Detection of Energy Consumption and Performance of Third-party Applications using Machine Learning and Artificial Intelligence

22_23-J 87

Project Proposal Report

Rahman S.H.

IT19189086

B.Sc. (Hons) Degree in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

June 2022

DECLARATION

I declare that this is my own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
Rahman S.H.	IT19189086	

The supervisor/s should certify the proposal report with the following declaration.
The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Date: 13/10/2022

Supervisor: Ms. Sanjeevi Chandrasiri (See Appendix for Acknowledgment)

Co-Supervisor: Ms. Madhuka Nadeeshani (See Appendix for Acknowledgment)

ABSTRACT

Massive amounts of data are transferred from one party to another in the modern world. Every day, technology evolves at a rapid pace, from the end-user to the developer. As 'good' technology evolves, so does the 'bad' side, which could be a critical factor in terms of security and efficiency. As a result, a user is encouraged to use the technology before dismissing it, and regrets accompany the first attempts. As a result, an anomaly detection and prevention framework and application could prevent third-party software from causing any harm to the user's device before it is installed. This application/framework will provide information about the third-party application and detect its security measures, potential harm to the device in terms of energy consumption, and whether the specifications match. This brief overview will introduce a Non-Specific Technology Risk Calculator, a tool that allows a user to detect a third-party application's security performance and energy consumption prior to installation using Artificial Intelligence, with specific parameters and standards such as machine learning. The mechanism will also provide the user with an in-depth analysis of the application prior to installation, as well as the option to understand the current technological process in the detection of energy consumption and the performance of the third-party application from the user's perspective.

Keywords: Artificial Intelligence, Compatibility, NSTRC, Performance, User Specifications

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

NSTRC	Non-Specific Technology Risk Calculator
AI	Artificial Intelligence

1 Introduction

1.1 Background

In the evolving nature of technology, it is evident that human beings move with technology as soon as it adapts. With the evolution of technology comes the convenience of work and usage of technology to automate tasks, set reminders, collect invoices, track assets, price comparisons, and more. As a user, you would not have to waste hours doing simple financial activities now that technology is available. You can pay your bills instantly with a few clicks.

When it comes to applications, users have a tendency to select the correct application needed for their specific purposes, and the process of the choice simply depends on many factors. Application performance monitoring is one of the most important tools used today to ensure that business applications remain operational. It ensures that your apps are available when users need them, whether they are desktop-based or web-based. When a user decides to install an application, they are prone to look at the risks and problems that may occur after installation. A third-party application might possess too many advertisements that would decrease the user's experience, or an application might have potential security and performance risks that could use up and make the user's phone vulnerable to attacks. When it comes to a software engineer, having to switch to a different language might take ages to set up on their PC due to performance decrement and energy consumption issues. As a result, the software engineer would either assume that their laptop has a malfunction or the developing language has a particular glitch in the runtime environment. Hence, time consumption and inefficiencies may occur.

With the rising challenges faced above, Artificial Intelligence can make a key impact to address these issues. AI technology has emerged as one of the most important tools for achieving the goal. However, due to the dynamic nature and variation in real-world problems and data, developing an effective AI model is a difficult task. Artificial intelligence (AI) is a broad field of computer science concerned with creating intelligent machines capable of performing tasks that normally require human intelligence. From a philosophical standpoint, AI has the potential to help people live more meaningful lives without having to work as hard, as well as manage the

vast network of interconnected individuals, businesses, states, and nations in a way that benefits everyone.

Based on the issues raised above, in this paper, a deep comprehensive and analytical view of detection of performance and energy consumption of third-party applications using Machine Learning and Artificial Intelligence, specifically dealing with detection of User Device performance facts and Third-Party Application Performance and Compatibility. Thus, the primary goal is to explain the fundamentals of numerous AI techniques and their application to the progress of computing and decision-making in order to meet the performance and compatibility detection requirements in third-party applications. First, a literature survey of previous research by other researchers is presented in Section 1.1. Research Gaps and the Research problem is specified in Sections 1.3 and 1.4. The objectives of the proposed solution are presented in Section 2, and the methodology of the proposed solution is specified in Section 3.

1.2 Literature survey

Detection of the User Device performance facts and Third-Party Application Performance and Compatibility can be measured in several ways. Several studies utilising various types of methodologies have documented the various ways.

