Research Statement

Asadullah Hill Galib

I am a new Ph.D. student (CSE Fall 2020) at Michigan State University with an interest in contributing to the domain of **applied machine learning and data mining**. I have gone through some influential faculties, courses, projects, and research those help to develop my interests. Basically, my interests grew up from my undergraduate studies. In the course of going through AI and ML, their broad-ranging applications and opportunities attracted me a lot. Thereby, I have found it as a promising domain to work on. I have hands-on research experiences in my interests.

This research statement is organized as follows: The first section discusses my works in the area of machine learning and data mining, with a focus on my master's thesis. The first section also includes current research works. The second section discusses my work in other areas in general. Both sections include plans for future research in the respective areas.

1 Research Works in Machine Learning and Data Mining

I have worked on **machine learning and data mining in cybersecurity, health-science, text analysis, and software engineering**. Currently, I am working on two research projects on machine learning and data mining in health-science and co-authorship networks.

1.1 Significant features in malware detection using machine learning (MS Thesis)

To brief about my prior research works, I would love to start with my master's thesis. In my MS thesis, I worked on Android malware detection using machine learning techniques. My thesis title is - *Significant Features Analysis For Android Malware Detection Using Machine Learning Techniques*.

Rationale behind the thesis: Typical malware detection techniques, like static analysis, dynamic analysis, hybrid analysis use full features set to classify malware. But the number of features is growing exceedingly with the growth of the Android system. It makes it complex and would misguide classifiers by over-fitting of data. So, rather than using full features set, analyzing significant features would help to reduce complexity as well as increase large scale malware detection. Several works also suggest that significant features can effectively classify malware.

Approach: The study aim at analyzing significant features for Android malware detection with consideration of maintaining performance. In doing so, Permissions, API Calls, and ensemble features are analyzed separately in this study to assess the individual impact and overall performance of each type of feature. An approach is proposed in this study to analyze and identify significant features. In the proposed approach, the features are incrementally analyzed using several feature selection techniques. Incrementally analyzed means all the techniques are used to evaluate 1 to full features set. Then, I plotted the performance metrics with respect to the number of features. According to the plot, the turning point, from which point the performance is about to constant, can be evaluated. As per the incremental feature selection, a well-suited feature selection technique is selected and a minimal number of significant features are identified. Afterward, a correlation-based feature elimination strategy is applied for further reduction of significant features.

Results and Implications: Experiments on two benchmark data sets indicate that the proposed approach can notably reduce the features set. The reduced set of significant features can perform relatively close to the full set of features in terms of accuracy, recall, f-1 performance, AUC. It also compares the performance of this approach with the related works and this work outperforms most of the works regarding malware detection rate. Furthermore, it reports the top significant features in malware detection.

Finally, it suggests that significant ensemble features are more effective than significant Permissions and API Calls. Also, significant API Calls perform better than significant Permissions. Like this study can reduce 73 API Calls to 20 API Calls, 114 permissions to 33 permissions, and 187 ensemble features to 25 features without affecting the performance notably. This study signifies that significant features would be useful in classifying Android malware effectively while maintaining detection performance. These findings would accelerate large scale malware detection with consideration of performance.

Publications: Basically, I have analyzed three parts here, significant permissions, significant API Calls, significant ensemble features. Among them, significant ensemble features analysis has the most potential, as it reduced the features set most efficiently, 187 to 25 only. I have **one conference publication** on significant API Calls analysis in **SEKE 2020** (The 32nd International Conference on

Software Engineering Knowledge Engineering, 2020) [1]. And, I have a plan to submit the other two parts of my thesis for publication in December.

Also, I have **one journal** [2] and **one conference publication** [3] on malware analysis. Basically, at the beginning of my thesis, I have scrutinized more than eighty related research papers. According to that, I have two systematic literature reviews, one is the extended version of the other.

Future Directions: In future work, the approach will be evaluated using different and large data sets. That will omit biases arisen from data sets. Apart from the Permissions and API Calls, other static and dynamics features will be incorporated for better performance. Features like app metadata, intents, system call, etc. will be used in future work. Besides, Reinforcement learning and Deep learning-based feature selection techniques can be used for reducing complexity and discernible feature exploration.

1.2 Predicting GitHub Issues lifetime using text analysis and machine learning:

The research project is about software analytics and text analysis. There, machine learning and topic modeling (text analysis) are incorporated in predicting the lifetime of GitHub Issues.

Topic modeling is the process of extracting keywords from documents to characterize and distinguish them from other documents. It is a process applied to summarize, compare, and analyze a large corpus of text. In the software engineering domain, it has been applied to mine repositories and extracts valuable insight into the important properties and aspects of the project and its developers. One such important aspect of current project management efforts is the prediction of issue lifetime. This study conducts topic modeling on GitHub issues to observe patterns in the extracted topics and their performance as a feature for predicting the lifetime of issues. It is observed that issues from a large collection of projects can yield distinguishable and comprehensible topics. In terms of predictive performance, the prediction model with topic modeling performs better than the previous approach, with a high increase in precision and f1- measure. Evaluating these findings helps establish topic modeling as a viable feature in issue-based software development processes.

Future Directions: With its promising results, this research project can be expanded upon. Firstly, the breadth of the cross-project analysis will be increased using more projects. Secondly, the implementation of LDA will be expanded, by considering its non-deterministic properties. Lastly, the intrinsic significance of individual topics in the the predictive model will be further inferred with the inclusion of empirical studies with GitHub issues.

