big data and cognitive computing

MDPI

Article

DSpamOnto: An Ontology Modelling for Domain-Specific Social Spammers in Microblogging

Malak Al-Hassan, Bilal Abu-Salih * and Ahmad Al Hwaitat

King Abdullah II School of Information Technology, The University of Jordan, Amman 11942, Jordan; m_alhassan@ju.edu.jo (M.A.-H.)
* Correspondence: b.abusalih@ju.edu.jo

Abstract: The lack of regulations and oversight on Online Social Networks (OSNs) has resulted in the rise of social spam, which is the dissemination of unsolicited and low-quality content that aims to deceive and manipulate users. Social spam can cause a range of negative consequences for individuals and businesses, such as the spread of malware, phishing scams, and reputational damage. While machine learning techniques can be used to detect social spammers by analysing patterns in data, they have limitations such as the potential for false positives and false negatives. In contrast, ontologies allow for the explicit modelling and representation of domain knowledge, which can be used to create a set of rules for identifying social spammers. However, the literature exposes a deficiency of ontologies that conceptualize domain-based social spam. This paper aims to address this gap by designing a domain-specific ontology called DSpamOnto to detect social spammers in microblogging that targes a specific domain. DSpamOnto can identify social spammers based on their domain-specific behaviour, such as posting repetitive or irrelevant content and using misleading information. The proposed model is compared and benchmarked against well-proven ML models using various evaluation metrics to verify and validate its utility in capturing social spammers.

Keywords: social spammers; ontology; DSRM; microblogging; machine learning; OSNs

check for updates

Citation: Al-Hassan, M.; Abu-Salih, B.; Al Hwaitat, A. DSpamOnto: An Ontology Modelling for Domain-Specific Social Spammers in Microblogging. *Big Data Cogn. Comput.* **2023**, *7*, 109. https://doi.org/10.3390/bdcc7020109

Academic Editors: Zuchao Li and Min Peng

Received: 10 May 2023
Revised: 25 May 2023
Accepted: 31 May 2023
Published: 2 June 2023

1. Introduction

The advent of Online Social Networks (OSNs) has reformed the way we communicate and share information. However, the lack of regulations and oversight on these platforms has opened the door for malicious actors to engage in various fraudulent and harmful activities [1]. This has led to the rise of social spam, which refers to the dissemination of unsolicited and low-quality content, such as fake profiles, phishing scams, and clickbait links. Social spammers aim to deceive and manipulate users, tarnishing the reputation of individuals and organizations and degrading the overall quality of experience on these platforms. Despite ongoing efforts to combat social spam, it continues to be a pervasive problem, with almost half of social media users reporting an increase in spam content in their feeds [2]. Social spam can have a number of negative consequences for both individuals and businesses. For individuals, social spam can be annoying and time-consuming to deal with. It can also lead to the spread of malware or phishing scams, which can damage computers and steal personal information. For businesses, social spam can damage reputations and lead to lost customers [3].

Social spam can be a nuisance for users, and it can also be used to spread malware, spread misinformation, and damage businesses. Social media platforms are working to combat social spam, but it is a constantly evolving problem. Machine learning techniques can be used to detect social spam by analysing patterns in data and identifying characteristics that distinguish spam from legitimate content. Various machine learning algorithms, such as decision trees, random forests, support vector machines, and deep learning models, can be trained on labelled datasets of spam and non-spam content to learn to recognize patterns in the data and make accurate predictions [4–7]. For example, some features that can be used for social spam detection include the content of the message, the user's profile information, the frequency of posting, and the behaviour of the user's social network. By analysing these features, machine learning models can identify spam messages and distinguish them from legitimate content. However, one of the inadequacies of machine learning techniques for social spam detection is the potential for false positives and false negatives. False positives occur when legitimate users are incorrectly labelled as spammers, while false negatives occur when social spammers are incorrectly labelled as legitimate users. Another weakness is the need for large amounts of labelled data to train accurate models, which can be difficult to obtain in some cases. Additionally, spammers can use techniques to evade machine learning detection, such as using natural language variations, changing account information frequently, and employing sophisticated tactics to make their spam messages appear more legitimate [8–10]. Therefore, it is important to regularly update machine learning models and to use them in conjunction with other techniques, such as human review and rule-based systems, to improve the accuracy of social spam detection.

On the other hand, incorporating ontologies for detecting social spammers has some advantages over machine learning techniques. Ontologies allow for the explicit modelling and representation of domain knowledge, which can be used to create a set of rules for identifying social spammers. These rules can be based on the characteristics of known social spammers and can be refined as more data becomes available. This can help ontologies to identify spam that is not explicitly labelled as such. For example, consider the following spam comment: "*This is the best product I've ever used! I highly recommend it to anyone*". A machine learning algorithm would need to be trained on a dataset of known spam and non-spam comments, using features such as the language used, sentiment, and context. The model would then predict whether the comment is spam or non-spam based on these features. In this case, the comment would likely be classified as non-spam, as it contains positive sentiment and does not appear to be promoting a specific product or service. In contrast, an ontology-based approach would involve defining a set of rules or concepts related to spam comments and then applying these rules to the comment in question. For example, the ontology might include rules about the use of superlatives or generic recommendations, which could flag this comment as spam.

However, the literature exposes a deficiency of ontologies that conceptualise domain-based social spam. The current social spam ontologies are only able to conceptualise social spammers based on general characteristics, rather than domain-specific ones. In other words, the current social spam ontologies are designed to identify social spammers based on generic characteristics such as the frequency of posting, the number of friends/followers, the rate of interaction, etc. These ontologies are not able to identify spammers that use more advanced tactics to target specific domains or industries. These techniques require a more sophisticated analysis of the content and context of the messages so as to extract domain-specific social features of social users and extract those who exhibit malicious activities that target specific domains. In this paper, we aim to conceptualise the domain-specific social spam in microblogging by means of designing a domain ontology, namely DSpamOnto. DSpamOnto is a domain-specific ontology designed to detect social spam in microblogging that targets a specific domain. It is developed to provide a better understanding of the behaviour of social spammers who focus on a particular domain, such as politics or health. By using DSpamOnto, it is possible to identify social spammers based on their domain-specific behaviour, such as posting repetitive or irrelevant content, using misleading information or tactics, and targeting specific domain(s). The study follows the Design Science Research Methodology (DSRM) and integrates two ontology development techniques, METHONTOLOGY [11] and Cyc 101 [12], to design, implement, and evaluate the ontology. The process involves determining the domain and scope of the ontology, reusing existing ontologies, developing a conceptual model, and evaluating the ontology. While the domain-specific social spamming is still evolving, this ontology serves as a starting point to facilitate future efforts in creating more advanced versions of the ontology.

Section 2 covers the current attempts to identify social spammers. In Section 3, the research methodology used in this study is described. Sections 4 and 5 discuss the development, execution, and assessment of DSpamOnto. Section 6 discusses the experimental results. Finally, Section 7 concludes the paper and highlights key areas for future research.

2. Related Works

Social spam refers to unwanted and unsolicited messages, comments, or posts on social media platforms that aim to deceive, mislead, or defraud users. Social spam can take various forms, such as fake profiles, misleading links, spam comments, and fraudulent promotions [1]. Social spam is a growing problem on social media platforms and can have serious consequences for users, such as identity theft, malware infections, financial losses, and reputation damage [13]. The rise of social spam can be attributed to the increasing popularity of social media and the ease with which spammers can create fake accounts and automate spamming activities. According to a study by Barracuda Networks, a leading provider of cloud-enabled security solutions, social media platforms are the most common targets for spam attacks. The study found that Twitter is the most targeted social media platform, accounting for 90% of all social spam attacks, followed by Facebook and Instagram [14].

The authors of [15] proposed a bagging-based approach for detecting spam emails. The approach uses two sources of information: an email's content and its metadata. The content of the email is extracted using a natural language processing (NLP) tool, and the metadata of the email is extracted using a spam filter. The extracted features are then used to train a classifier to distinguish between spam and non-spam emails. The classifier is evaluated on a dataset of real spam emails and is shown to outperform state-of-the-art methods. Abu-Salih et al. [1] developed an intelligent system for detecting social spam in microblogging. The system uses a variety of features, including the content of the tweet, the user who posted the tweet, and the network of users who have retweeted the tweet. The system is evaluated on a dataset of real social spam tweets and is shown to outperform state-of-the-art methods. The authors of [16] incorporated a domain ontology to extract features from reviews, such as the use of positive and negative words, the use of superlatives, and the use of first-person pronouns. The features are then used to train a classifier to distinguish between fake and real reviews. The classifier is evaluated on a dataset of real and fake reviews and is shown to outperform state-of-the-art methods. Jabardi et al. [17] developed an ontology-based approach for detecting fake Twitter accounts. The approach uses a domain ontology to extract features from Twitter profiles, such as the number of followers, the number of followed accounts, the number of tweets, and the use of hashtags. The features are then used to train a classifier to distinguish between fake and real Twitter accounts. The classifier is evaluated on a dataset of real and fake Twitter accounts and is shown to outperform state-of-the-art methods. In [18], the authors developed an ontology-based approach that involves creating an ontology that captures the characteristics of fake accounts and using this ontology to analyse Twitter data. Hussain et al. [19] proposed an ontology-based approach for filtering spam URLs. The approach uses a domain ontology to extract features from URLs, such as the domain name, the top-level domain, and the URL path. The features are then used to train a classifier to distinguish between spam and non-spam URLs. The classifier is evaluated on a dataset of real spam URLs and is shown to outperform state-of-the-art methods.

