DS-PR											
KNN	**0.482**	0.322	**0.581**	0.421	**0.087**	0.001	0.007	0.000	0.000	0.000	0.000
NB	**0.516**	0.365	-0.366	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.533**	0.162	**0.563**	0.434	**0.090**	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.533**	0.296	**0.567**	0.435	**0.090**	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.085	**0.159**	-0.092	**0.317**	**0.055**	0.000	0.000	0.000	0.000	0.000	0.000
DS-CO											
KNN	**0.591**	0.322	0.582	0.435	0.087	0.001	0.007	0.000	0.000	0.000	0.000
NB	**0.516**	0.369	**0.389**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.672**	0.151	0.564	-0.039	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.672**	0.289	0.572	-0.041	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.574**	**0.000**	0.213	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DS-ST											
KNN	**0.381**	0.194	**0.462**	0.283	0.068	0.038	-0.007	0.000	0.000	0.000	0.220
NB	**0.392**	0.293	0.306	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.362**	0.082	**0.431**	0.310	0.057	-0.053	0.064	0.000	0.000	0.000	0.220
RF	**0.362**	0.168	**0.443**	0.303	0.063	-0.053	0.064	0.000	0.000	0.000	0.220
SVM lin	**0.000**	0.017	0.000	**0.032**	0.000	0.017	0.000	0.000	0.000	0.000	0.220
DS-GR											
KNN	**0.591**	0.322	0.582	0.435	0.087	0.001	0.007	0.000	0.000	0.000	0.000
NB	**0.516**	0.369	0.389	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.672**	0.151	0.564	-0.039	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.672**	0.290	0.570	-0.041	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.574**	0.000	0.213	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 48: MCCs Values Achieved When Testing the Features Separately for Skeptical Acceptance Under Several Semantics: Training with the *balanced-kwt-train* Dataset and Testing Only the Grounded-Graphs of the *balanced-kwt-test* Dataset

Classifier	Degree	Katz	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	Is SC	is Sym.	is Ir.
DS-PR											
KNN	**0.216**	0.089	**0.206**	0.136	**0.120**	0.057	0.008	0.000	0.000	0.000	0.000
NB	**0.211**	0.153	-0.147	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.247**	0.051	**0.081**	0.096	**0.192**	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.247**	0.086	**0.171**	0.091	**0.192**	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.000	**0.066**	0.000	**0.147**	**0.192**	0.000	0.000	0.000	0.000	0.000	0.000
DC-CO											
KNN	**0.216**	0.089	0.206	0.136	0.120	0.057	0.008	0.000	0.000	0.000	0.000
NB	**0.211**	0.153	**-0.147**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.247**	0.056	0.083	0.096	0.192	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.247**	0.097	0.169	0.094	0.192	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.000**	**0.066**	0.000	0.147	**0.192**	0.000	0.000	0.000	0.000	0.000	0.000
DC-ST											
KNN	**0.162**	0.042	**0.108**	0.072	0.051	0.035	-0.053	0.000	0.000	0.000	0.000
NB	**0.176**	0.121	0.114	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.185**	0.020	**0.045**	0.041	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.185**	0.039	**0.095**	0.039	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.237**	0.000	0.158	**0.000**	0.107	0.000	0.000	0.000	0.000	0.000	0.000
DC-GR											
KNN	**0.605**	0.287	**0.702**	0.546	0.531	0.000	0.000	0.000	0.000	0.000	0.000
NB	**0.217**	0.180	0.246	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.766**	0.186	0.571	-0.026	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.766**	0.394	0.696	-0.026	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.492**	0.000	**0.721**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 49: MCCs Values Achieved When Testing the Features Separately for Credulous Acceptance Under Several Semantics: Training with the *balanced-kwt-train* Dataset and Testing Only the ER-Graphs of the *balanced-kwt-test* Dataset

Classifier	Degree	Katz	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	Is SC	is Sym.	is Ir.
DS-PR											
KNN	**0.198**	0.117	**0.277**	0.204	**0.223**	0.101	0.004	0.000	0.000	0.000	0.000
NB	**0.235**	0.173	-0.138	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.246**	0.086	**0.133**	0.156	**0.223**	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.246**	0.145	**0.244**	0.156	**0.223**	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.087	**0.236**	-0.002	**0.234**	**0.223**	0.000	0.000	0.000	0.000	0.000	0.000
DS-CO											
KNN	**0.605**	0.287	**0.702**	0.546	0.531	0.000	0.000	0.000	0.000	0.000	0.000
NB	**0.217**	0.180	**0.246**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.766**	0.186	0.572	-0.026	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.766**	0.395	0.698	-0.026	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.492**	**0.000**	**0.721**	0.000	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
DS-ST											
KNN	**-0.008**	0.014	**0.045**	0.034	**0.206**	0.043	-0.202	0.000	0.000	0.000	0.000
NB	**0.124**	0.082	0.084	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.015**	0.014	**0.030**	0.017	**0.117**	-0.143	-0.259	0.000	0.000	0.000	0.000
RF	**0.015**	0.020	**0.053**	0.020	**0.117**	-0.143	-0.259	0.000	0.000	0.000	0.000
SVM lin	**0.000**	0.000	0.000	**0.047**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DS-GR											
KNN	**0.605**	0.287	**0.702**	0.546	0.531	0.000	0.000	0.000	0.000	0.000	0.000
NB	**0.217**	0.180	**0.246**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.766**	0.189	0.572	-0.026	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.766**	0.393	0.700	-0.026	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.492**	0.000	**0.721**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 50: MCCs Values Achieved When Testing the Features Separately for Skeptical Acceptance Under Several Semantics: Training with the *balanced-kwt-train* Dataset and Testing Only the ER-Graphs of the *balanced-kwt-test* Dataset

