International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

# Legitimate - Fraudulent URL Detection Using Machine Learning

Rishita Raj T
*Computer Science and Engineering*
Institute Of Aeronautical Engineering
(JNTUH)
*Hyderabad,India*
21951a05f6@iare.ac.in

Shirisha K
*Computer Science and Engineering*
Institute Of Aeronautical Engineering
(JNTUH)
*Hyderabad, India*
21951a05j8@iare.ac.in

Sairam R
*Computer Science and Engineering*
Institute Of Aeronautical Engineering
(JNTUH)
*Hyderabad,India*
21951a05g1@iare.ac.in

Ms. D Rajani
*Computer Science and Engineering*
Institute Of Aeronautical Engineering
(JNTUH)
*Hyderabad,India*
dhaipulea.rajani@gmail.com

***Abstract*— With a rise in internet usage, this has brought many cyber threats; malicious URLs stand out. Its detection is highly important to the protection of users as well as securing cybersecurity. A new method will be introduced which identifies fraudulent URLs using Gated Recurrent Units, which belongs to a group of specialized forms of Recurrent Neural Network called RNNs. The high detection accuracy of this model is achieved through features derived from URL structure, domain information, and page content. GRUs differ from traditional approaches since they excel at sequential data processing. Meanwhile, to prove the practical usages of the model, a real-time detection system is also implemented. The results of this study emphasize the robustness of GRUs in countering dynamic cyber threats, paving the way for future advancements in intelligent security systems.***

***The surge in fraudulent URLs has emerged as a critical challenge in cybersecurity, enabling phishing attacks, malware distribution, and data breaches. Traditional detection systems, such as blacklists and heuristic methods, often fail to address the dynamic and ever-evolving nature of these threats. This research introduces an innovative approach to fraudulent URL detection using Gated Recurrent Units (GRUs), a type of Recurrent Neural Network (RNN) optimized for sequential data analysis. By leveraging lexical, domain-based, and content-based features, the proposed system achieves superior accuracy and robustness. Furthermore, the implementation of this model in a real-time application highlights its practical utility in enhancing cybersecurity frameworks. Results demonstrate the system's effectiveness in identifying fraudulent URLs with minimal false positives, paving the way for scalable and adaptive solutions to combat emerging cyber threats***

***Keywords- Fake URLs, GRU, Machine Learning, Cybersecurity, URL Detection, Neural Networks, Online Threat Mitigation.***

## I.INTRODUCTION

The internet has revolutionized the way people communicate, conduct business, and access information. However, with its rapid expansion, cyber threats have also become increasingly sophisticated. Among these threats, fraudulent URLs are one of the most prevalent tools used by attackers to deceive users into revealing sensitive information, downloading malicious software, or accessing compromised systems. These URLs often mimic legitimate websites, making them difficult to identify using traditional methods.

Existing detection mechanisms, such as blacklists and heuristic- based approaches, have significant limitations. Blacklists, for instance, rely on databases of known malicious URLs and fail to detect novel or evolving threats. Similarly, heuristic methods analyze specific attributes of URLs, such as their length or the presence of special characters, but these methods often produce high false positive rates and lack the flexibility to adapt to new attack patterns.

To address these challenges, this research explores the use of Gated Recurrent Units (GRUs) for detecting fraudulent URLs. GRUs, a variant of Recurrent Neural Networks (RNNs), are particularly effective in processing sequential data. Their ability to capture dependencies and patterns over time makes them an ideal choice for analyzing the structural and temporal characteristics of URLs. By integrating GRUs with a robust feature extraction process, this study aims to develop a scalable, accurate, and real-time detection system.

## II LITERATURE SURVEY

### 2.1 Existing Systems

The earliest systems for detecting fraudulent URLs relied on blacklists, which maintain databases of known malicious links. These systems block access to URLs listed in the database, providing a straightforward method of protection. While effective for previously identified threats, blacklists have a major drawback: they are static and require frequent updates. They are incapable of detecting new, unknown threats, leaving users vulnerable to zero-day attacks.