In our previous work [20] we proposed an ontology-based approach that involves constructing a domain ontology, applying machine learning techniques for domain term extraction and clustering, and using the ontology to identify domain-specific topics from Twitter data. In addition, we developed an ontology-based approach that involves constructing a credibility ontology, applying a credibility score algorithm to the ontology, and using the ontology to identify credible information sources in social Big Data [21]. This study is different from our previous works and other seminal works because it aims to create a proof-of-concept domain-specific social spam ontology that incorporates the key entities and ideas that characterise social spammers and their domain-specific activity. This study recognises the key concepts and connections related to domain-specific social spam, such as the kinds of spam messages and tactics for targeting domains. Table 1 shows a review of some recent ontology-based social spam detection approaches.

Table 1. A review of some recent ontology-based social spam detection approaches.

Ref	Year	Social Platform	Method	Evaluation Metric(s)	Limitation(s)
[15]	2022	Email	Ontology, and bagging-based approach	accuracy, precision, recall, and F1 score.	-The paper is only evaluated on a dataset of real spam emails. This means that the approach may not be able to detect spam emails that are written in a different style than the emails in the dataset. -The paper does not address the problem of false positives.
[1]	2022	Twitter	NLP, ML models	accuracy, precision, recall, and F1 score.	-As social spam becomes more sophisticated, the system may need to be updated to maintain its accuracy.
[16]	2022	online reviews for e-commerce platforms.	Linguistic, POS tagging, and SWRL rules	accuracy, precision, recall, and F1 score.	-The classification results may be sensitive to the choice of ontology and SWRL rules. -The SWRL rules may not be able to capture all of the relevant relationships between the concepts in the ontology.
[17]	2020	Twitter	SWRL	precision, recall, and F1-score,	-It only focuses on detecting fake accounts on Twitter and may not be applicable to other social media platforms. -The ontology used in the study may not capture all possible characteristics of fake accounts, and the study may not have captured all instances of fake accounts on Twitter.
[18]	2018	Twitter	Ontology and clustering techniques	precision, recall, and F1-score	The approach is not robust to changes in the fake Twitter account landscape.
[19]	2018	Spam URL	Ontology and ML classifiers	accuracy, precision, recall, and F1 score.	-The study may not have captured all instances of spam URLs in different languages. -The study also does not provide information on the scale of the dataset used in the evaluation.
[20]	2018	Twitter	Ontology and ML classifiers	Precision, recall, and F1-score	-The present ML methods presume that uncertainty and incompleteness do not affect Twitter classification accuracy. These aspects might exist.
[20]	2018	Twitter	Ontology and SWRL	Case study, Precision, recall, and F1-score	-The developed ontology is limited in terms of the number of classes and relationships.
[22]	2019	Social Media	Ontology and LSTM	Accuracy, precision, recall, F1-score	-The study did not compare the proposed approach with other state-of-the-art methods for spam detection.

3. Research Methodology

This study utilizes the Design Science Research Methodology (DSRM) approach, which was developed as a standard research paradigm in the Information Systems (IS) field to provide researchers with a framework for creating constructs, models, methods, and instantiations [23]. Design Theory was established by Gregor et al. [24] and Venable [25] in the same context, with the former presenting the necessary components for communicating a design theory, while the latter simplified the formulation and addressed some of the contentious issues in Design Theory. Peffers et al. [26] argue that DSRM is an effective approach to designing and evaluating artifacts that solve real-world problems, providing a systematic framework for conducting research that is both rigorous and relevant to practitioners.

As depicted in Figure 1, this methodology is iterative, meaning that feedback from each activity is used to inform and refine subsequent activities. The ultimate goal is to create a design theory that can be generalized and applied to other similar problems and to create an artifact that solves the identified problem practically and effectively. The proposed methodology consists of the following seven main activities:

1. **Problem identification and motivation**: This involves identifying and defining the problem that the research aims to solve. This activity is critical because it sets the stage for the entire research process, including the subsequent activities involved in designing and developing a solution to the problem. Relevant activities include:
   - *An analysis of the literature is conducted to find ontologies that are relevant to domain-specific social spam.*
   - *The literature exposes a deficiency of ontologies that conceptualise domain-based social spam. The current social spam ontologies are only able to conceptualise generic-based social spammers.*
2. **Define the objectives for a solution:** This involves formulating a clear and concise set of objectives for the proposed solution. The objectives are derived from the problem identification and motivation activity and aim to guide the design and development of the artifact. Relevant activities include:
   - *This study aims to develop a proof-of-concept domain-specific social spam ontology that integrates the key entities and concepts that capture social spammers and their domain-specific behaviour.*
   - *Identify the key concepts and relationships relevant to domain-specific social spam, such as the types of spam messages and the strategies used by spammers to target specific domains.*
3. **Design and Development:** This entails creating an artifact or solution to address the problem identified in the earlier activities. This activity requires the researcher to use existing theories, frameworks, and best practices to design and develop the artifact that meets the objectives set in the previous activity. Relevant activities include:
   - *Design and develop the ontology using established ontology engineering methodologies, such as the NeOn methodology or the Methodology framework, and by following established best practices and design patterns.*
   - *This phase will create an artifact in the form of an ontology (DSpamOnto).*
   - *The ontology's elements, such as its concepts (classes), attributes (properties), restrictions (facets), and instances (individuals), will be identified by examining academic and technical sources.*
   - *The software Protégé is utilized to create the ontological depiction of the elements of the ontology.*
4. **Demonstration**: This includes showcasing and evaluating the developed artifact to stakeholders and experts to demonstrate its effectiveness and utility in solving the identified problem. Relevant activities include:
   - *To demonstrate the proposed methodology, a prototype ontology is developed as a proof-of-concept. The ontology's ability to provide a better comprehension of the social spammer's behaviours.*
   - *To demonstrate the effectiveness of the ontology by applying it to real-world use cases and scenarios, such as analysing social media data for domain-specific spam or evaluating the effectiveness of different spam detection algorithms.*

5. **Evaluation**: The goal of this activity is to determine whether the developed artifact meets the objectives set in the earlier activities and provides a satisfactory solution to the identified problem. The evaluation process can take different forms, such as a user study, a controlled experiment, or a case study. Relevant activities include:
    - *Evaluate the effectiveness of the ontology and its impact on the domain of interest by conducting empirical studies, surveys, or other types of evaluation and feedback mechanisms.*
    - *Various evaluation metrics are integrated to evaluate ontology.*
6. **Communication**: It involves disseminating the research findings, knowledge, and insights gained from the design and development of the artifact to the wider community of stakeholders, including academics, practitioners, and researchers. Relevant activities include:
    - *This manuscript discusses information regarding the need for, the design approach of, and usefulness of the developed artifact, facilitating the exchange of information.*

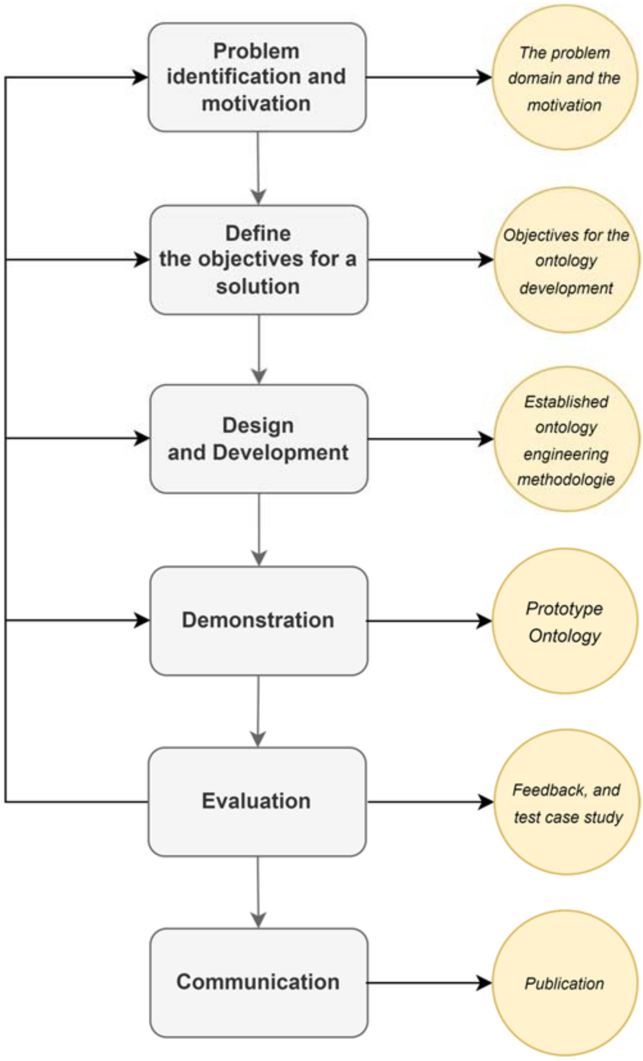
Problem identification and motivation
The problem domain and the motivation
Define the objectives for a solution
Objectives for the ontology development
Design and Development
Established ontology engineering methodologie
Demonstration
Prototype Ontology
Evaluation
Feedback, and test case study
Communication
Publication

Figure 1. DSRM methodology (Prepared by the authors based on the discussion provided in [26].

