Malware Detection Framework Using Hybrid Deep Learning Algorithms

MSc Cybersecurity
Research Project

Ayaan Khan
Student ID: X21199728

School of Computing

National College of Ireland

Supervisor: Michael Pantridge

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Ayaan Khan		
Student ID:	X21199728		
Programme:	MSc Cybersecurity	**Year:**	2022-2023
Module:	MSc Research Project and Internship		
Supervisor:	Michael Pantridge		
Submission Due Date:	18-09-2023		
Project Title:	Malware Detection Framework Using Hybrid Deep Learning Algorithms		
Word Count:	7152	**Page Count**	26

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Ayaan Khan

Date: 18-09-2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Malware Detection Framework Using Hybrid Deep Learning Algorithms

Ayaan Khan
x21199728

Abstract: -

As the complexity and stealth of malware approaches continue to advance, conventional signature-based detection technologies have challenges in keeping up with these developments. Considering this, scholars and professionals have resorted to employing sophisticated machine learning methodologies, including deep learning, to augment the precision of malware identification.

During the data preparation stage, the initial binary data is converted into appropriate input formats that are compatible with deep learning models. The process of feature extraction includes the retrieval of static information, such as opcode sequences, as well as dynamic features, such as system call sequences, from samples of malicious software. The classification step ultimately utilizes an ensemble methodology for decision-making, wherein the collective outputs of distinct deep learning models are amalgamated to formulate a conclusive forecast.

The initial findings indicate that the utilization of hybrid deep learning techniques in the Malware Detection Framework leads to enhanced detection performance in comparison to using individual models. The utilization of the ensemble technique significantly improves the overall resilience of the detection system, efficiently discerning both established and emerging malware variations.

Keywords: Malware detection, deep learning, hybrid models.

1. Introduction: -

1.1 Background: -

One of the most crucial aspects of contemporary technical equipment is security. Important information can be lost, stolen, or misused without inadequate security. While most internet-based programmes are safe to use, some are dangerous. Malware is a term for software specifically designed to harm a user’s data. Malicious software is nothing new. Malware has a long history that dates to the 1970s (Costin and Zaddach, 2018). The purpose of malware development is to hurt users. To combat malware, security professionals are continually searching for solutions. Malware can delete private information, steal information, track the user, and many other damaging activities. According to research, the entire cost of malware attacks was $3 trillion USD in 2015 and is projected to increase to $6 trillion USD by 2022 (Rimon and Haque, 2022). Therefore, it is critical to find and remove malware files before they do damage to the system. One of the most crucial cybersecurity duties is protecting digital systems against viruses. This is because a single attack can corrupt data and cause significant losses. Security experts must thus constantly develop new defences against malware (Genç, Lenzini and Ryan, 2018).

1.2 Importance: -

Traditional malware detection methods, such as signature-based methods, have difficulties identifying novel and undiscovered malware varieties. Computer systems, networks, and user data are under attack from the complexity and amount of malware (Rimon and Haque, 2022). By autonomously deriving intricate patterns and attributes from malware samples, deep learning algorithms present a viable solution and enable more precise detection. In comparison to standalone algorithms, the usage of hybrid deep learning algorithms, which incorporate various deep learning approaches, has demonstrated enhanced detection accuracy (Genç, Lenzini and Ryan, 2018). Hybrid algorithms can improve malware detection by using the advantages of many models. The processing of vast and complicated information, such as binary files, network traffic, and system logs essential to malware detection, makes this particularly crucial.

To stop harm from being done and to lessen the effects of malware assaults, real-time detection is essential. Fast and effective detection may be provided by hybrid deep learning systems, allowing for quick reactions to new threats (Vinayakumar et al., 2019). The capacity to identify and react in real-time is crucial, given the escalating speed and sophistication of malware assaults. Deep learning algorithms also benefit from generalisation and flexibility. They are flexible to changing malware landscapes because they can efficiently detect new and hidden malware samples (Rimon and Haque, 2022). This is essential because hackers frequently alter their methods to avoid discovery. The efficiency of hybrid deep learning algorithms in detecting malware is influenced by their capacity to generalise successfully to fresh samples.