Classifier	Degree	Katz	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	Is SC	is Sym.	is Ir.
					DS-PR						
KNN	**0.515**	0.175	0.196	0.333	0.027	0.000	0.000	0.000	0.000	0.000	0.000
NB	0.392	0.134	-0.106	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.573	0.104	**0.178**	0.313	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	0.573	0.164	**0.201**	0.308	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.000	**0.217**	0.000	**0.330**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
					DC-CO						
KNN	0.515	0.175	**0.196**	0.333	0.027	0.000	0.000	0.000	0.000	0.000	0.000
NB	0.392	0.134	**-0.106**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.573	0.104	0.179	0.313	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	0.573	0.164	0.201	0.307	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.000**	**0.217**	**0.000**	**0.330**	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
					DC-ST						
KNN	**0.519**	0.157	0.196	0.336	**0.027**	0.000	0.000	0.000	0.000	0.000	0.000
NB	0.162	0.125	0.107	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.577	0.098	**0.176**	0.316	**-0.034**	0.000	0.000	0.000	0.000	0.000	0.000
RF	0.577	0.148	**0.196**	0.311	**-0.034**	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.439	0.000	0.105	**0.000**	0.039	0.000	0.000	0.000	0.000	0.000	0.000
					DC-GR						
KNN	0.417	0.340	**0.416**	**0.483**	0.377	0.000	0.000	0.000	0.000	0.000	0.000
NB	**0.331**	0.194	**0.180**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.515	0.210	0.402	-0.022	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	0.515	0.343	0.419	-0.022	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.494**	0.000	**0.133**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 51: MCCs Values Achieved When Testing the Features Separately for Credulous Acceptance Under Several Semantics: Training with the *balanced-kwt-train* Dataset and Testing Only the Barabasi-Graphs of the *balanced-kwt-test* Dataset

Classifier	Degree	Katz	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	Is SC	is Sym.	is Ir.
DS-PR											
KNN	**0.388**	0.321	**0.415**	**0.483**	**0.377**	0.000	0.000	0.000	0.000	0.000	0.000
NB	**0.303**	0.185	-0.169	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.512**	0.200	**0.400**	0.489	**0.385**	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.512**	0.326	**0.420**	0.489	**0.385**	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.104	**0.023**	-0.078	**0.460**	**0.025**	0.000	0.000	0.000	0.000	0.000	0.000
DS-CO											
KNN	**0.417**	0.340	**0.416**	**0.483**	0.377	0.000	0.000	0.000	0.000	0.000	0.000
NB	**0.331**	0.194	**0.180**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.515**	0.211	0.402	-0.022	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.515**	0.348	0.420	-0.022	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.494**	**0.000**	**0.133**	**0.000**	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
DS-ST											
KNN	**0.333**	0.306	**0.381**	**0.471**	**0.369**	0.000	0.000	0.000	0.000	0.000	0.000
NB	**0.192**	0.164	0.169	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.433**	0.193	**0.362**	**0.487**	**0.360**	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.432**	0.307	**0.375**	**0.486**	**0.374**	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.000**	0.000	0.000	**0.172**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DS-GR											
KNN	**0.417**	0.340	**0.416**	**0.483**	0.377	0.000	0.000	0.000	0.000	0.000	0.000
NB	**0.331**	0.194	**0.180**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.515**	0.207	0.402	-0.022	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.515**	0.346	0.420	-0.023	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.494**	0.000	**0.133**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 52: MCCs Values Achieved When Testing the Features Separately for Skeptical Acceptance Under Several Semantics: Training with the *balanced-kwt-train* Dataset and Testing Only the Barabasi-Graphs of the *balanced-kwt-test* Dataset