4. DSpamOnto: An Ontology Design for Social Spam

There are several methodologies that can be adopted to design and construct domain ontologies including (i) top-down methodology—which involves starting with a high-level conceptual framework, such as a philosophical theory or a domain-specific taxonomy, and then refining it through iterative feedback from domain experts and by adding more detailed concepts and relationships; (ii) bottom-up methodology—involves starting with a large collection of individual concepts and facts, such as those extracted from natural language texts or existing databases, and then clustering them into more abstract categories based on their similarities and differences; and (iii) mixed methodology—incorporates a mixture of top-down and bottom-up methodologies in the ontology design. This study integrates a top-down methodology, namely METHONTOLOGY [11] with a mixed-based methodology, namely Cyc 101 [12] to construct DSpamOnto. The procedure is divided into four steps: (i) identifying the topic and extent of the ontology; (ii) ontology reuse; (iii) conceptual model creation; and (iv) ontology evaluation. In addition, we use the "Protégé" tool to build the ontology. Protégé allows for the interaction with other reasoning tools and incorporates business principles for inference. Protégé also supports the most current WWW Consortium RDF and OWL 2 Web Ontology Language standards. The following subsections go over the actions that were taken to build DSpamOnto.

4.1. Identifying the Domain and Extent of the Ontology

This stage specifies the domain that the ontology will conceptualise as well as the queries that the designed ontology will address. Table 2 demonstrated our response to the queries used to identify the ontology's domain and scope.

Table 2. Queries used to identify the domain and extent of the ontology.

Query	Response
What is the domain that the ontology will conceptualise?	In this paper, we aim to conceptualise domain-specific social spam in microblogging by means of designing a domain ontology, namely DSpamOnto.
What is the objective and intent of this ontology?	The purpose of DSpamOnto is to furnish a better understanding of a special category of social spammers, which usually targets a certain domain
Who will benefit from this ontology?	Academic scholars and corporate practitioners interested in developing a conceptualised model for social spammers can benefit from this ontology. Based on the advancements in this environment, the suggested ontology can be repurposed and expanded in the future with additional ideas, properties, and examples.
What are the important issues that integrated knowledge in ontology can answer?	○ What are the main concepts that define and conceptualise domain-based social spammers? ○ How the proposed ontology can be incorporated to prevent and/or mitigate the domain-specific social spam? ○ How can the ontology be used to model the behaviour of users at the content and profile levels?

4.2. Ontology Reuse

The literature reports certain attempts to develop ontologies to capture and conceptualise the behaviour of social spammers. For example, Halawi et al. [18] used a set of ontology-based rules to identify spam tweets and users. To distinguish a fake account from a genuine account, ontology engineering and semantic web rule language rules are used in the work proposed by Jabardi et al. [17]. These papers provide detailed descriptions of ontologies that are designed to capture the features and characteristics of social spammers, such as their behavioural patterns, network structures, and content features. Despite the importance of such efforts, there has been no attempt to develop an ontology for domain-specific social spam dedicated to standardising and formalising the specified domain knowledge to the best of our knowledge. Nevertheless, this study benefits from other seminal works to develop a fine-grained ontology that conceptualises domain-specific social spam.

4.3. Development of a Conceptual Model

Designing a conceptual model of a certain ontology comprises the following actions:

1. **Enumerate key terms in the ontology**: Identifying and listing out the critical terms or concepts that are relevant to the domain being modelled. These terms should represent the main concepts, attributes, and relationships within the domain, and they form the basis of the ontology's vocabulary. Enumerating key terms is an important step in ontology engineering, as it provides a foundation for creating a structured and standardized representation of the knowledge within a specific domain.
2. **Define classes and their hierarchy:** Defining classes and class hierarchy involves identifying the key concepts and entities that are relevant to the domain and organizing them into a hierarchical structure. The class hierarchy in the ontology could be organized in a top-down or bottom-up approach, depending on the ontology engineer's preference and the nature of the domain being modelled. For example, a top-down approach might begin with a general class such as "domain-based social spam" and then create subclasses for different types of social spam, such as "phishing," "clickbait," and "fake news". A bottom-up approach might begin with specific classes such as "spam message" and "spam campaign" and then group them under a more general class such as "social spam". The class hierarchy should aim to create a structured and organized representation of the knowledge within the social spam domain, allowing for more effective spam detection and prevention systems to be developed.
3. **Define class properties—slots:** After defining the classes and class hierarchy in an ontology, the next step is to define class properties or slots. These slots represent the attributes of the classes and the relationships between them. Class properties can be categorized into two main types: object properties and datatype properties. Object properties connect two classes or entities in the ontology. Datatype properties describe the characteristics of a class or entity.
4. **Define the facets of slots:** Each class property or slot has facets that define its type and value. The physical type defines the kind of data that can be assigned to the slot. For example, a "message_content" property in a social media ontology could have a physical type of "string" because the content is made up of text. A "user_followers" property could have a physical type of "integer" because the number of followers is a whole number. Defining class properties and their facets is a critical step in ontology engineering because it helps to create a precise and detailed representation of the relationships and attributes within the domain.
5. **Create instances:** Populating the ontology with individual values for each class is the last step in ontology design. These instances represent real-world entities that are relevant to the ontology domain. The instances can be created manually or through automated processes such as data extraction or machine learning algorithms. Once the instances are created, they can be used for various tasks such as data analysis, decision-making, and knowledge discovery.

Our DSpamOnto, which includes concepts, relationships, attributes, and examples, was designed using the steps outlined above. A comprehensive investigation of various academic papers and corporate reports was conducted in order to extract the technical terminology required to build the ontology [1,4,27–30]. Protégé is used to develop DSpamOnto. Protégé is one of the most widely used ontology editors for building and managing ontologies. It is an open-source platform that provides a user-friendly interface for creating, editing, and visualizing ontologies. OWL-DL is a language for describing and modelling ontologies that is part of the Web Ontology Language (OWL) family. OWL-DL is a decidable subset of OWL, which means that automated reasoning can be used to infer knowledge and validate ontologies. This makes it a popular choice for building and sharing ontologies in many different domains. When developing a DSpamOnto, using a logical ontology language such as OWL-DL can help ensure that the ontology is well-defined, consistent, and can be easily shared and reused by others. The followings are the main classes of DSpamOnto and their descriptions:

SocialUser: A social user is a person who uses social media platforms to communicate, share content, and connect with others. Social users can be individuals, organizations, or groups, and they can use social media for various purposes, such as networking, entertainment, information sharing, or marketing.

DomainBasedStatus: This concept indicates if a certain user's feature is domain-dependent or domain-independent. A domain-dependent feature targets a particular domain. For example, social spammers might post contents on social media pages related to health and fitness that are irrelevant or spammy but include links to their own weight loss product. For example, they might post comments such as "I lost 30 pounds in just 2 weeks! Check out this amazing product!" with a link to their own website. A domain-independent feature is a generic feature that does not convey an interest in a certain domain.

DomainOfInterest: A domain of knowledge is a specific area of expertise or subject matter that is defined by a set of related concepts, topics, and practices. Examples of domains of knowledge include sports, politics, science, medicine, education, art, and business, among others. Each domain has its unique characteristics, language, and methods of inquiry that are used to study and understand its subject matter.

DynamismStatus: Dynamic social features of social spammers are those that are constantly changing over time and can be influenced by external factors such as trending topics, events, and user behaviour. Examples of dynamic social features include the frequency and timing of tweets, the use of hashtags and mentions, the types of content shared, and the engagement levels of the user's followers. Static social features of social spammers, on the other hand, are those that remain relatively constant over time and are not as easily influenced by external factors. Examples of static social features include the number of followers, the account creation date, the profile picture and bio, and the account verification status.

SocialFeature: Social spam features can be categorized into different types based on the characteristics of the data they analyse. Some common types of social spam features include Content-based features, Graph-based features, User-based features, Metadata-based features, etc.

SocialUserType: In the context of social media, the "SocialUserType" class in our ontology is designed to classify users based on their legitimacy or illegitimacy. While the ontology captures the knowledge and relationships related to social spam detection, the actual classification of users into "legitimate" or "illegitimate" categories is determined through the integration of our ontology-based spam detection model with external classification mechanisms.

The integration process involves leveraging external techniques such as machine learning algorithms, rule-based systems, or human review processes. These techniques analyse various factors, including user behaviour, content analysis, and other relevant indicators, to determine the legitimacy of a user based on predefined criteria. For example, machine learning algorithms can be trained on labelled datasets of known legitimate and illegitimate users, using features such as posting frequency, interaction patterns, content characteristics, and social network behaviour. The trained models can then predict the legitimacy of new users by evaluating these features against the learned patterns. Similarly, rule-based systems can be constructed using predefined rules that capture the characteristics and behaviours associated with legitimate or illegitimate users.