The development of a malware detection system employing hybrid deep learning algorithms is a research area with significance for academia as well as business (Rudin, 2019). Academically, it enhances knowledge of deep learning methods for malware detection and contributes to cybersecurity. In the workplace, it may be used to improve the security of computer systems, networks, and IoT gadgets, helping businesses and people safeguard their digital assets.

1.3 Research question and objectives: -

The research study aims to explore the effectiveness of Hybrid Deep Learning Algorithms in malware detection frameworks.

The research questions of the study are:

- What is the need for a malware detection framework in the current environment?
- What is the role and importance of hybrid deep learning algorithms?
- Can Hybrid deep learning algorithms strengthen the efficiency of malware detection?
- What are the challenges that must be considered using hybrid deep learning algorithms in the malware detection framework?

The following are the objectives of addressing the research question and achieving the research aims:

- To examine the need for a malware detection framework
- To discuss the role of hybrid deep learning algorithms.
- To analyse the use of hybrid deep learning algorithms in malware detection
- To explore the challenges and concerns of using hybrid deep learning algorithms in malware detection framework

1.4 Limitations: -

While creating a malware detection framework with hybrid deep learning algorithms has the potential to increase malware detection's precision and effectiveness, there are still specific issues that need to be resolved and research gaps that need to be filled. The accessibility and representativeness of the malware samples utilised for training and evaluation may be a drawback of the experiment (Jang-Jaccard and Nepal, 2014). Another possible area for development is the suggested framework's generalizability to other malware kinds and developing attack tactics. Even though deep learning algorithms have demonstrated an excellent capacity to generalise, there may still be limits in the ability to identify novel and undiscovered malware variants or sophisticated attack paths that were absent from the training dataset.

1.5 Assumptions: -

The following assumptions are made:

- To guarantee the efficacy and generalizability of the proposed system, the dataset should encompass a wide range of malware kinds, including known and undiscovered strains.

- The chosen assessment parameters, such as accuracy, precision, recall, and F1 score often used have inherent limitations. They cannot fully reflect the efficacy of detection, such as the ability to identify malware variants that have yet to be observed.

- The results and conclusions from the analyses on the chosen dataset apply to actual malware detection settings. It is acknowledged, nonetheless, that the performance of the suggested framework might change when used with various datasets or operational settings.

- The study will abide by ethical standards and principles, which include getting the required approvals and protecting private rights while gathering and using malware samples and related data.

1.6 Outline the structure of the report: -

Chapter	**Description**
1. Introduction	This chapter introduces the chapter and its motivation in detail. It has highlighted the research aims, objectives as well as research questions.
2. Related work	This chapter has reviewed several pieces of literature on malware detection frameworks using hybrid deep learning algorithms.
*3.*Research Methodology	This chapter has discussed the research procedure as an evaluation methodology in detail.
4. Design Specification	This chapter has identified and presented the techniques, framework or architecture that underlie the implementation and the association requirements.
5. Implementation	This chapter has chapter discussed the implementation of the proposed solution.
6. Evaluation	This chapter has presented a detailed analysis of the results and significant findings of the study. It has also evaluated the implications of these findings from academic and practitioner perspectives.

*7.*Conclusion and Future	This chapter has concluded the overall study has presented future work on malware detection framework using hybrid deep learning algorithms.

2. Related Work: -

2.1 Overview of Malware Detections: -

Malware detection is critical with the ubiquity of malware on the Internet as it functions as an early warning system for computer security against malware and cyber threats. According to (Smmarwar, Gupta and Kumar, 2022), malware detection stops hackers from accessing the computer and ensures data security. To infect the computer, malicious software disguises itself as a trustworthy application. Vinayakumar et al. (2019) highlighted phishing emails, bogus installers, infected attachments, and phishing websites as the most frequent methods of installation. The authors revealed that hackers make persuading consumers to install malware appealing. Aslan and Yilmaz (2021) believe that people often are unaware of the actual appearance of malware because of how legitimate looking the programme seems. The authors highlighted how malicious software is loaded on a computer. Malware conceals itself after installation in a variety of computer folders. If the virus is sophisticated enough, it can have direct access to the operating system and starts to record private information and encrypt files. In this consideration, the developers or specialists develop a malware detection framework to prevent such consequences of heavy loss. As per Aslan and Yilmaz (2021), it efficiently identifies malware using various methods and techniques.

