PDF DEFENDER: A LEARNING APPROACH TO PDF MALWARE RECOGNITION

PROJECT REPORT

Submitted by

AKSHAYA ANCY. P — **Register No.: 20TD0104**

GOHILA. S — **Register No.: 20TD0122**

SWATHI. D — **Register No.: 20TD0171**

Under the guidance of

Dr. V. KAVITHA

in partial fulfillment of the requirements for the degree

of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJIV GANDHI COLLEGE OF ENGINEERING AND TECHNOLOGY

PUDUCHERRY-607402

MAY - 2024

RAJIV GANDHI COLLEGE OF ENGINEERING AND TECHNOLOGY

PONDICHERRY UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the project work entitled "**PDF DEFENDER: A LEARNING APPROACH TO PDF MALWARE RECOGNITION**" is a bonafide work done by **AKSHAYA ANCY P [20TD0104], GOHILA S [20TD0122], SWATHI D [20TD0171]** in partial fulfillment of the requirement, for the award of B.Tech Degree in Computer Science and Engineering by Pondicherry University during the academic year 2023-2024.

PROJECT GUIDE **HEAD OF THE DEPARTMENT**

Submitted for the University Examination held on ……

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We are very thankful and grateful to our beloved guide, **Dr. V. KAVITHA** whose great support in valuable advices, suggestions and tremendous help enabled us in completing our project. He has been a great source of inspiration to us.

We also sincerely thank our Head of the Department, **Dr. R. G. SURESH KUMAR** whose continuous encouragement and sufficient comments enabled us to complete our project report.

We thank all our **Staff members** who have been by our side always and helped us with our project. We also sincerely thank all the lab technicians for their help as in the course of our project development.

We would also like to extend our sincere gratitude and grateful thanks to our **Dr. E. VIJAYAKRISHNA RAPAKA** for having extended the Research and Development facilities of the department.

We are grateful to our Chairman **Shri M. K. RAJAGOPALAN** He has been a constant source of inspiration right from the beginning.

We would like to express our faithful and grateful thanks to our Administrator officer, **Shri. V. BHASKARAN.**

We wish to thank our **family members and friends** for their constant encouragement, constructive criticisms and suggestions that has helped us in timely completion of this project.

TABLE OF CONTENT

PDF DEFENDER : A LEARNING APPROACH TO PDF MALWARE RECOGNITION

ABSTRACT

PDF malware refers to malicious software or code that is embedded within PDF (Portable Document Format) files, which are commonly used for sharing and distributing information. The proposed system represents a significant leap forward in the ongoing battle against PDF malware, addressing a critical vulnerability inherent in existing approaches. This vulnerability lies in the susceptibility of current systems to evasive variants of PDF-based malware that can adeptly bypass machine learning-based classifiers. What sets our solution apart is its innovative use of LSTM (Long Short-Term Memory) algorithm. In contrast to the conventional system, our approach integrates a pre-trained model into the detection process.

This strategic incorporation significantly reduces the time required for training the system without compromising on accuracy. By leveraging LSTM, we have created an advanced and efficient solution specifically tailored for the detection of image-based malware within PDF files. In rigorous comparative testing, our system has demonstrated remarkable performance improvements. It not only surpasses the accuracy of the current system but also achieves faster training times, showcasing its efficacy in handling the evolving challenges posed by PDF-based malware threats. The integration of LSTM further enhances the robustness of our approach, providing an advanced defense mechanism against the dynamic and constantly changing landscape of PDF malware.

In summary, our pioneering system not only elevates security measures against malicious PDF files but also introduces a more streamlined and responsive solution to tackle the persistent challenges posed by the sophisticated and ever-changing nature of PDF-based malware threats. This represents a significant stride forward in the field of cybersecurity, offering a more effective and adaptive defense against the constantly evolving tactics employed by malicious actors in the digital realm.

Keywords: Variational Autoencoder (VAE), PDF-based malware threats

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	:	Artificial Intelligence
ML	:	Machine Learning
DL	:	Deep Learning
ANN	:	Artificial Neural Network
CNN	:	Convolutional Neural Network
RNN	:	Recurrent Neural Network
LSTM	:	Long Short-Term Memory
GRU	:	Gated Recurrent Unit
GAN	:	Generative Adversarial Network
SVM	:	Support Vector Machine
NLP	:	Natural Language Processing
GPT	:	Generative Pre-trained Transformer
OCR	:	Optical Character Recognition
Adam	:	Adaptive Moment Estimation
GPU	:	Graphics Processing Unit
TPU	:	Tensor Processing Unit
AP	:	Average Precision
MAP	:	Mean Average Precision
MSE	:	Mean Squared Error
CE	:	Cross-Entropy
KL Divergence	:	Kullback-Leibler Divergence
ReLU	:	Rectified Linear Unit

ORGANIZATION OF THE CHAPTERS

In Chapter 1, we introduce about the project concept and give an overview idea about theproject. The main need of reviewing the overview about the project is to give a clear idea about the project to the reader. We also have given a need for study which focuses on the main requirements of the project. The objective of the project is also discussed in this chapter.

In Chapter 2, we discuss about the detailed description of existing systems by analysis the literature survey of the existing techniques. We also then presented about the techniques and methods. In our proposed method we also listed out the advantages of using our proposed method. Then we presented the differences between the existing system and proposed systemby stating the advantages of our proposed system.

In Chapter 3, we made a system analysis of the methods we propose. Here we have discussed about the existing system. Existing propose a framework to automatically and in real- time perform credibility analysis of posts on social media, based on three levels of credibility: Text, User, and Social So, we have also discussed about the problem definition of the project that must be discussed in order to give a clear proposed system**.**

In Chapter 4, we listed the hardware requirements and software requirements of our project. The understanding of the hardware and software requirements is required in order to make a clear planning for our implementation stage. We have also listed out the software's that arebeing used in our project and have also given a brief description of that software's

In Chapter 5, we have given our proposal. By analysing the existing system and the problem definition, we have given the proposed system for the project. we identify the credibility of tweets by using text credibility, user credibility, social credibility and emotional credibility (sentiment polarity) for a particular topic, we add weight to authors content using this credibility.

In Chapter 6, we have given a brief conclusion about the project. We have discussed about the disadvantages of the existing system and the advantages of proposed system and given a conclusionof the project.

CHAPTER 1

1. INTRODUCTION

1.1 OVERVIEW

The term "PDF malware" refers to malicious software or code that is embedded within files using the Portable Document Format (PDF), a widely adopted format for sharing and distributing information. PDF files are commonly exploited by malicious actors who may take advantage of vulnerabilities in PDF readers or employ social engineering techniques to create deceptive PDFs containing hidden malware [25]. This malware can be in the form of embedded scripts, links, or executable code. When an unsuspecting user opens what appears to be a harmless PDF file, the embedded malware is triggered, potentially leading to security breaches, data theft, or compromise of the system.

In the context of combating PDF malware, it has become increasingly important to employ advanced detection techniques. These techniques include hybrid algorithmic approaches and image-based analysis, which aim to identify and neutralize threats within PDF documents. By leveraging innovative algorithms and analysing the content of PDF files at an image level, security measures can be enhanced to keep pace with the evolving sophistication of PDF-based malware[23].

The broader category of "malware" encompasses various types of malicious software intentionally crafted to cause harm or exploit vulnerabilities in computer systems, networks, and devices [12]. This includes viruses, worms, trojan horses, ransomware, spyware, and adware, each designed for specific malicious purposes. Cybercriminals commonly deploy malware through deceptive means, such as phishing emails, malicious websites, or compromised software downloads, to infiltrate and compromise the security of a target system.

Once inside a system, malware can execute a range of malicious activities, including stealing sensitive information, disrupting system functions, or providing unauthorized access to attackers. Effectively combating malware involves implementing robust cybersecurity measures, including the use of antivirus software, firewalls, and regular system updates [26]. These measures work together to detect, prevent, and mitigate the impact of various types of malware threats, helping to safeguard digital ecosystems and protect against the potentially devastating consequences of malicious software attacks.

1.2 TYPES OF MACHINE LEARNING

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

1.2.1 SUPERVISED LEARNING

Supervised learning is the type of machine learning in which machines are trained using well labelled training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output. In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher. Supervised learning is a process of providing input data as well as correct output data to the machine learning model [27]. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

In the real-world, supervised learning can be used for Risk Assessment, Image classification, Fraud Detection, spam filtering, etc. Supervised learning is a type of machine learning in which machines learn from known datasets (set of training examples), and then predict the output [22]. A supervised learning agent needs to find out the function that matches a given sample set.

TYPES OF SUPERVISED LEARNING ALGORITHMS

REGRESSION

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

- Linear Regression
- Non-Linear Regression
- Bayesian Linear Regression
- Regression Tree
- Polynomial regression

LINEAR REGRESSION

Linear regression is one of the easiest and most popular machine learning algorithms [6]. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc[22].

NON-LINEAR REGRESSION

Nonlinear regression is a form of regression analysis in which data is fit to a model and then expressed as a mathematical function. Simple linear regression relates two variables (X and Y) with a straight line (y = mx + b), while nonlinear regression relates the two variables in a nonlinear (curved) relationship.

BAYESIAN LINEAR REGRESSION

Bayesian regression methods are very powerful, as they not only provide us with point estimates of regression parameters [6], but rather deliver an entire distribution over these parameters. This can be understood as not only learning one model, but an entire family of models and giving them different weights according to their likelihood of being correct.

REGRESSION TREE

A regression tree is basically a decision tree that is used for the task of regression which can be used to predict continuous valued outputs instead of discrete outputs. In Decision Trees for Classification, and observed how the tree asks right questions at the right node in order to give accurate and efficient classifications [16].

POLYNOMIAL REGRESSION

Polynomial Regression is a regression algorithm that models the relationship between a dependent(y) and independent variable(x) as nth degree polynomial. It is also called the special case of Multiple Linear Regression in ML. Because of adding some polynomial terms to the Multiple Linear regression equation to convert it into Polynomial Regression [6]. It is a linear model with some modification in order to increase the accuracy.

CLASSIFICATION

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc. Some of the classification algorithms are:

- K-Nearest Neighbour
- Decision Tree
- Support vector machine
- Logistic Regression
- Random Forest

K-NEAREST NEIGHBOR

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning techniques. It stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using this algorithm. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

DECISION TREE

A decision tree is a decision support tool that uses a tree like model of decisions and their possible consequences, including chance event outcomes, resources costs, and utility. Decision trees are commonly used in operation research, specifically in decision analysis, to help identify a strategy most likely to reach goal, but are also a popular tool in machine learning.

SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear or nonlinear classification, regression, and even outlier detection tasks. SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection [16]. SVMs are adaptable and efficient in a variety of applications because they can manage high-dimensional data and nonlinear relationships.

LOGISTIC REGRESSION

Logistic regression comes under the supervised learning techniques. It is used for predicting the categorical dependent variable using a given set of the independent variable. The outcome must be a categorical or discrete value. It is used for solving classification problems.