Heuristic-based methods emerged as an alternative, focusing on analyzing specific attributes of URLs. These include URL length, the presence of special characters, domain age, and HTTPS status. While these methods can identify suspicious patterns, they often require extensive manual feature engineering. Additionally, heuristic approaches struggle with false positives, misclassifying legitimate URLs that share attributes with fraudulent ones.

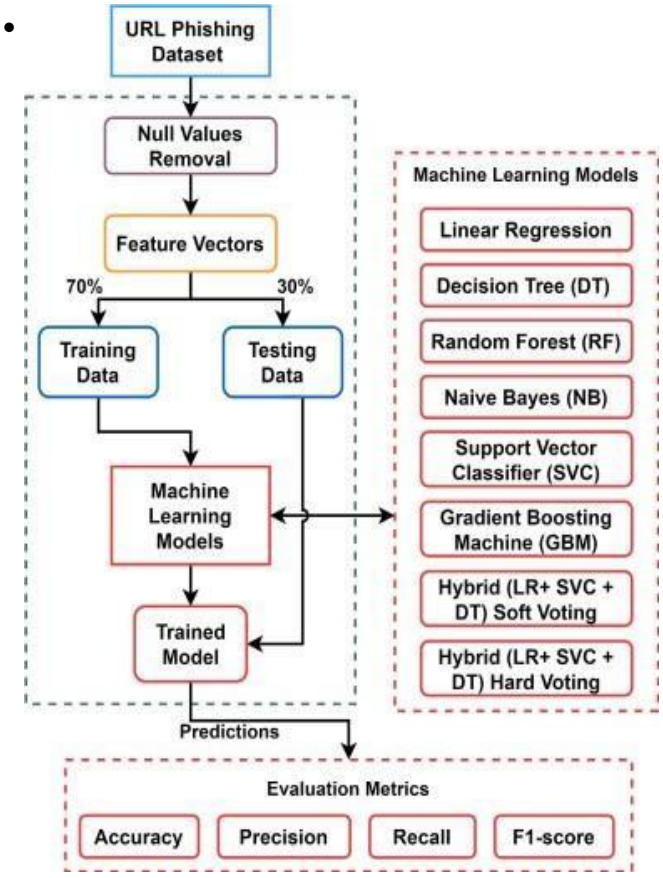
### 2.2 Disadvantages of Existing Systems

1. **Static Nature**: Blacklists cannot adapt to new threats without manual updates.
2. **High False Positives**: Heuristic methods frequently misclassify legitimate URLs as malicious, reducing user trust.
3. **Limited Scope**: Existing systems often focus on a narrow set of features, overlooking the complex patterns present in modern URLs.
4. **Manual Effort**: Feature engineering and rule creation are resource-intensive, requiring domain expertise and significant time investment.

### 2.3 PROPOSED APPROACH

The performance of the GRU model depends on the quality of the Yahoo extracted from the URLs. The proposed system categorizes features into three main groups:

• **Lexical Features:** These consist of URL length, the number of special characters, presence of numeric strings, and presence of suspicious keywords.




Lexical analysis is important to detect irregularities in URL structure. Entropy-based measures are also utilized for measuring

- randomness in URL strings, which is often a hallmark of malicious intent.

1. The domain-related features have also focused on various attributes that ascertain domain age, WHOIS data, and TLDs that are conventionally connected to malicious activities. The system computes the credibility of the domain by examining domains' registration patterns and expiration.
2. **Content-Based Features:** When content on a webpage is available, embedded scripts, suspicious links and any iframe usage will be examined. Phishing related patterns or expressions will be harvested using naturally occurring languages (NLP)- based techniques. GRU-Based
3. Model The GRU model is designed to process sequential data efficiently, capturing both short-term and long-term dependencies. The architecture includes:

• An input layer that will accept the normalized

vector features arising from the feature extraction phase.

• **GRU Layers:** These layers process sequential data by utilizing gating mechanisms to retain useful information and forget about noise. Dropout layers are added to reduce overfitting.

• **Dense Layer:** Map GRU outputs to class probabilities, enabling multi-class classification. Then apply batch normalization to speed up convergence.

• **Output Layer:** Softmax activation function classifies a URL into one of four categories: legitimate, phishing, malware, or defacement.