Among previous proposal for Detecting Anomalous Energy Consumption in Android Applications, Marco Couto, Tiago Carção, Jácome Cunha and João Paulo Fernandes proposed a tool for detecting anomalous energy emitting from Android applications which could cause harm to a user's device [1]. As part of a research project that supports various devices, a dynamically calibrated model for energy consumption for the Android ecosystem was initiated. To monitor application execution, the model is used as an API: first, they instrument the application source code so that it can relate energy consumption to the application source code; second, a statistical approach based on fault-localization techniques is used to localise abnormal energy consumption in the source code.

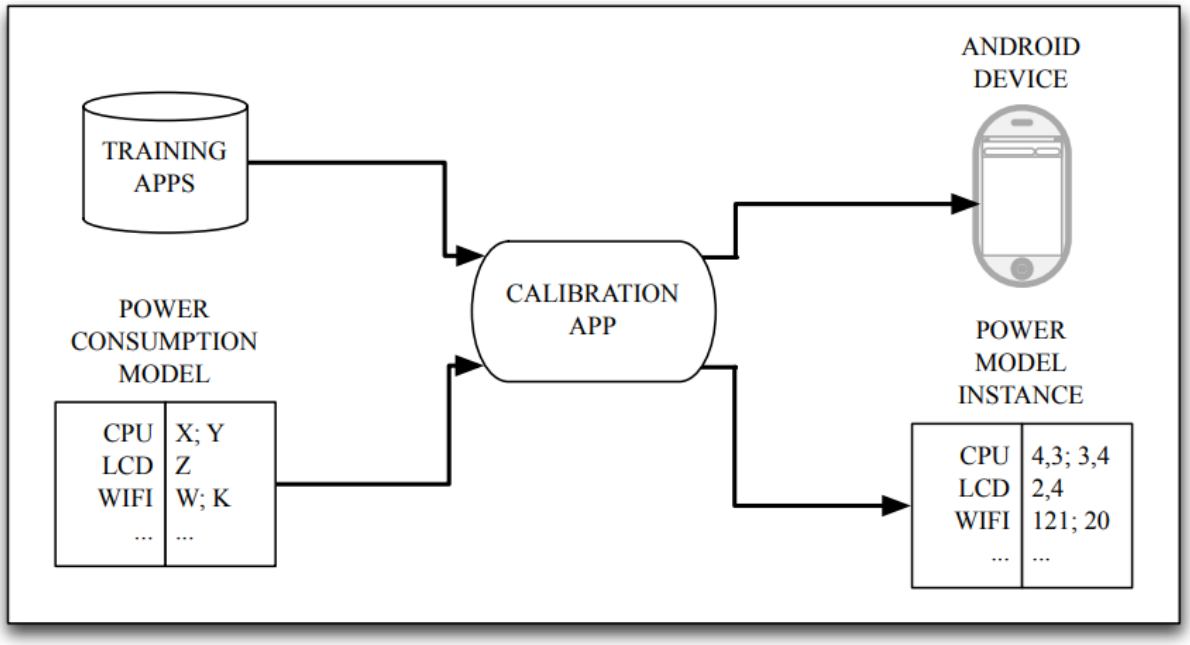


Figure 1.1: Architecture to dynamically calibrate a power model for an application,

In another research conducted by Yassine Himeur, Abdullah Alsalemi, Faycal Bensaali & Abbes Amira [2], they proposed a Novel Approach for Detecting Anomalous Energy Consumption Based on Micro-Moments and Deep Neural Networks. The paper introduces a novel approach to detecting abnormal energy consumption, in which power consumption observations are divided into five major classes, denoted as follows: "good usage," "turn on a device," "turn off a device," "excessive consumption," and "consumption while outside."