Classifier	Degree	Katz	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	Is SC	is Sym.	is Ir.
DS-PR											
KNN	**0.194**	0.080	**0.213**	**0.146**	**0.099**	-0.019	-0.009	0.000	0.000	0.000	0.000
NB	**0.201**	0.131	-0.190	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.217**	0.056	**0.154**	0.135	**0.110**	-0.022	0.000	0.000	0.000	0.000	0.000
RF	**0.214**	0.098	**0.190**	0.135	**0.110**	-0.022	0.000	0.000	0.000	0.000	0.000
SVM lin	0.000	**0.010**	0.000	**-0.045**	**0.014**	0.000	0.000	0.000	0.000	0.000	0.000
DC-CO											
KNN	**0.194**	0.080	**0.213**	**0.146**	0.099	-0.019	-0.009	0.000	0.000	0.000	0.000
NB	**0.201**	0.131	**-0.190**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.217**	0.054	0.157	0.135	0.110	-0.022	0.000	0.000	0.000	0.000	0.000
RF	**0.214**	0.100	0.200	0.135	0.110	-0.022	-0.023	0.000	0.000	0.000	0.000
SVM lin	**0.000**	**0.010**	**0.000**	**-0.045**	**0.014**	0.000	0.000	0.000	0.000	0.000	0.000
DC-ST											
KNN	**0.175**	0.034	**0.154**	**0.146**	**0.074**	0.013	0.009	0.000	0.000	0.000	0.032
NB	**0.164**	0.105	0.147	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.225**	0.031	**0.092**	**0.120**	**-0.009**	0.000	0.000	0.000	0.000	0.000	0.032
RF	**0.225**	0.051	**0.120**	**0.120**	**-0.009**	0.000	0.000	0.000	0.000	0.000	0.032
SVM lin	**0.238**	0.000	0.074	**0.000**	0.070	0.000	0.096	0.000	0.000	0.000	0.032
DC-GR											
KNN	**0.606**	0.267	**0.509**	**0.486**	0.425	0.000	0.000	0.000	0.000	0.000	0.000
NB	**0.317**	0.240	**0.403**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.631**	0.198	0.513	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.629**	0.315	0.588	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.637**	0.000	**0.317**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 53: MCCs Values Achieved When Testing the Features Separately for Credulous Acceptance Under Several Semantics: Training with the *balanced-kwt-train* Dataset and Testing Only the Stable-Graphs of the *balanced-kwt-test* Dataset

Classifier	Degree	Katz	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	Is SC	is Sym.	is Ir.
DS-PR											
KNN	**0.265**	0.161	**0.333**	**0.303**	**0.237**	-0.019	0.019	0.000	0.000	0.000	0.000
NB	**0.246**	0.192	-0.209	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.243**	0.111	**0.259**	**0.265**	**0.256**	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.243**	0.168	**0.348**	0.270	**0.256**	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.223	**0.272**	0.033	**0.333**	**0.261**	0.000	0.000	0.000	0.000	0.000	0.000
DS-CO											
KNN	**0.606**	0.267	**0.509**	**0.486**	0.425	0.000	0.000	0.000	0.000	0.000	0.000
NB	**0.317**	0.240	**0.403**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.631**	0.206	0.519	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.630**	0.314	0.583	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.637**	**0.000**	**0.317**	**0.000**	**0.000**	0.000	0.000	0.000	0.000	0.000	0.000
DS-ST											
KNN	**0.049**	0.014	**0.105**	**0.021**	**0.107**	0.035	-0.227	0.000	0.000	0.000	0.093
NB	**0.115**	0.083	0.067	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.042**	0.003	**0.100**	**0.043**	**0.114**	0.024	-0.212	0.000	0.000	0.000	0.093
RF	**0.040**	0.012	**0.121**	**0.035**	**0.114**	0.024	-0.212	0.000	0.000	0.000	0.093
SVM lin	**0.000**	0.000	0.000	**-0.018**	0.000	0.000	0.000	0.000	0.000	0.000	**0.093**
DS-GR											
KNN	**0.606**	0.267	**0.509**	**0.486**	0.425	0.000	0.000	0.000	0.000	0.000	0.000
NB	**0.317**	0.240	**0.403**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.631**	0.208	0.516	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	**0.631**	0.323	0.584	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	**0.637**	0.000	**0.317**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 54: MCCs Values Achieved When Testing the Features Separately for Skeptical Acceptance Under Several Semantics: Training with the *balanced-kwt-train* Dataset and Testing Only the Stable-Graphs of the *balanced-kwt-test* Dataset

C. Overview Test Results for Individual Features and Rank for *kwt-train* and *kwt-test*

Classifier	Degree C.	Katz C.	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	is SC	is Sym.	is Ir.
					DC-PR						
KNN	0.975	**0.956**	0.977	**0.851**	**0.145**	0.000	0.177	0.000	0.000	0.000	0.000
NB	0.281	0.324	0.296	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.977**	0.889	**0.978**	0.817	0.000	0.000	**0.212**	0.000	0.000	0.000	0.000
RF	**0.977**	0.926	**0.978**	0.820	0.000	0.000	0.200	0.000	0.000	0.000	0.000
SVM lin	0.230	-0.076	0.103	0.814	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rank	2	3	1	4	6	10	5	10	10	10	10
					DC-CO						
KNN	0.975	**0.956**	0.977	**0.851**	**0.145**	0.000	0.177	0.000	0.000	0.000	0.000
NB	0.281	0.324	0.296	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.977**	0.889	0.978	0.817	0.000	0.000	**0.212**	0.000	0.000	0.000	0.000
RF	**0.977**	0.924	**0.979**	0.820	0.000	0.000	0.192	0.000	0.000	0.000	0.000
SVM lin	0.230	-0.076	0.103	0.814	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rank	2	3	1	4	6	10	5	10	10	10	10
					DC-ST						
KNN	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
NB	0.034	0.025	0.015	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.033	0.016	-0.001	0.007	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RF	0.020	0.024	-0.001	0.007	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SVM lin	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rank	-	-	-	-	-	-	-	-	-	-	-
					DC-GR						
KNN	0.971	**0.969**	0.970	-0.001	0.000	**0.721**	0.711	0.000	0.000	0.000	0.000
NB	-0.157	0.507	-0.151	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.976**	0.959	**0.973**	-0.009	0.000	**0.721**	0.721	0.000	0.000	0.000	0.000
RF	**0.976**	0.967	0.972	-0.009	0.000	**0.721**	0.716	0.000	0.000	0.000	0.000
SVM lin	0.938	0.917	0.970	0.000	0.000	**0.721**	**0.721**	0.000	0.000	0.000	0.000
Rank	1	3	2	10	10	4	5	10	10	10	10