By integrating these external classification mechanisms with our ontology, we can utilize the knowledge represented in the ontology to enhance the accuracy and effectiveness of the classification process. The ontology provides the necessary conceptual foundation for organizing and representing the domain-specific knowledge related to social spam detection, enabling the integration of these external techniques. It is important to note that the specific techniques and algorithms used for classification may vary depending on the implementation and requirements of the spam detection system. The ontology serves as a guiding framework that helps define the relevant concepts, relationships, and rules for the classification process.

To offer a thorough understanding of the core classes as well as the interconnected objects and data characteristics, Figures 2–4 depict some of the classes in DSpamOnto. For example, Figure 2 shows the SocialFeature class which is a subclass of DomainSocialSpam class—the main class. SocialFeature class is subcategorized into content-based, graph-based, and metadata-based features. Each social feature might be domain-dependent or domain-independent, and if it is domain dependent, then it should be linked to one of the domains indicated in the ontology.

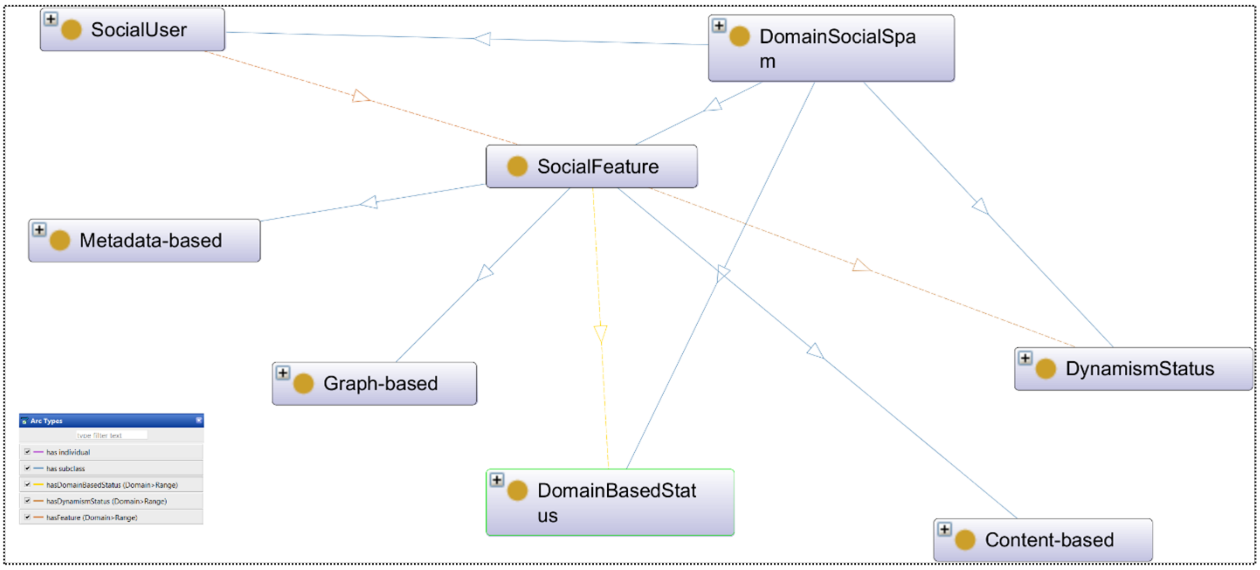

Figure 2. A snapshot of SocialFeature class and its interrelated classes and subclasses in DSpamOnto.

Figure 3 demonstrates the subclasses of the Content-based class. Content-based social spam behaviour involves the use of misleading or irrelevant content in tweets to trick users into clicking on spammy links or engaging with fraudulent content.

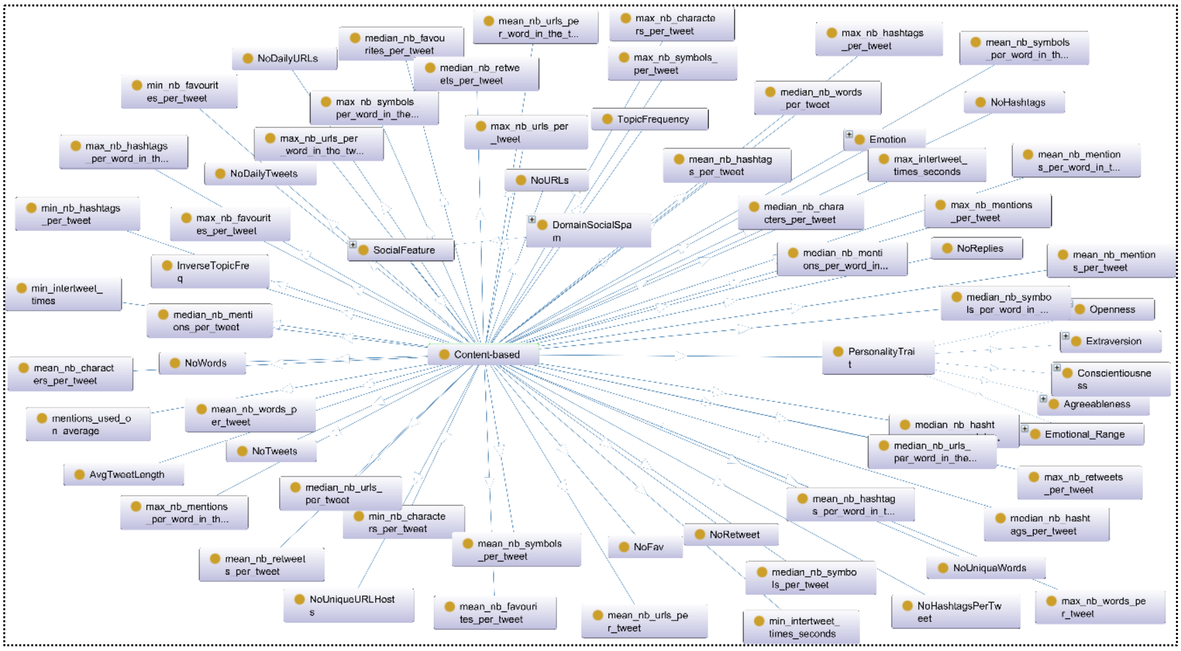
mean_nb_urls_per_word_in_the_t...
max_nb_characters_per_tweet
median_nb_favourites_per_tweet
NoDailyURLs
median_nb_retweets_per_tweet
max_nb_symbols_per_tweet
max_nb_hashtags_per_tweet
mean_nb_symbols_per_word_in_th...
min_nb_favourites_per_tweet
max_nb_symbols_per_word_in_the...
median_nb_words_per_tweet
NoHashtags
TopicFrequency
max_nb_urls_per_tweet
max_nb_hashtags_per_word_in_th...
max_nb_urls_per_word_in_the_tw...
Emotion
NoDailyTweets
NoURLs
mean_nb_hashtags_per_tweet
max_intertweet_times_seconds
mean_nb_mentions_per_word_in_t...
min_nb_hashtags_per_tweet
max_nb_favourites_per_tweet
median_nb_characters_per_tweet
max_nb_mentions_per_tweet
DomainSocialSpam
SocialFeature
NoReplies
median_nb_mentions_per_word_in...
mean_nb_mentions_per_tweet
InverseTopicFreq
min_intertweet_times
median_nb_symbols_per_word_in_...
Openness
median_nb_mentions_per_tweet
Extraversion
NoWords
Content-based
PersonalityTrait
mean_nb_characters_per_tweet
Conscientiousness
Agreeableness
mentions_used_on_average
mean_nb_words_per_tweet
median_nb_hasht...
Emotional_Range
median_nb_urls_per_word_in_the...
NoTweets
AvgTweetLength
max_nb_retweets_per_tweet
median_nb_urls_per_tweet
mean_nb_hashtags_per_word_in_t...
max_nb_mentions_per_word_in_th...
min_nb_characters_per_tweet
median_nb_hashtags_per_tweet
mean_nb_symbols_per_tweet
NoFav
NoRetweet
mean_nb_retweets_per_tweet
NoUniqueWords
median_nb_symbols_per_tweet
NoUniqueURLHosts
mean_nb_favourites_per_tweet
mean_nb_urls_per_tweet
min_intertweet_times_seconds
NoHashtagsPerTweet
max_nb_words_per_tweet

Figure 3. A snapshot of Content-based class and its interrelated classes and subclasses in DSpamOnto.

Figure 4 demonstrates the Metadata-based class and its subclasses. This class and its subclasses capture the use of metadata, such as hashtags, mentions, or trending topics, to promote spammy content or engage in deceptive activities.