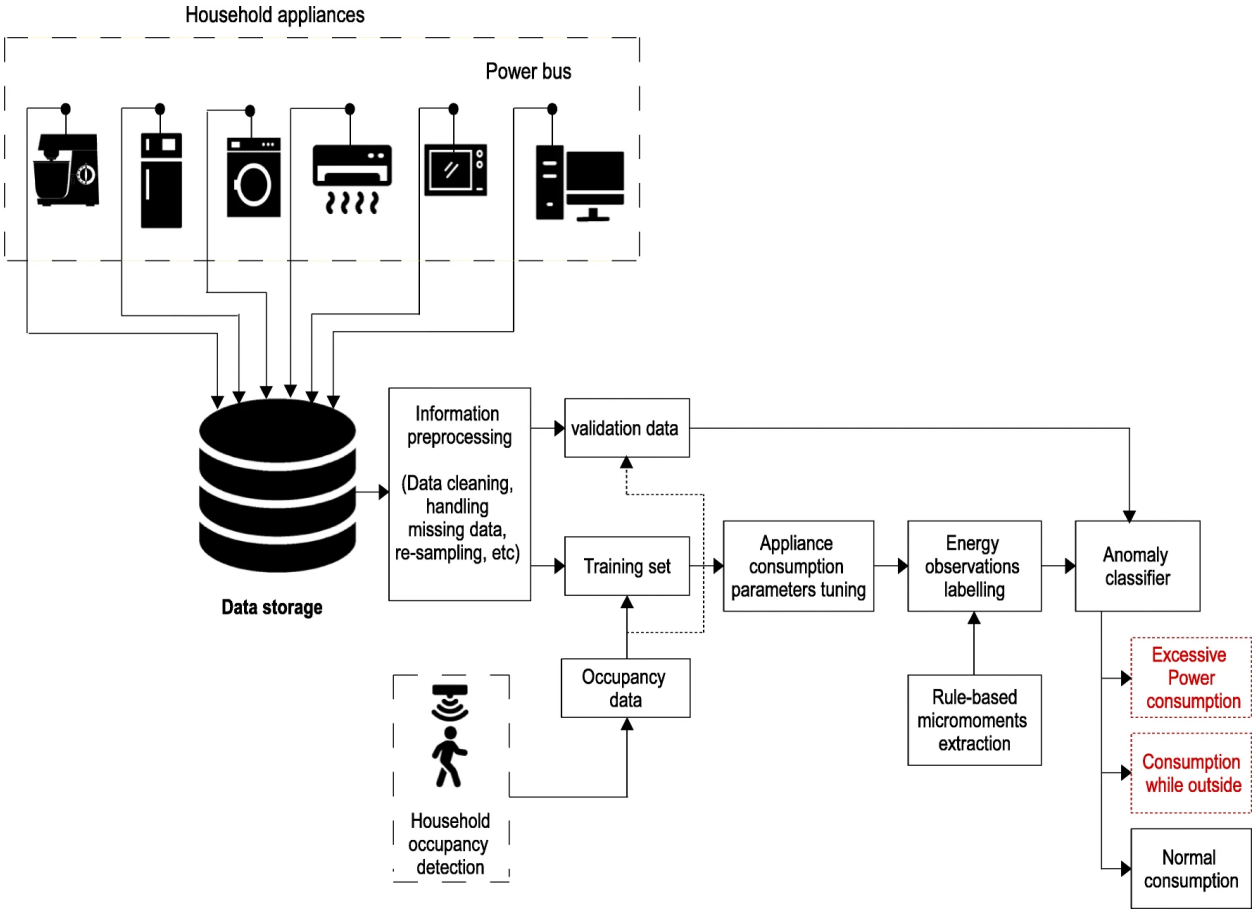


Figure 1.2: General flowchart of the proposed system for Detecting Anomalous Energy Consumption Based on Micro-Moments and Deep Neural Networks

The use of micro-moments to extract features from energy consumption signals is introduced. Building sensors and utility sub-meters provide the latter. To be more specific, features are extracted based on individual occupancy patterns in order to detect anomalous end-user consumption. Furthermore, Google uses micro-moment features to model consumer behaviour

for marketing applications. To the best of our knowledge, despite the fact that they have recently been investigated as relevant features that characterise consumer needs and thus the properties of the marketers required to be a part of, the use of micro-moments for energy applications, particularly anomalous energy consumption, has not yet been explored.

1.3 Research Gap

According to the literature review, previous researchers have been extremely helpful in determining the current problem. Both reviews identify a common problem related to the performances of an application and detect the anomalies of a specific third-party application.