RANDOM FOREST

Random Forest is a popular machine learning algorithm that belongs to the supervised learning techniques [4]. It can be used for both classification and regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifies to solve a complex problem and to improve the performance of the model.

1.2.2 UNSUPERVISED LEARNING

Unsupervised learning is a type of machine learning in which models are trained using unlabelled dataset and are allowed to act on that data without any supervision. In unsupervised learning, the algorithms are trained with data which is neither labelled nor classified [27]. In unsupervised learning, the agent needs to learn from patterns without corresponding output values.

Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning of having the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format. Unsupervised Learning Algorithms allow users to perform more complex processing tasks compared to supervised learning. Although, unsupervised learning can be more unpredictable compared with other natural learning methods [27]. Unsupervised learning algorithms include clustering, anomaly detection, neural networks, etc.

TYPES OF UNSUPERVISED LEARNING ALGORITHMS

CLUSTERING

Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities [23]. Some types of clustering are:

- Hierarchical clustering
- K-means clustering
- Principal Component Analysis

HIERARCHICAL CLUSTERING

Hierarchical clustering is an algorithm which builds a hierarchy of clusters. It begins with all the data which is assigned to a cluster of their own. Here, two close cluster are going to be in the same cluster [23]. This algorithm ends when there is only one cluster left.

K-MEANS CLUSTERING

K means it is an iterative clustering algorithm which helps you to find the highest value for every iteration. Initially, the desired number of clusters are selected. In this clustering method, you need to cluster the data points into k groups [17]. A larger k means smaller groups with more granularity in the same way. A lower k means larger groups with less granularity.

The output of the algorithm is a group of "labels." It assigns data point to one of the k groups. In k-means clustering, each group is defined by creating a centroid for each group. The centroids are like the heart of the cluster, which captures the points closest to them and adds them to the cluster.

PRINCIPAL COMPONENT ANALYSIS

In case you want a higher-dimensional space. You need to select a basis for that space and only the 200 most important scores of that basis. This base is known as a principal component. The subset you select constitute is a new space which is small in size compared to original space. It maintains as much of the complexity of data as possible.

ASSOCIATION

An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset [28]. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

- Anomaly Detection
- Apriori Algorithm

ANOMALY DETECTION

Anomaly detection also known as outlier detection is the process of finding data points within a dataset that differs from the rest. Common applications of anomaly detection

include fraud detection in financial transactions, fault detection and predictive maintenance.[28]

APRIORI ALGORITHM

Apriori is part of the association rule learning algorithms, which sit under the unsupervised branch of Machine Learning. This is because Apriori does not require us to provide a target variable for the model. Instead, the algorithm identifies relationships between data points subject to our specified constraints [28].

1.2.3 REINFORCEMENT LEARNING

Reinforcement learning (RL) is the science of decision making. It is about learning the optimal behaviour in an environment to obtain maximum reward. This optimal behaviour is learned through interactions with the environment and observations of how it responds, similar to children exploring the world around them and learning the actions that help them achieve a goal [1]. In the absence of a supervisor, the learner must independently discover the sequence of actions that maximum the reward. This discovery process is akin to a trail-and-error search. The quality of actions is measured by not just the immediate reward they return, but also the delayed reward they might fetch. As it can learn the actions that result in eventual success in an unseen environment without the help of a supervisor, reinforcement learning is a very powerful algorithm [4].

1.3 CHALLENGES IN MACHINE LEARNING

Machine learning is a rapidly growing field that has the potential to revolutionize the way we live and work. However, it also presents a number of challenges that researchers and practitioners need to overcome [1]. One of the biggest challenges is obtaining high-quality and sufficient data to train models. Without enough data, models may not be able to accurately predict or classify new inputs. Data preprocessing is another crucial step in machine learning, but it can be challenging as it often involves cleaning, normalizing, and transforming data to make it usable for training models [11].

Overfitting and underfitting are also common challenges in machine learning, where overfitting occurs when a model is too complex and is trained to fit the training data too well, resulting in poor generalization to new data, whereas underfitting occurs when a model is too simple and is not able to capture the underlying patterns in the data. High-dimensional data can also present challenges, as it can lead to increased computation time and memory requirements, which can be addressed by using dimensionality reduction techniques [15].

Choosing the right model for a given task and evaluating the performance of a model can also be challenging tasks. Additionally, avoiding bias in the data and the model is another challenge that needs to be considered in machine learning.

TRANSFER LEARNING ALGORITHM

The concept of transfer learning in machine learning involves leveraging knowledge gained from one task and applying it to another related task. In transfer learning, a model pretrained on a large dataset for a specific task is adapted to perform a different but related task with a smaller dataset. This approach is particularly beneficial when acquiring extensive labelled data for a new task is challenging or impractical. Transfer learning algorithms typically involve two main steps: pre-training and fine-tuning [20].

During pre-training, a neural network is trained on a large dataset for a source task, such as image classification. The knowledge gained during this phase is captured in the network's weights. In the fine-tuning step, the pretrained model is adjusted or fine-tuned using a smaller dataset for the target task, adapting the learned features to the new context [11].

One popular application of transfer learning is in image recognition. For example, a model pre-trained on a dataset like ImageNet, which contains a wide variety of images and categories, can be fine-tuned for a more specific task like recognizing specific objects in medical images. Transfer learning is advantageous in scenarios where labelled data is scarce, as it allows the model to benefit from knowledge gained in other domains [11]. This approach has found success in various fields, including computer vision, natural language processing, and speech recognition, contributing to the development of more accurate and efficient models in real-world applications.

Transfer learning is a technique in machine learning where a model trained on one task is used as the starting point for a model on a second task. This can be useful when the second task is similar to the first task, or when there is limited data available for the second task. By using the learned features from the first task as a starting point, the model can learn more quickly and effectively on the second task [15]. This can also help to prevent over fitting, as the model will have already learned general features that are likely to be useful in the second task.

Many deep neural networks trained on images have a curious phenomenon in common:

in the early layers of the network, a deep learning model tries to learn a low level of features, like detecting edges, colours, variations of intensities, etc. Such kind of features appear not to be specific to a particular dataset or a task because no matter what type of images are processing either for detecting a lion or car. In both cases, having to detect these low-level features. All these features occur regardless of the exact cost function or image dataset [14]. Thus, learning these features in one task of detecting lions can be used in other tasks like detecting humans.

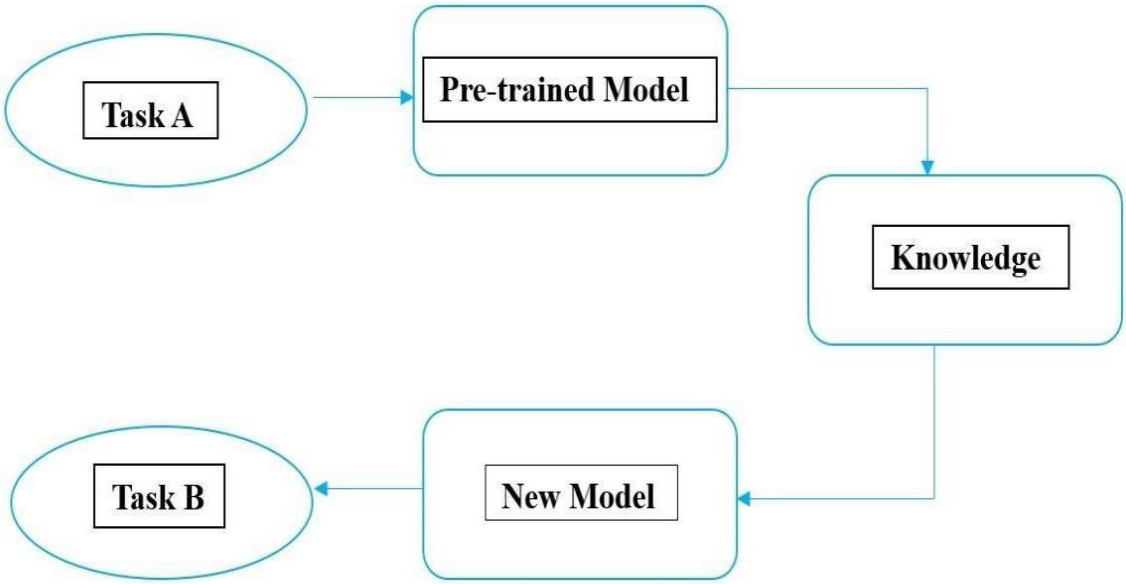

Fig 1.1 Transfer Learning

From fig 1.1, This is what transfer learning is. Nowadays, it is very hard to see people training whole convolutional neural networks from scratch, and it is common to use a pretrained model trained on a variety of images in a similar task, e.g. models trained on ImageNet (1.2 million images with 1000 categories) and use features from them to solve a new task. When dealing with transfer learning, and came across a phenomenon called the freezing of layers. A layer, it can be a CNN layer, hidden layer, a block of layers, or any subset of a set of all layers, is said to be fixed when it is no longer available to train. Hence, the weights of freeze layers will not be updated during training. While layers that are not frozen follows regular training procedure [24].

When using the transfer learning in solving a problem, by select a pre-trained model as our base model. Now, there are two possible approaches to using knowledge from the pretrained model. The first way is to freeze a few layers of the pre-trained model and train

other layers on our new dataset for the new task. The second way is to make a new model, but also take out some features from the layers in the pre-trained model and use them in a newly created model. In both cases, by taking out some of the learned features and try to train the rest of the model [23]. This makes sure that the only feature that may be the same in both of the tasks is taken out from the pre-trained model, and the rest of the model is changed to fit the new dataset by training.

ADVANTAGES OF TRANSFER LEARNING

- Speed up the training process: By using a pre-trained model, the model can learn more quickly and effectively on the second task, as it already has a good understanding of the features and patterns in the data.
- Better performance: Transfer learning can lead to better performance on the second task, as the model can leverage the knowledge it has gained from the first task.
- Handling small datasets: When there is limited data available for the second task, transfer learning can help to prevent overfitting, as the model will have already learned general features that are likely to be useful in the second task.

DISADVANTAGES OF TRANSFER LEARNING

- Domain mismatch: The pre-trained model may not be well-suited to the second task if the two tasks are vastly different or the data distribution between the two tasks is very different.
- Overfitting: Transfer learning can lead to overfitting if the model is fine-tuned too much on the second task, as it may learn task-specific features that do not generalize well to new data.
- Complexity: The pre-trained model and the fine-tuning process can be computationally expensive and may require specialized hard.

1.4 OBJECTIVE OF THE PROJECT WORK

The primary objective of the PDF malware prediction project is to develop an advanced and robust system for anticipating and mitigating the risks associated with malicious software embedded within Portable Document Format (PDF) files. The project aims to address a critical vulnerability in existing systems by focusing on the prediction and detection of evasive variants capable of circumventing traditional machine learning-based classifiers. To achieve this, the project employs a hybrid algorithmic approach,

incorporating a pre-trained model to significantly reduce training time while maintaining high accuracy.