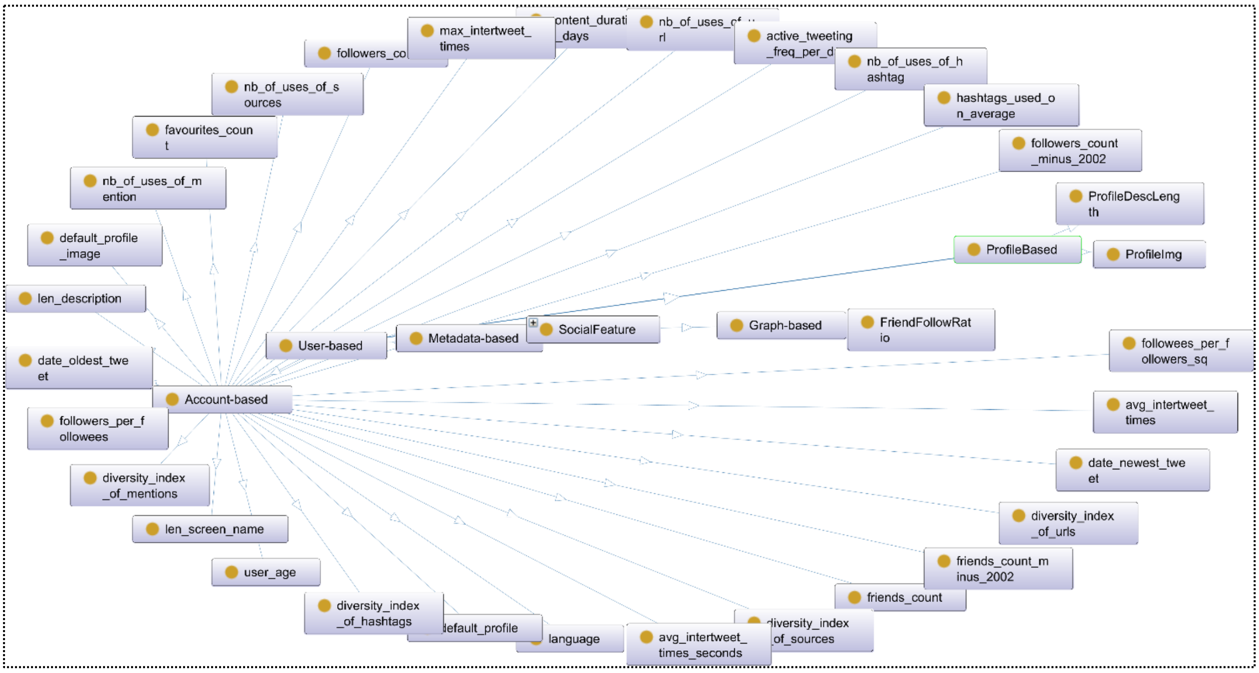

Figure 4. A snapshot of Metadata-base class and its interrelated classes and subclasses in DSpamOnto.

In our ontology-based spam detection model, the handling of domains plays a critical role in capturing the specific characteristics and behaviours of social spammers within different domains. We would like to provide a more explicit and detailed explanation of how domains are handled in our proposed ontology. The goal of our ontology, DSpamOnto, is to detect social spam in microblogging platforms by focusing on a specific domain, such as politics or health. To achieve this, we incorporate domain-specific rules and concepts into the ontology. By considering the unique characteristics and behaviours associated with different domains, DSpamOnto can effectively identify social spammers within those domains. This approach enables a more targeted and accurate detection of social spam, as it takes into account the specific patterns and activities exhibited by spammers in each domain.

For instance, let us consider the domain of the politics. In DSpamOnto, we can include rules that capture the politically biased language, dissemination of misinformation and untrustworthy sentiments and emotions about political candidates, or promotion of politically motivated agendas. These rules help in identifying social spammers who engage in such behaviours within the political domain. Similarly, in the health domain, DSpamOnto focuses on detecting spammers who spread false health claims, promote unverified medical products, or employ fraudulent practices related to health and wellness. The ontology incorporates domain-specific rules and concepts that help in identifying and distinguishing social spammers within the health domain.

By incorporating domain-specific rules and concepts, DSpamOnto ensures that the detection of social spam is tailored to the unique characteristics and behaviours within each domain. This approach enhances the accuracy and effectiveness of spam detection by considering the context and relevance of the content being analysed.

5. Ontology Design Evaluation

Ontology evaluation is a crucial step in the development and implementation of an ontology. There have been several reports in the literature on ontology evaluation measures [31–35]. Ontology evaluation establishes the quality of an ontology as well as whether its constraints and standards have been met. The following subsection discusses the evaluation metrics incorporated in this study.

5.1. Five Criteria Evaluation Metric

This research employs Yu et al.'s [31] evaluation technique, which is based on five criteria:

- **Consistency:** This refers to the absence of contradictions or conflicting information within an ontology. A consistent ontology ensures that any logical inferences made using the ontology are reliable and accurate. In order to check consistency, automated reasoning tools can be used to check that the axioms (statements) in the ontology do not lead to any logical contradictions. If a reasoner can find a contradiction in the ontology, it indicates that the ontology is inconsistent. Consistency checking is particularly important for large and complex ontologies, where it can be difficult to manually detect contradictions. Ensuring consistency can also help identify errors or gaps in the ontology's design and implementation, allowing for refinement and improvement. To guarantee that DSpamOnto is logically coherent, it has been reasoned using the FaCT++, HermiT, Pellet, Pellet (Incremental), RacerPro, and TrOWL reasoners. The reasoners looked at the class, object, and data property structures, class/object property claims, and the presence of the same entities in the ontology. The DSpamOnto contains no contradictory truths.
- **Completeness:** This refers to whether or not the ontology covers all the concepts and relationships relevant to its intended domain. In other words, completeness assesses whether the ontology includes all the necessary knowledge required to support the intended tasks. To assess completeness, the proposed ontology is compared against a set of requirements or a benchmark. The benchmark is created based on the domain experts' knowledge and includes all the concepts and relationships that the ontology is expected to cover. If an ontology satisfies the completeness criterion, it means that it includes all the necessary concepts and relationships required for its intended use. However, if the ontology is incomplete, it indicates that some aspects of the domain are not captured, which can lead to incorrect or incomplete inferences. Despite the importance of the current efforts, there has been no attempt to develop an ontology for domain-specific social spam dedicated to standardising and formalising the specified domain knowledge to the best of our knowledge. Nevertheless, this study benefits from other seminal works to develop a fine-grained ontology that conceptualises domain-specific social spam. It is essential to note that completeness is relative to the intended use of the proposed ontology, and it is not possible to achieve absolute completeness. Ontologies can always be improved and extended as new knowledge is acquired, and new use cases emerge.
- **Conciseness:** An ontology is considered concise if it contains the minimum number of concepts and relationships necessary to represent the domain it is intended to model. A concise ontology is easier to understand and use, and it has no redundancy. If an ontology is not concise, it will be difficult for users to find the information they need and to understand how the ontology works. To avoid redundancy and ensure conciseness, the developed ontology was crafted with the goal of providing concise information about the domain-based social spammers.
- **Expandability:** Expandability is the ability of an ontology to be easily modified to add new concepts and relationships. An ontology needs to be expandable because the domain it is intended to model is often dynamic and ever-changing. DSpamOnto is designed to be extensible and interoperable. It can be easily modified by adding, removing, or altering axioms. DSpamOnto also aligns with the four extensibility principles: ontology term reuse, ontology semantic alignment, ontology design patterns (ODP) usage for new term generation and existing term editing, and community extensibility [36].

- **Sensitiveness:** An ontology is considered sensitive if any change to the ontology could affect the core of the ontology. DSpamOnto is flexible and open to amendments, which means that it can accommodate changes and updates as illustrated in the Expandability evaluation metric.

5.2. Ontology-Level Evaluation

Ontology-level evaluation is a process of assessing the quality of an ontology. This can be done by looking at the ontology's structure, its semantics, and its usability. The structure of an ontology refers to the way that the ontology is organized. The semantics of an ontology refers to the meaning of the ontology's terms and relationships. The usability of an ontology refers to how easy it is to use the ontology. There are a number of different ways to evaluate an ontology. Some common methods include [37,38]:

- **The size of vocabulary (SOV):** is a metric used to measure the size or extent of an ontology's vocabulary. It is defined as the total number of definitions in the ontology, including classes, individuals, and properties. The $\boldsymbol{SOV}$ can be formulated as follows:

$$\boldsymbol{SOV} = |\boldsymbol{C}| + |\boldsymbol{I}| + |\boldsymbol{P}|$$

where $|\boldsymbol{C}|$ is the number of classes, $|\boldsymbol{I}|$ is the number of individuals, and $|\boldsymbol{P}|$ is the number of properties in the ontology. DSpamOnto contains 147 classes, 1500 individuals, and 75 properties, thus the $\boldsymbol{SOV}$ = 147 + 1500 + 75 = 1722. This metric provides a measure of the richness and complexity of an ontology's vocabulary. A larger $\boldsymbol{SOV}$ indicates that the ontology includes a greater number of concepts, relationships, and properties, and is therefore likely to be more complex and comprehensive. However, a larger $\boldsymbol{SOV}$ may also make the ontology more difficult to use and maintain. Therefore, the $\boldsymbol{SOV}$ should be considered in conjunction with other evaluation metrics to gain a comprehensive understanding of an ontology's complexity.