The majority of the researchers have used physical devices such as sensors, IoT devices and utility submeters in order to carry out the above specifications and determine the root cause of the failing third party application's energy consumption and performance. A collection of training programmes that target all aspects of the power model. The training programmes also change the state of each component (over its entire range) while keeping the other constant. As a result, the energy consumption is measured by that specific component in that state. Previous studies do not mention any form of a user interface, where users are able to access information regarding the third-party application and determine whether to proceed with installation or not. Moreover, the above studies do not classify the importance of having the technology targeted for the convenience of a user, and the detection and collection of those data to interpret the working performance of the application. For our project, our main aim is to develop and achieve the objectives and provide a solution to the problem. We propose a 'Non-Specific Technology Risk Calculator,' a framework, desktop, and mobile application. We provide software that can help a user determine the working performance and compatibility of a third-party application, as well as provide detailed information about what the application is about and what risks/drawbacks may occur if the user proceeds with the installation.

Comparison of past research with current proposed research solutions.

	Desktop and Mobile Application.	**In-built development environment.**	**Identify performance issues and display them to users.**	**User-friendly UI**	**Report Generation**
Past Research A [1]	✘	✘	✘	✘	✘
Past Research B [2]	✘	✘	✘	✘	✘
Proposed System	✓	✓	✓	✓	✓

Table 1.1 Comparison of past research with current proposed research solutions.

1.4 Research Problem

Before installing an application, a user may question its performance and risk. When developing or building software, a software developer considers performance to be one of the most important factors. It would be more beneficial for a user, developer, or anyone else who uses the software if we could see how the third-party software we are about to install will balance the performance and capacity of our device/machine. In some cases, a user may come to regret installing a faulty third-party application, which not only reduces the performance of his/her device but also threatens the security and privacy of the user's device.

Anomaly detection suffers from a variety of issues, difficulties, and challenges that impede progress in this field. These issues are mostly domain-specific, and they are as follows:

- There is frequently no clear definition of normal and abnormal energy consumption.
- There is no clear border between normal and abnormal behaviours.
- There are no unified metrics deployed for performance assessment.
- There is a lack of ground-truth data; each group of researchers uses their own data.

As a result, reproducing the outputs of existing solutions becomes extremely difficult; as a result, it is impossible to determine which frameworks represent the state-of-the-art in various scenarios.

Other problems may be as follows in terms of performance and energy consumption after installing third-party applications on any device:

- Any PC that has third-party software installed will not function properly and will not support anything until the specific software is uninstalled.
- After installing the third-party programme, the PC's performance and tasks slow down, affecting the performance of other software.
- Malware and virus attacks originating from third-party applications may pose security and privacy risks that the user is unaware of.

- As software engineers and members of the software development team, we are stressed by the lack of performance of device machines due to disc space or limited RAM capacity, which increases time consumption for everyone on the team.

2 Objectives

2.1 Main Objective

The main goal is to create and define a framework/gateway tool that will assist both users and developers in identifying issues in third-party applications/language stacks prior to installation. The tool will not only detect the performance of a third-party application, but it will also offer suggestions and feedback, as well as test the app's ability to protect a user/developer from potential device and security issues. According to a general user's perspective, the tool ensures that the third-party application installs an application without complications or compromising the importance of safety, security, and potentially harmful distractions.

2.2 Specific Objective

The tool's first goal will be to detect the User Device Specifications, such as RAM speed, CPU speed, and so on, and then use that information to compare compatibility with the Third-Party Application. This function will be supported by the tool on both desktop and mobile devices.

The tool will first detect the user's device performance with relevant parameters (RAM speed, CPU speed, etc.) as well as the performance and security of the third-party application. The third-party data will be gathered by collecting data from the web library with the same trained machine learning model. The two specifications are then compared for compatibility, and the user is shown statistics/recommendations. The user now has the option of proceeding with installation after reviewing the learned specifications, requirements, and compatibility. This is done to warn the user about the third-party application before installing it and to allow the user to decide whether the third-party application is safe to install on his/her device.

The detection of the performance and energy consumption for the tool will have specific objectives summarised as follows:

- A user, upon his/her choice, will proceed to install a third-party application from the Application Store (Google Play Store for Android and App Store for iOS). The tool (application which will be preinstalled) will be running in the background.

- Minutes prior to installation, a warning message will be issued along with a full report of the specifications, for both desktop and mobile, initiating the process and providing a full report on the harm and damages the application can cause to the user's device.
- The tool will run a deep scanning of the machine (Desktop/Mobile) and obtain its specifications (RAM capacity, processing speed etc.) and display it to the user.
- The tool will prompt the user to choose between proceeding with the application or providing suggestions for options.