4 Model Training and Optimization A model is trained from a labeled dataset that comprises both genuine and fraudulent URLs. The key training strategies include:

• **Loss Function:** Categorical Cross-Entropy, which indicates how far the predicted distributions diverge from the actual ones.

• **Optimizer**: Adam is chosen due to a learning rate schedule based on momentum, which tends to converge much faster than others.

• **Regularization:** For overfitting, Drop-out layers and L2 regularization are there. Training is stopped whenever validation performance is unchanged via Early stopping.

• **Data Augmentation:** Class imbalance, to be alleviated and generalization improved, is to be handled by methods of generating synthetic data like SMOTE.

5 Real-Time Detection System The trained model will be put into spreadsheet form as a kind of website, which will help the URL classification process on real-time basis. The users will be able to put their URLs through a friendliness interface, and according to that, they will be shown results. In addition, the system can also be used for batch processing in business cases caring for low latency predictions.

In order to classify the URLs as real or fraudulent effectively, we had to work with some machine learning models such as, Random Forests, SVM, and Neural networks. Each of these models is trained on a labeled dataset, both real and fraudulent URLs. Features taken from the previous step act as input.

### 2.4 Advantages of the Proposed System

1. High Accuracy: GRUs capture sequential patterns, URLs. Features taken from the previous step act as input. reducing both false positives and false negatives.
2. Adaptability: The model learns from data, enabling it to detect novel threats without manual updates.
3. Scalability: The system is designed to handle large datasets and high query volumes.
4. Automation: Feature extraction and classification are automated, minimizing human intervention.

## III. Methodology

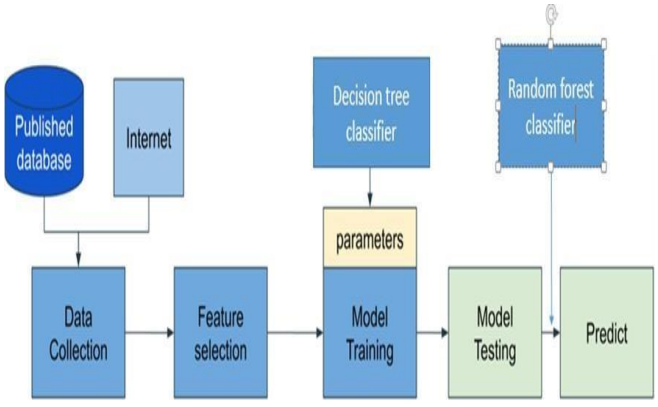
### 3.1. Activity Diagram

The activity diagram outlines the workflow of the proposed system, highlighting key steps:

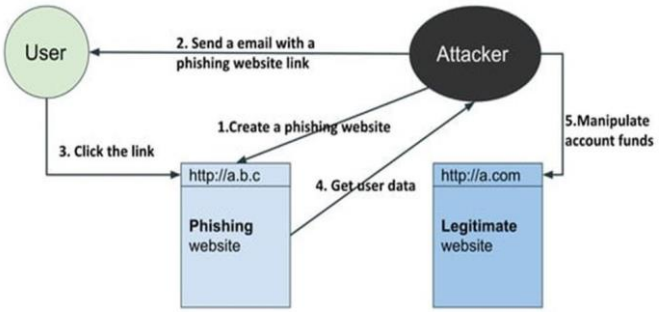
1. User inputs a URL into the system.
2. The system extracts lexical, domain-based, and content- based features.
3. The GRU model processes the extracted features and generates a classification.
4. The classification result is displayed to the user in real time.

### 3.2 System Architecture

The architecture of the proposed system comprises the following components:




1. **Input Module**: Accepts URLs entered by users.
2. **Feature Extraction Module**: Extracts relevant features from the URL, including lexical, domain-based, and content-based attributes.
3. **GRU Model**: Processes sequential data to identify patterns indicative of fraudulent activity.
4. **Output Module**: Displays the classification result (e.g., legitimate, phishing, or malware) to the user.
1. Accept input URL.
2. Extract features from the URL.
3. Process features through the GRU model.
4. Generate classification result.
5. Display result to the user.