According to Jang-Jaccard and Nepal (2014), malware detection is a key component of cybersecurity that help to detect, stop, warn about, and respond to malware attacks. Jang-Jaccard and Nepal (2014) discussed Check summation, application allow listing, and signature-based detection as examples of conventional malware detection methods. Lu et al. (2021) discussed that advanced malware detection solutions, on the other hand, use machine learning and artificial intelligence (AI) to search for and identify new and undiscovered malware threats proactively. The authors discussed that a subset of machine learning's deep learning algorithms had drawn attention for their capacity to analyse vast volumes of data and spot intricate patterns. Real-time malware detection is essential to recognise threats and take appropriate action quickly. As per Haq, Khan and Akhundzada (2021), real-time detection is a good application for deep learning models, which are renowned for their speedy data processing. Real-time protection is made possible by hybrid deep learning algorithms, increasing the detection system's speed and effectiveness. Another author, Rimon and Haque (2022), discussed that dynamic deep learning models use deep learning in conjunction with heuristic techniques to independently detect and eliminate contemporary malware. These solutions offer a proactive defence system and adapt to changing threats.

2.2 Hybrid Deep Learning Algorithms: -

According to Alzaylaee, Yerima and Sezer (2020), hybrid learning models combine the two types to use each's advantages. Bayesian deep learning, Bayesian GANs, and Bayesian conditional GANs are a few instances of hybrid models. Thanks to hybrid learning models, Costin and Zaddach (2018) revealed that deep learning with uncertainty may now be incorporated into a broader range of business situations. Better performance and model explain ability can be achieved in this way, which might lead to more broad adoption. Other authors, Lu et al. (2021), highlighted that as deep learning begins to be incorporated into probabilistic programming languages, expect to see more deep learning techniques obtain Bayesian analogues. When compared to conventional approaches, our method demonstrated improved detection performance. Another study by Naeem, Alshammari and Ullah (2022) used a deep convolutional neural network-based transfer learning method to classify malware photos on Android devices. The hybrid approach has great success identifying and categorising malware samples. However, the hybrid model had higher computational needs than the single model and traditional machine learning methods. Research by Akhtar and Feng (2022) used the CNN-LSTM method, a hybrid deep learning methodology, for real-time virus detection. As per the authors, the suggested CNN-LSTM model increased the malware detection accuracy. Other hybrid classifiers and machine learning algorithms were also tested in the study, but the CNN-LSTM model produced the best outcomes. Another research by Costin and Zaddach (2018) suggested a hybrid deep learning approach for Internet of Things (IoT) device virus detection. This system combines many deep learning techniques and produces encouraging outcomes in identifying multi-class threats, including attacks.

2.3 Impact of Hybrid Deep Learning Algorithms on Malware Detection Framework: -

According to Muzaffar et al. (2022), the use of deep learning algorithms to improve the accuracy and efficiency of malware detection systems, a key element of cybersecurity, has been studied by researchers. The authors demonstrated that a hybrid deep learning technique combines two or more deep learning models to improve the efficiency of the malware detection system. Research by Lu et al. (2020) explored that a hybrid deep learning approach that combines a deep belief network

(DBN) with a gated recurrent unit (GRU) has been proposed for the aim of identifying Android malware. This method has been shown to be more accurate than earlier deep learning models. The authors also discussed that creating a hybrid deep network architecture for Android virus detection was spurred on by the success of deep learning methods in feature representation learning. As per Zhu et al. (2021), an analysis of deep learning's application to Android malware detection highlights the significance of the technology in preventing malware threats and lowering the need for human expert involvement. The study of several layers of representations and the creation of scalable models that can process enormous quantities of data are made possible by deep learning.