Table 55: MCCs Values Achieved When Testing the Features Separately for Credulous Acceptance Under Several Semantics: Using the *kwt-train* and *kwt-test* Datasets

Classifier	Degree C.	Katz C.	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	is SC	is Sym.	is Ir.
					DS-PR						
KNN	0.848	**0.910**	0.884	**0.598**	**0.624**	**0.166**	0.119	0.000	0.000	0.000	0.000
NB	0.377	0.584	0.442	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.923	0.872	0.921	0.563	**0.624**	**0.166**	0.159	**0.165**	0.000	0.000	0.000
RF	**0.924**	0.892	**0.922**	0.566	**0.624**	**0.166**	**0.160**	**0.165**	0.000	0.000	0.000
SVM lin	0.889	0.897	0.717	0.475	0.465	0.125	0.150	**0.165**	0.000	0.000	0.000
Rank	1	3	2	5	4	6	8	7	10	10	10
					DS-CO						
KNN	0.971	**0.969**	0.970	-0.001	0.000	**0.721**	0.711	0.000	0.000	0.000	0.000
NB	-0.157	0.507	-0.151	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.976**	0.958	**0.973**	-0.009	0.000	**0.721**	**0.721**	0.000	0.000	0.000	0.000
RF	**0.976**	0.968	**0.973**	-0.008	0.000	**0.721**	**0.721**	0.000	0.000	0.000	0.000
SVM lin	0.938	0.917	0.970	0.000	0.000	**0.721**	**0.721**	0.000	0.000	0.000	0.000
Rank	1	3	2	10	10	4	5	10	10	10	10
					DS-ST						
KNN	0.014	0.048	0.023	0.019	0.059	0.069	0.083	0.000	0.000	0.000	0.000
NB	-0.010	0.070	-0.020	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.007	0.037	0.004	0.020	0.000	0.000	0.075	0.000	0.000	0.000	0.000
RF	0.007	0.039	0.005	0.023	0.000	0.000	0.075	0.000	0.000	0.000	0.000
SVM lin	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rank	-	-	-	-	-	-	-	-	-	-	-
					DS-GR						
KNN	0.971	**0.969**	0.970	-0.001	0.000	**0.721**	0.711	0.000	0.000	0.000	0.000
NB	-0.157	0.507	-0.151	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.976**	0.958	**0.973**	-0.009	0.000	**0.721**	**0.721**	0.000	0.000	0.000	0.000
RF	**0.976**	0.968	0.972	-0.008	0.000	**0.721**	**0.721**	0.000	0.000	0.000	0.000
SVM lin	0.938	0.917	0.970	0.000	0.000	**0.721**	**0.721**	0.000	0.000	0.000	0.000
Rank	1	3	2	10	10	4	5	10	10	10	10

Table 56: MCCs Values Achieved When Testing the Features Separately for Skeptical Acceptance Under Several Semantics: Using the *kwt-train* and *kwt-test* Datasets

Classifier	Degree C.	Katz C.	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	is SC	is Sym.	is Ir.
DC-PR	2	3	1	4	6	10	5	10	10	10	10
DC-CO	2	3	1	4	6	10	5	10	10	10	10
DC-GR	1	3	2	10	10	4	5	10	10	10	10
DS-PR	1	3	2	5	4	6	8	7	10	10	10
DS-CO	1	3	2	10	10	4	5	10	10	10	10
DS-GR	1	3	2	10	10	4	5	10	10	10	10
AVG Rank	1,33	3,00	1,67	7,17	7,67	6,33	5,50	9,50	10,00	10,00	10,00
Order of Testing	1	3	2	6	7	5	4	8			

Table 57: Rank Values for Classification with the *kwt-train* and *kwt-test* Datasets