- **Edge node ratio (ENR):** *ENR* is a metric used in ontology evaluation to measure the connectivity and complexity of an ontology's structure. The *ENR* measures the ratio of edges to nodes in the ontology. In other words, it measures how many relationships (edges) exist for each concept (node) in the ontology:

$$ENR = E/N \tag{1}$$

where *E* is the total number of edges and *N* is the total number of nodes in the ontology. A high *ENR* value indicates that the ontology has a large number of relationships between concepts, which can make it more complex and difficult to understand. On the other hand, a low *ENR* value indicates that the ontology has relatively few relationships, which can make it more simple and easier to understand. *ENR* is a useful metric for understanding the overall structure of an ontology, but it should be used in combination with other metrics to gain a comprehensive understanding of an ontology's complexity. For example, an ontology may have a high *ENR* value but a small vocabulary size, indicating that it has many relationships between a small number of concepts. In this case, the ontology may still be relatively simple to use and understand. The value of DSpamOntolgy *ENR* is about '1', demonstrating a reasonable and straightforward domain ontology.

- **Tree impurity (TIP):** *TIP* is a metric used in ontology evaluation to measure the degree of ambiguity or uncertainty in the ontology's structure. The TIP metric is typically used for tree-structured ontologies, where each concept has only one parent and one or more children. *TIP* measures the degree to which concepts in the ontology are "impure" or ambiguous, in the sense that they have multiple child concepts that are not related. The *TIP* metric is based on the Gini impurity measure used in decision tree algorithms and can be calculated as follows:

$$TIP = 1 - \sum\left(p^2\right) \tag{2}$$

where *p* is the proportion of child concepts belonging to a particular category, such as a specific branch of the tree. The *TIP* metric ranges from 0 to 1, with a value of 0 indicating a perfectly pure ontology (all child concepts belong to the same category), and a value of 1 indicating a completely impure ontology (each concept has an equal number of children in each category). *TIP* can be a useful metric for identifying areas of an ontology that may be difficult to use or understand due to ambiguity or inconsistency. High TIP values can indicate that certain concepts in the ontology are poorly defined or have multiple conflicting interpretations. *TIP* of the DSpamOnto equals "0.4", indicating a less complicated ontology and implying a comparatively minor departure from the rooted tree in the inheritance hierarchy.

- **The entropy of ontology graph (EOG):** *EOG* measures the information content and complexity of the ontology's graph structure. The *EOG* metric is based on the concept of information entropy, which is a measure of the uncertainty or randomness of a set of data. *EOG* measures the amount of uncertainty or randomness in the ontology's graph structure, taking into account both the number of nodes and edges in the graph and the distribution of connections between them. The *EOG* metric can be calculated using the following formula:

$$EOG = -\sum_{x=1}^{n} p(x) log_2(p(x)) \tag{3}$$

where $p(x)$ is the probability of a particular connection type in the graph, such as the probability that two nodes are connected by a certain type of edge. The *EOG* metric ranges from 0 to $log_2(N)$, where N is the total number of nodes in the graph. The value of $p(x)$ is calculated by dividing the vertex's degree, or # properties linked with that class, by the sum of all degrees of V for each vertex x in the graph. In particular, $p(x_i)$ can be calculated as:

$$p(x_i) = \frac{\deg(x_i)}{\sum_{v \in V} \deg(x)} \tag{4}$$

A higher *EOG* value indicates a greater degree of uncertainty or randomness in the graph structure, while a lower *EOG* value indicates a more structured and predictable graph.

The value of *EOG* for DSpamOnto is almost one, demonstrating that the class structure of DSpamOnto is adequate and reasonable.

5.3. Class-Level Evaluation

Brewster et al. [39] developed a class-level evaluation metric called the "Ontology Design Quality Measure" (ODQM), which is used to evaluate the complexity and quality of individual classes in ontology. ODQM is based on a set of certain metrics that evaluate different aspects of a class's design, including its size, complexity, and relationships to other classes:

- **The number of classes (NOC)**: NOC metric measures the total number of classes defined in the ontology, which can give an indication of the ontology's breadth and depth in terms of the concepts it covers. The total number of classes in DSpamOnto is 147, indicating a reasonably satisfactory ontology. However, this number is anticipated to be extended considering that this is a new ontology; thus, DSpamOnto will be further expanded and populated.
- **The number of properties (NOP)**: NOP measures the richness and complexity of the ontology's relationships and attributes. The number of properties in DspamOnto is approximately 170, which shows solid reasoning.
- **The number of root classes (NORC)**: NORC is a metric used in ontology evaluation to measure the total number of classes in the ontology that have no superclasses. These classes are considered to be at the top of the ontology hierarchy and are referred to as root classes. DspamOnto's NORC is around 60, where the root classes represent distinct but related concepts.
- **Relationship richness (RR)**: RR takes into account both the number and types of relationships defined between classes. A rich ontology relationship structure typically

indicates a more comprehensive and well-structured representation of the domain being modelled. The relationships defined between classes can include various types of relationships such as subclass, part-of, instance-of, and many others. RR of DspamOnto is around 0.7, indicating richness in terms of facts rooted in DspamOnto conceptual representation.

5.4. SWRL Rules

SWRL (Semantic Web Rule Language) is a rule-based language for the Semantic Web that allows the creation of rules and logical expressions to represent knowledge [40,41]. In ontology evaluation, SWRL rules can be used to check if the ontology is consistent and satisfies certain constraints. SWRL rules can be used to express complex constraints that cannot be represented using OWL (Web Ontology Language) alone. To evaluate an ontology using SWRL rules, the rules are first defined and then applied to the ontology using a reasoner. The reasoner checks if the rules are satisfied by the ontology and reports any violations or inconsistencies. The use of SWRL rules in ontology evaluation can help ensure that the ontology is accurate, complete, and consistent and that it satisfies the intended requirements and constraints.

The following is a set of SWRL rules that are used for social spam detection based on the developed ontology:

Listing 1 in shows a rule that takes into account several factors that may indicate domain-specific social spam, such as the posting frequency, user type, sentiment score, emotion, content similarity, and user behaviour. The values of these features are estimated based on the domain-based behaviour of the user. If the conditions of this rule are met, the rule will infer that the social media account in question is likely a spammer and will create a SocialSpamDetectionResult with the classification of “spammer”. Note that this rule also includes some SWRL built-in functions to perform mathematical and logical operations, such as swrlb:greaterThan, swrlb:equal, and swrlb:lessThan.

Listing 1: An example of a SWRL rule to detect a social spammer.

```
DomainSocialSpammer(?x) ∧
      hasPostingFrequency(?x, ?freq) ∧
      hasUserType(?x, ?type) ∧
      hasSentimentScore(?x, ?sentiment) ∧
      hasEmotion(?x, ?emotion) ∧
      hasContentSimilarity(?x, ?similarity) ∧
      hasUserBehavior(?x, ?behavior) →
            SocialSpamDetectionResult(?x, "spammer")
                ^ (swrlb:greaterThan(?freq, 100)
                ^ swrlb:equal(?type, "bot")
                ^ swrlb:lessThan(?sentiment, -0.5)
                ^ swrlb:equal(?emotion, "anger")
                ^ swrlb:greaterThan(?similarity, 0.8)
                ^ swrlb:equal(?behavior, "malicious"))
```

To capture the temporal factor, Listing 2 shows a rule that detects domain-specific social spammers who have created more than 20 posts, tweets, or comments in a specific time frame and who have not received any comments on their posts or tweets in that same time frame. The rule uses various built-in SWRL functions, such as Subtract, DateTime, and Count, to calculate time differences and count the number of posts or tweets by a particular user. The rule also checks if any comments have been received on the user’s posts or tweets by using the NotExists operator. If the rule conditions are satisfied, the user is marked as a spammer.

Listing 2: An example of a SWRL rule to detect a social spammer.

```
DomainSocialSpammer(?s) ∧
     CreatedAt(?t, ?d) ∧
     DateTime(?d, ?y, ?m, ?day, ?h, ?min, ?sec, ?zone) ∧
     Subtract(2023, ?y, ?y_diff) ∧
     Subtract(3, ?m, ?m_diff) ∧
     GreaterThan(?y_diff, 0) ∧
      Or(GreaterThan(?m_diff, 0), And(Equal(?m_diff, 0), GreaterThan(14, ?day))) ∧
     Count(?s, ?c) ∧ GreaterThan(?c, 20) ∧
     NotExists(?p, Comment(?p, ?s, ?c2) ∧
     CreatedAt(?p, ?d2) ∧
     DateTime(?d2, ?y2, ?m2, ?day2, ?h2, ?min2, ?sec2, ?zone2) ∧
     Subtract(?y_diff, ?y2_diff, ?y_diff_diff) ∧
     Subtract(?m_diff, ?m2_diff, ?m_diff_diff) ∧
     Or(GreaterThan(?y_diff_diff, 0), And(Equal(?y_diff_diff, 0), GreaterThan(14,
?day_diff)))) →
             DomainSocialSpammer (?s)
```

In the Listing 3, DomainSocialUserType represents a class of social users, and SocialSpammer is a class of users that have been identified as social spammers. The rule checks for a user(p)that has an inverse topic frequency of value "1", and also has at least one of the following domain-specific characteristics: has an excessive embedded URLs and has an excessive number of hashtags. If the user's social content meets these conditions, then the user is classified as domain social spammer.