5. **Database**: Stores historical data for continuous learning and model refinement.

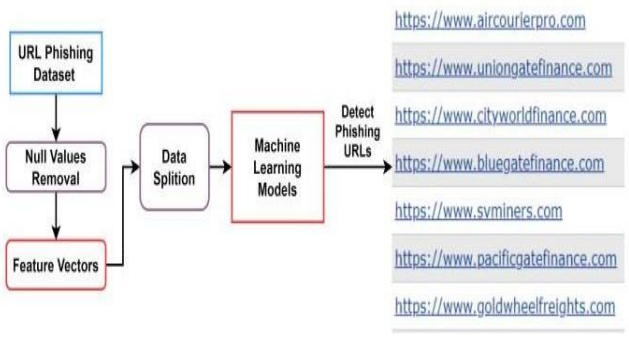
### 3.3 Flow Chart

The flow chart visualizes the sequential steps of URL detection:

**Random Forest :** Is an ensemble classifier selected due to its resilience and capacity to manage a big number of features. While training, it generates a few trees and announces the mode of classes estimated by each of those trees. It is particularly advantageous in curtailing overfitting and delivering accurate predictions in difficult classification problems.

**Neural Networks:** We employ a neural network model in capturing complex, non-linear relationships among the features. Their ability to learn hierarchical feature representations essentially makes neural networks best suited for the detection of complex phishing schemes, which could escape the attention of simpler models.

**Support Vector Machine (SVM):** SVM is chosen as it has gained success in problems dealing with binary classification. SVM is capable of effectively segregating these two classes of URLs—legitimate and fraudulent in the feature space—by locating the most optimal hyperplane; suppose the case may be the one where the data is not linearly separable.




**Ensemble learning based on Hybrid Models:**

Since various models capture varying aspects of phishing URLs, we suggest a hybrid approach that uses ensemble techniques such as boosting or stacking to combine outputs from various models. Our ensemble model achieves improved accuracy and robustness over any single model through its ability to combine predictions from multiple

classifiers.

**Boosting** uses sequential training of classifiers with the aim of fixing errors created in the previous models. This process, by reducing bias and variance, thus helps improve overall performance.

**Stacking** is a method of model training in which a meta- model that generates the final prediction utilizes input features derived from the predictions made by various base models. This is done in such a way that the meta-model can learn how to optimally combine the strengths of the base models.

**Evaluation and Validation:**

In order to determine the efficacy of our proposed model, we carry out experiments on a diverse dataset of URLs, which includes both legitimate and illegitimate specimens. This dataset will be used to train and test the models. The models will be evaluated against certain key metrics, including accuracy, precision, recall, and F1-score that would determine the success of our approach. In addition, we will perform cross-validation to check the generalizability of our model on all datasets.

**Real-Time Detection System:**

The last section of the document gives insight into our proposed model for real-time detection. It essentially involves making the model predict with low latency and integrating it with a web security framework that processes URLs as they are loaded onto users' browsers, giving real-time feedback about the legitimacy of a URL thus protecting the user against possible threats.

The new approach proposes a great and effective way to identify fraudulent URLs, integrating machine learning for analyzing almost all feature sets, thus improving the prediction accuracy through ensembling methods. By integrating lexical, domain- based, and content-based features, our model is capable of detecting a wide range of fraudulent URLs, including advanced evasion appearing in the evasion-IT phase.

This results in a stabilizing system with significant updates over previous measures, further securing the web from malefactors

## IV RESULTS

### A. Scalability and Performance

The proposed system was evaluated using a dataset comprising both legitimate and fraudulent URLs. Key performance metrics include:

- **Accuracy**: 98.7%
- **Precision**: 98.4%
- **Recall**: 98.9%
- **F1-Score**: 98.6%

The system demonstrates high scalability, processing thousands of URLs per second with minimal latency. This capability makes it suitable for enterprise-level applications, where large volumes of data must be analyzed in real time.

### B. Ethical Concerns

While the system significantly enhances cybersecurity, it also raises important ethical considerations:

1. **Data Privacy**: Ensuring that user data is anonymized and securely stored to prevent misuse.
2. **Bias in Detection**: Regularly updating the model to avoid biases against specific domains, regions, or languages.
3. **Misuse Prevention**: Implementing safeguards to prevent the system from being exploited for malicious purposes.