By leveraging innovative techniques, such as hybrid algorithms and image-based analysis, the project seeks to enhance the efficiency of malware detection within PDF files. The overarching goal is to provide a proactive defence mechanism against the evolving landscape of PDF-based malware threats, thereby bolstering overall cybersecurity measures and safeguarding users from potential security breaches, data theft, and system compromises associated with malicious PDF files.

1.5 NEED OF THE PROJECT WORK

The need for the project work on PDF malware prediction arises from the growing and persistent threat landscape posed by malicious software embedded within Portable Document Format (PDF) files. As PDFs are widely used for document sharing and distribution, they have become a common vector for cyber-attacks. Malicious actors exploit vulnerabilities in PDF readers or employ sophisticated social engineering techniques to create deceptive PDFs containing hidden malware. Traditional approaches to malware detection often fall short in effectively identifying evasive variants that can bypass machine learning-based classifiers. Hence, the project addresses a critical need in the field of cybersecurity.

By developing an advanced and proactive PDF malware prediction system, the project aims to fill the gap in existing security measures and provide a more robust defence against the evolving tactics employed by cybercriminals. The hybrid algorithmic approach, including the integration of a pre-trained model and image-based analysis, is designed to enhance the accuracy and efficiency of malware detection, significantly reducing training times without compromising on security. The ultimate goal is to contribute to the development of cutting-edge tools that can better protect users and organizations from the potential security risks associated with malicious PDF files, including unauthorized access, data theft, and system compromise.

CHAPTER 2

2. LITERATURE SURVEY

2.1 TITLE : An Attention Mechanism for Combination of CNN and VAE for Image-Based Malware Classification.

AUTHOR : Tuan Van Dao; Hiroshi Sato; Masao Kubo

YEAR 2022

DESCRIPTION

In the face of the escalating number and sophistication of malware, there is a pressing need for efficient detection and neutralization methods. Traditional approaches, relying on malware signatures or behaviours, often demand substantial computational resources for feature engineering. Recognizing the potential of machine learning in addressing these challenges, recent studies have explored the application of various techniques to identify and classify malware families [2]. Although combining multiple state-of-the-art methods has gained popularity, the challenge lies in determining an optimal and efficient combination. Complex neural network architectures have demonstrated improved classification performance, but they come at the cost of increased resource requirements.

They introduced a novel lightweight architecture that merges small Convolutional Neural Networks with an advanced Variational Autoencoder, augmented by channel and spatial attention mechanisms [2]. Through comprehensive experiments, the proposed approach showcases superior performance and efficiency compared to other cutting-edge techniques when applied to both unbalanced and balanced Malimg datasets [29]. This innovative combination offers a promising solution for addressing the evolving landscape of malware threats while mitigating the computational burden associated with intricate neural network architectures.

CONCLUSION

We are going to use the MALIMG DATASET from this research. The architecture of CNN algorithm is taken for our further works. And the using of lightweight architecture, combining small Convolutional Neural Networks with a Variational Autoencoder and incorporating attention mechanisms, emerges as a promising solution.

2.2 TITLE : Self-Attentive Models for Real-Time Malware Classification

AUTHOR : Qikai Lu; Hongwen Zhang; Husam Kinawi; Di Niu

YEAR 2022

DESCRIPTION

In the realm of cybersecurity, the classification of malware holds paramount importance, providing crucial insights into the nature of threats and facilitating the development of effective countermeasures. The challenge intensifies in the context of real-time malware classification, particularly considering the high network throughputs of modern networks. They introduced two self-attention transformer-based classifiers, namely SeqConvAttn and ImgConvAttn, as innovative alternatives to the prevailing Convolutional Neural Network (CNN) classifiers [3].

Going beyond this, the work proposes a file-size-aware two-stage framework that integrates these transformer-based models, allowing for a nuanced control of the trade-off between accuracy and latency in real-time classification scenarios. To evaluate the efficacy of these proposed designs, extensive experiments are conducted on three diverse malware datasets: the Microsoft Malware Classification Challenge (BIG 2015), and two subsets derived from the BODMAS PE malware dataset, namely BODMAS-11 and BODMAS-49[5]. Results demonstrate that the transformer-based designs outperform traditional CNN-based models in terms of classification accuracy.

Moreover, the two-stage framework showcases its capability to reduce average model inference latency while concurrently maintaining superior accuracy. This research addresses the critical need for real-time malware classification in the face of modern network challenges, offering a promising solution that aligns with the demands of both accuracy and latency in the dynamic landscape of cybersecurity [3].

CONCLUSION

We are going to include Self-Attention Transformer-Based Classifiers, File-Size-Aware Two-Stage Framework and The Evaluation on Diverse Malware Datasets. Usage of diverse malware datasets, such as the Microsoft Malware Classification Challenge (BIG 2015), BODMAS-11, and BODMAS-49 subsets. The implementation of Performance Metrics and Real-Time Malware Classification for achieving a balance between accuracy and latency in the context of modern networks.

2.3 TITLE : A New Method for Malware Detection Based on Energy

AUTHOR : Liang Ge

YEAR 2021

DESCRIPTION

The escalating growth in the volume of malware, coupled with the deployment of sophisticated evasion and obfuscation techniques, has significantly impeded traditional signature-based approaches to cybersecurity. Machine learning-based methods have emerged as a promising alternative, offering faster analysis times and greater resilience against evasion tactics. In this work, presenting a semi-supervised malware detection and classification system centred around energy [15]. The unique aspect of this system lies in its reliance on anomaly free training data, aiming to detect and identify diverse malware types in test data.

Additionally, they introduced a novel approach by back-propagating gradients of the energy score concerning the code, generating a gradient map that facilitates the localization of malware within the code. To quantitatively assess the effectiveness of the method, extensive testing was conducted on an open-source dataset comprising approximately 20 thousand Portable Executable (PE) samples, analysed through static analysis [9]. The results of malware detection achieved through this method are comparable to those of other academic works in the current state of the art.

Notably, the models employed in the proposed method yield interpretable results, providing security analysts with valuable insights for a better understanding of malware behaviour [9]. The research contributes to the ongoing efforts in developing robust and interpretable malware detection techniques, offering a novel approach that leverages energy based models and gradient maps to enhance the accuracy and interpretability of malware detection systems.

CONCLUSION

We are going to use Semi-Supervised Malware Detection System to explore the implementation of a semi-supervised malware detection and classification system cantered around energy, anomaly-Free Training Data to Assess how this methodology contributes to the detection and identification of diverse malware types in test data. And the Robust and Interpretable Techniques are also used to leverage the contribution of this research to ongoing efforts in developing robust and interpretable malware detection techniques.

2.4 TITLE : A Hierarchical based ensemble classifier for behavioural malware detection using Machine Learning

AUTHOR : Muhammad Jamil Hussain; Ahmad Shaoor; Salman Baig; Abrar Hussain; Syed

YEAR 2022

DESCRIPTION

In the face of an escalating threat landscape posed by malware, safeguarding data security and confidentiality has become a paramount concern. Recognizing the growing significance of malware detection, this research introduces an efficient behavioural malware detection system designed specifically for Portable Executable (PE) files. The approach relies on machine learning classifiers to discern and categorize potential threats. They utilize the Blue Hexagon Open Dataset for Malware Analysis (BODMAS) [6], a recently published dataset spanning from August 2019 to September 2020, to both train and evaluate the proposed design. The proposed methodology unfolds in two stages.

In the initial stage, a binary classifier employs a random forest algorithm to detect whether a PE file is malicious or benign. Subsequently, the second stage employs a multi-class ensemble-based voting classifier to identify the family of malware. This ensemble comprises K-nearest neighbour (KNN) [21], support vector machine (SVM) [16], random forest, decision tree, and gradient boosting classifiers, each contributing with equal weights. Notably, the binary classifier achieves a remarkable accuracy of 99.48%, demonstrating its efficacy in distinguishing between malicious and non-malicious PE files. The ensemble-based classifier in the second stage attains an accuracy of 92.49%, showcasing its ability to identify the specific family of malware.

CONCLUSION

We are going to include efficient Behavioural Malware Detection System for PE Files to adopt the concept of an efficient behavioural malware detection system designed specifically for Portable Executable (PE) files and the Machine Learning Classifiers to Explore the use of machine learning classifiers, particularly the random forest algorithm for binary classification in the initial stage. We also plan to use the Blue Hexagon Open Dataset for Malware Analysis (BODMAS) to both train and evaluate your proposed design. This dataset covers malware samples from August 2019 to September 2020.

2.5 TITLE: PDF Malware Detection Based on Optimizable Decision Trees

AUTHOR : Qasem Abu Al-Haija, Ammar Odeh and Hazem Qattous

YEAR 2022

DESCRIPTION

In the landscape of digital document sharing, Portable Document Format (PDF) files stand out as one of the most widely used formats, making them an attractive target for hackers seeking to exploit them as infection vectors [11]. Security threats often emerge when malicious actors hide harmful code within apparently innocuous PDF documents, leading to the creation of PDF malware. Addressing this issue requires advanced techniques to differentiate between benign and malicious files. Research studies have consistently shown that machine learning methods offer efficient detection mechanisms against such PDF-based threats [14]. They introduced a novel detection system designed to analyse PDF documents, distinguishing between benign and malware-infected files.

The system leverages the power of the AdaBoost decision tree with optimal hyperparameters, trained and evaluated on a comprehensive and modern dataset named Evasive-PDFMal2022[13]. The investigational assessment underscores the effectiveness of this lightweight yet accurate PDF detection system, achieving an impressive prediction accuracy of 98.84% with a minimal prediction interval of 2.174 μ Sec. These results demonstrate the system's ability to outperform other state-of-the-art models in the same research domain. In essence, the proposed model presents a highly effective solution for uncovering PDF malware, offering superior detection performance and minimal detection overhead. This research contributes to the ongoing efforts in developing robust cybersecurity solutions, particularly in the realm of PDF file security, and showcases a promising tool for safeguarding against malicious threats hidden within PDF documents [13].

CONCLUSION

We are going to use PDF Malware Detection System to Implement a novel detection system specifically designed to analyse PDF documents and AdaBoost Decision Tree to Leverage the power of the AdaBoost decision tree with optimal hyperparameters as the core of your detection system. The evasive-PDFMal2022 Dataset is used to Utilize the comprehensive and modern dataset named Evasive-PDFMal2022 for training and evaluating your PDF malware detection system and High Prediction Accuracy to Consider the impressive prediction accuracy of 98.84% achieved by the proposed system, along with a minimal prediction interval of 2.174 μ Sec.

