Implementation of open source (Smishing Detection) AI

Summary (Harrison)

This report will discuss the implementation of open-source smishing detection, the purpose of this report is to express the steps needed for the implementation of the smishing detection. There will be various academic journals and sources utilised to help reach the desired outcome of the report. The report will follow a structure that includes an introduction, a purpose, literature review, case studies, discussion and a conclusion. The conclusion will explore the final findings of implementing an open source (smishing detection) AI.

Introduction (Harrison)

Background

It has been explored by the source of Misha (2021, pg.14) who indicates that over the last ten years there has been a huge significant increase in the number of fraudulent messages and attempts to deceive victims has increased by almost 40 percent according to Salloum (2022, pg.6020), this statistic highlights this is a serious problem that is only continuing to grow and become more severe. A smishing attack and detection are defined as the fraudulent practice of sending texts or email (or any other form of message) pretending to be sent from a trusted source, in the hope of gaining personal information or money.

Purpose of the Research

The goal of this research report is to discover where open- source AI has been implemented effectively for smishing detection through case studies and evidence. From this, using those case studies and evidence to have a comprehensive understanding of implementing open source (smishing) AI.

Literature Review (Harrison)

Overview of Smishing

The definition of smishing is referred to in the source of 'International Journal of Advanced Computer Science' and Samad (2023, pg.14) in which it states that smishing is an engineering attack that will implement fake mobile texts that are aimed to trick victims into opening and downloading files, which will contain malware and infect the victim's device. Smishing exploits the trust that users place in text messages received from seemingly legitimate sources. These fraudulent messages often impersonate reputable institutions, such as banks or government agencies, to deceive recipients into disclosing sensitive personal information or clicking on malicious links. The impact of smishing can be severe, leading to unauthorised access to personal accounts, financial loss, and identity theft. The rise of smishing is closely linked to the increasing reliance on mobile devices for communication and the proliferation of SMS as a vector for cyber-attacks (Kosinski 2024).

Role of AI in Cybersecurity

Artificial intelligence (AI) is significantly enhancing cybersecurity measures, in particular through machine learning algorithms and natural language processing which is the development of algorithms to improve communication between humans and computer language. Advanced AI algorithms can be implemented to analyse vast amounts of data to identify patterns and anomalies which can be indicative of smishing attempts, this improves detection accuracy and response times. An example of this can be seen through AI examining linguistic features of text messages to differentiate between legitimate and fraudulent messages. In 2021 Ulfath et al utilised machine learning based techniques to identify smishing attacks and successfully identified patterns between smishing and legitimate SMS messages, indicating how useful AI can be in combating SMS Phishing attacks.

Case Studies/ Examples of Successful Use Cases

Case Study 1: [Barclays Bank] (Harrison)

- In this case study, Barclay's Bank a global institute was suffering from a significant number of smishing cases that was specifically targeting their customer base. The attacks would deceive the customer by sending fraudulent messages that appeared to come from Barclay's.
- The implementation that Barclay undertook numerous steps to implement smishing, it first included
- 1) Data collection like what we have highlighted in this report. 2) From that they also completed a preparation phase in processing that included tokenization and data cleaning. 3) Barclays also then added a feature extraction in which keyword analysis and N-grams analysis was implemented. 4) The next step included adding Machine Learning Models in the classification techniques and implementing deep learning models. 5) The final step was for Barclay's to implement model training and evaluation, in which handling imbalanced data and incorporating more advanced language.
- The results of this implementation included a high level of accuracy, 96% of the messages can be identified as correct and the F1- Score was approximately a 93%. After this, there was a lot of support from the customers in their feedback portals as they respected Barclay's for taking proactive steps to protect their clients.
- In my research it has been made clear that the challenges did include finding the balance for what was a fraudulent message and what was a genuine message. Additionally, the constantly evolving world of technology meant that the attacker's capabilities are only growing and forming more advanced attack styles. Resulting in the challenge for Barclay's to constantly update their system.

