CommentSecure: YouTube Spam Comment Filtering using Ensemble Learning

DISSERTATION

Submitted in partial fulfillment of the requirements of the MTech in Data Science and Engineering Degree programme

By

(Jeevan Madhukar Chavan)

(2021SC04033)

Under the supervision of

(Savio Coelho)

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

Pilani , INDIA

(March, 2024)

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
FIRST SEMEST 2023-24

Dissertation Title: CommentSecure: YouTube Spam Comment Filtering using Ensemble Learning

Name of Student: Jeevan Madhukar Chavan

Name of Supervisor: Savio Coelho

ID No. of Student: 2021SC04033

EVALUATION DETAILS

EC No.	Component	Weightage	Comments	Marks Awarded
1	Abstract Outline	10%	Abstract – Outline Document Abstract is well-constructed, providing a concise yet comprehensive overview of the project. It effectively captures the problem statement, methodology, and anticipated outcomes, setting a clear expectation for readers	100%
2.	Mid-Sem Progress Seminar Viva Work Progress	 10% 5% 15%	Introduction is commendable, laying a strong foundation by effectively highlighting the significance of spam comment filtering on YouTube. The literature review is well-structured, data handling and modeling section could benefit from more specific details, particularly regarding dataset information, transparent bias handling, and a clear presentation of data preprocessing steps	100% 95% 95%

Organizational Mentor Name:	Savio Coelho
Qualification	BE IT- 13 Years of experience
Designation & Address	Vice President - JPMorgan Chase & co, Mumbai
Email Address	savio.coelho84@gmail.com
Signature	**Savio Coelho**
Date	**30 Jan 2024**

Abstract

Keywords: Comment moderation, Classification techniques, Naive Bayes, KNN, SVM, ensemble learning

YouTube has played a important role as a social media platform for video content sharing since its publishment. However, the platform's open nature exposes users to malicious intentions, such as the dissemination of malware and profanity, often manifesting in the comment section. Despite the presence of YouTube's built-in spam control tool, its efficacy falls short in addressing the dynamic landscape of malicious and spam content within comments.

In response to this challenge, our project conducts a comprehensive evaluation of top-performance classification techniques, with a focus on incorporating ensemble learning methods to fortify YouTube's comment moderation capabilities. Through a rigorous statistical analysis, Support Vector Machine, K-Nearest Neighbors and Naive Bayes, emerge as statistically equivalent performers.

Building upon these findings and the power of ensemble learning, we introduce " CommentSecure" – an innovative system designed to filter comments on YouTube in real-time. This system represents a cutting-edge solution, leveraging advanced classification techniques, including ensemble learning, to effectively identify and mitigate spam, profanity and malicious content within the dynamic comment environment of YouTube.

Chapter 1

Introduction

In recent times, a significant portion of internet activity has shifted towards social media platforms, a trend primarily driven by the appealing content they offer. The underlying reasons for this shift can be attributed to the preferences of individuals, particularly the youth, who exhibit a tendency to favor shorter formats such as tweets, videos, and concise posts over longer forms like e-books and blogs. Social media platforms align seamlessly with these preferences, offering a variety of short-form content, including tweets, videos, posts, and comments.

Current statistics highlight Facebook and YouTube as the most frequented social platforms, with YouTube boasting an impressive 1.5 billion monthly active users. Viewers on this platform spend an average of over an hour daily exclusively on mobile devices. Despite variations in statistics, Amazon Web Traffic Statistics ranks YouTube as the leading social media platform. The continuous evolution of social media platforms involves the incorporation of intriguing features designed for optimal user benefit. Notably, YouTube exemplifies this trend by enabling users to share in advertisement income through models such as Count per Click. Consequently, most YouTubers have gained substantial influence, surpassing their traditional media counterparts. Beyond monetary gains, this influence is also attributed to the freedoms, reduced content restrictions, and the ability to be selective about video content.

The current state of internet speed and accessibility enhances the appeal of YouTube videos as a compelling option for online viewing. However, despite these advantages, YouTube faces notable drawbacks that impact the quality of its content. A significant concern is the prevalence of malicious and spam content within the video comment section. Although the option to disable comments exists, the inherent value of user interaction prompts the continued enablement of comments. Comments contribute positively to shared videos by offering improvement suggestions. The decision to maintain comments on YouTube necessitates an acknowledgment that a considerable number may be classified as spam.