1.3 Child mortality classification with pre-birth factors using machine learning techniques

In my undergrad program, there are not enough opportunities to conduct research. Despite that, I have worked on two research projects. One of them is *The Influences of Pre-birth Factors in Early Assessment of Child Mortality using Machine Learning Techniques*. It is the very first research project of mine.

The early assessment of patterns and trends in causes of child mortality help decision-makers assess needs, prioritize interventions, and monitor progress. Post-birth factors of the child, such as real-time clinical data, health data of the child, etc. are frequently used in child mortality studies. However, in the early assessment of child mortality, pre-birth factors would be more practical and beneficial than the post-birth factors.

That research project aims at incorporating pre-birth factors, such as birth history, maternal history, reproduction history, socioeconomic condition, etc. for classifying child mortality. To assess the relative importance of the features, Information Gain (IG) is employed. For classifying child mortality, four machine learning algorithms are evaluated. Results show that the classification achieved an AUC score of 0.947 in classifying child mortality which outperformed the clinical standards. In terms of accuracy, precision, recall, and f-1 score, the results are also notable and uniform. In developing countries like Bangladesh, the early assessment of child mortality using pre-birth factors would be effective and feasible as it avoids the uncertainty of the post-birth factors.

Future Directions: Future works will involve the study of other decisive factors, such as psychological factors, social factors, demographic factors, etc. for child mortality classification. Also, incorporating physical and health information of children in classifying child mortality will be performed in the future.

1.4 Analyzing co-authorship network and top authors using machine learning (current work)

In this research project, I have to extract a network of co-authorship relations between researchers identified to be among the top researchers in machine learning (i.e., those in the top-300 highest h-index according to Google scholar and have collaborated at least once with each other). Basically, I am experimenting with eigenvectors and page rank algorithm for centrality measures which might direct to the top k authors.

1.5 Heart Disease Prediction and Factors Analysis (current work)

This project aims at predicting heart disease effectively with consideration of performance measures and significant factors/attributes analysis. So, it addresses two aspects: how well can we predict heart disease and how the factors influence heart disease prediction. Several machine learning models and data mining techniques will be employed here for prediction and feature analysis. UCI repository: Cleveland database will be used here.

2 Research Works in Other Fields

Apart from the machine learning and data mining field, I have worked on software metrics, audio privacy, software maintenance, and evolution.

2.1 GodExpo: an automated god structure detection tool for Golang

This research project aims at detecting God Class for Golang. God Class is a class that threatens maintainability and understand-ability of code by performing most of the work alone. Various tools exist that can detect God Class of Java or C++ programs, however, there is no existing tool for detecting God Class(Structure) in Golang. This research project presents a tool entitled GodExpo to detect God Structures in Golang programs by calculating metrics namely Weighted Method Count, Tight Class Cohesion, and Access to Foreign Data. In addition, GodExpo can provide a version-wise result to observe the evolution of God structures. To evaluate GodExpo, an experiment has been conducted on several versions of two open-source Golang projects and the tool successfully found God structures in all versions of those projects. I have **one publication** regarding this research project [4]

2.2 ProtectMe: An Approach to Provide Audio Privacy in Real-time

This research project ensures audio communication with privacy in real-time using a modification algorithm. Manuscript in Preparation. In this project, an approach has been developed to provide the participants with audio communication full audio privacy in real-time. It will completely hide the vocal identity without compromising any information of the audio stream. This approach does not involve any extra hardware mechanism or data analysis to provide the service. In either way, it is very much a user-friendly and inexpensive solution for the users to preserve their security in audio conversation.

2.3 Optimizing Search Space in Code Smells Detection using a Novel Metric

This research project tries to reduce search space in code smells detection using a novel metric called - NCPC, while maintaining the performance of code smells detection. Manuscript in Preparation. Typical code smell detection approaches search code smell in all source files, and this process continues in multiple phases of the development life cycle. That process may computationally complex for real-life large-scale projects due to the vast size of the search space. In this study, a simple search space reduction approach is proposed for code smell detection based on a novel software evolution metric of change history information. The proposed approach is evaluated on 11 popular and large-scale projects from GitHub using a code smells dataset of four code smells. The results have shown that the proposed metric significantly reduces the search space while detecting a sound percentage of the actual code smell. It is also analyzed that this approach performs considerably better in detecting Blob, Feature Envy, and Divergent Change, while depicting relatively poor performance for Parallel Inheritance. In the future, other common code smells and more large-scale projects will be analyzed using this approach.

Reference (My Publications)

[1] **Galib, A. H.**, Hossain, B. M. (2020, July). Significant API Calls in Android Malware Detection (Using Feature Selection Techniques and Correlation Based Feature Elimination). In Proceedings of the 32nd International Conference on Software Engineering Knowledge Engineering (pp. 566-571).

[2] **Galib, A. H.**, Hossain, B. M. (2020). A Review on Hybrid Analysis using Machine Learning for Android Malware Detection. Dhaka University Journal of Applied Science Engineering (DUJASE), Volume 5(1 & 2). Manuscript in press.

[3] **Galib, A. H.**, Hossain, B. M. (2019, December). A Systematic Review on Hybrid Analysis using Machine Learning for Android Malware Detection. International Conference on Innovation in Engineering and Technology (ICIET) 2019.

[4] Yasir, R. M., Asad, M., **Galib, A. H.**, Ganguly, K. K., Siddik, M. S. (2019, May). GodExpo: an automated god structure detection tool for Golang. In 2019 IEEE/ACM 3rd International Workshop on Refactoring (IWoR) (pp. 47-50). IEEE.