D. Overview Test Results for Individual Features and Rank for *pbbg-train* and *pbbg-test*

Classifier	Degree C.	Katz C.	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	is SC	is Sym.	is Ir.
DC-PR											
KNN	0.554	0.347	0.518	**0.507**	-0.006	0.306	0.151	0.000	0.000	0.000	0.000
NB	0.302	0.202	0.308	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.607	0.236	**0.537**	0.499	**0.429**	0.417	0.431	0.000	0.000	0.000	0.000
RF	0.607	**0.351**	**0.537**	0.495	**0.429**	0.417	0.431	0.000	0.000	0.000	0.000
SVM lin	**0.616**	0.246	0.440	0.000	**0.429**	**0.428**	**0.433**	0.000	0.000	0.000	0.000
Rank	1	7	2	3	5	6	4	10	10	10	10
DC-CO											
KNN	0.554	0.347	0.518	**0.507**	-0.006	0.306	0.151	0.000	0.000	0.000	0.000
NB	0.302	0.202	0.308	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.607	0.237	0.536	0.499	**0.429**	0.417	0.431	0.000	0.000	0.000	0.000
RF	0.607	**0.349**	**0.540**	0.495	**0.429**	0.417	0.431	0.000	0.000	0.000	0.000
SVM lin	**0.616**	0.246	0.440	0.000	**0.429**	**0.428**	**0.433**	0.000	0.000	0.000	0.000
Rank	1	7	2	3	5	6	4	10	10	10	10
DC-ST											
KNN	0.537	0.311	0.489	**0.495**	0.015	**0.352**	0.154	0.000	0.000	0.000	0.000
NB	0.274	0.178	0.279	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.586	0.205	0.507	0.486	**0.412**	0.336	0.342	0.000	0.000	0.000	0.000
RF	0.586	**0.319**	**0.511**	0.482	**0.412**	0.336	0.342	0.000	0.000	0.000	0.000
SVM lin	**0.590**	0.085	0.414	0.000	0.000	0.321	**0.441**	0.000	0.000	0.000	0.000
Rank	**1**	**7**	**2**	**3**	**5**	**6**	**4**	10	10	10	10
DC-GR											
KNN	0.665	0.456	0.692	0.625	0.000	0.206	0.189	0.000	0.000	0.000	0.000
NB	0.380	0.211	0.427	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.702**	0.339	0.696	**0.636**	0.000	0.000	**0.215**	0.000	0.000	0.000	0.000
RF	**0.702**	**0.463**	**0.708**	0.635	0.000	0.000	**0.215**	0.000	0.000	0.000	0.000
SVM lin	**0.702**	0.000	0.552	0.000	0.000	**0.254**	0.000	0.000	0.000	0.000	0.000
Rank	2	4	1	3	10	5	6	10	10	10	10

Table 58: MCCs Values Achieved when Testing the Features Separately for Credulous Acceptance under Several Semantics Using the *pbbg-train* and *pbbg-test* Datasets

Classifier	Degree C.	Katz C.	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	is SC	is Sym.	is Ir.
DS-PR											
KNN	0.609	0.411	0.606	**0.545**	-0.024	**0.137**	0.155	0.000	0.000	0.000	0.000
NB	0.365	0.215	0.365	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.630	0.284	0.616	0.543	**0.470**	0.136	**0.384**	0.000	0.000	0.000	0.000
RF	0.630	**0.423**	**0.623**	0.541	**0.470**	0.136	**0.384**	0.000	0.000	0.000	0.000
SVM lin	0.619	0.124	0.455	0.000	0.000	0.136	0.000	0.000	0.000	0.000	0.000
Rank	1	5	2	3	4	7	6	10	10	10	10
DS-CO											
KNN	0.665	0.478	0.691	0.625	0.000	0.206	0.189	0.000	0.000	0.000	0.000
NB	0.380	0.209	0.426	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.702**	0.361	0.697	**0.635**	**0.549**	0.166	**0.457**	0.000	0.000	0.000	0.000
RF	**0.702**	**0.488**	0.708	0.634	**0.549**	0.166	**0.457**	0.000	0.000	0.000	0.000
SVM lin	0.679	0.000	0.531	0.000	0.000	**0.376**	0.000	0.000	0.000	0.000	0.000
Rank	2	5	1	3	4	7	6	10	10	10	10
DS-ST											
KNN	0.251	0.166	0.379	0.214	0.202	-0.171	0.095	0.000	0.000	0.000	-0.012
NB	0.193	0.160	0.187	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.344**	0.116	0.430	0.252	**0.210**	**0.165**	**0.204**	0.000	0.000	0.000	0.000
RF	**0.344**	**0.169**	0.424	**0.267**	**0.210**	**0.165**	**0.204**	0.000	0.000	0.000	0.000
SVM lin	0.144	0.137	0.218	0.098	0.000	0.014	0.000	0.000	0.000	0.000	0.000
Rank	2	6	1	3	4	7	5	10	10	10	10
DS-GR											
KNN	0.665	0.456	0.692	0.625	0.000	0.206	0.189	0.000	0.000	0.000	0.000
NB	0.380	0.211	0.427	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.702**	0.343	0.695	**0.636**	0.000	0.000	**0.215**	0.000	0.000	0.000	0.000
RF	**0.702**	**0.465**	0.707	0.635	0.000	0.000	**0.215**	0.000	0.000	0.000	0.000
SVM lin	**0.702**	0.000	0.552	0.000	0.000	**0.254**	0.000	0.000	0.000	0.000	0.000
Rank	2	4	1	3	10	5	6	10	10	10	10

