**Mini Project Report on**



**DEEP LEARNING-BASED MALWARE DETECTION**



**Submitted in partial fulfilment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Ayush om Srivastava**  **2017316**

***Under the Mentorship of***

***Dr. Mohammad Wazid***

**professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**July-2023**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“DEEP LEARNING-BASED MALWARE DETECTION”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Dr. Mohammad Wazid, professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Ayush om Srivastava 2017316

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **01-02** |
| Chapter 2 | Literature Survey | **02-05** |
| Chapter 3 | Methodology | **06-08** |
| Chapter 4 | Result and Discussion | **09-10** |
| Chapter 5 | Conclusion and Future Work | **11** |
|  | References |  |

**Chapter 1**

**Introduction**

In the following sections, a brief introduction and the problem statement for the work has been included.

* 1. **Introduction**

Deep learning (DL) is a representational learning method with multiple levels of representation, each of which transforms one representation level to a higher level, allowing very complex tasks to be learned. Deep learning can help solve complex problems facing the machine learning community. It has been shown to be useful in the search for patterns in high data and has many applications in government, industry, and science. Learning representation allows us to feed raw data and explore the representation needed for classification or research. Machine learning is used to improve the performance of computer systems and data processing in many fields such as medicine, science, robotics, content filtering and web recommendations, search engines, social networking and e-commerce platforms, and products such as cameras. and smartphones. However, traditional machine learning methods have some limitations with raw data.

The preeminent undermining issue is the different malware that can ambush and harm your system. Malware is one of the most genuine security perils on the Internet these days. In truth, most Web issues such as spam e-mails and refusal of benefit assaults have malware as their crucial cause. That's, computers that are compromised with malware are habitually organized together to create botnets, and various ambushes are moved utilizing these noxious, attacker-controlled frameworks. To deal with the unused malware made, present day techniques to recognize and expect any hurt caused by them.

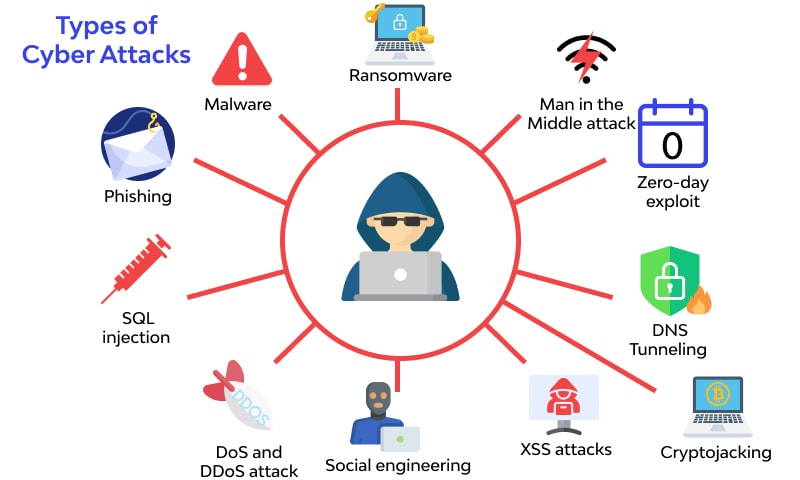
Cyber-attacks are the most urgent problem of today's technological world. The term refers to exploiting a security vulnerability in a system for malicious purposes such as hacking, modifying, or destroying the system. Malware is an example of a cyber-attack. Malware is a program or instruction designed to harm a computer, user, business, or computer system [1]. The term "malware" covers a wide range of threats, including viruses, Trojans, ransomware, spyware, adware, malware, wipers, scary software, and more. Malware, by definition, is any code that cannot run without the user's knowledge or consent.

This study specifically demonstrates that detection of computer traffic problems can improve the security of computer systems by calculating the difference between the detection results of the use of malware analysis and the integration of machine learning algorithms (Naive Byes, SVM, J48, RF and plan).