2.6 TITLE : Design and Analysis of Machine Learning Based Technique for Malware Identification and Classification of Portable Document Format Files

AUTHOR : Sultan S. Alshamrani

YEAR 2022

DESCRIPTION

In the realm of system security, modern antivirus software often falls short in providing adequate protection against malicious Portable Document Format (PDF) files, posing a potential threat to computer systems [19]. To address this vulnerability, they introduced a novel PDF malware classification system based on machine learning (ML). The unique aspect of this system lies in its dual approach to inspecting PDF files, employing both statistical and dynamic analysis. This dual-method approach enhances the accuracy of identifying the true nature of a document. Crucially, the method is non-signature-based, making it potentially effective in distinguishing obscure and zero-day malware threats. The experiment involves the deployment of five different classifier algorithms, and the best-fit approach is determined by evaluating key metrics such as true positive rate (TPR), precision, false positive rate (FPR), false negative rate (FNR), and F1-score for each algorithm

To validate the effectiveness of the proposed system, a comprehensive comparison is conducted with previously existing PDF classification systems. Additionally, they implement a malicious attack on the proposed system to simulate obfuscation of malicious code within a PDF file, a tactic often used by attackers. The results indicate that the proposed approach, particularly when utilizing the random forest (RF) [4] classifier, achieved an F1-measure of 0.986, surpassing the state-of-the-art systems where the F1-measure was 0.978. This demonstrates the effectiveness of the proposed approach in identifying malware embedded within PDF files, offering a more robust solution compared to existing systems. Overall, they present a significant contribution to the field of cybersecurity by introducing an advanced PDF malware classification system that enhances accuracy and resilience against sophisticated threats [4].

CONCLUSION

We are using a machine learning approach and utilizing the random forest classifier and using tools for statistical and dynamic analysis.

2.7 TITLE : EvadeRL: Evading PDF Malware Classifiers with Deep Reinforcement Learning**.**

AUTHOR : Zhengyang Mao,1Zhiyang Fang,2Meijin Li,1and Yang Fan

YEAR 2022

DESCRIPTION

In the era of information digitization, PDF files have become a significant carrier of malicious documents, presenting a challenge for executable file detection technologies. While machine learning-based classifiers have shown improved effectiveness in detecting PDF malware, adversaries continually develop countermeasures, such as generating adversarial examples, to evade detection [23]. In contrast to previous works that aimed to highlight vulnerabilities in learning-based detection models, they focus on addressing the computational demands associated with stochastic manipulations applied in existing research.

The work introduces EvadeRL, a novel framework designed to automatically generate adversarial examples based on double deep Q-Network. The approach involves an EvadeRL agent that selects a series of actions to modify PDF files, utilizing classification results and observations from the environment to calculate the approximate value of each action. Through interaction with the environment, experiences gained are stored to train the decision network. Subsequently, the agent can generate adversarial examples against a target detector by adopting optimal behaviours post-training [20]. Notably, they contributed to the sustainability of evasion attacks through online fine-tuning, marking it as the first in the field to focus on evolving malware. Experimental results demonstrate that EvadeRL achieves a high evasion rate against PDF malware detectors, surpassing other approaches in terms of execution cost, robustness against hardened detectors, and sustainability against both evolving malware and detectors [20]. The research sheds light on the intricate dynamics of evasion tactics in the realm of PDF malware detection, providing valuable insights and a pioneering framework to enhance our understanding of adversarial challenges in the cybersecurity landscape.

CONCLUSION

We are using the novel framework called EvadeRL, designed to address challenges in PDF malware detection. Autonomous Adversarial Example Generation for the Recognition of approach involving an EvadeRL agent that autonomously selects actions to modify PDF files, leveraging classification results and environmental observation.

2.8 TITLE : An Analysis of Machine Learning-Based Android Malware Detection Approaches

AUTHOR: R. Srinivasan1, S Karpagam2, M. Kavitha1 and R. Kavitha

YEAR 2022

DESCRIPTION

The proliferation of Android apps has brought about a corresponding increase in Android malware, posing a significant threat to mobile ecosystems. With Android phones constituting a substantial 72.2 percent of all smartphone sales, the potential impact of malware operations on these devices is substantial [8]. Hackers employ various tactics such as credential theft, eavesdropping, and malicious advertising to compromise the security of Android phones. Recognizing the growing menace, researchers have delved into Android malware detection from diverse perspectives, proposing hypotheses and methodologies. They specifically focus on machine learning (ML)-based techniques as effective tools for identifying Android malware [10].

ML approaches have proven their efficacy by enabling the creation of classifiers from sets of training cases, eliminating the need for explicit signature definitions in malware detection. They provided a comprehensive examination of existing ML-based Android malware detection approaches, highlighting the power and promise of machine learning and genetic algorithms in this context. By exploring the Android system architecture, security mechanisms, and malware categorization, they contributed to a deeper understanding of the landscape and underscores the potential of advanced technologies in combatting the evolving challenges posed by Android malware [5]. This serves as a valuable resource for researchers and practitioners seeking insights into the ongoing efforts to enhance the security of Android devices against the growing threat of malware.

CONCLUSION

We are using machine learning (ML)-based techniques, in the absence of explicit signature definitions, suggests that integrating ML into our detection system could be effective we are taking Android system architecture, genetic algorithms and security mechanisms are used in adapting Android malware detection system and designing a robust solution.

2.9 TITLE : Ransomware Detection and Classification Strategies

AUTHOR : Aldin Vehabovic, Nasir Ghani, Elias Bou-Harb, Jorge Crichigno, Aysegul Yayimli

YEAR 2022

DESCRIPTION

Ransomware, utilizing encryption methods to render data inaccessible to legitimate users, has emerged as a formidable cyber threat, causing significant damage to governments, corporations, and private users. The proliferation of various ransomware families has prompted researchers to devise and propose a multitude of detection and classification schemes. In response to the escalating nature of these cyber threats, many of these schemes leverage advanced machine learning techniques for processing and analysing real-world ransomware binaries and action sequences [7].

These categories include network-based approaches, host-based approaches, forensic characterization methods, and authorship attribution techniques. Each category addresses ransomware detection from distinct perspectives, providing a holistic overview of the diverse strategies employed by researchers and cybersecurity professionals [17]. The survey not only outlines existing methodologies but also sheds light on key facilities and tools used for ransomware analysis. However, despite the progress made in ransomware detection, they also highlight open challenges that remain in this space. These challenges encompass the need for continuous adaptation to evolving ransomware variants, the development of more resilient and accurate detection methods, and the enhancement of collaborative efforts to address this everchanging cybersecurity threat landscape [7]. Overall, the work serves as a valuable resource for researchers, practitioners, and policymakers in the ongoing effort to combat the menace of ransomware.

CONCLUSION

We are using the ransomware detection and classification methods and incorporating advanced machine learning techniques for processing real-world ransomware binaries and action sequences, aligning with the evolving nature of cyber threats and adapting to new ransomware variants, refining detection accuracy, and fostering collaborative efforts in the cybersecurity community.

2.10 TITLE : Comparative Analysis of Machine Learning Models for PDF Malware Detection: Evaluating Different Training & Testing Criteria

AUTHOR : Bilal Khan1, Muhammad Arshad2, Sarwar Shah Khan3

YEAR 2022

DESCRIPTION

As the transfer of files becomes more prevalent, the proliferation of maliciously coded documents has given rise to sophisticated cyber-attacks. Portable Document Format (PDF) files, in particular, have become a significant vector for malware due to their adaptability and widespread usage [24]. The challenge in detecting malware within PDF files lies in their capacity to incorporate various harmful elements such as embedded scripts, exploits, and malicious URLs. They address this challenge through a comparative analysis of machine learning (ML) techniques, specifically Naive Bayes (NB) [18], K-Nearest Neighbour (KNN), Average One Dependency Estimator (A1DE), Random Forest (RF), and Support Vector Machine (SVM) for PDF malware detection.

They utilize a dataset sourced from the Canadian Institute for Cyber-security and employs two different testing criteria: percentage splitting and 10-fold cross-validation. The performance of each ML technique is assessed using key metrics such as F1-score, precision, recall, and accuracy [18]. As PDF files continue to be a prevalent medium for cyber threats, this contributes to the ongoing efforts to enhance cybersecurity by leveraging machine learning techniques for effective malware detection and mitigation.

CONCLUSION

We are using machine learning techniques for malware detection. The comparative analysis of Naive Bayes, K-Nearest Neighbour, Average One Dependency Estimator, Random Forest, and Support Vector Machine offers a nuanced understanding of their effectiveness in mitigating the diverse elements found in malicious PDFs. Evaluating and choosing appropriate ML techniques for PDF malware detection. Utilizing metrics like Fl-score, precision, recall, and accuracy for comprehensive evaluation Contributing to the continuous improvement of cybersecurity measures as PDF-based threats evolve.

LITERATURE SURVEY:

Sl.no	Title	Author	Year	Dataset	Advantage	Disadvantage
1	Learn2Evade: Learning-Based Generative Model for Evading PDF Malware Classifiers	Ho Bae, Younghan Lee	2021	PDFrate-v2	The effectiveness of generated evasive PDF malware variants	It primarily targets specific classifiers
2	An Attention Mechanism for Combination of CNN and VAE for Image-Based Malware Classification	Tuan Van Dao, Hiroshi Sato	2022	unbalanced and balanced Malimg datasets	Complex malware variants can be detected	Balancing complexity and performance is a challenge
3	Self-Attentive Models for Real-Time Malware Classification	Qikai Lu1, Hongwen Zhang	2022	BODMAS PE malware dataset, BODMAS -11 and BODMAS -49. W	Flexibility and improved performance	Efficiency and low latency are crucial
4	PDF Malware Detection Based on Optimizable Decision Trees	Qasem Abu Al-Haija, Ammar Odeh and Hazem Qattous	2022	Evasive-PDFMal2022	The effectiveness of generated evasive PDF malware variants	The need more adaptive and resilient detection strategies
5	LinRegDroid: Detection of Android Malware Using Multiple Linear Regression Models-Based Classifiers	Durmuş Özkan Şahin, Sedat Akleylek	2022	Drebin dataset,	Enhanced accuracy and efficiency	The complex and non-linear characteristics of Android malware detection

Fig. 2.1 Summary of the literature survey

CHAPTER 3

3. EXISTING SYSTEM

3.1 EXISTING SYSTEM OVERVIEW

This research paper delves into the challenging problem of identifying adversarial examples against machine learning (ML) models specifically designed for classifying malicious Portable Document Format (PDF) files. Adversarial examples, small perturbations to inputs that can alter the predictions of ML models, have been studied extensively in the context of image processing and, more recently, in applications like malware classification. The unique complexity of PDF data structures, coupled with the added constraint that the generated PDFs must exhibit malicious behaviour, makes this problem particularly intricate. To tackle this challenge, the article proposes a variant of generative adversarial networks (GANs) tailored to generate evasive variant PDF malware, maintaining original malicious behaviour while avoiding detection by existing classifiers.

This research paper approach leverages the target classifier as a secondary discriminator and incorporates a new feature selection process that includes unique features extracted from malicious PDF files. The evaluation of this technique involves testing it against three representative PDF malware classifiers (Hidost'13, Hidost'16, and PDFrate-v2) and examining its effectiveness against commercial Antivirus engines from Virus Total—a first-of-its-kind analysis against such engines.

The results demonstrate the model's remarkable speed in finding evasive variants for all selected seeds against state-of-the-art PDF malware classifiers. This outcome raises significant security concerns in the face of adversaries, emphasizing the potential vulnerability of existing PDF malware classifiers. The article not only contributes to advancing our understanding of adversarial attacks on PDF malware classifiers but also underscores the need for robust defences to mitigate the security risks associated with such sophisticated attacks.