Case Study 2: [Detecting Smishing Attacks Using Feature Extraction and Classification Techniques] (Mitchell)

- The framework of this study uses machine learning and natural language processing techniques to detect Smishing threats. Feature extraction is performed using Term Frequency-Inverse Document Frequency, which measures the importance of words by evaluating their frequency in individual messages relative to the entire dataset, and N-grams, which capture sequences of words (unigrams are single words, bigrams are pairs of consecutive words). Feature selection is done using the Analysis of Variance test, a statistical method that identifies significant features by comparing the variance between groups. The classification of messages as phishing or legitimate is achieved using several machine learning algorithms, including Support Vector Machine, which finds the optimal hyperplane to separate classes; Random Forest, which constructs multiple decision trees for improved accuracy; XGBoost, an optimised gradient boosting algorithm; and AdaBoost, which combines weak classifiers to form a strong one. The dataset, sourced from the UCI Machine Learning Repository, contains labelled instances of legitimate and phishing SMS. The study finds that Support Vector Machine's with a radial basis function kernel achieves the highest accuracy in detecting smishing (Ulfath et al 2021).
- The features for detecting smishing in this study are derived using Term Frequency-Inverse Document Frequency and N-grams. Term Frequency-Inverse Document Frequency measures the importance of words by evaluating their frequency within individual SMS messages and across the entire dataset, highlighting significant terms. N-grams capture sequences of words (unigrams and bigrams) to understand the context within the messages. The top 10,000 features based on Term Frequency-Inverse Document Frequency scores are selected. Feature selection is then performed using the Analysis of Variance test, which identifies significant features by comparing the variance between phishing and legitimate SMS groups. Features with p-values below specified thresholds (0.05, 0.01, 0.001) are chosen for classification. The classification is carried out using machine learning algorithms including Support Vector Machine, Random Forest, XGBoost, and AdaBoost, each applied to the selected features to distinguish between phishing and legitimate messages (Ulfath et al 2021).
- The framework was evaluated using a dataset from the UCI Machine Learning Repository, split into training and testing sets with an 80:20 ratio. The performance of the classifiers was measured using tenfold cross-validation and metrics such as accuracy, precision, recall, and F1-score. The Support Vector Machine with a radial basis function kernel achieved the highest accuracy of 98.39%, outperforming other classifiers. The effectiveness of Support Vector Machine is attributed to its ability to handle high-dimensional spaces and draw clear margins between classes. This high accuracy indicates that the proposed framework is effective in detecting smishing, providing a robust tool for identifying phishing SMS messages (Ulfath et al 2021).
- One major challenge was handling the unstructured and nonlinear nature of SMS text data, which complicates the differentiation between phishing and legitimate messages. This was addressed by using feature extraction techniques such as Term Frequency-Inverse Document Frequency and N-grams, transforming the raw text into structured

data suitable for analysis. Another significant challenge was the computational expense of processing large datasets. To solve this, the study implemented the Analysis of Variance test for feature selection, which reduced the feature set to only statistically significant features, improving computational efficiency without compromising detection accuracy (Ulfath et al 2021).

Discussion (Mitchell)

Key Findings

The key findings from case study one is, the need to ensure the data collection stage is extremely important as seen in Barclay case despite completing an in-depth data collection they do still list the natural language and ability to decipher what messages are genuine and what messages are fraudulent. Additionally, another key finding from this case study (Barclay) was the feature importance and how keywords and the presence of a URL (specifically a fraudulent URL) was always an important feature.

The key findings from case study two highlight the effectiveness of using Term Frequency-Inverse Document Frequency and N-grams for feature extraction in detecting smishing attacks. Term Frequency-Inverse Document Frequency identifies the importance of terms within SMS messages, while N-grams capture contextual sequences of words, improving the accuracy of smishing detection. The Analysis of Variance test for feature selection proved valuable in reducing computational complexity by focusing on statistically significant features, which enhances the efficiency of the detection process. Additionally, the Support Vector Machine with a radial basis function kernel demonstrated strong performance, achieving high accuracy in distinguishing between phishing and legitimate messages. Integrating these techniques into the Smishing Detection app can significantly enhance its ability to identify and prevent phishing attempts.

Best Practices

- Implement thorough data collection processes to gather a diverse set of SMS messages.
- Include data preparation steps such as tokenization and data cleaning.
- Use Term Frequency-Inverse Document Frequency for assessing word importance.
- Apply N-grams to capture sequences of words and provide context.

- Utilise the Analysis of Variance for feature selection, focusing on statistically significant features.

- Deploy machine learning algorithms like Support Vector Machine with a radial basis function kernel, Random Forest, XGBoost, and AdaBoost for classification.

- Implement model training and evaluation with techniques such as tenfold cross-validation.

- Handle imbalanced data effectively to improve model performance.

- Regularly update the detection system to address evolving attack techniques.

- Optimise computational efficiency by reducing the feature set to significant features

Challenges and Solutions

Balancing Genuine and Fraudulent Messages

Distinguishing between genuine and fraudulent messages is a major challenge. To address this, advanced feature extraction techniques, such as keyword analysis and N-grams, were used. These methods help identify critical features and patterns to improve a system's ability to differentiate between legitimate and fraudulent messages.