Listing 3: An example of a SWRL rule to detect a social spammer.

```
SocialUserType(?p) ∧
       InverseTopicFreq(?p, "1") ∧
       (hasExcessiveLinks(?p, true) ∨ hasExcessiveHashtags (?p, true)) →
             DomainSocialSpammer (?p)
```

Listing 4 illustrates a rule that identifies a person (p) as a spammer if they are friends with a person who has many followers, they have tweeted with a high frequency of spam keywords, their tweet has negative sentiment and contains an emotion other than joy or love, and their tweet contains many hashtags, mentions, and links (at least five in total).

Listing 4: An example of a SWRL Rule to detect a social spammer.

```
SocialUserType (?p) ∧
       FriendsWith(?p, ?f) ∧
       UserHasManyFollowers(?f) ∧
        HighFrequencyOfSpamKeywordsInTweet(?t, ?k) ∧ TweetHasNegativeSentiment(?t) ∧
       EmotionDetectedInTweet(?t, ?e) ∧ (?e != "joy" ^ ?e != "love") ∧
TweetContainsManyHashtags(?t) ∧
       TweetContainsManyMentions(?t) ∧
       TweetContainsManyLinks(?t) ∧
       ?nlinks + ?nmentions + ?nhashtags >= 5) →
             DomainSocialSpammer(?p, ?t)
```

6. Experimental Results

This section demonstrates the effectiveness of the proposed ontology. A benchmark comparison with other state-of-the-art machine learning models is conducted over a labelled dataset. The following sections discuss the incorporated dataset, evaluation metrics, and experimental results.

6.1. Dataset Selection and Preprocessing

The MIB dataset is a collection of Twitter accounts, consisting of both fake and legitimate accounts, published by the Institute of Informatics and Telematics (IIT), part of the Italian National Research Council (CNR) [42]. The dataset includes a set of features that are commonly used to distinguish between fake and legitimate accounts, such as the number of followers, the number of tweets, the age of the account, and the frequency of tweets. The dataset is intended to be used for research on social media spam detection and has been widely used by researchers in the field. It provides a valuable resource for developing and evaluating machine learning models and other algorithms for detecting social media spam. Table 3 shows certain statistics for MIB dataset.

Table 3. Statistics of incorporated MIB dataset.

Group Name	#Accounts	#Tweets	Year
social spambots	3457	428,542	2014
traditional spambots	433	5,794,931	2013
fake followers	3351	196,027	2012

The incorporated dataset is pre-processed to enhance quality and to prepare it for the conducted experiment. The pre-processing includes data cleaning to eliminate noisy and irrelevant data and data normalization to standardize the data and reduce variations. This step also includes case folding, stemming, and stop word removal.

6.2. A Comparison with ML Classifiers

The proposed model is also evaluated with various machine learning classifiers, namely Naïve Base (NB), Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), K Neighbours (KNN), and Linear Regression (LR). To assess the performance of the classification models, the data is split into training, validation, and testing sets. The training set is used to train the model, the validation set is used for hyperparameter tuning, and the testing set is used for performance evaluation. Several evaluation metrics are used in this experiment including Accuracy, Precision, Recall, and F1 score.

6.3. Results

Table 4 reports the results of the conducted experiment. The table shows the performance comparison results of different machine learning models on a given dataset based on several evaluation metrics such as accuracy, precision, recall, and F1 score. Each row represents a specific model, and the columns represent the corresponding evaluation metrics. From the table, it can be observed that the proposed method has the highest accuracy of 0.8029, followed closely by the RF model with an accuracy of 0.7972. The decision tree (DT) model also performs well with an accuracy of 0.7943. The lowest accuracy is observed for the logistic regression (LR) model with an accuracy of 0.6999. This is mostly due to specific assumptions that may cause the NB and LR models to perform poorly. NB and LR, in particular, frequently presume feature independence; as a result, they are unable to learn from feature interactions [43,44]. As a result, situations with high correlations between features imply that NB and LR classifiers are unable to produce appropriate estimations due to this strong assumption.

Regarding precision, the proposed method has a precision score of 0.7088, which is slightly better than the RF model with a precision score of 0.7055. The lowest precision is observed for the NB model with a precision score of 0.597. For recall, the proposed method has the highest score of 0.816, followed by the RF model with a recall score of 0.799. The lowest recall score is observed for the LR model with a score of 0.5801. Finally, for the F1 score, the proposed method has the highest score of 0.7579, followed by the RF model with a score of 0.7486. The lowest F1 score is observed for the LR model with a score of 0.5925.

Table 4. Comparison Results.

Model	Accuracy	Prec.	Recall	F1
LR	0.6999	0.6072	0.5801	0.5925
NB	0.7268	0.597	0.8903	0.7049
KNN	0.7378	0.6542	0.6558	0.6535
SVM	0.7607	0.6946	0.685	0.677
DT	0.7943	0.7189	0.7506	0.7338
RF	0.7972	0.7055	0.799	0.7486
Proposed method	0.8029	0.7088	0.816	0.7579

7. Conclusions

Social spam is a persistent problem on online social networks that can harm individuals and businesses. Machine learning and ontology-based approaches have been proposed to detect social spam. Machine learning techniques analyse data patterns and characteristics to distinguish spam from legitimate content, while ontologies explicitly model domain knowledge to create rules for identifying social spammers. However, both approaches have their limitations, such as false positives and false negatives in machine learning, and the inability to conceptualize domain-specific social spam in ontologies. Nevertheless, ongoing research in this area aims to develop more accurate and effective approaches to combat social spam on online social networks.

This study intends to fill this need by developing DSpamOnto, a domain-specific ontology for detecting social spammers in microblogging that target a certain domain. DSpamOnto can detect social spammers based on domain-specific activity such as publishing repeated or unrelated material and utilising false information. To test and confirm its utility in catching social spammers, the suggested model is evaluated and benchmarked against well-proven ML models using several assessment measures.

Future work in this area should focus on developing more effective and efficient techniques for detecting social spam in microblogging platforms, with a particular emphasis on domain-specific spam. By developing better detection techniques, it may be possible to reduce the negative impact of social spam on users and businesses and improve the overall quality of experience on microblogging platforms. The following are the proposed areas that will be investigated in the future:

- Incorporating DSpamOnto with machine learning techniques: The proposed domain-specific ontology, DSpamOnto, can be integrated with machine learning algorithms to improve the accuracy of social spam detection in microblogging platforms. By combining the explicit domain knowledge represented by the ontology with the pattern recognition capabilities of machine learning algorithms, more effective and efficient spam detection models can be developed.
- Developing a large labelled dataset for DSpamOnto: In order to train accurate machine learning models using DSpamOnto, a large labelled dataset of domain-specific spam and non-spam content is required. This can be a challenging task as it requires manual labelling of a large amount of data. However, developing such a dataset is crucial for the development and evaluation of machine learning models for detecting domain-specific social spam.
- Testing DSpamOnto on different microblogging platforms: The proposed ontology should be tested on different microblogging platforms to evaluate its effectiveness and applicability in different contexts. This will help to identify any platform-specific issues that may arise when using DSpamOnto and to develop solutions to address them.
- Developing strategies to combat evasive social spammers: As noted in the literature, spammers can use various techniques to evade detection, such as using natural language variations and changing account information frequently. Therefore, it is im-

portant to develop strategies to detect and combat these evasive social spammers. This may involve using more sophisticated machine learning algorithms or incorporating additional features into DSpamOnto to identify these tactics.