In this context, we present a thorough performance assessment of various widely recognized machine learning techniques designed for the automated filtration of undesirable comments. Our primary objective is to identify effective methods and configurations suitable for integration into an online tool aimed at detecting inappropriate text comments on YouTube.

1.1 Problem Statement

A significant challenge encountered on YouTube pertains to renowned channel owners grappling with issues in their video comment sections. This challenge manifests in the form of spam comments strategically designed to self-promote videos or propagate viruses and malware, ultimately resulting in the disabling of the comment section for the affected channel.

1.2 Objectives

- To develop a spam comment monitoring System which will detect disseminate viruses and malwares links in comment section of channel.
- To detect the self-promoting / Abuse comments in video sections
- To Label comments as Spam or Ham.
- To protect users from malicious activities
- To show that ensemble method is feasible to get more accurate results

1.3 Motivation:

The motivation behind ensemble learning lies in addressing the limitations of individual models. Different models may capture different aspects of the underlying patterns in the data or make different errors. By combining these diverse models, ensemble methods aim to create a more reliable and generalizable prediction.

1.4 Ensemble Learning

The ensemble method is constructed by combining the predictions of the individual Naive Bayes, KNN, and SVM classifiers. This fusion is achieved through a well-considered aggregation technique, such as voting or averaging, to harness the strengths of each algorithm and mitigate their individual weaknesses. The implementation is carried out using Python, leveraging popular machine learning libraries.

Chapter 2

Literature Survey

G. Mishne, D. Carmel and R. Lempel, "Blocking Blog Spam with Language Model Disagreement," AIRWeb vol. 5, pp. 1-6, 2005[1].

The exploration of comment-related issues has been ongoing for over a decade, with noteworthy contributions from various researchers. Mishne, Carmel, and Lempel's early work in 2005 addressed blog spam by creating a language-based model that required no training. Their model, though relatively simple, utilized language context to identify spam in blog comments. Despite a 7.5% false positive rate and an 11% false negative rate, the study's limited corpus raised concerns about its practicality on a broader scale.

Ashwin Rajadesingan and Anand Mahendran, "Comment Spam Classifica- tion in Blogs through Comment Analysis and Comment-Blog Post Relation- ships,"

Rajadesingan and Mahendran focused on classifying comment spam in blogs, recognizing the persistent issue of spammers targeting commenting systems. Their innovative approach involved considering blog post-comment relationships and previously-unexplored features. Through their experimental approach, the methodology demonstrated notable success in achieving a spam detection accuracy of 94.82%, showcasing precision at 96.50% and recall at 95.80%.

A. Sureka, "Mining user comment activity for detecting forum spammers in YouTube," arXiv preprint arXiv:1103.5044, 2011[3].

Sureka's 2011 study explored the domain of YouTube comment spamming, employing a text-mining approach for the automated detection of spam activity. The methodology flagged users as potential spammers based on behaviors such as making multiple comments on the same video or consistently posting comments on unrelated videos. Despite achieving a noteworthy accuracy in spam flagging, questions arose about the method's efficiency, primarily because it was tested on a relatively small sample dataset

In summary, these studies collectively contribute to the evolving understanding of combating comment spam in various online platforms, showcasing diverse methodologies and addressing limitations in detection models.

Chapter 3

Requirements and Analysis

A Software Requirements Specification (SRS) is presented as a comprehensive description of the software system designed for filtering spam comments on YouTube. It encompasses both functional and non-functional requirements, along with a set of use cases delineating user interactions that the software must facilitate.

3.1 Hardware Requirements:

To ensure optimal performance, the system is designed with the following hardware specifications:

Processor: Intel core I3

Speed: 2.0 GHz

RAM: 4 GB

Hard Disk: 100 GB min

Keyboard: Standard Keyboard

Mouse: Touch or Button Mouse

Monitor: LED Monitor with a resolution of 1920x1080 or higher

3.2 Software Requirements:

The software components of the system include the following:

Operating System: Windows 10

Programming Language: Python

Software Version: Python 3.8 or above

Development Environment: PyCharm 2023 or the latest version

Front End: HTML5, CSS3

Functional Requirements:

Comment Retrieval:

The system must fetch comments from YouTube for analysis.