PROBLEM DEFINITION

The problem at the core of this project is the persistent and intricate challenge of -detecting malware concealed within PDF files. As PDFs are extensively utilized for

information sharing, they become a prime target for malicious activities, exploiting vulnerabilities within PDF readers or employing social engineering techniques to embed hidden malware. A critical issue arises in the development of evasive variants of PDF malware that can successfully circumvent conventional machine learning-based classifiers. This poses a substantial threat to digital security, as conventional methods may fall short in effectively identifying and neutralizing these sophisticated and adaptive threats.

The project sets out to define and address this critical problem by proposing a novel hybrid algorithm and Variational Autoencoder (VAE) approach for image-based malware detection in PDFs. The goal is to significantly enhance the accuracy and efficiency of malware detection, ultimately contributing to the broader mission of fortifying cybersecurity measures against the evolving and complex landscape of pdf-based malware threats.

3.2 ARCHIETECTURE DIAGRAM OF EXISTING SYSTEM

The architecture diagram of PDF malware prediction using Convolutional Neural Network (CNN) algorithm typically involves a multi-layered structure designed to effectively process and analyse the characteristics of PDF files for malware detection. At the core of the architecture is the CNN, a deep learning model specifically suited for image-based tasks.

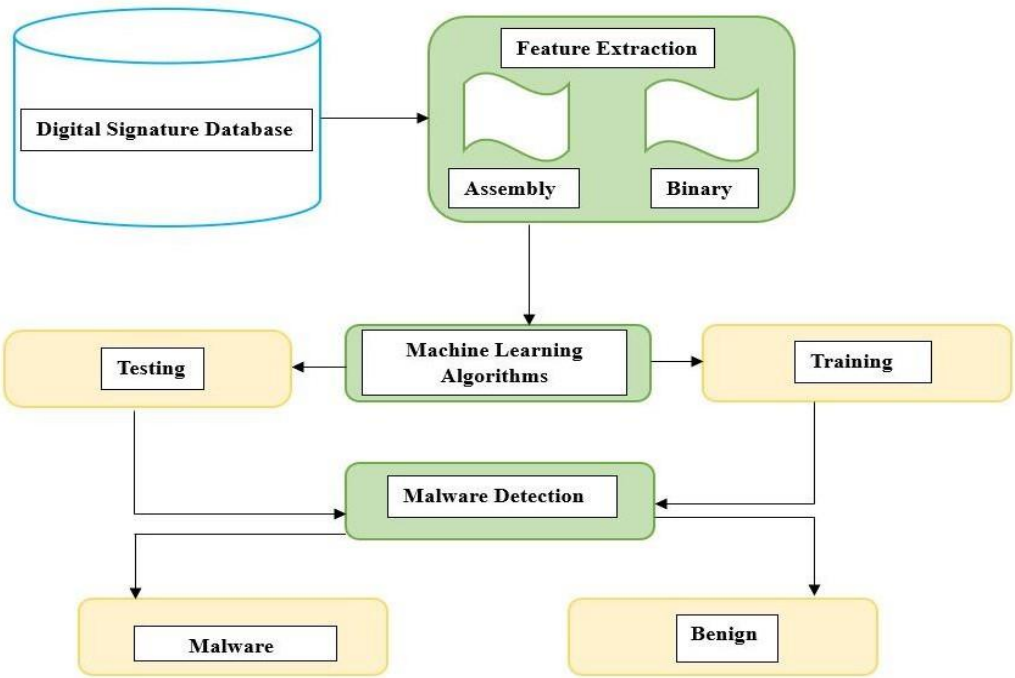

Fig: 3.1 Architecture diagram of existing system

From Fig:3.1, The initial layers of the CNN are dedicated to feature extraction, employing convolutional and pooling layers to capture hierarchical representations of the input data. In the context of PDF malware prediction, the input data may include pixel information or other relevant features extracted from the PDF files.

The architecture may also include pre-processing steps to convert PDF files into a format suitable for input into the CNN. Following feature extraction, the network incorporates fully connected layers for further abstraction and learning complex patterns within the data. The final layer typically consists of a SoftMax activation function, enabling the model to provide probability scores for different malware classes. In addition to the CNN layers, the architecture may integrate normalization techniques, dropout layers for regularization, and possibly auxiliary components like recurrent neural networks (RNNs) for sequential data analysis if the PDF malware prediction involves temporal aspects.

The training process involves optimizing the model's parameters using labelled datasets comprising both benign and malicious PDF files. The resulting trained CNN architecture can then be used for predicting whether a given PDF file contains malware based on learned patterns and features. Overall, the architecture diagram of PDF malware prediction using the CNN algorithm illustrates a comprehensive and sophisticated approach to enhancing security measures against evolving threats in the digital landscape.

CHAPTER 4

4. SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

- Processor : Intel i7 up to 4.0 GHz Speed
- RAM : 8 GB
- Hard Disk : 512 GB SSD
- Keyboard : Standard Windows Keyboard
- Mouse : Two or Three Button Mouse

4.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 10 Home
- Language : Python (version 3.7 or above)
- Front End : React Native (version 0.74)
- Frame Work : Flask (version 32.0.0)
- Database : MySQL (version 8.0)
- Anaconda Prompt (Managing Python Packages) (version 4.3)

CHAPTER 5

5. PROPOSED SYSTEM

5.1 PROPOSED SYSTEM OVERVIEW

The proposed system for detecting PDF malware hinges on the utilization of the Long Short-Term Memory (LSTM) algorithm, a specialized neural network architecture tailored for analysing sequential data. LSTMs excel in capturing temporal dependencies and patterns, making them particularly effective in discerning the subtle variations indicative of malicious elements within PDF files. By integrating LSTM into the system, it gains the capability to learn and adapt to the dynamic nature of PDF-based malware, thereby enhancing its predictive accuracy over time. The LSTM's ability to retain information over long periods ensures that the system can recognize evolving threats by capturing even the most intricate alterations in PDF file structures, thus fortifying cybersecurity defences against the ever-changing landscape of digital threats.

In essence, the strategic incorporation of LSTM within the proposed system not only underscores its adaptability to dynamic patterns inherent in PDF malware but also accentuates its efficacy in accurately detecting and classifying malicious elements. By leveraging the robust capabilities of LSTM networks, the system achieves a heightened level of sophistication in cybersecurity, bolstering defences against evolving threats in the digital realm. This approach represents a significant advancement in the field, offering a proactive and adaptive solution to combat the persistent challenges posed by PDF-based malware threats.

5.2 ARCHITECTURE DIAGRAM OF PROPOSED SYSTEM

The architecture diagram, Fig 5.1 illustrates the structural components and flow of information within a system or model. Typically, it depicts various layers, modules, or components and their interactions or connections. Each element in the diagram represents a specific function or task, and the connections between them show how data or information flows through the system. This visual representation provides a comprehensive overview of the system's design and operation, aiding in understanding its functionality, dependencies, and relationships between different parts. Architecture diagrams are valuable tools for

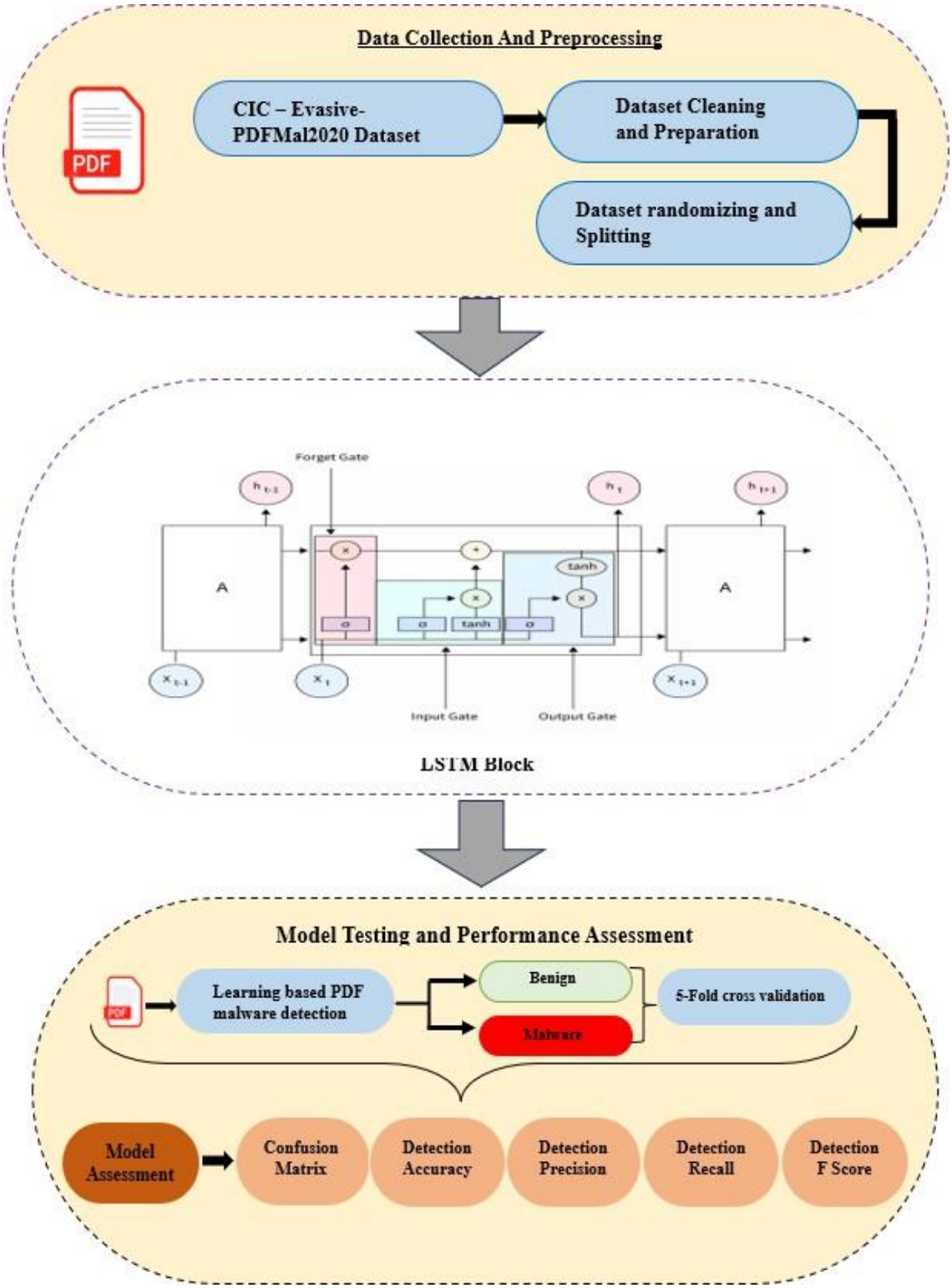

Fig 5.1 Architecture diagram of proposed system

communication, documentation, and analysis, enabling stakeholders to grasp the system's complexity and design principles effectively. They serve as blueprints for development, troubleshooting, and optimization, guiding engineers and developers throughout the implementation process.