Evolving Attack Techniques

The continuous evolution of smishing tactics requires detection systems to be regularly updated. To keep pace with new and sophisticated attack methods, it is important that detection systems undergo frequent updates, and continuous model training to maintain its effectiveness against emerging threats.

Handling Unstructured and Nonlinear Text Data

The unstructured and nonlinear nature of text data complicates analysis. To manage this, feature extraction methods like Term Frequency-Inverse Document Frequency and N-grams were applied. These techniques convert raw text into structured data, making it more suitable for analysis and improving the accuracy of message classification.

Conclusion (Mitchell)

Summary of Key Points

The research into implementing open-source AI for smishing detection has identified several key techniques and strategies. Term Frequency-Inverse Document Frequency and N-grams for feature extraction, combined with Analysis of Variance for feature selection, proved highly effective in identifying and processing terms and sequences in SMS messages. Machine learning algorithms, especially the Support Vector Machine with a radial basis function kernel, demonstrated high accuracy in differentiating phishing and legitimate SMS messages, achieving an accuracy rate of 98.39% in one study. The Barclays Bank case study highlighted the importance of thorough data collection, preparation, and advanced feature extraction methods, achieving an accuracy rate of 96% and an F1-score of 93%.

Challenges such as balancing genuine and fraudulent messages, evolving attack techniques, and handling unstructured text data can be addressed through advanced feature extraction techniques and continuous model updates. Best practices include comprehensive data collection and preparation, employing advanced feature extraction and selection methods, and using machine learning algorithms. Regular updates and model training are crucial to maintaining the system's effectiveness against new smishing tactics. Overall, the research demonstrates that open-source AI can be effectively used for Smishing Detection.

Recommendations for Smishing Detection

Optimise Support Vector Machine with Radial Basis Function Kernel

The Smishing Detection app should implement a Support Vector Machine with a radial basis function kernel. This can be done by collecting an extensive dataset of SMS messages, both legitimate and fraudulent. Then, preprocess the data with techniques like tokenization and data cleaning. Next, apply feature extraction methods such as Term Frequency-Inverse Document Frequency and N-grams to convert the text into numerical features. Finally, train the Support Vector Machine model on this processed data, fine-tuning parameters like the penalty parameter and kernel coefficient to achieve optimal performance and high accuracy in detecting smishing messages.

Improve Feature Extraction with Term Frequency-Inverse Document Frequency and N-grams

The smishing detection app should continue using and enhancing Term Frequency-Inverse Document Frequency and N-grams for feature extraction. To further improve detection accuracy, consider combining these techniques with other advanced methods, such as incorporating contextual embeddings from models like BERT, a pre-trained language model that captures deep contextual relationships in text by processing words in both directions simultaneously. These methods can capture more complex patterns within the text. By integrating Term Frequency-Inverse Document Frequency and N-grams with such techniques, Smishing Detection can gain a deeper understanding of the content and context of SMS messages, overall leading to better detection.

Implement Regular System Updates and Model Training

The smishing detection app should incorporate a system for regular updates and model retraining. To do this we can, establish a process for continuously collecting new data and feedback on detected smishing attempts. Use this new data to retrain the detection model, ensuring it adapts to evolving smishing tactics. Implement mechanisms for automated model updates and monitoring to keep the system effective against emerging threats.

References

- Kosinski S (10 June 2024) ‘What is smishing (SMS phishing)?’ , IBM, Accessed 6 August 2024.

- Mishra, S.(2021). DSmishSMS-A System to Detect Smishing SMS. *Neural Computing and Applications*. doi:https://doi.org/10.1007/s00521-021-06305-y.

- Samad, S.R.A. (2023). SmishGuard: Leveraging Machine Learning and Natural Language Processing for Smishing Detection. *International Journal of Advanced Computer Science and Applications*, [online] 14(11). doi:https://doi.org/10.14569/IJACSA.2023.0141160.

- Salloum, S. (2022). A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques. *IEEE Access*, 10, pp.65703–65727. doi:https://doi.org/10.1109/access.2022.3183083.

- Ulfath R, Sarker I, Chowdhury M, Hammoudeh M (2021) Detecting Smishing Attacks Using Feature Extraction and Classification Techniques, International Conference on Big Data, IoT, and Machine Learning, vol 95 https://doi.org/10.1007/978-981-16-6636-0_51