Preprocessing:

Comments undergo preprocessing to enhance their quality before analysis. Includes cleaning and normalization processes.

Spam Filtering:

Implement a state-of-the-art spam filtering mechanism using Python to difference between spam and ham comments. Utilize the latest machine learning libraries and techniques for improved accuracy.

User Interaction:

Provide a modern and responsive user interface for users to view videos, post comments, and interact with the system seamlessly.

Non-Functional Requirements:

Performance:

The system must perform efficiently on the specified hardware, ensuring a smooth user experience. Aim for low latency in comment processing.

Programming Language:

Python 3.8 or above is the designated programming language for system development.

Development Environment:

Utilize PyCharm 2023 or the latest version as the preferred Integrated Development Environment (IDE) for coding and project management.

Use Cases:

User logs in and views videos:

The system allows users to log in and view videos on YouTube.

User posts a comment:

Users can actively engage with the platform by posting comments on videos.

System fetches comments:

The system retrieves comments from YouTube for analysis.

Comments undergo preprocessing:

Prior to analysis, comments undergo advanced preprocessing steps for enhanced efficiency.

Spam filtering using Python:

Implement a cutting-edge spam filtering mechanism in Python to categorize comments as spam or non-spam.

Chapter 4

System Architecture and Use Diagram

4.1 System Architecture

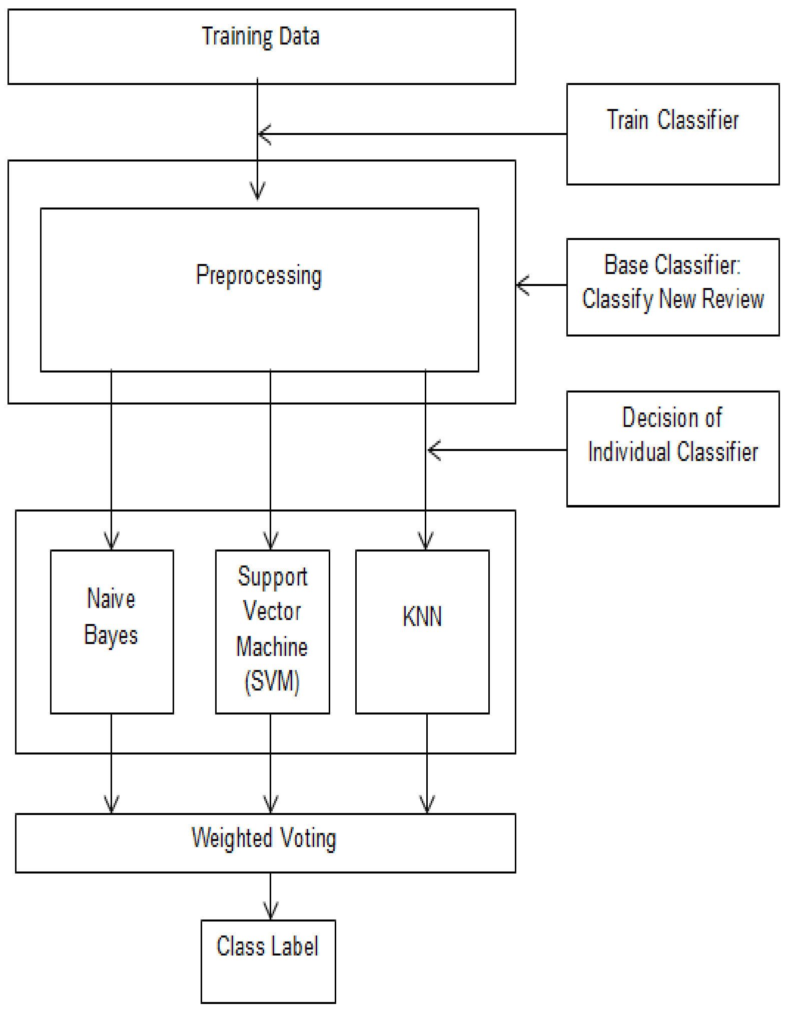

Training Data
Train Classifier
Preprocessing
Base Classifier: Classify New Review
Decision of Individual Classifier
Naive Bayes
Support Vector Machine (SVM)
KNN
Weighted Voting
Class Label

Figure 4.1: System Architecture

The outlined diagram provides an overview of the comprehensive architecture proposed for the system. The initial phase of the methodology involves the preparation of datasets for analysis, and these datasets are sourced from YouTube using the YouTube Data API.