Table 59: MCCs Values Achieved when Testing the Features Separately for Skeptical Acceptance under Several Semantics Using the *pbbg-train* and *pbbg-test* Datasets

Classifier	Degree C.	Katz C.	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	is SC	is Sym.	is Ir.
DC-PR	1	7	2	3	5	6	4	10	10	10	10
DC-CO	1	7	2	3	5	6	4	10	10	10	10
DC-ST	1	7	2	3	5	6	4	10	10	10	10
DC-GR	2	4	1	3	10	5	6	10	10	10	10
DS-PR	1	5	2	3	4	7	6	10	10	10	10
DS-CO	2	5	1	3	4	7	6	10	10	10	10
DS-ST	2	6	1	3	4	7	5	10	10	10	10
DS-GR	2	4	1	3	10	5	6	10	10	10	10
AVG Rank	1,50	5,63	1,50	3,00	5,88	6,13	5,13	10,00	10,00	10,00	10,00
Order of Testing	1	5	2	3	6	7	4	-	-	-	-

Table 60: Rank Values for Classification with the *pbbg-train* and *pbbg-test* Datasets

E. Overview Test Results for Individual Features and Rank for *pbbg-train* and *ICCMA-450-test*

Classifier	Degree C.	Katz C.	Closeness C.	Betweenness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	is SC	is Sym.	is Ir.
					DC-PR						
KNN	0.173	0.071	0.212	0.104	0.118	-0.127	0.393	0.000	0.000	0.000	0.000
NB	-0.008	-0.006	0.190	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.397	0.037	0.345	**0.108**	**0.484**	0.307	**0.396**	0.000	0.000	0.000	0.000
RF	0.397	**0.078**	0.374	0.103	**0.484**	0.307	0.386	0.000	0.000	0.000	0.000
SVM lin	**0.416**	0.057	**0.476**	0.000	**0.484**	**0.312**	0.296	0.000	0.000	0.000	0.000
Rank	**3**	**7**	**2**	**6**	**1**	**5**	**4**	**10**	**10**	**10**	**10**
					DC-CO						
KNN	0.173	**0.071**	0.212	0.104	0.118	-0.127	0.393	0.000	0.000	0.000	0.000
NB	-0.008	-0.006	0.190	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.397	0.056	0.345	**0.108**	**0.484**	0.307	**0.396**	0.000	0.000	0.000	0.000
RF	0.397	0.070	0.374	0.103	**0.484**	0.307	**0.396**	0.000	0.000	0.000	0.000
SVM lin	**0.416**	0.057	**0.476**	0.000	**0.484**	**0.312**	0.296	0.000	0.000	0.000	0.000
Rank	**3**	**7**	**2**	**6**	**1**	**5**	**4**	**10**	**10**	**10**	**10**
					DC-ST						
KNN	0.018	-0.006	0.164	0.070	-0.008	-0.065	**0.225**	0.000	0.000	0.000	0.000
NB	-0.092	-0.018	0.102	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.351	0.044	0.205	**0.081**	**0.445**	-0.001	0.064	0.000	0.000	0.000	0.000
RF	0.351	**0.058**	0.233	0.078	**0.445**	-0.001	0.064	0.000	0.000	0.000	0.000
SVM lin	**0.354**	0.031	**0.357**	0.000	0.000	**0.222**	0.065	0.000	0.000	0.000	0.000
Rank	**3**	**7**	**2**	**6**	**1**	**5**	**4**	**10**	**10**	**10**	**10**
					DC-GR						
KNN	0.196	-0.007	0.226	**0.384**	0.000	-0.072	**0.437**	0.000	0.000	0.000	0.000
NB	0.054	0.017	0.081	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.431	**0.053**	0.340	0.161	0.000	0.000	0.007	0.000	0.000	0.000	0.000
RF	**0.432**	0.001	**0.373**	0.158	0.000	0.000	0.007	0.000	0.000	0.000	0.000
SVM lin	0.422	0.000	0.363	0.000	0.000	**0.183**	0.041	0.000	0.000	0.000	0.000
Rank	2	6	4	3	10	5	1	10	10	10	10

Table 61: MCCs Values Achieved when Testing the Features Separately for Credulous Acceptance under Several Semantics Using the *pbbg-train* and *ICCMA-450* Datasets.