3 Methodology

3.1 Requirements Gathering

Before we begin the implementation phase of our research project, we must gather requirements that are relevant to achieving our proposed system goal. There are a few ways to collect requirements.

- Refer relevant research papers.
- Complete an analysis of the potential harm it could cause to any user.
- Conduct a survey to identify the common issues related to downloading third-party applications, for both ordinary users and developers.

Referring to relevant research articles can help us identify our research topic area and advise us on how to proceed successfully with our research. Studying the procedure related to third-party applications is extremely beneficial in achieving our research goal and gaining a clear understanding of workflow. Conducting informal interviews with relevant users and developers will provide us with identifying the root cause of the problem and help to reach new trending information. It will also help us develop our research project further.

3.2 System Diagram

The proposed system will be capable of identifying potential harm and risks of installing a third-party application or developing language by first detecting any key performance issues and be able to obtain data to be stored in real time. The functionalities can be divided as follows:

1. Third-party application installation and detection of performance and compatibility check.
2. Identify application parameters and compare compatibility with device's working performance.
3. Obtain user device data specifications and store them in the system database.
4. Compare specifications and generate an overall report of the data found.

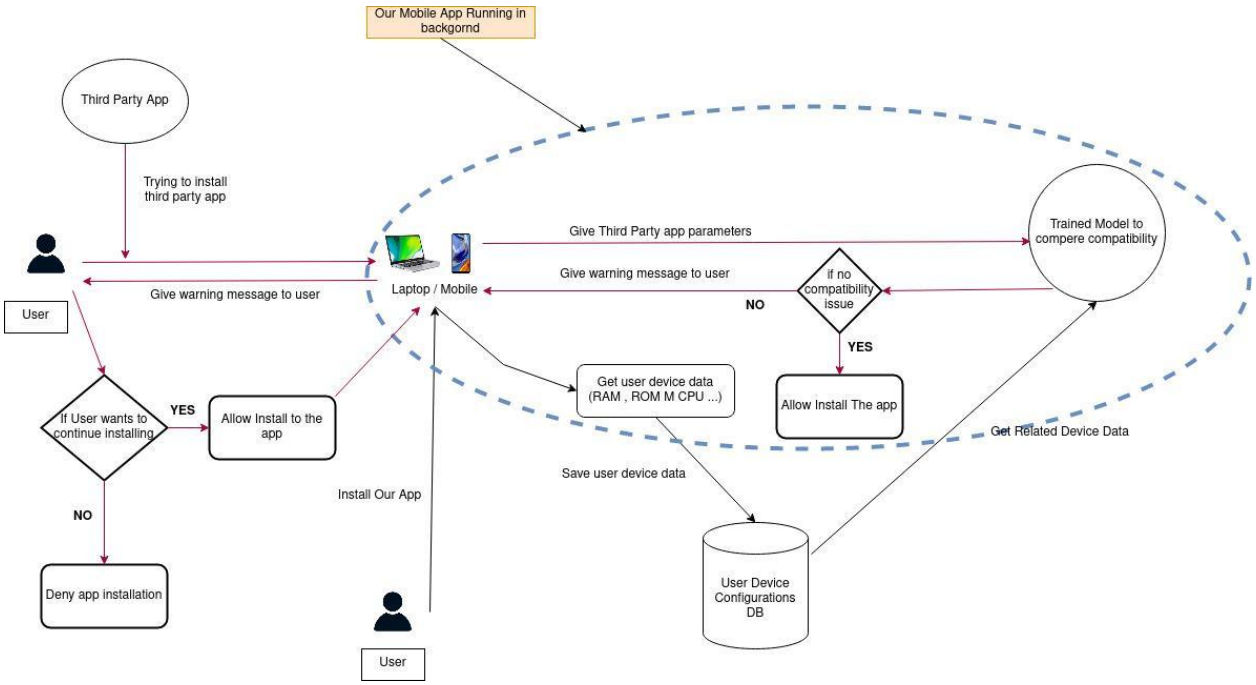


Figure 3.1 Overall system diagram for the detection of a third-party application for a performance and compatibility check.

3.2.1 Third-party application installation and detection of performance and compatibility check.

First, the user upon his/her choice will install a particular third party application. The desktop or mobile tool (NSTRC) will be running in the background, and will automatically detect that an application is being installed into the device. The main task is to make the detector compatible with all types of Operating systems (Android, iOS, etc.) that will work for both mobile and desktop devices.