The proposed architecture diagram for PDF malware prediction represents a systematic and comprehensive framework designed to detect and mitigate threats embedded within Portable Document Format (PDF) files. At its core is a hybrid algorithm that integrates diverse machine learning techniques for enhanced accuracy and adaptability. The architecture encompasses several key components, starting with a pre-processing module that transforms raw PDF data into a format suitable for analysis.

Feature extraction layers follow, utilizing advanced methods such as Convolutional Neural Networks (CNNs) to capture intricate patterns and characteristics inherent in PDF files. The architecture further incorporates a hybrid algorithm layer, strategically combining deep learning models and traditional machine learning approaches. This hybridization aims to exploit the strengths of each algorithm type, facilitating a more nuanced and effective detection process. Additionally, the system may involve heuristic analysis, anomaly detection, or feature engineering components, contributing to a multifaceted approach in identifying potential malware signatures.

The final classification layer leverages the insights gained from the hybrid algorithm to predict whether a given PDF file is benign or malicious. The architecture diagram's adaptability to dynamic PDF-based threats means it can respond effectively to new and evolving malware tactics that target PDF files. By integrating various techniques into a cohesive framework, the system can address the limitations of individual methods, creating a robust defence mechanism. Its modular design allows for scalability, meaning it can grow and adapt to handle increasing volumes of threats.

Moreover, the incorporation of future advancements in machine learning and cybersecurity ensures the architecture remains relevant and effective in combating emerging PDF malware threats. In essence, this architecture provides a comprehensive and forward-looking solution to the constantly evolving landscape of PDF-based cyber threats.

5.3 MODULES DESCRIPTION

- Data collection
- Pre-processing
- Feature extraction
- Modal creation
- Prediction

5.3.1 DATA COLLECTION

The data for this project is sourced from the URL provided by the University of New Brunswick's Canadian Institute for Cybersecurity (CIC). Specifically, the dataset named "pdfmal-2022" is utilized as the foundation for training, validating, and testing the proposed malware classification system. This dataset, curated by the CIC, is likely to comprise a diverse collection of PDF files with embedded malware instances. The inclusion of real-world, representative samples is crucial for the system to effectively learn and generalize patterns indicative of malicious software. The dataset is expected to encompass variations in the types of malwares, encapsulating the evolving nature of cyber threats.

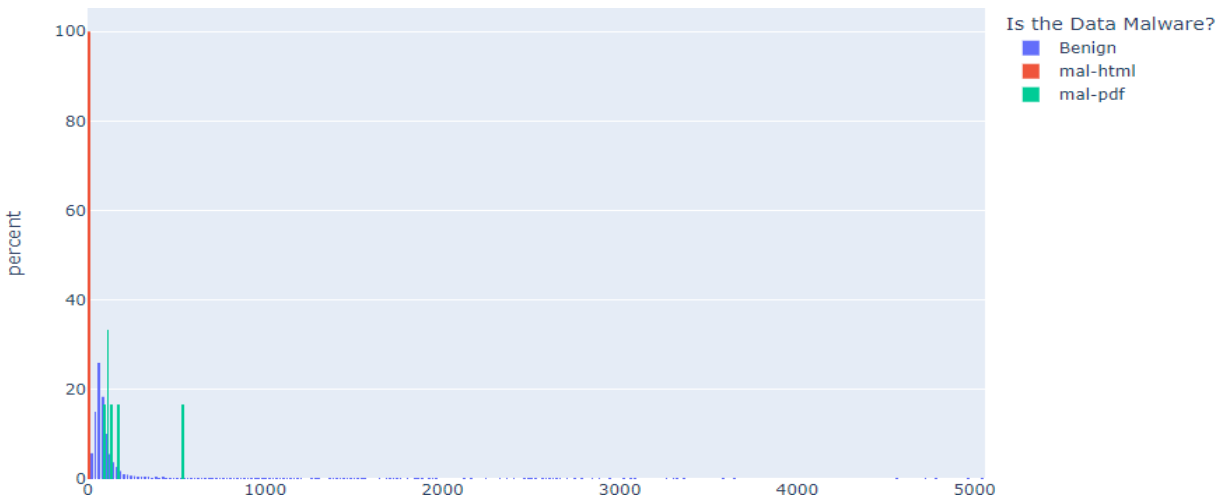

Fig 5.2 Data collection

Leveraging this dataset is fundamental for evaluating the system's performance under realistic conditions, ensuring that it is robust and adaptable to the dynamic landscape of PDF-based malware. The commitment to using datasets from reputable sources like the CIC

underscores the project's dedication to creating a reliable and effective malware classification solution.

The graph visually depicts the prevalence of malware in PDF files compared to both benign PDFs and malware in HTML files. The data suggests that malware in PDFs constitutes a significant portion of the detected threats, potentially indicating a targeted exploitation of the PDF format by malicious actors. Conversely, while malware in HTML files also pose a notable threat, it appears to be comparatively lower in frequency than malware embedded within PDF documents. This insight underscores the importance of robust detection mechanisms specifically tailored for PDF files, given their widespread use and susceptibility to exploitation.

Additionally, it highlights the need for comprehensive cybersecurity measures across various file formats to mitigate the risks posed by malware. Understanding these trends can inform the development of effective strategies to safeguard against malicious activities targeting both PDF and HTML files, thereby bolstering overall digital security.

5.3.2 PRE- PROCESSING

The pre-processing of the dataset is a crucial step in preparing the raw data from the "pdfmal-2022" dataset, obtained from the University of New Brunswick's Canadian Institute for Cybersecurity, for effective use in the malware classification system. This phase involves several key procedures to enhance the quality and suitability of the data. Initial steps typically include data cleaning, which addresses any inconsistencies, missing values, or irregularities in the dataset.

	filename	obj	endobj	stream	endstream	xref	trailer	startxref	Page	Encrypt	...	AA	OpenAction	AcroForm	JBIG2Decode	RichMedia	Launch	EmbeddedFile	XFA	Colors_2_24	label
0	999875	73	73	25	25	2	2	2	7	0	...	0	0	0	0	0	0	0	0	0	Benign
1	999870	122	122	46	46	2	2	2	22	0	...	0	0	0	0	0	0	0	0	0	Benign
2	999858	66	66	19	19	2	2	2	12	0	...	0	0	0	0	0	0	0	0	0	Benign
3	999854	72	72	22	22	2	2	2	12	0	...	0	0	0	0	0	0	0	0	0	Benign
4	999331	70	70	19	19	2	2	2	11	0	...	0	0	0	0	0	0	0	0	0	Benign

Fig 5.3 Preprocessing

Subsequently, the data may be subjected to normalization or standardization to ensure uniformity in the scale of features Feature extraction techniques may also be applied to capture essential characteristics or patterns indicative of malware. Additionally, the dataset is likely partitioned into training, validation, and testing sets to enable the evaluation of the classification system's performance. This pre-processing phase is pivotal for mitigating potential biases, reducing noise, and optimizing the dataset for training the machine learning model. The meticulous handling of data during this stage significantly contributes to the overall effectiveness and generalizability of the malware classification system.

5.3.3 FEATURE EXTRACTION

In the process of feature extraction, duplicate values are identified and removed to enhance the efficiency and accuracy of the analysis. Duplicate values, when present in the dataset, can skew the results by inflating the importance of certain features or introducing redundancy in the information. By eliminating duplicates, the feature extraction stage ensures that each unique characteristic or attribute is appropriately represented and considered during analysis.

This helps in streamlining the dataset and reducing computational overhead during subsequent stages of the analysis. Additionally, removing duplicate values helps in improving the interpretability of the extracted features, as it ensures that only relevant and distinct information contributes to the final analysis. Overall, the removal of duplicate values in feature extraction plays a crucial role in producing reliable and meaningful insights from the data, facilitating more accurate decision-making processes in various domains.

5.3.4 MODAL CREATION

The Long Short-Term Memory (LSTM) algorithm, a specialized form of recurrent neural network (RNN), excels at capturing long-term dependencies in sequential data. Unlike traditional RNNs, LSTMs incorporate gated mechanisms to selectively retain or discard information over time. At each time step, an LSTM unit receives input and previous hidden state information, then adjusts the flow of data using input, forget, and output gates.

This enables LSTMs to effectively capture complex temporal patterns, making them ideal for tasks like natural language processing, time series analysis, and malware detection.

```
duplicated_rows = data.loc[np.where(data.duplicated())]
print(duplicated_rows.head())
```

```
      filename  obj  end obj  stream  endstream  xref  trailer  startxref  \
2779    591955   63       63      22         22     2        2          2
2781    591949   68       68      20         20     2        2          2
2783    591942   44       44      12         12     2        2          2
4309    499051   85       85      28         28     1        1          1
4314    499038   51       51      12         12     1        1          1

      Page  Encrypt  ...  AA  OpenAction  AcroForm  JBIG2Decode  RichMedia  \
2779    10        0  ...   0           0         0            0          0
2781    12        0  ...   0           0         0            0          0
2783     8        0  ...   0           0         0            0          0
4309    15        0  ...   0           0         0            0          0
4314    12        0  ...   0           0         0            0          0

      Launch  EmbeddedFile  XFA  Colors_2_24   label
2779       0             0    0            0  Benign
2781       0             0    0            0  Benign
2783       0             0    0            0  Benign
4309       0             0    0            0  Benign
4314       0             0    0            0  Benign
```

Fig 5.4 Modal creation

Through its unique architecture, the LSTM algorithm addresses the vanishing gradient problem encountered in traditional RNNs, allowing it to learn and retain relevant information over extended periods, thus enhancing its suitability for various sequential data analysis tasks.

a. Input gate i_t, forget gate f_t, and output gate o_t calculations:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

b. Candidate cell state $\tilde{C}_t$ calculation:

$$\tilde{C}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

c. Cell state C_t update using input, forget, and candidate gates:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

d. Hidden state h_t calculation using the output gate and the updated cell state:

$$h_t = o_t \odot \tanh(C_t)$$

In an LSTM (Long Short-Term Memory) unit, the calculation of the hidden state $\mathbf{h_t}$ at a given time step \(t \) involves several interconnected components, including gates and memory cells, which facilitate the model's ability to capture and retain information over long sequences. These components include the input gate $\boldsymbol{i_t}$, forget gate $\mathbf{f_t}$, output gate $\mathbf{o_t}$,

and the candidate cell state $\mathbf{C_t}$ Each gate is responsible for controlling the flow of information within the LSTM unit.

$$h_t = \sigma\left(W_{hx}x_t + W_{hh}h_{t-1} + b\right)$$

The input gate determines how much new information is added to the cell state, the forget gate regulates what information is discarded from the previous cell state, and the output gate determines how much of the cell state is exposed as the output. Additionally, the candidate cell state represents the new candidate values that could be added to the cell state. Through these interconnected mechanisms, LSTM units are capable of effectively learning and retaining information over long sequences, making them particularly well-suited for tasks involving temporal dependencies and patterns.