Classifier	Degree C.	Katz C.	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	is SC	is Sym.	is Ir.
DS-PR											
KNN	0.194	0.001	0.195	**0.091**	-0.002	-0.072	**0.377**	0.000	0.000	0.000	0.000
NB	0.064	0.029	0.073	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.399	**0.051**	0.307	**0.091**	**0.390**	**0.190**	0.075	0.000	0.000	0.000	0.000
RF	0.399	0.010	**0.341**	0.087	**0.390**	**0.190**	0.075	0.000	0.000	0.000	0.000
SVM lin	0.207	0.004	0.334	0.000	0.000	0.187	0.036	0.000	0.000	0.000	0.000
Rank	**1**	**7**	**4**	**6**	**2**	**5**	**3**	**10**	**10**	**10**	**10**
DS-CO											
KNN	0.190	0.002	0.218	**0.163**	0.000	-0.074	**0.423**	0.000	0.000	0.000	0.000
NB	0.067	**0.029**	0.093	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.439	-0.050	0.335	0.162	**0.438**	**0.189**	0.083	0.000	0.000	0.000	0.000
RF	0.439	0.014	**0.367**	0.159	**0.438**	**0.189**	0.083	0.000	0.000	0.000	0.000
SVM lin	0.171	0.000	0.349	0.000	0.000	0.182	0.040	0.000	0.000	0.000	0.000
Rank	**1**	**7**	**4**	**6**	**2**	**5**	**3**	**10**	**10**	**10**	**10**
DS-ST											
KNN	-0.024	-0.102	0.002	-0.002	**0.099**	-0.316	-0.096	0.000	0.000	0.000	0.343
NB	-0.061	-0.052	0.097	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.027	-0.081	0.055	0.010	-0.060	-0.084	**0.055**	0.000	0.000	0.000	0.000
RF	**0.119**	-0.063	0.074	0.015	-0.060	-0.084	**0.055**	0.000	0.000	0.000	0.000
SVM lin	0.031	**0.037**	**0.121**	**0.024**	0.000	0.339	-0.008	0.000	0.000	0.000	0.000
Rank	**4**	**7**	**3**	**8**	**5**	**2**	**6**	**10**	**10**	**10**	**1**
DS-GR											
KNN	0.196	-0.007	0.226	**0.384**	0.000	-0.072	0.437	0.000	0.000	0.000	0.000
NB	0.054	**0.017**	0.081	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.431	-0.059	0.339	0.161	0.000	0.000	0.007	0.000	0.000	0.000	0.000
RF	**0.432**	0.004	**0.374**	0.159	0.000	0.000	0.007	0.000	0.000	0.000	0.000
SVM lin	0.422	0.000	0.363	0.000	0.000	**0.183**	0.041	0.000	0.000	0.000	0.000
Rank	**2**	**5**	**4**	**3**	**10**	**6**	**1**	**10**	**10**	**10**	**10**

Table 62: MCCs Values Achieved when Testing the Features Separately for Skeptical Acceptance under Several Semantics Using the *pbbg-train* and *ICCMA-450* Datasets.

Classifier	Degree C.	Katz C.	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	is SC	is Sym.	is Ir.
DC-PR	3	7	2	6	1	5	4	10	10	10	10
DC-CO	3	7	2	6	1	5	4	10	10	10	10
DC-ST	3	7	2	6	1	5	4	10	10	10	10
DC-GR	2	6	4	3	10	5	1	10	10	10	10
DS-PR	1	7	4	6	2	5	3	10	10	10	10
DS-CO	1	7	4	6	2	5	3	10	10	10	10
DS-ST	4	7	3	8	5	2	6	10	10	10	1
DS-GR	2	5	4	3	10	6	1	10	10	10	10
AVG Rank	2,38	6,63	3,13	5,50	4,00	4,75	3,25	10,00	10,00	10,00	8,88
Order of Testing	1	7	2	6	4	5	3	-	-	-	8

Table 63: Rank Values for Classification with the *pbbg-train* and *ICCMA-450* Datasets

F. Overview Test Results for Individual Features and Rank for *ICCMA-450-train* and *ICCMA-450-test*

Classifier	**Degree C.**	**Katz C.**	**Close-ness C.**	**Between-ness C.**	**Size Sccs**	**SCCs**	**AVG Deg.**	**is Aperiod**	**is SC**	**is Sym.**	**is Ir.**
DC-PR											
KNN	**0.519**	**0.159**	0.500	**0.365**	0.546	0.113	0.076	**0.259**	**0.125**	0.000	**0.214**
NB	-0.025	0.135	0.225	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.474	0.147	0.429	0.022	**0.580**	0.303	**0.408**	0.000	0.000	0.000	0.000
RF	0.438	0.171	0.542	0.025	**0.580**	**0.618**	**0.408**	0.000	0.000	0.000	0.000
SVM lin	0.010	0.000	**0.596**	-0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rank	4	9	2	6	3	1	5	7	10	11	8
DC-CO											
KNN	**0.519**	**0.159**	0.500	**0.365**	0.546	0.113	0.076	**0.259**	**0.125**	0.000	**0.214**
NB	-0.025	0.135	0.225	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.474	0.132	0.451	0.022	**0.580**	0.303	0.408	0.000	0.000	0.000	0.000
RF	0.436	0.173	0.535	0.023	**0.580**	**0.618**	**0.424**	0.000	0.000	0.000	0.000
SVM lin	0.010	0.000	**0.596**	-0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rank	4	9	2	6	3	1	5	7	10	11	8
DC-ST											
KNN	**0.651**	0.115	0.473	**0.270**	**0.555**	0.137	0.045	**0.253**	**0.120**	0.000	**0.220**
NB	-0.041	**0.132**	0.213	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.626	0.103	0.404	0.019	0.000	0.219	**0.619**	0.000	0.000	0.000	0.000
RF	0.645	0.118	**0.523**	0.020	0.000	**0.574**	**0.619**	0.000	0.000	0.000	0.000
SVM lin	0.000	0.022	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rank	1	9	5	6	4	3	2	7	10	11	8
DC-GR											
KNN	0.258	**0.168**	0.399	**0.543**	**0.263**	-0.085	0.301	**0.117**	**0.056**	0.000	**0.025**
NB	0.143	0.099	0.142	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.471	0.134	0.377	0.034	0.000	**0.283**	0.309	0.000	0.000	0.000	0.000
RF	**0.472**	0.135	**0.530**	0.034	0.000	0.055	**0.375**	0.000	0.000	0.000	0.000
SVM lin	0.000	0.000	0.000	-0.004	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rank	3	7	2	1	6	5	4	8	9	11	10