The Malware Detection module is responsible for analyzing the data it collects and trains to determine whether a particular software or network connection poses a security issue. For example, imagine a learning machine that can explain the logic behind the patterns it sees. Machine-learning-trained algorithms can use information about their performance on previous tasks to hone their prediction skills and use that information to make changes.

Globally, cybercriminals pose a serious threat to businesses, universities, governments and individuals by using malware and stealing confidential information. Every day, thousands of scammers use malware to try to break into networks, steal data or move money. Therefore, protecting the security of sensitive information has become an important issue in the scientific community. This study aims to provide a general framework for detecting malicious programs and protecting personal information from hackers using data mining and machine learning techniques. In this article, we analyze signature-based and negative-based signatures to develop effective and efficient malware classification and detection. Experiments show that the proposed method is better than its alternative.

Modern malware has become more sophisticated and creative and threatens the security of today's websites. Figure 1 shows the types of cyber-attacks in the digital world or cyberspace. Malware is software designed to harm a computer or network, such as spying on users or stealing their money. Malware attacks have become more common and now even affect IoT medical devices, environmental and business management. Modern spyware is difficult to detect because it constantly changes its code and behavior. The growth of malware renders beacon-based protection ineffective. Instead, general protection is required.



**Chapter 2**

**Literature Survey**

In this chapter some of the major existing work in these areas has been reviewed:

The proliferation of computers, smartphones, and other connected devices has made the world vulnerable to cyberattacks. In response to the growth of malware activity, many malware detection methods have emerged. Researchers use a variety of big data and machine learning techniques when trying to identify malicious code. Traditional machine learning-based malware detection requires a lot of uptimes but can identify malware that looks good. The architecture may become obsolete due to the popularity of modern machine learning algorithms such as deep learning. In this study, we explore various malware detection and classification techniques. Researchers have developed machine learning and deep learning techniques to analyze patterns with malicious intent.

Armaan (2021) explains and evaluates the accuracy of various models. No application created for a digital platform can do its job without knowledge. There are many cyber risks, so be careful to protect data. Although feature selection is difficult when creating any model, machine learning is a method that paves the way for accurate predictions. This approach requires a solution that is flexible enough to handle non-transactional data. To control and prevent future attacks, we must identify malware and establish new rules and models for the generation of malware strains, as shown in Table 1. To find patterns, IT Security professionals can use malware analysis tools. Having a system to identify malware patterns and identify vulnerabilities benefits the cybersecurity industry. This tool helps monitor security alerts and prevent malware attacks. If the malware is dangerous, we need to remove it before it can infect any further. Malware analysis has become popular because it can help organizations reduce the impact of the growth of malware threats and the progression of malware attacks.

A screenshot of a computer

Description automatically generated

Chowdhury (2018) proposed a practical malware discovery approach that employs a machine-learning classification method. We investigated whether altering many parameters might increase the exactness with which malware is classified. N-gram and API call capabilities were joined into our approach. The exploratory assessment affirmed the viability and steadfastness of our proposed procedure. Future work will center on merging a expansive number of highlights to extend discovery exactness whereas diminishing untrue positives. Execution comes about for competing approaches are appeared in Table 2; our Chowdhury approach was clearly prevalent.

**A screenshot of a graph

Description automatically generated**

Currently, the expansion of malevolent programs poses a critical risk to worldwide steadiness. Within the 1990s, as the number of interconnected computers detonated, so did the predominance of malevolent program, which in the long run driven to the broad conveyance of malware. Different defensive measures have been taken in reaction to this wonder. Shockingly, current shields cannot keep up with present day dangers that malware creators have made to foil security programs. In a long time, researchers' center on malware location inquire about has moved toward ML algorithm strategies. In this term paper, we show a defensive instrument that assesses three ML algorithm approaches to malware location and chooses the foremost fitting one. Agreeing to insights, the choice tree approach has the most extreme discovery precision (99.01%) and the least untrue positive rate (FPR; 0.021%) on a little dataset.