Here, σ represents the sigmoid activation function, tanh represents the hyperbolic tangent function, ⊙ represents element-wise multiplication, and b represent weight matrices and biases, respectively, for each gate and cell. These equations govern the flow of information through an LSTM unit, allowing it to learn and retain information over long sequences, thereby addressing the vanishing gradient problem often encountered in traditional RNNs.

$$h_t = \sigma\left(W_{hx}x_t + W_{hh}h_{t-1} + b\right)$$

CHAPTER 6

6. IMPLEMENTATIONS

6.1 LIBRARIES USED

```
import os
import pandas as pd
from wordcloud import WordCloud
import matplotlib.pyplot as plt

import re
import nltk
from nltk.corpus import stopwords
import PyPDF2
import io
import zipfile

import fitz  # PyMuPDF

from pdf2image import convert_from_path
from IPython.display import display, Image
```

Fig.6.1 Libraries

From fig 6.1 it explains the libraries

- **os**: Provides a way to use operating system dependent functionality.
- **pandas (as pd)**: An open-source library providing high-performance, easy-to-use data structures, and data analysis tools
- **wordcloud (WordCloud)**: A tool for creating word clouds from text.
- **matplotlib.pyplot (as plt)**: A plotting library for creating static, interactive, and animated visualizations in Python
- **re**: Offers operations for Python regular expressions.
- **nltk**: A leading platform for building Python programs to work with human language data
- **PyPDF2**: A library for reading PDF files and extracting text content.
- **io**: Provides the Python interfaces to stream handling.
- **zipfile**: Allows reading and writing ZIP archive files.
- **fitz (PyMuPDF)**: A Python binding for MuPDF – a lightweight PDF and XPS viewer.
- **pdf2image**: Converts PDF files into images.
- **IPython.display**: Tools for displaying objects in Python scripts, shell, and notebooks.

6.2 PRE-PROCESSING

```
# Function to extract text from PDF
def extract_text_from_pdf(pdf_path):
    with open(pdf_path, 'rb') as file:
        reader = PyPDF2.PdfFileReader(file)
        text = ''
        for page_num in range(reader.numPages):
            page = reader.getPage(page_num)
            text += page.extract_text()
    return text
```

Fig.6.2 Code for Extract Text from Pdf

From the fig 6.2 it explains the Python code snippet for a function named extract_text_from_pdf. This function is designed to extract text from a PDF file using the PyPDF2 library. Here's a breakdown of the code:

- **Function Definition**: extract_text_from_pdf(pdf_path) defines the function with one parameter, pdf_path, which is the path to the PDF file.
- **Opening the PDF**: The with open(pdf_path, 'rb') statement opens the PDF file in binary read mode.
- **Reading the PDF**: PyPDF2.PdfFileReader(file) is used to read the PDF file content.
- **Extracting Text**: An empty string text is initialized, and a for loop iterates over all the pages using reader.numPages. For each page, page.extract_text() extracts the text and appends it to text .
- **Return Value**: The function returns the text variable, which contains all the extracted text from the PDF

This function is useful for automating the process of extracting text from PDF files for further processing or analysis.

6.3 CODE TO PRE-PROCESS TEXT

```
def preprocess_text(text):
    # Remove non-alphanumeric characters
    text = re.sub(r'[^a-zA-Z0-9\s]', '', text)

    # Convert to lowercase
    text = text.lower()

    # Tokenization and removal of stop words
    stop_words = set(stopwords.words('english'))
    tokens = nltk.word_tokenize(text)
    filtered_tokens = [word for word in tokens if word.isalnum() and word not in stop_words]

    return ' '.join(filtered_tokens)
```

Fig 6.3 Code to Pre-Process Text

From the fig 6.3 it explains about the Function Definition

The function extract_text_from_pdf is defined. It takes a single argument, pdf_path, which represents the path to a PDF file.

- **Opening the PDF File:** The with open(pdf_path, 'rb') statement opens the specified PDF file in binary read mode ('rb'). This allows us to read the content of the file.
- **Reading PDF Content:** The PyPDF2.PdfFileReader(file) creates a reader object to read the PDF content. We can use this object to access information about the PDF, such as the number of pages (reader.numPages).
- **Extracting Text from Pages:** The for loop iterates over each page in the PDF. For each page, page.extract_text() extracts the text content from that page. The extracted text is then appended to the text variable.
- **Returning the Extracted Text:** Finally, the function returns the concatenated text containing all the extracted text from the entire PDF.

This function is useful for tasks like text mining, searching for specific keywords, or analysing the content of PDF documents programmatically.

6.4 TOKENIZATION AND REMOVAL OF STOP WORDS

```
stop_words = set(stopwords.words('english'))
tokens = nltk.word_tokenize(text)
filtered_tokens = [word for word in tokens if word.isalnum() and word not in stop_words]
```

Fig 6.4 Tokenization and Removal of Stop Words

From the fig 6.4 it explains the Python code snippet that demonstrates how to filter out stopwords from a given text using the Natural Language Toolkit (nltk). Here's a breakdown of the code:

- **stop_words:** A set of English stopwords provided by nltk, which are common words like 'the', 'is', 'in', etc., that are often filtered out before processing text.
- **tokens:** A list of words created by tokenizing the input text, which means breaking the text into individual words or tokens.
- **filtered_tokens:** A list comprehension that filters out any tokens that are not alphanumeric (using word.isalnum()) and are not in the stop_words set

This code is typically used in text analysis and natural language processing to clean and prepare text data for further tasks such as sentiment analysis, topic modeling, or machine learning

6.5 PREPROCESS ALL PDF FILES IN A DIRECTORY

```
def preprocess_directory(directory_path):
    preprocessed_texts = []
    for filename in os.listdir(directory_path):
        if filename.endswith(".pdf"):
            pdf_path = os.path.join(directory_path, filename)
            preprocessed_text = preprocess_pdf(pdf_path)
            preprocessed_texts.append(preprocessed_text)
    return preprocessed_texts
```

Fig 6.5 Preprocess All Pdf Files in A Directory

From the fig 6.5 it explains about the Python function named preprocess_directory. This function is designed to preprocess text from PDF files within a specified directory. Here's a breakdown of its functionality:

- Function Definition: It defines a function preprocess_directory that takes directory_path as an argument.
- Text Preprocessing: It initializes an empty list preprocessed_texts to store the pre-processed text from each PDF.
- PDF File Handling: The function iterates over all files in the given directory, processes only those ending with ".pdf", and appends the preprocessed text to the list.
- Return Value: After processing all PDF files, it returns the list preprocessed_texts containing the preprocessed texts.

This function relies on another function preprocess_pdf, which is not shown in the image but is assumed to handle the actual preprocessing of the PDF text.
The preprocess_directory function is a useful way to automate the preprocessing of multiple PDFs in a batch.

6.6 FUNCTION TO CONVERT PDF TO PNG

```
def convert_pdf_to_png(pdf_path, output_directory):
    images = convert_from_path(pdf_path, output_folder=output_directory, fmt='png')
    return images
```

Fig 6.6 Function to Convert PDF To PNG

From the fig 6.6 it explains about the Python function named convert_pdf_to_png. Here's a breakdown of its components:

- Function Name: convert_pdf_to_png
- Parameters:
  - pdf_path: The file path of the PDF to be converted.
  - output_directory: The directory where the PNG images will be saved.

- Process:
    - The function uses a method convert_from_path to convert the PDF located at pdf_path into PNG images.
    - The images are saved in the output_directory.
- Return Value: The function returns the variable images, which contains the converted PNG images.

This function is useful for automating the conversion of PDF files into PNG format, which can be helpful in various applications such as document processing or web development.

CHAPTER 7

7. RESULT AND DISCUSSION

Performance metrics are essential tools for assessing the effectiveness of machine learning models. Accuracy, the most intuitive metric, provides an overall measure of correctness by quantifying the proportion of correctly classified instances. However, in scenarios with imbalanced class distributions, accuracy may not adequately capture the model's performance, leading to misleading conclusions. Precision and recall offer more nuanced insights by focusing on specific classes. Precision measures the proportion of true positive predictions among all positive predictions, emphasizing the model's ability to avoid false positives.

In contrast, recall, also known as sensitivity, quantifies the proportion of true positive predictions among all actual positive instances, highlighting the model's ability to identify relevant instances of a class. Loss, on the other hand, guides the optimization process during training by quantifying the discrepancy between predicted and true values, aiding in model refinement. Evaluating a model using a combination of these metrics offers a comprehensive understanding of its performance and suitability for the task at hand, ensuring robust and reliable results.

7.1 ACCURACY

Accuracy is a fundamental performance metric used to assess the overall effectiveness of a machine learning model. It measures the proportion of correctly

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

classified instances among the total number of instances evaluated by the model. A high accuracy score indicates that the model's predictions closely match the actual labels, reflecting its ability to make correct decisions across different classes or categories.

Accuracy is particularly valuable for tasks where each class is of equal importance and there is a balanced distribution of instances among classes. However, it's important to note that accuracy alone may not provide a complete picture of a model's performance,

especially in scenarios with imbalanced class distributions or when different classes have varying levels of significance.

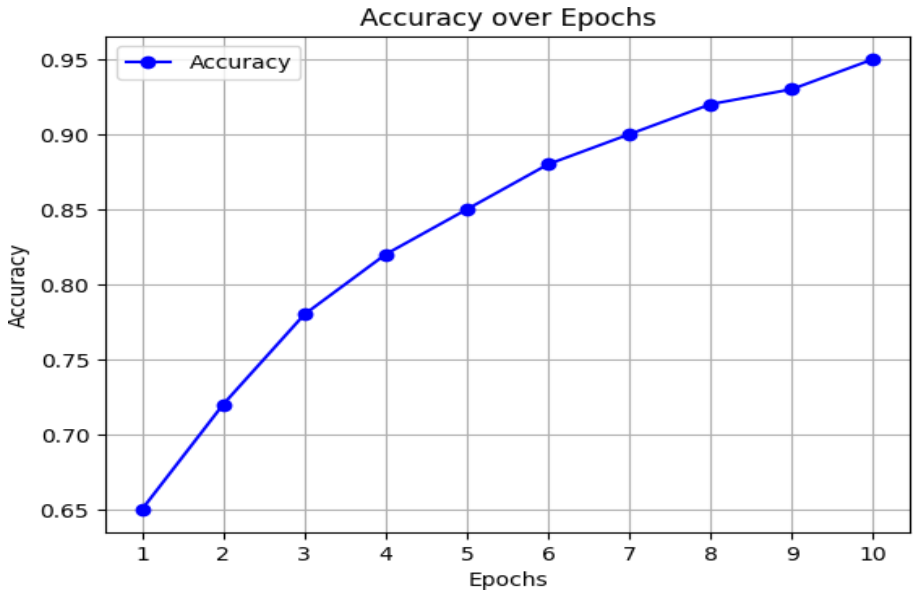

Fig 7.1 Accuracy graph

7.2 CONFUSION MATRIX

A confusion matrix is a performance measurement tool used in machine learning to evaluate the accuracy of a classification model. It provides a comprehensive summary of the model's predictions by tabulating the actual class labels against the predicted class labels.