In the consolidated dataset, the "Classification" field plays a crucial role, serving as a key indicator to determine whether a comment falls into the "spam" or "ham" category. During the data processing phase, special attention is given to extracting and isolating the comments column, treating it as a dedicated vector for specific processing. The methodology views comments as bags of words, deliberately avoiding additional preprocessing to preserve the originality of the text. Each word is considered as a set of terms, with each term forming a word that consists of two or more alphabetical characters, underscores, or numbers.

For the classification task, Naive Bayes ,Support Vector Machine, and KNN algorithms are employed. This multi-algorithmic approach enhances the system's capacity to effectively distinguish between spam and non-spam comments, contributing to a robust and accurate classification process..

4.2 Use-Case Diagram

In the context of a user-admin-system scenario, the interaction involves three primary actors: the user, the admin, and the system.

User:

The user has several functionalities, including the ability to log in, view videos, and post comments. Logging in is a prerequisite for personalized interactions, while viewing videos and posting comments contribute to the user's engagement with the platform.

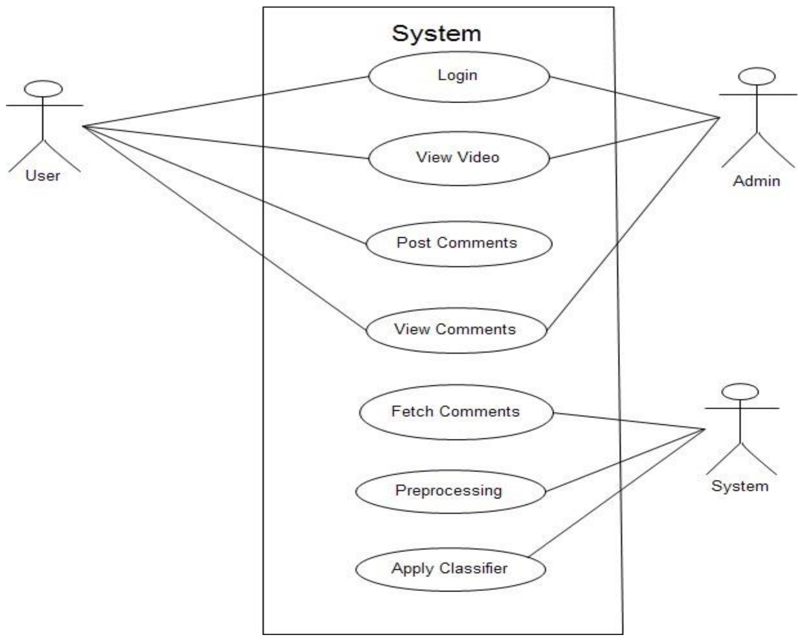
Admin:

The admin actor oversees administrative functions within the system. While specific details are not outlined, the admin likely possesses elevated privileges for managing user accounts, content, or handling issues related to user interactions.

System:

The system, acting as the underlying infrastructure, performs several crucial tasks in response to user actions. It fetches comments from the database, preprocesses the comments, and applies a classifier to distinguish between spam and non-spam content. This integral process ensures a streamlined and quality user experience.

These interactions can be summarized as follows:



User Actions:

Login: Users initiate a secure login process to access personalized features and engage with the platform.

View Video: Users have the capability to browse and view videos on the platform, contributing to their overall user experience.

Post Comment: Users can actively engage by posting comments on videos, fostering community interaction.

System Actions:

Fetch Comment: The system retrieves comments from the database, ensuring that the most up-to-date information is presented to users.

Preprocess: Before analysis, comments undergo preprocessing, possibly involving steps such as cleaning, normalization, or structuring to enhance the efficiency of subsequent tasks.

Apply Classifier: The system employs a classifier to evaluate comments, distinguishing between spam and non-spam content. This step contributes to maintaining the quality and relevance of user-generated content.

This delineation clarifies the distinct roles and functionalities of the user, admin, and system in this interactive scenario. The user engages with the platform, the admin oversees administrative functions, and the system diligently manages data, ensuring a seamless and secure user experience.