Malware proceeds to create and proliferate at a disturbing rate. Nur (2019) compared three ML classifiers to analyze and measure the location precision of the ML classifier that utilized inactive examination to extricate highlights based on PE data. As a bunch, we prepared machine learning calculations to recognize unsafe versus generous data. The DT machine learning method accomplished 99% precision, as outlined in Table 2, making it the foremost effective classifier we inspected. This experiment demonstrated the potential of inactive examination based on PE data and chosen key information highlights to realize the most noteworthy discovery precision and the foremost precise portrayal of malware. Malicious programs and their dangers, or “malware,” got to be progressively common and advanced as the Web created. Their quick scattering over the Web has given malware creators with get to a wide assortment of malware era devices. Each day, the reach and advancement of malware develops. This think about centered on analyzing and measuring classifier execution to superior get it how machine learning works. Inactive investigation extricated highlights from the recuperated PE record and library data; six classifiers based on ML procedures were assessed. It was prescribed that ML frameworks be prepared and tried to decide whether a record is hurtful. Exploratory results confirmed that the irregular woodland strategy is ideal for information categorization, with 99.4 percent precision. It appeared that the PE library was congruous with inactive investigation which focusing on as it were some properties may make strides malware location and characterization. The most advantage is that it is less likely that noxious computer programs will be introduced by mischance, as clients can check a file’s legitimacy some time recently opening it.

**"Deep Android Malware Detection" by Wei Yang, Lei Luo, Ming Li et al. (2017):**

This document provides an in-depth study of Android malware detection. It uses stacked autoencoders to extract advanced features from malware. Classification with Dalvik bytecode and SoftMax classifier for an Android application.

**"End-to-End Malware Classification Using Deep Convolutional Neural Networks", Rajaraman, Arvind, and Jae woo Kang (2017):**

The authors present an end-to-end approach to classify malware using deep neural networks (CNNs). They convert binary data into grayscale images and train CNNs to classify them as malware or positive images.

**"Malware Classification Using Long-Term Memory Networks", Pascanu, Razvan et al. (2015):**

This article examines the use of short-term temporal (LSTM) networks, a type of recurrent neural network, in malware classification. Demonstrates that LSTM networks can detect good neighbors in the system of API calls to detect malware.

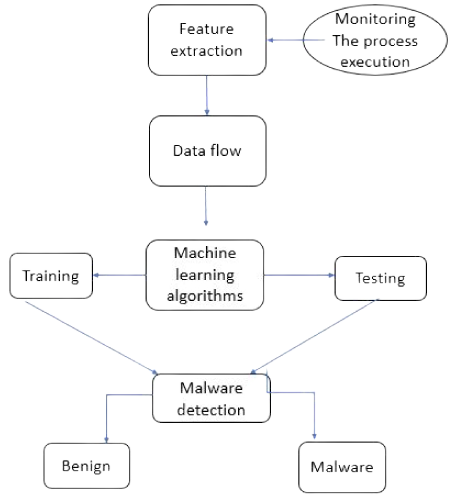
**"Deep Learning-Based Malware Detection Using Two-Dimensional Binary Features", Nataraj, Lakshmanan, et al. (2011):**

The authors present a deep learning method for malware classification using convolutional neural networks (CNN). They represent binary data as 2D images based on opcode n-grams and train CNNs for classification.

**Chapter 3**

**Methodology**

This case study presents the individual steps and components of effective machine learning for malware detection and classification, explores the challenges and limitations of this study, and examines new developments and trends in the field with an emphasis on technology education. The proposed research methodology for this study is given below. Figures 3 and 4 show the end-to-end workflow for better understanding of applying machine learning to malware detection.

****

**Figure3:** Deep-leaning ML malware detection method.

**A diagram of software testing

Description automatically generated**

**Figure 4:** Process representation

**About dataset:**

This study is based on data provided by the Canadian Cybersecurity Research Institute. This document contains several files containing datasets for different types of malwares. Such wheels can be used to train many models. About 51 different malware families were found in the sample. Contains more than 17,394 data points from various sources; the file has 279 lines and 17,394 lines.