3.2.2 Identify application parameters and compare compatibility with device's working performance.

Once the tool is running in the background, it identifies the parameters of the third-party application. The specifications obtained from the tool will be stored in the database for analysis and future references. The tool will use specific algorithms derived using Artificial Intelligence (AI) to detect the specifications and make it ready for a comparison test. Specification of the third-party application that can be detected by the tool are as follows:

- Webpage Weight.
- Number of HTTP requests.
- Webpage Load time.
- Plugins.
- Render-blocking JavaScript and CSS.
- Under-optimised images.
- Poorly constructed responsive design.
- Overuse of advertisements.
- Server setup.

3.2.3 Obtain user device data specifications and store them in the system database.

The tool is now ready to make use of the above mentioned statistical information related to the device data and the third-party application, and hence, complete the comparison using Artificial Intelligence, with the help of complex algorithms. Precise rule-based procedures, known as algorithms, that the tool can follow step by step to decide how to respond intelligently. The information obtained will now be stored in the user device configurations database, and the statistical information will be provided to detect and compare the harms and risks the third-party application could cause if installation proceeds.

3.2.4 Compare specifications and generate an overall report of the data found.

A model, as shown in the system diagram will be trained to compare specifications and provide an overall report display to the user. The detector will not only detect the specifications but also retrieve information from a web app portal to obtain specifications from similar third-party applications and suggest them to the user.

3.1 Implementation & Technologies

The implementation of this proposed system is currently scheduled to use current trending technologies. The implementation of our proposed system will be divided into smaller sections, each of which will be developed separately. Once the smaller sections have been completed, the system will proceed to the testing environment process.

For the detection of specification for performance and compatibility check, Artificial Intelligence will be used as a mechanism to perform the detection process with calculated algorithms fit for its purpose. Python and Pycharm will be used to create algorithms using Machine and Deep Learning, hence to train a pre-model for user device configurations.

GitLab will be used as a version control and collaboration, and our desktop/mobile application will consist of the backend (Python and Pycharm) and the frontend interface (ReactJS). The database will be Mongo DB which is an ideal storage medium for the user device and third-party application configurations, and Visual Studio Code will be used as it is the perfect IDE for web application development and cloud hosting development.

4 PROJECT REQUIREMENTS

4.1 Functional Requirements

The detection tool for performance and compatibility of third-party applications with user device is necessary to comply with all functional requirements. Functional requirements of this system consist of:

- The tool should be a free downloadable application.
- The tool should be able to detect the device specifications and store it in the local database.
- The tool should be able to obtain the information regarding the third-party application using trained models with AI to detect any anomalies present in the application.
- The tool should collect all necessary information to perform a comparison, performance and compatibility check.
- The tool should be able to provide suggestions and alternatives if the current third-party application is harmful to the user's device/machine.
- The tool should provide an option whether to proceed with the installation of the third-party application or move towards the choice of another similar application.

4.2 Non-Functional Requirements

- High Availability – The AI model should be acquired as soon as the user installs a third-party application, and the system should obtain the accuracy of the specifications of the application as soon as after all procedures have been completed.
- Efficiency of performance – The tool must be able to detect the anomalies within a short period of time.
- User-friendliness – Enhanced user interface will provide enormous support for users and developers to do important relevant duties.

5 GANTT CHART

Column1	Column	Column	Column	Column	Column	Column	Column	Column	Column1	Column1	Column1	Column1	Column12
Task	*June*	*July*	*August*	*September*	*October*	*November*	*December*	*January*	*February*	*March*	*April*	*May*	*June*
Identify a research problem													
Study about the reason for existing problem													
Gather requirements													
Feasibility studies													
Analyzing existing research according to scope													
Analyzing the requirements													
Topic evaluation and document preparation													
Helped a prepare a charter documentation													
Project proposal preparation													
Proposal presentation													
SRS document preparation													
Identify the core fucntion													
Implementation													
Progress presentation (50%)													
Database & web application development													
Testing evaluation													
Progress presentation (90%)													
System testing													
Final report presentation													
Final presentation and viva													
		Completed			Not Completed				On Progress				