Chapter 5
Methodology and Algorithms

5.1 Methodology

The project involves a comprehensive evaluation of three classification techniques: Naive Bayes, KNN, and SVM. These techniques are selected based on their widespread use and effectiveness in text classification tasks. Ensemble learning is introduced to harness the strengths of each individual classifier, creating a more robust and accurate system.

5.1.1 Data Collection:

To conduct a comprehensive evaluation of comment moderation techniques, a diverse and representative dataset of YouTube comments was collected. This dataset included comments with varying content types, such as text, emojis, and multimedia elements. Efforts were made to ensure the inclusion of both benign and malicious comments to create a balanced training and testing set.

5.1.2 Preprocessing:

The collected dataset underwent thorough preprocessing to standardize and clean the text data. This involved tasks such as lowercasing, removal of special characters, and stemming to ensure consistency and reduce dimensionality. Emphasis was placed on maintaining the context and semantics of the comments during preprocessing to preserve the nuances of user-generated content.

5.1.3 Feature Extraction:

For the classification models, the preprocessed comments were transformed into numerical feature vectors. Techniques such as TF-IDF were employed to represent the importance of words within the comments. This feature extraction process aimed to convert textual information into a format suitable for training machine learning models.

5.1.4 Model Selection:

Three established classification technique were selected for their well-documented effectiveness in handling text classification tasks. The implementation of these algorithms was carried out using widely-used machine learning libraries, with meticulous attention given to tuning their hyperparameters through cross-validation. This iterative tuning process aimed to optimize the performance of each algorithm, ensuring that they are well-adapted and finely calibrated for the specific requirements of the text classification task at hand.

5.1.5 Ensemble Learning:

Ensemble learning was introduced to harness the collective intelligence of multiple classifiers. A weighted voting system was employed, where the decisions of Naive Bayes, KNN, and SVM were combined to produce a final classification. The weights assigned to each classifier were optimized based on their individual performance on the validation set.

5.1.6 Evaluation Metrics:

To evaluate the efficacy of the models, we employed established classification metrics, encompassing precision, recall, and F1 score. Precision serves as a metric to assess the accuracy of positive predictions, while recall evaluates the model's capacity to capture all positive instances. The F1 score, being a harmonic mean of precision and recall, offers a balanced measure that considers both aspects of the model's performance. These metrics collectively provide a comprehensive analysis of the models' classification performance, offering insights into their precision, ability to capture relevant instances, and overall balanced effectiveness

5.1.7 Experimental Setup:

To assess the generalization performance of the models, we partitioned the dataset into training and testing sets. During the training phase, cross-validation was implemented to mitigate overfitting and enhance model robustness. Furthermore, a dedicated validation set played a pivotal role in fine-tuning hyperparameters and optimizing ensemble learning strategies. This approach ensured a comprehensive evaluation of the models, promoting their ability to generalize well to unseen data and optimizing their performance through meticulous tuning.

5.1.8 Real-time Implementation:

The final ensemble learning model was integrated into the "CommentSecure" system for real-time YouTube comment moderation. The system continuously monitors and filters incoming comments, leveraging the trained classifiers to swiftly identify and mitigate spam, profanity, and malicious content.

5.1.9 Ethical Considerations:

Throughout the methodology, ethical considerations were prioritized, and steps were taken to ensure the responsible use of the models. Bias detection and mitigation strategies were implemented to prevent the algorithms from exhibiting discriminatory behavior.

This detailed methodology aimed to provide a transparent and replicable approach to evaluating and implementing comment moderation techniques on the YouTube platform. The combination of rigorous model selection, ensemble learning, and ethical considerations contributes to the robustness and reliability of the proposed system, "CommentSecure."

5.2 Algorithms

To enhance the effectiveness of spam detection in YouTube videos, we employ a combination of algorithms, namely KNN , SVM and Naïve Bayes of these algorithms, we strive to achieve improved efficiency and accuracy in classifying comments, effectively differentiating between spam and legitimate content.

5.2.1 Naive Bayes

Naive Bayes, a widely adopted and straightforward machine learning algorithm, derives its foundation from Bayes' theorem, a probabilistic principle named after the esteemed statistician Thomas Bayes. This algorithm finds extensive application in classification tasks, especially in domains such as natural language processing and spam filtering. The term "Naive" in its nomenclature stems from the assumption of independence among features, simplifying computations, albeit with the acknowledgment that this assumption may not consistently align with real-world.