**Preprocessing:**

Data stored as binary code in the file system and the file itself is an unprocessed executable. We prepare them before researching them. Opening executables requires a protected environment or virtual machine (VM). PEiD software automatically decompresses executable files.

**About Feature Extraction:**

Twentieth-century datasets often contain tens of thousands of features. In recent years, as feature counts have grown, it has become clear that the resultant machine learning model has been overfit. To address this problem, we built a smaller set of features from a larger set; this technique is commonly used to maintain the same degree of accuracy while using fewer features. The goal of this study was to refine the existing dataset of dynamic and static features by keeping those that were most helpful and eliminating those that were not valuable for data analysis.

**How features are selected:**

After completing feature extraction, which involved the discovery of more features, feature selection was performed. Feature selection was a crucial process for enhancing accuracy, simplifying the model, and reducing overfitting, as it involved choosing features from a pool of newly recognized qualities. Researchers have used many feature classification strategies in the past to identify dangerous code in software. As the feature rank technique is very effective at picking the right features for building malware detection models, it was extensively employed in this study.

**Model Selection:**

Choose a deep learning method suitable for malware detection. Usually convolutional neural networks (CNNs), recurrent neural networks (RNNs), or both are used. The architecture must be able to detect the unique patterns and characteristics of malware.

**Model Training:**

Introduces deep learning models selected using a previous dataset. Distribute data for training and validation. During training, the model learns to distinguish malware from good examples by adjusting its internal parameters based on recorded data.

**Analysis and Hyperparameter Tuning:**

Evaluating the performance of training models using separate datasets. Measures such as accuracy, precision, recall, and F1 score can be used to evaluate the performance of the model. If necessary, adjust hyperparameters such as learning rate, batch size, or network architecture to improve the performance of the model.

**Deployment and real-time detection:**

Once the model performs satisfactorily, it can be used for real-time malware detection. It uses input models (like files or network connections) and uses training models to classify it as malicious or harmless. Additional post-processing steps such as clustering or behavioral analysis can be used to improve detection accuracy.

**Continuous Improvement:**

Malware is constantly evolving, so updating and repurposing deep learning models is crucial. As new malware models emerge, they should be added to the training database to ensure the model stays up to date against the threat.

**Chapter 4**

**Result and Discussion**

Following is the count of all the benign and malware present in the sample dataset.

The bar graph shown consists of the whole dataset before dividing it into test and train sets. The blue bar or 0 shows benign software whereas the orange bar or 1 shows malware.

A blue and orange rectangular bars

Description automatically generated

The following Correlation matrix shows the relation between each column to determine the best features to be extracted from the dataset for training and testing our Model.

A close-up of a computer screen

Description automatically generated

The following graphs show the training and testing loss as well as accuracy. You can refer it as follows:

A graph with blue dots

Description automatically generated

A graph with red dots

Description automatically generated

Before Optimizing the model, the achieved accuracy on the current dataset is

**Test loss: 1.404659 Test accuracy: 98.595341%**

After Optimizing using keras.optimizers , the achieved accuracy on the same dataset is

**Test loss: 0.000911. Test accuracy: 98.595341%**

**Chapter 5**

**Conclusion and Future Work**

This project uses deep learning for malware detection to create models that can use the properties of the data used in this project to provide information about whether the software fed into it is correct code or malware. The training model is more accurate than expected and the data model used for training includes 1,00,000 software models, some chosen to train the model, and some used to test the samples.

In the future, I will use my coding skills and knowledge to implement protection mechanisms that can detect and prevent malware originating from a particular software or multiple software. Obviously, in most cases it can be used for anti-virus or security deployment purposes to detect and prevent damage to the user's body from undetected malware.

**References**

[1] MDPI open access journals. (https://www.mdpi.com)

[2] Hindwi research journals (https://www.hindawi.com)

[3] Research gate flowcharts and diagrams.