Author Contributions: Conceptualization, M.A.-H. and B.A.-S.; methodology, M.A.-H. and B.A.-S.; software, M.A.-H. and B.A.-S.; validation, M.A.-H., B.A.-S. and A.A.H.; formal analysis, M.A.-H. and B.A.-S.; investigation, M.A.-H. and B.A.-S.; resources, M.A.-H. and B.A.-S.; data curation, A.A.H.; writing—original draft preparation, M.A.-H. and B.A.-S.; writing—review and editing, M.A.-H., B.A.-S. and A.A.H.; visualization, B.A.-S.; supervision, M.A.-H.; project administration, M.A.-H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to express our sincere appreciation to the referees for their valuable and insightful comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abu-Salih, B.; Qudah, D.A.; Al-Hassan, M.; Ghafari, S.M.; Issa, T.; Aljarah, I.; Alqahtani, S. An intelligent system for multi-topic social spam detection in microblogging. *J. Inf. Sci.* **2022**. [CrossRef]
2. Zantal-Wiener, A. 47% of Social Media Users Report Seeing More Spam in Their Feeds, Even as Networks Fight to Stop It. 2019. Available online: https://blog.hubspot.com/marketing/social-media-users-seeing-more-spam (accessed on 3 August 2020).
3. Barati, R. Security Threats and Dealing with Social Networks. *SN Comput. Sci.* **2022**, *4*, 9. [CrossRef]
4. Rodrigues, A.P.; Fernandes, R.; Shetty, A.; Lakshmanna, K.; Shafi, R.M. Real-time twitter spam detection and sentiment analysis using machine learning and deep learning techniques. *Comput. Intell. Neurosci.* **2022**, *2022*, 5211949. [CrossRef] [PubMed]
5. Shen, H.; Liu, X.; Zhang, X. Boosting Social Spam Detection via Attention Mechanisms on Twitter. *Electronics* **2022**, *11*, 1129. [CrossRef]
6. Rao, S.; Verma, A.K.; Bhatia, T. Hybrid ensemble framework with self-attention mechanism for social spam detection on imbalanced data. *Expert Syst. Appl.* **2023**, *217*, 119594. [CrossRef]
7. Ghanem, R.; Erbay, H. Spam detection on social networks using deep contextualized word representation. *Multimed. Tools Appl.* **2023**, *82*, 3697–3712. [CrossRef]
8. Shams, R.; Mercer, R.E. Supervised classification of spam emails with natural language stylometry. *Neural Comput. Appl.* **2016**, *27*, 2315–2331. [CrossRef]
9. Çıtlak, O.; Dörterler, M.; Doğru, İ.A. A survey on detecting spam accounts on Twitter network. *Soc. Netw. Anal. Min.* **2019**, *9*, 35. [CrossRef]
10. Concone, F.; Re, G.L.; Morana, M.; Das, S.K. SpADe: Multi-Stage Spam Account Detection for Online Social Networks. *IEEE Trans. Dependable Secur. Comput.* **2022**, 1–16. [CrossRef]
11. Fernández-López, M.; Gómez-Pérez, A.; Juristo, N. Methontology: From ontological art towards ontological engineering. In Proceedings of the 1997 AAAI Spring Symposium, Palo Alto, CA, USA, 24–26 March 1997.
12. Lenat, D.; Guha, R. Building large knowledge-based systems: Representation and inference in the CYC project. *Artif. Intell.* **1993**, *61*, 4152.
13. Herath, T.B.G.; Khanna, P.; Ahmed, M. Cybersecurity Practices for Social Media Users: A Systematic Literature Review. *J. Cybersecur. Priv.* **2022**, *2*, 1–18. [CrossRef]
14. Networks, B. Spear Phishing: Top Threats and Trends. 2020. Available online: https://www.barracudamsp.com/resources/reports/spear-phishing-threats-and-trends/#:~:text=Spear%20phishing%20is%20a%20threat,business%20email%20compromise%2C%20and%20blackmail (accessed on 15 March 2023).
15. Agrawal, D.; Deepak, G. OntoSpammer: A Two-Source Ontology-Based Spam Detection Using Bagging. In *Innovations in Electrical and Electronic Engineering: Proceedings of ICEEE 2022—2022 9th International Conference on Electrical and Electronics Engineering (ICEEE), Alanya, Turkey, 29–31 March 2022*; Springer: Berlin/Heidelberg, Germany, 2022; Volume 2, pp. 145–153.
16. Vidanagama, D.; Silva, A.; Karunananda, A. Ontology based sentiment analysis for fake review detection. *Expert Syst. Appl.* **2022**, *206*, 117869. [CrossRef]
17. Jabardi, M.H.; Hadi, A.S. Ontology Meter for Twitter Fake Accounts Detection. *Int. J. Intell. Eng. Syst.* **2021**, *14*, 410–419. [CrossRef]
18. Halawi, B.; Mourad, A.; Otrok, H.; Damiani, E. Few are as Good as Many: An Ontology-Based Tweet Spam Detection Approach. *IEEE Access* **2018**, *6*, 63890–63904. [CrossRef]
19. Hussain, M.; Ahmed, M.; Khattak, H.A.; Imran, M.; Khan, A.; Din, S.; Ahmad, A.; Jeon, G.; Reddy, A.G. Towards ontology-based multilingual URL filtering: A big data problem. *J. Supercomput.* **2018**, *74*, 5003–5021. [CrossRef]

20. Abu-Salih, B.; Wongthongtham, P.; Kit, C.Y. Twitter mining for ontology-based domain discovery incorporating machine learning. *J. Knowl. Manag.* **2018**, *22*, 949–981. [CrossRef]
21. Wongthongtham, P.; Salih, B.A. Ontology-based approach for identifying the credibility domain in social Big Data. *J. Organ. Comput. Electron. Commer.* **2018**, *28*, 354–377. [CrossRef]
22. Jain, G.; Sharma, M.; Agarwal, B. Spam detection in social media using convolutional and long short term memory neural network. *Ann. Math. Artif. Intell.* **2019**, *85*, 21–44. [CrossRef]
23. Alan, R.H.; Salvatore, T.M.; Jinsoo, P.; Sudha, R. Design science in information systems research. *MIS Q.* **2004**, *28*, 75–105.
24. Jones, D.; Gregor, S. The anatomy of a design theory. *J. Assoc. Inf. Syst.* **2007**, *8*. [CrossRef]
25. Venable, J. Rethinking Design Theory in Information Systems. In *Design Science at the Intersection of Physical and Virtual Design*; vom Brocke, J., Ed.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 136–149.
26. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. A Design Science Research Methodology for Information Systems Research. *J. Manag. Inf. Syst.* **2007**, *24*, 45–77. [CrossRef]
27. Borse, D.; Borse, S. State of the art on Twitter spam detection. In *Applied Computational Technologies: Proceedings of ICCET*; Springer Nature: Berlin/Heidelberg, Germany, 2022; pp. 486–496.
28. Sun, N.; Lin, G.; Qiu, J.; Rimba, P. Near real-time twitter spam detection with machine learning techniques. *Int. J. Comput. Appl.* **2020**, *44*, 338–348. [CrossRef]
29. Vives, L.; Tuteja, G.S.; Manideep, A.S.; Jindal, S.; Sidhu, N.; Jindal, R.; Bhatt, A. A novel hybrid approach of gravitational search algorithm and decision tree for twitter spammer detection. *Int. J. Mod. Phys. C* **2021**, *33*, 2250060. [CrossRef]
30. Deng, L.; Wu, C.; Lian, D.; Wu, Y.; Chen, E. Markov-Driven Graph Convolutional Networksfor Social Spammer Detection. *IEEE Trans. Knowl. Data Eng.* **2022**. [CrossRef]
31. Yu, J.; Thom, J.A.; Tam, A. Requirements-oriented methodology for evaluating ontologies. *Inf. Syst.* **2009**, *34*, 766–791. [CrossRef]
32. Alani, H.; Brewster, C. Metrics for Ranking Ontologies. 2006. Available online: https://eprints.soton.ac.uk/262603/1/Alani-EON06.pdf (accessed on 15 March 2023).
33. D'Aquin, M.; Schlicht, A.; Stuckenschmidt, H.; Sabou, M. Criteria and evaluation for ontology modularization techniques. In *Modular Ontologies*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 67–89.
34. Dellschaft, K.; Staab, S. Strategies for the Evaluation of Ontology Learning. *Ontol. Learn. Popul.* **2008**, *167*, 253–272.
35. Zavitsanos, E.; Paliouras, G.; Vouros, G.A. Gold Standard Evaluation of Ontology Learning Methods through Ontology Transformation and Alignment. *IEEE Trans. Knowl. Data Eng.* **2010**, *23*, 1635–1648. [CrossRef]
36. Yongqun, H.; Zuoshuang, X.; Jie, Z.; Yu, L.; James, A.O.; Edison, O. The eXtensible ontology development (XOD) principles and tool implementation to support ontology interoperability. *J. Biomed. Semant.* **2018**, *9*, 3.
37. Srinivasulu, S.; Sakthivel, P.; Balamurugan, E. Measuring the ontology level and class level complexity metrics in the semantic web. *Int. J. Adv. Comput. Eng. Netw.* **2014**, *2*, 68–74.
38. Ajami, H.; Mcheick, H. Ontology-Based Model to Support Ubiquitous Healthcare Systems for COPD Patients. *Electronics* **2018**, *7*, 371. [CrossRef]
39. Brewster, C.; Alani, H.; Dasmahapatra, S.; Wilks, Y. Data Driven Ontology Evaluation. 2004. Available online: https://www.researchgate.net/publication/37537072_Data_Driven_Ontology_Evaluation (accessed on 15 March 2023).
40. Hassanpour, S.; O'Connor, M.J.; Das, A.K. Exploration of SWRL rule bases through visualization, paraphrasing, and categorization of rules. In *Rule Interchange and Applications: International Symposium*; Springer: Las Vegas, NV, USA, 2009.
41. Horrocks, I.; Peter, F.; Harold, B.; Said, T.T.; Benjamin, G.; Mike, D. SWRL: A semantic web rule language combining OWL and RuleML. *W3C Memb. Submiss.* **2004**, *21*, 1–31.
42. Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; Tesconi, M. Fame for sale: Efficient detection of fake Twitter followers. *Decis. Support Syst.* **2015**, *80*, 56–71. [CrossRef]
43. Zulkarnain, N.Z.; Meziane, F. Ultrasound reports standardisation using rhetorical structure theory and domain ontology. *J. Biomed. Informatics* **2019**, *100*, 100003. [CrossRef] [PubMed]
44. Elhenawy, M.; El-Shawarby, I.; Rakha, H. *Modeling the Perception Reaction Time and Deceleration Level for Different Surface Conditions Using Machine Learning Techniques*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 131–142.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.