Machine Learning Based Approach to Recommend MITRE ATT&CK Framework for Software Requirements and Design Specifications

AIMS (Artificial Intelligence, Machine Learning, and Security) Research Group

Nicholas Lasky and Benjamin Hallis | Department of Computer Science | Advisor: Dr. Mounika Vanamala

University of Wisconsin Eau Claire

Introduction

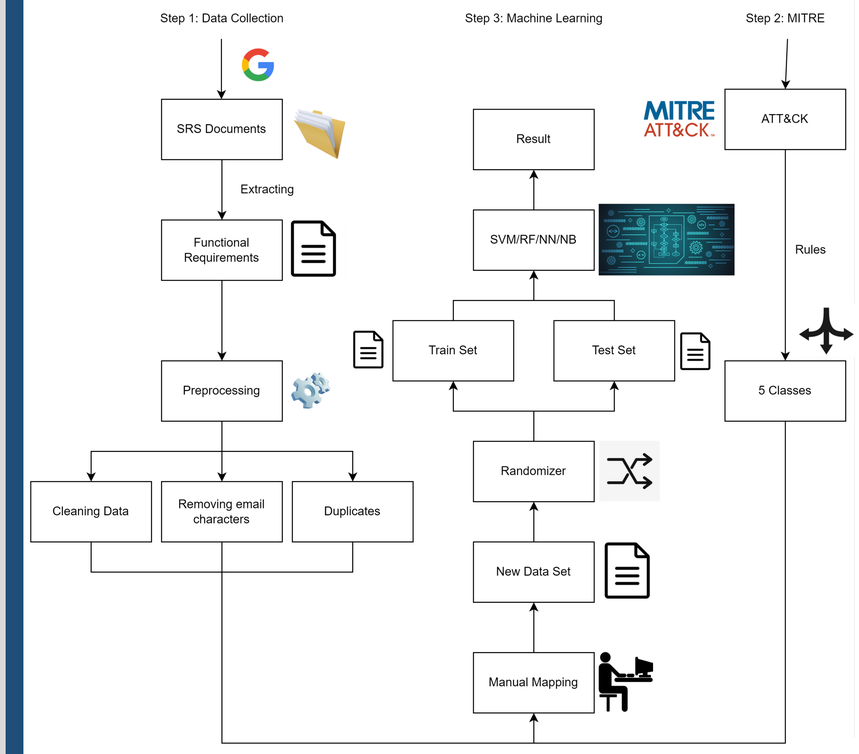
What is the problem, and why should people care?

- As cyber-attacks increase in volume and sophistication, the current state of cybersecurity solutions is inadequate
- According to Red Canary, Advanced Persistent Threat (APT) attacks have increased from approximately 500 attacks per year in 2009 to almost 2,500 APT attacks per year in 2019
- The lack of timely detection and response is mainly caused by the insufficient support of attack action correlations and prediction to allow for proactive intrusion, investigation, and mitigation
- There are lots of different ways to try and get past security, and it is difficult to create a system that protects against every single one
- With the vast amount of data available online, It is a necessary to integrate security-related activities and deliverables into each phase of the software development life cycle

Background/Our proposed research

- MITRE has developed MITRE ATT&CK, a public knowledge-based library of adversary tactics and techniques
- MITRE has been used by several organizations in their products and processes
- Although MITRE ATT&CK TTP has been used in various ways, none of the above uses help software developers utilize these techniques and tactics in developing secure software in the early stages of the SDLC, such as requirements and design
- With the large amount of information provided in MITRE, it is difficult for software developers to go through ATT&CK TTP's manually and identify those relevant to the software they are developing
- We propose to develop a recommender system that recommends tactics, techniques, and procedures relevant to the system under development based on software requirements and design documents