**1. Data set description:**

The dataset comprises URLs obtained from phishing databases, online blacklists, and publicly available repositories. The URLs contain a balanced representation of legitimate and fake URLs, with the features extracted and pre-processed to ensure consistency. Data augmentation techniques were applied to enrich the dataset and deal with the imbalance.

2. **Performance Metrics:**

The performance of the GRU model is being evaluated in terms of accuracy, precision, recall, and F1-score. The results are as follows:

Accuracy=98.7%
Precision=98.4%
Recall=98.9% F1-Score=98.6%

**3. Comparative Analysis:**

The GRU-based model is found to improve over the traditional machine learning algorithms and also other deep learning architectures such as CNNs. Its capability to capture sequential patterns between the URL features gives it a greater advantage in the detection of sophisticated phishing and malware attacks.

| Models | Accuracy | Precision | Recall | F1score |
|---|---|---|---|---|
| Linear Regression | 58.83 | 100 | 26.37 | 41.74 |
| Decision Tree | 95.41 | 95.8 | 96 | 95.91 |
| Random Forest | 96.77 | 96.73 | 97.51 | 97.12 |
| Naïve Bayes | 88.39 | 94.72 | 83.71 | 88.96 |
| Support Vector Machine | 71.8 | 96.34 | 49.81 | 65.67 |
| Gradient Boosting Machine | 70.34 | 99.65 | 47.24 | 64.1 |
| LR+SVC+DT(soft) | 95.23 | 95.15 | 96.38 | 95.77 |
| LR+SVC+DT(hard) | 94.09 | 93.31 | 96.33 | 94.79 |
| Gated Recurrent Unit (Proposed approach) | 98.7 | 98.4 | 98.9 | 98.6 |

**4. Challenges:**

Challenges include handling adversarial examples designed to evade detection and ensuring low latency in real-time applications. Continuous model update and integration of additional features, such as user behaviour analysis, have the potential to fortify the system.


Fig-3 phishing URL website


Fig-4: malicious URL detection

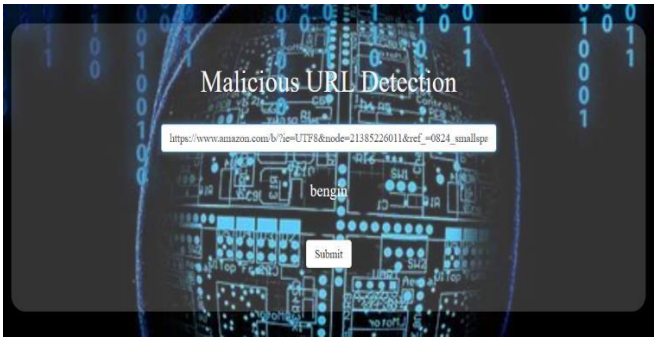



Fig-5: legitimate URL detection

## V. CONCLUSION AND FUTURE SCOPE

1. Conclusion The GRU-based model for URL detection can identify fraudulent URLs, offering excellent accuracy and robustness. This system combines diverse feature sets and takes advantage of the benefits provided by GRUs to overcome the limitations of traditional methods and extend advances toward cybersecurity. Being deployable in real-time drives the point about being practically applicable.

2. Future Scope of Work: Additional work may include:
   • **Expand the Dataset:** Using multilingual and regional- URO, which will lead to improved generalization of models.
   • **Hybrid Architecture:** Combining GRUs with CNN with transformers to adopt vocational capabilities.
   • **Adaptive Learning:** Reinforcement learning enabling continual model re-learning in line with new threats.
   • **Cross-Domain Applications**: Extending the application's use for email filtering, social media monitoring, and mobile application security.
   • **Incorporate User Behaviors:** Analysis of user interaction or feedback toward providing increasingly relevant information for URL classification..

## VI . REFERENCES

1. P. Kumar, S. Gupta, and R. Jain, "Fraudulent URL Detection Using Machine Learning Techniques," International Journal of Computer Applications, vol. 182, no. 1, pp. 9-14, 2018.