Here are the key details about the Naive Bayes algorithm:

Probabilistic Model:

Naive Bayes operates on the principles of probability. It calculates the probability of a given data point belonging to a particular class based on its features.

Assumption of Independence:

The "naive" assumption in Naive Bayes is that the features used to describe an observation are conditionally independent, given the class label. This simplifies the calculation and reduces the number of parameters to estimate.

Training Process:

During the training phase, the algorithm estimates the prior probabilities of each class and the likelihood of each feature given the class. These probabilities are then used to make predictions during the testing phase.

Classification:

To classify a new data point, the algorithm calculates the probability of each class given the observed features using Bayes' theorem. The class with the highest probability is assigned as the predicted class.

Strengths:

Naive Bayes is computationally efficient, especially for high-dimensional data. It performs well with a small amount of training data and is resistant to overfitting.

Limitations:

The independence assumption may not hold in some cases. It tends to be less accurate compared to more complex models when the data has strong dependencies among features.

5.2.2 KNN

K-Nearest Neighbors (KNN) is a machine learning algorithm classified under instance-based or lazy learning, commonly employed for both classification and regression tasks. The fundamental concept behind KNN lies in the assumption that similar instances within a feature space are likely to share similar labels or outcomes.

This algorithm relies on a distance metric, often the Euclidean distance, to measure the similarity between data points. The parameter 'k' in KNN represents the number of nearest neighbors considered when making predictions. A smaller 'k' enhances sensitivity to local variations, while a larger 'k' results in a smoother decision boundary.

For classification tasks, KNN determines the majority class among the 'k' nearest neighbors to predict the class. In regression tasks, the algorithm calculates the average or other measures of the 'k' nearest neighbors' values for prediction.

While KNN offers simplicity and adaptability, careful consideration is required when selecting the optimal 'k' value and the appropriate distance metric, as these factors significantly impact the algorithm's performance

5.2.3 SVM

The Support Vector Machine stands as a robust machine learning classifier utilized for filtering spam comments on platforms like YouTube. Here are the fundamental aspects associated with leveraging SVM for this specific application

Feature Representation:

Comments on YouTube can be represented as feature vectors, where each feature corresponds to different aspects of the comment, SVM leverages these feature vectors to learn the characteristics that differentiate spam from legitimate comments.

Training Process:

During the training phase, SVM learns from labeled data, adjusting its parameters to find the optimal hyperplane that maximizes the margin between different classes. The training data typically consists of examples of both spam and non-spam comments.

Handling Imbalanced Data:

SVM provides flexibility in handling imbalanced datasets, where the number of spam comments may be significantly different from non-spam comments. Techniques like adjusting class weights can be employed to address this imbalance.

Tuning Parameters:

SVM has parameters, such as the regularization parameter (C) and the choice of kernel, that may require tuning to achieve optimal performance. Grid search or other optimization techniques can be applied to find the best combination of parameters.

Real-time Classification:

Once trained, SVM can efficiently classify new comments in real-time, making it suitable for dynamic platforms like YouTube where comments are continuously posted.

Chapter 6

Conclusion

In conclusion, we have determined to utilize a fusion of the Naive Bayes Support Vector Machine algorithm and , K-nearest neighbors through an ensemble method to discern between spam and ham comments in YouTube videos. This strategic amalgamation aims to address shortcomings observed in existing systems, fostering the anticipation of enhanced efficiency and accuracy in distinguishing whether a given comment is spam or ham.

References

[1] G. Mishne, D. Carmel and R. Lempel, "Blocking Blog Spam with Language Model Disagreement," AIRWeb vol. 5, pp. 1-6, 2005.

[2] C. Smitashree and J. G. Breslin, "User Sentiment Detection: a YouTube Use Case," The 21st National Conference on Artificial Intelligence and Cognitive Science, 2010.

[3] A. Sureka, "Mining user comment activity for detecting forum spammers in YouTube," arXiv preprint arXiv:1103.5044, 2011.

[4] P. Heymann, G. Koutrika, and H. Garcia-Molina, "Fighting spam on social web sites: A survey of approaches and future challenges", IEEE Internet Computing, 11(6):36–45,2007