Flow Chart



Phases

Phase 1 – Data Collection

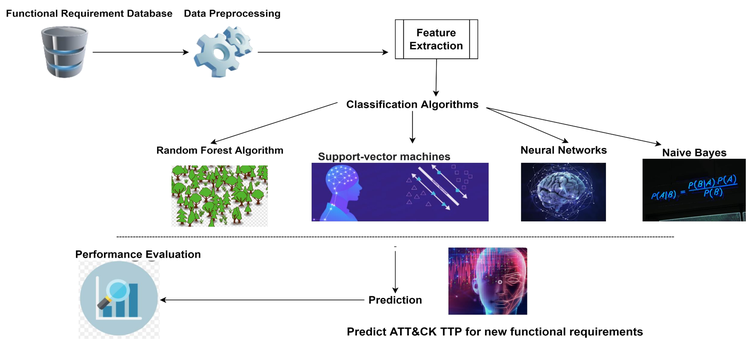
- Gathered functional SRS from online documents/PDFs using Google
- Obtained roughly 600 unique SRS and manually mapped them to the MITRE taxonomy database
- Removed any duplicates during collection

Phase 2 – MITRE Classification

Phase 3 – Manual Mapping

- 12 of the 14 MITRE branches used
- 5 total classes
- C1: initial access, execution, and impact
- C2: resource development and command and control
- C3: persistence and defense evasion
- C4: privilege escalation, credential access, and lateral movement
- C5: collection and exfiltration
- Manual mapping done in Excel and converted to a CSV file to apply machine learning algorithms

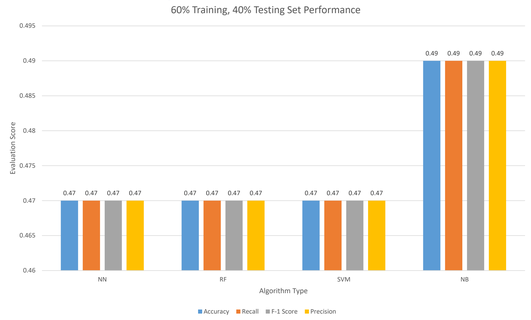
Phase 4 – ML Algorithms, Randomizer, and Data Splits



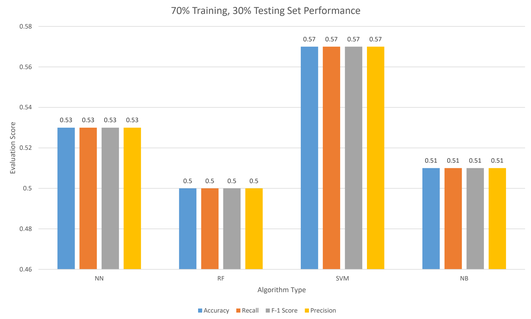
- 4 classification algorithms applied: random forest, SVM, neural network, Naïve Bayes
- Randomizer – pull random SRS into training and testing sets to provide equal distribution of requirements for both sets
- 3 data splits: 60% training, 40% testing; 70% training, 30% testing; 80% training, 20% testing
- Our goal: determine which algorithm and data split provides the best performance

Results

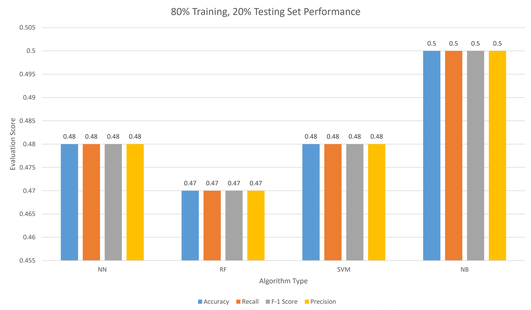
60% Training, 40% Testing



70% Training, 30% Testing



80% Training, 20% Testing



Conclusion and Future Work

- 70% training and 30% testing data split performed best
- On average, SVM performed the best, followed closely by Naïve Bayes, neural network, and random forest
- Gaps between other splits not noticeably different due to the relatively small data set
- Accuracy of each algorithm influenced by manual mapping – done by multiple people
- For the future, we need to obtain more functional requirements so that the machine would get better and better at training itself by providing it with more data
- Incorporate more algorithms to train and evaluate sets
- After consistent high accuracies, in the future, we could craft a tool for companies to put their functional requirements into automatically to show the mappings, potential threats, and mitigations to make their functional requirements even more secure

Acknowledgements

We would like to thank the Office of Research and Sponsored Programs at the University of Wisconsin – Eau Claire for funding this project. We also want to acknowledge the contributions of the AIMS research lab, including Dr. Rushit Dave, Dr. Jim Seliya, and Dr. Benjamin Fine.

We thank the Office of Research and Sponsored Programs for supporting this research, and Learning & Technology Services for printing this poster.