Figure 5.1 Gantt Chart

6 Work Breakdown Structure

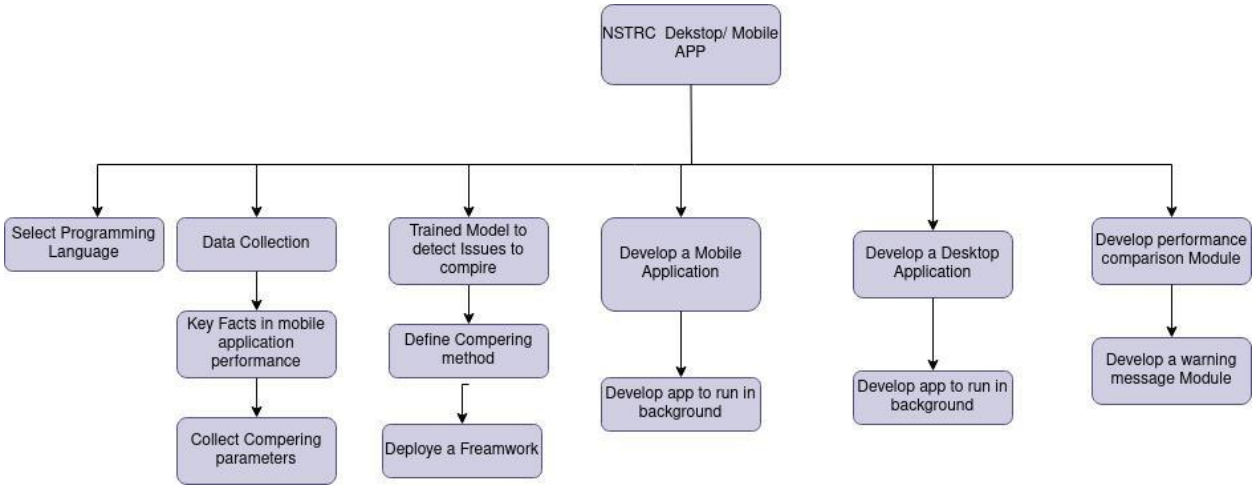


Figure 6.1 Overall Work Breakdown Structure

7 Commercialization

The NSTRC application is going to use different types of users: Mobile application users, desktop application users, and web application developers. The application is to prevent harm to the user device or the web applications deployed on servers. The application will be released version-wise in the first step of the framework part and will be released for the most popular programming language according to proven statistical information.

In the current times, many user devices are struggling because of installed and invalid compatible third-party apps. Therefore, the NSTRC application will be able to help many users to protect their devices. Currently, there are many intern and trainee developers initiating their career into the technology field. As a result, they will need to be supported to write code in the proper way for proper functioning. Our tool will be able to provide them with the necessary convenience and application in order to carry out the functionalities. When the first release is available, we will make the entire application available for free open source. As time moves forward, new releases with versions will be implemented with better features and trusted resources.

As a desktop mobile application, we need to get closer to the user in order to reach our target. GIGABIT is a short and memorable name, and we designed a logo that is also simple and recognizable. The version is also embedded into the logo to make users aware of the version they are using in, and update it whenever a new release has been made.

Figure 7.1 NSTRC Logo

8 References

[1] Couto, Marco & Carção, Tiago & Cunha, Jácome & Fernandes, João & Saraiva, João. (2014). Detecting Anomalous Energy Consumption in Android Applications. 77-91. 10.1007/978-3-319-11863-5_6.

[2] Himeur, Y., Alsalemi, A., Bensaali, F. et al. A Novel Approach for Detecting Anomalous Energy Consumption Based on Micro-Moments and Deep Neural Networks. Cogn Comput 12, 1381–1401 (2020). https://doi.org/10.1007/s12559-020-09764-y

[3] Techvera. (2021). Factors That Affect the Performance of Your Computer. [online] Available at: https://techvera.com/factors-that-affect-the-performance-of-your-computer/.

[4] Arslan, A. (2022). 6 Specs Android Power Users Must Have in Their Phones. [online] MUO. Available at: https://www.makeuseof.com/android-power-users-phone-specs/ [Accessed 12 Oct. 2022].

[5] Boucher, P. (2020). STUDY Panel for the Future of Science and Technology EPRS | European Parliamentary Research Service. [online] Available at: https://www.europarl.europa.eu/RegData/etudes/STUD/2020/641547/EPRS_STU(2020)641547_EN.pdf.