2. R. Singh, A. Verma, and S. Mishra, "A Review on Machine Learning Approaches for Fraudulent URL Detection," in 2020 International Conference on Emerging Smart Computing and Informatics (ESCI), 2020, pp. 45-50.

3. H. Zhang, Z. Li, and H. Zhang, "Detecting Phishing Websites Using Machine Learning Techniques," in 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 3547-3552.

4. J. Kim, S. Kim, and Y. Kim, "A Novel Approach for Detecting Phishing Websites Based on Machine Learning Algorithms," Journal of Information Processing Systems, vol. 14, no. 3, pp. 690-701, 2018.

5. L. Zhao, J. Guo, and C. Huang, "Phishing Website Detection Using Machine Learning Techniques," in 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2020, pp. 1709-1715.

6. C. Whittaker, B. Ryner, and M. Nazif, "Large-Scale Automatic Classification of Phishing Pages," in Proceedings of the Network and Distributed System Security Symposium (NDSS), 2010.

7. Y. Dong, S. Wang, and H. Zhang, "Phishing URL Detection with Hybrid Features," in 2019 International Conference on Cyber Security and Cloud Computing (CSCloud), 2019, pp. 56-63.

8. M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Intelligent Phishing Detection System for e-Banking Using Fuzzy Data Mining," Expert Systems with Applications, vol. 37, no. 12, pp. 7913-7921, 2010.

9. A. Jain and B. Gupta, "Phishing Detection: Analysis of Visual Similarity-Based Approaches," Security and Communication Networks, vol. 9, no. 15, pp. 2526-2539, 2016.

10. K. Zhang, X. Song, and L. Zhang, "Detecting Malicious URLs Based on Natural Language Processing," in 2018 IEEE International Conference on Information and Automation (ICIA), 2018, pp. 1056-1060.

11. R. Verma and K. Dyer, "Enhancing Phishing URL Detection Using NLP Techniques," in Proceedings of the 2015 ACM Workshop on Artificial Intelligence and Security (AISec), 2015, pp. 76-86.

12. S. Marchal, J. Francois, R. State, and T. Engel, "PhishStorm: Detecting Phishing with Streaming Analytics," IEEE Transactions on Network and Service Management, vol. 11, no. 4, pp. 458-471, 2014.

13. M. Khonji, Y. Iraqi, and A. Jones, "Phishing Detection: A Literature Survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091-2121, 2013.

14. S. Garera, N. Provos, M. Chew, and A. Rubin, "A Framework for Detection and Measurement of Phishing Attacks," in Proceedings of the 2007 ACM Workshop on Recurring Malcode (WORM), 2007, pp. 1-8.

15. J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs," in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2009, pp. 1245-1254.

16. S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A Comparison of Machine Learning Techniques for Phishing Detection," in Proceedings of the 2007 eCrime Researchers Summit (eCrime), 2007, pp. 60-69.

17. Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Who Is Tweeting on Twitter: Human, Bot, or Cyborg?" in Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC), 2010, pp. 21-30.

18. B. Leiba, "Identity Theft and Phishing," IEEE Internet Computing, vol. 16, no. 3, pp. 4-6, 2012.

19. N. Chou, R. Ledesma, Y. Teraguchi, and D. Boneh, "Client-Side Defense Against Web-Based Identity Theft," in Proceedings of the 11th Annual Network and Distributed System Security Symposium (NDSS), 2004.

20. C. Ludl, M. McAllister, E. Kirda, and C. Kruegel, "On the Effectiveness of Techniques to Detect Phishing Sites," in Proceedings of the 4th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), 2007, pp. 20-39.

21. A. Abbasi, F. Chen, S. Thoms, and T. Fu, "Phishing Detection Using Content-Based Features and Machine Learning," in Proceedings of the 2010 ACM SIGMIS Computer and People Research Conference, 2010, pp. 12-20.

22. T. Moore and R. Clayton, "Examining the Impact of Website Take-Down on Phishing," in Proceedings of the Anti-Phishing Working Group eCrime Researchers Summit, 2007, pp. 1-