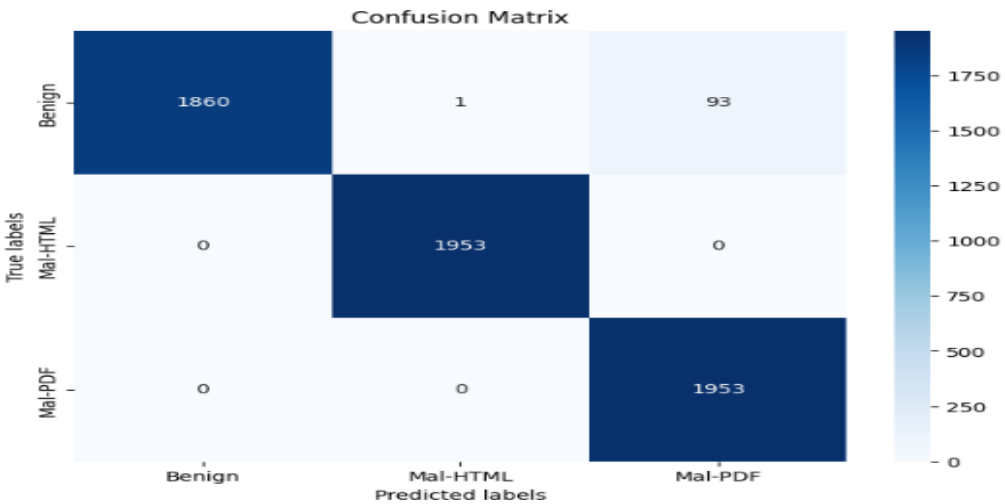

Fig 7.2 Confusion Matrix

The matrix is structured as a grid, with rows representing the true classes and columns representing the predicted classes. Each cell in the matrix corresponds to the number of instances that belong to the true class and were predicted as belonging to the predicted class. This arrangement allows for a detailed analysis of the model's behaviour, revealing the types of errors it makes.

From the confusion matrix, various performance metrics such as accuracy, precision, recall, and F1-score can be derived, providing insights into the model's strengths and weaknesses across different classes. Overall, the confusion matrix serves as a powerful diagnostic tool, guiding model refinement and optimization efforts to enhance classification performance.

7.3 COMPARISON OF PARAMETERS

Classification Report:

	precision	recall	f1-score	support
Benign	1.00	0.95	0.98	1954
Mal-HTML	1.00	1.00	1.00	1953
Mal-PDF	0.95	1.00	0.98	1953
accuracy			0.98	5860
macro avg	0.98	0.98	0.98	5860
weighted avg	0.98	0.98	0.98	5860

Fig 7.3 Comparison of parameters

Precision, recall, and F1 score are key performance metrics used to assess the effectiveness of classification models in machine learning. Precision quantifies the proportion of true positive predictions among all positive predictions made by the model, indicating its ability to avoid false positives. A high precision score signifies that the model makes accurate positive predictions with minimal false alarms. On the other hand, recall measures the proportion of true positive predictions among all actual positive instances in the dataset, illustrating the model's capability to capture all relevant instances of a class.

A high recall score indicates that the model effectively identifies most positive instances, minimizing the risk of missing relevant data points. The F1 score, a harmonic mean of precision and recall, provides a balanced assessment of a model's performance, considering both false positives and false negatives. It offers a single metric that reflects the trade-off between precision and recall, allowing for a comprehensive evaluation of the model's classification accuracy.

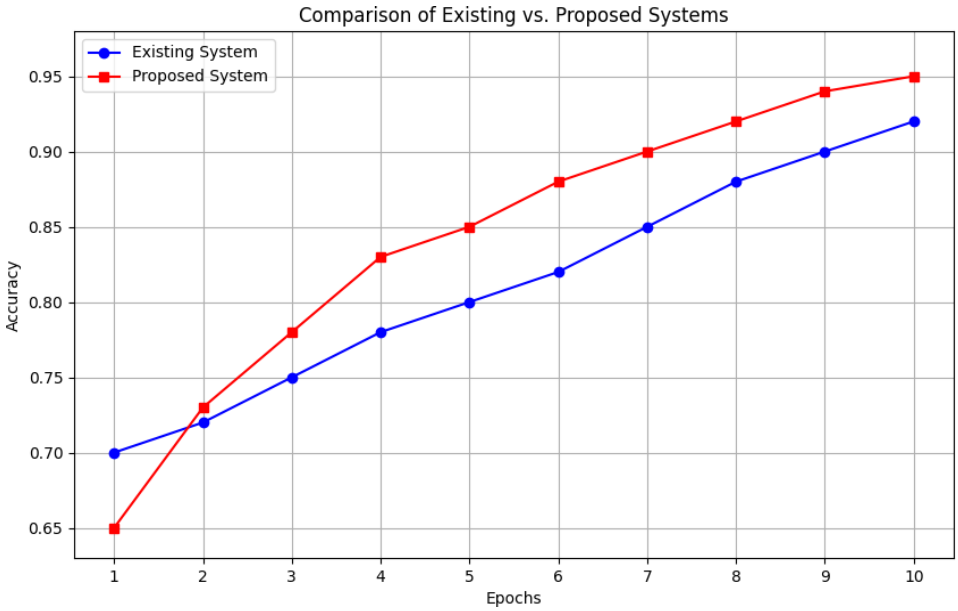

Fig 7.4 Comparisons of existing vs. proposed systems

CHAPTER 8

8. CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

In conclusion, the integration of the Long Short-Term Memory (LSTM) algorithm within the proposed system for detecting PDF malware signifies a significant leap forward in cybersecurity. By harnessing the LSTM's specialized capabilities in analysing sequential data and capturing temporal dependencies, the system gains a heightened ability to discern subtle variations indicative of malicious elements within PDF files. This strategic integration not only enhances the system's predictive accuracy over time but also fortifies cybersecurity defences against the ever-evolving landscape of digital threats. Ultimately, the utilization of LSTM represents a proactive and adaptive approach to combating PDF-based malware, offering a sophisticated solution to the persistent challenges faced in safeguarding digital information and networks. As such, the incorporation of LSTM marks a pivotal advancement in cybersecurity, paving the way for more effective and robust defences against emerging threats in the digital realm.

8.2 FUTURE ENHANCEMENTS

In future work, exploring ensemble techniques, refining training methodologies with diverse datasets, incorporating explainable AI techniques, and implementing mechanisms for continuous monitoring and adaptation are crucial. Ensemble methods could combine various detection models for improved performance, while diversified datasets would enhance generalization. Explainable AI could provide insights into the system's decision-making, boosting trust. Lastly, mechanisms for ongoing updates and feedback loops would ensure the system remains effective against evolving PDF malware threats.

REFERENCE

[1] J. Saxe and K. Berlin, “Deep neural network based malware detection using two dimensional binary program features,” in Proc. 10th Int Conf. Malicious Unwanted. Softw., 2015, pp. 11–20.

[2] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, “Droid-Sec: Deep learning in android malware detection,” ACM SIGCOMM Comput. Commun. Rev., vol. 44, no. 4, pp. 371–372, 2014.

[3] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda, “Scalable, behavior-based malware clustering,” in Proc. Netw. Distrib. Syst. Secur. Symp., 2009, vol. 9, pp. 8–11.

[4] J. Jang, D. Brumley, and S. Venkataraman, “Bitshred: Feature hashing malware for scalable triage and semantic analysis,” in Proc. 18th ACM Conf. Comput. Commun. Secur., 2011, pp. 309–320.

[5] M. Cova, C. Kruegel, and G. Vigna, “Detection and analysis of drive-bydownload attacks and malicious javaScript code,” in Proc. 19th Int Conf. World Wide Web, 2010, pp. 281–290.

[6] M. A. Rajab, L. Ballard, N. Lutz, P. Mavrommatis, and N. Provos, “CAMP: Content-agnostic malware protection,” in Proc. Netw. Distrib. Syst. Secur. Symp., 2013.

[7] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna, “COMPA: Detecting compromised accounts on social networks,” in Proc. Netw. Distrib. Syst. Secur. Symp., 2013.

[8] G. Stringhini, C. Kruegel, and G. Vigna, “Shady paths: Leveraging surfing crowds to detect malicious web pages,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., 2013, pp. 133–144.

[9] J. Schlumberger, C. Kruegel, and G. Vigna, “Jarhead analysis and detection of malicious Java applets,” in Proc. 28th Annu. Comput. Secur. Appl. Conf., 2012, pp. 249–257.

[10] P. Laskov and N. Šrndi´c, “Static detection of malicious JavaScript-bearing PDF documents,” in Proc. 27th Annu. Comput. Secur. Appl. Conf., 2011, pp. 373–382.

[11] G. Kakavelakis and J. Young, “Auto-learning of SMTP TCP transportlayer features for spam and abusive message detection,” in Proc. 25th Int. Conf. Large Installation Syst. Admin., 2011, p. 18.

[12] N. Šrndic and P. Laskov, “Detection of malicious PDF files based on hierarchical document structure,” in Proc. 28th Annu. Netw. Distrib. Syst. Secur. Symp., 2013, pp. 1–16.

[13] C. Smutz and A. Stavrou, “Malicious PDF detection using metadata and structural features,” in Proc. 28th Annu. Comput. Secur. Appl. Conf., 2012, pp. 239–248.

[14] C. Smutz and A. Stavrou, “When a tree falls: Using diversity in ensemble classifiers to identify evasion in malware detectors,” in Proc. Netw. Distrib. Syst. Secur. Conf., 2016.

[15] N. Šrndi´c and P. Laskov, “Hidost: A static machine-learning-based detector of malicious files,” EURASIP J. Inf. Secur., vol. 2016, no. 1, 2016, Art. no. 45.

[16] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” vol. 1050, p. 20, 2015.

[17] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial machine learning at scale,” 2016, arXiv:1611.01236.

[18] S. Alfeld, X. Zhu, and P. Barford, “Data poisoning attacks against autoregressive models,” in Proc. 30th AAAI Conf. Artif. Intell., 2016, pp. 1452–1458.

[19] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, “Adversarial attacks on neural network policies,” 2017, arXiv:1702.02284.

[20] W. Hu and Y. Tan, “Generating adversarial malware examples for blackbox attacks based on GAN,” 2017, arXiv:1702.05983.

[21] D. Maiorca, B. Biggio, and G. Giacinto, “Towards adversarial malware detection: Lessons learned from PDF-based attacks,” ACM Comput. Surv., vol. 52, no. 4, pp. 1–36, 2019.

[22] “SonicWall detects, reports dramatic rise in fraudulent PDF files in Q1 2019,” 2019. [Online]. Available: https://www.sonicwall.com/ news/sonicwall-detects-reports-dramatic- rise-in-fraudulent-pdf-files-inq1-2019/

[23] M. Heiderich, M. Niemietz, F. Schuster, T. Holz, and J. Schwenk, “Scriptless attacks: Stealing the pie without touching the sill,” in Proc. ACM Conf. Comput. Commun. Secur., 2012, pp. 760–771.