[6] www.linkedin.com. (n.d.). Major Reasons Behind Poor Performance Of An Application. [online] Available at: https://www.linkedin.com/pulse/major-reasons-behind-poor-performance-application-ambuj-tayal/ [Accessed 12 Oct. 2022].

[7] Feng, L., Xu, S., Zhang, L. et al. Anomaly detection for electricity consumption in cloud computing: framework, methods, applications, and challenges. J Wireless Com Network 2020, 194 (2020). https://doi.org/10.1186/s13638-020-01807-0

[8] Himeur, Y., Ghanem, K., Alsalemi, A., Bensaali, F. and Amira, A. (2021). Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current

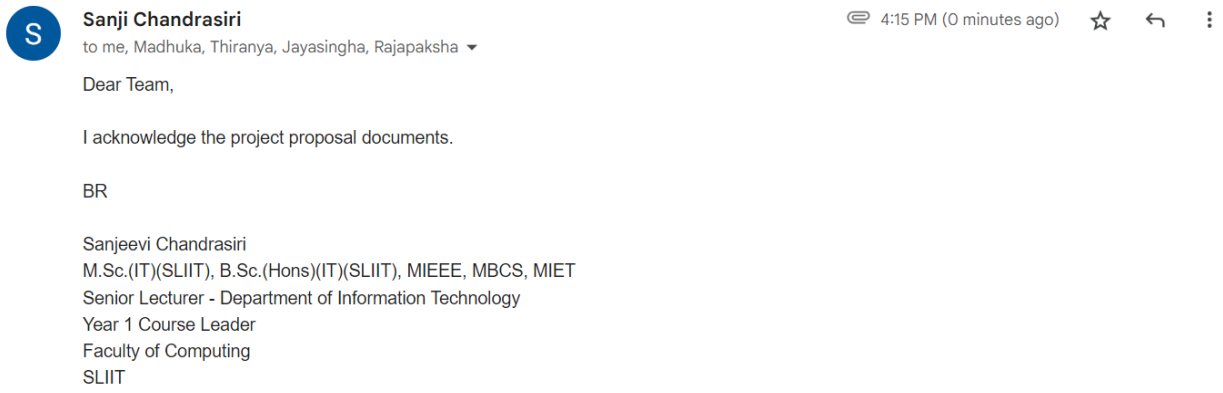
trends and new perspectives. Applied Energy, [online] 287, p.116601. doi:10.1016/j.apenergy.2021.116601.

[9] Eid, S., Makady, S. and Ismail, M. (2020). Detecting software performance problems using source code analysis techniques. Egyptian Informatics Journal, 21(4), pp.219–229. doi:10.1016/j.eij.2020.02.002.

[10] Hrci Marketing. (n.d.). Top 5 Common Performance Problems. [online] Available at: https://www.hrci.org/community/blogs-and-announcements/hr-leads-business-blog/hr-leads-business/2022/02/14/top-5-common-performance-problems [Accessed 15 Jul. 2022].

9 Appendix

Acknowledgement from Supervisor, Ms. Sanjeevi Chandrasiri



Sanji Chandrasiri
to me, Madhuka, Thiranya, Jayasingha, Rajapaksha

4:15 PM (0 minutes ago)

Dear Team,

I acknowledge the project proposal documents.

BR

Sanjeevi Chandrasiri
M.Sc.(IT)(SLIIT), B.Sc.(Hons)(IT)(SLIIT), MIEEE, MBCS, MIET
Senior Lecturer - Department of Information Technology
Year 1 Course Leader
Faculty of Computing
SLIIT

Figure 8.1 Acknowledgment from Project Supervisor

Acknowledgement from Supervisor, Ms. Madhuka Nadeeshani

Figure 8.2 Acknowledgment from Co-Supervisor

MN

Madhuka Nadeeshani <madhuka.n@sliit.lk>
To: Rahman S.H it19189086; Rajapaksha R.M it19156316; Thiranya M. A. R. it19129440 **+1 other**
Cc: Sanji Chandrasiri

Fri 10/14/2022 4:34 PM

[EXTERNAL EMAIL] ***This email has been received from an external source – please review before actioning, clicking on links, or opening attachments.***

Dear Team,

I endorse the project proposal documents.

Best Regards,
Madhuka