Table 64: MCCs Values Achieved when Testing the Features Separately for Credulous Acceptance under Several Semantics Using the *ICCMA-450-train* and *ICCMA-450-test* Datasets.

Classifier	Degree C.	Katz C.	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	is SC	is Sym.	is Ir.
DS-PR											
KNN	0.220	**0.143**	0.353	**0.473**	**0.223**	-0.069	0.226	**0.140**	**0.068**	0.000	**0.065**
NB	0.179	0.111	0.123	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.451	0.118	0.295	0.026	0.000	**0.230**	0.249	0.000	0.000	0.000	0.000
RF	**0.454**	0.135	**0.433**	0.027	0.000	0.020	**0.282**	0.000	0.000	0.000	0.000
SVM lin	0.000	0.000	0.000	-0.004	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rank	2	7	3	1	6	5	4	8	9	11	10
DS-CO											
KNN	0.244	**0.154**	0.413	**0.528**	**0.260**	-0.049	0.277	**0.127**	**0.060**	0.000	**0.041**
NB	0.182	0.120	0.166	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	**0.477**	0.131	0.344	0.029	0.000	**0.260**	0.284	0.000	0.000	0.000	0.000
RF	0.475	0.138	**0.529**	0.028	0.000	0.039	**0.322**	0.000	0.000	0.000	0.000
SVM lin	0.000	0.000	0.000	-0.004	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rank	3	7	2	1	5	6	4	8	9	11	10
DS-ST											
KNN	0.264	0.013	0.434	**0.052**	0.535	-0.375	**0.184**	0.000	0.000	0.000	0.000
NB	-0.030	**0.016**	0.029	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.434	-0.040	0.379	0.021	**0.688**	**0.728**	-0.209	**0.755**	0.000	0.000	**0.496**
RF	**0.436**	-0.035	**0.440**	0.027	**0.688**	**0.728**	-0.209	**0.755**	0.000	0.000	**0.496**
SVM lin	0.086	0.000	0.007	0.015	0.541	0.000	0.000	**0.755**	0.000	0.000	**0.496**
Rank	6	9	5	8	3	2	7	1	10	10	4
DS-GR											
KNN	0.258	**0.168**	0.399	**0.543**	**0.263**	-0.085	0.301	**0.117**	**0.056**	0.000	**0.025**
NB	0.143	0.099	0.142	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DT	0.471	0.138	0.377	0.034	0.000	**0.283**	**0.309**	0.000	0.000	0.000	0.000
RF	**0.472**	0.144	**0.532**	0.032	0.000	0.055	**0.309**	0.000	0.000	0.000	0.000
SVM lin	0.000	0.000	0.000	-0.004	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rank	3	7	2	1	6	5	4	8	9	11	10

Table 65: MCCs Values Achieved when Testing the Features Separately for Skeptical Acceptance under Several Semantics Using the *ICCMA-450-train* and *ICCMA-450-test* Datasets.

Classifier	Degree C.	Katz C.	Close-ness C.	Between-ness C.	Size Sccs	SCCs	AVG Deg.	is Aperiod	is SC	is Sym.	is Ir.
DC-PR	4	9	2	6	3	1	5	7	10	11	8
DC-CO	4	9	2	6	3	1	5	7	10	11	8
DC-ST	1	9	5	6	4	3	2	7	10	11	8
DC-GR	3	7	2	1	6	5	4	8	9	11	10
DS-PR	2	7	3	1	6	5	4	8	9	11	10
DS-CO	3	7	2	1	5	6	4	8	9	11	10
DS-ST	6	9	5	8	3	2	7	1	10	10	4
DS-GR	3	7	2	1	6	5	4	8	9	11	10
AVG Rank	3,25	8,00	2,88	3,75	4,50	3,50	4,38	6,75	9,50	10,88	8,50
Order of Testing	2	8	1	4	6	3	5	7	10		9

Table 66: Rank Values for Classification with the *ICCMA-450-train* and *ICCMA-450-test* Datasets