Naway and Li (2018) highlighted that a high-performance system had been created for virus detection that blends deep learning with feature selection techniques. This method shows how deep understanding might increase malware detection precision. However, the authors further revealed that the technique could be computationally costly. In this consideration, the authors discussed that deep learning algorithms can automatically extract data from malware samples. Other authors, Rimon and Haque (2022), stated that by merging deep learning models with other machine learning models, a hybrid deep learning method can increase the effectiveness of feature extraction. For feature extraction and malware classification, for instance, a hybrid deep learning framework that combines backpropagation (BP) and particle swarm optimisation (PSO) has been presented, and it obtained good accuracy with little computing cost. A similar kind of study conducted by Vinayakumar et al. (2019) stated that deep learning models are well-suited for real-time virus detection since they can swiftly handle vast volumes of data. The detection method may be made even faster and more effective using a hybrid deep learning algorithm, enabling real-time malware detection.

2.4 Challenges and Concerns for Malware Detection Framework Using Hybrid Deep Learning Algorithms: -

As per Smmarwar, Gupta, and Kumar (2022), hybrid deep learning techniques have shown much potential for improving malware detection accuracy. They also need to go beyond a few hurdles to develop malware detection technologies that are more effective and dependable. For malware detection, selecting relevant components that can effectively distinguish between harmful and benign software is required. Another author, Ullah, Srivastava, and Ullah (2022), said that Hybrid deep learning methods must leverage static and dynamic properties, such as API request patterns and statistical data, to improve detection accuracy. However, selecting the ideal collection of features and their weights could be challenging and need extensive testing and fine-tuning.

As per Vinayakumar et al. (2019), selecting crucial elements that can effectively distinguish between harmful and benign software is vital for hybrid deep learning systems. Attackers take advantage of this information to produce malware variants that are resistant to detection since different features have variable degrees of capacity to discriminate between harmful behaviours. Because of this, Vinayakumar et al. (2019) emphasised that most malware detection systems now use hybrid traits rather than a single kind to give a complete picture. Finding the perfect balance between feature weights and features may be challenging and need much testing and adjusting. Naeem, Alshammari, and Ullah (2022) stated that hybrid deep learning algorithms commonly mix multiple models, like CNNs and LSTMs, to benefit from one model's advantages in sequential data processing and image-based analysis. However, the complexity of these models grows with their integration, resulting in longer calculation times and, consequently, more significant resource requirements. It is critical to strike a balance between model complexity and detection performance. Ullah, Srivastava, and Ullah (2022) stated deep learning models, particularly those with intricate architectures, are sometimes referred to as "black boxes," making it challenging to understand how they make decisions. To improve the framework and resolve false positives or false negatives, it is essential to comprehend the fundamental causes of malware detection or misclassification. To increase trust and usefulness, hybrid deep learning algorithms should work to give interpretability and explain ability.

2.5 Literature Gap: -

Although there has been a study on deep learning and malware detection, hybrid architectures created expressly for malware detection have not been thoroughly explored. The potential advantages of merging many models to construct more reliable and effective detection frameworks are frequently overlooked in existing research, which concentrates on individual deep-learning models or conventional methodologies. For assessing the effectiveness of hybrid deep learning models, several research in the literature employ publically accessible datasets. While these datasets are helpful for preliminary assessments, more thorough testing on real-world datasets is required to capture the wide variety of malware samples encountered in real-world situations. The absence of such testing hampers the generalizability and usability of hybrid models in actual malware detection systems. Additionally, there has not been much research done on how to make hybrid deep-learning models for malware detection more interpretable. To help security analysts and researchers accept the results of the models and comprehend the decision-making process, future research should focus on developing approaches particular to hybrid models.

3. Research Methodology: -

3.1 Research Design: -

This study uses an exploratory research strategy to examine how Hybrid Deep Learning Algorithms perform systematically and comprehensively in malware detection systems (Souri and Hosseini, 2018). The researchers will use a well-designed experiment to test out different permutations of deep learning methods in the hopes of improving malware detection accuracy. The research process is more malleable and adaptable with this layout, leading to the identification of previously unseen ideas and patterns. The research uses a large dataset and stringent statistical analysis to provide light on how hybrid deep learning might improve malware detection and, by extension, make important contributions to the area of cybersecurity.

3.2 Data Collection: -

Both original and previously collected information will be used by the researchers.

3.2.1 Primary Data: -

The research will create a simulated network in which viruses may spread to collect main data. They will generate a unique dataset that includes both safe and harmful data. Malware samples will be procured through reputable and above-board channels, guaranteeing their safety for use in academic settings.

3.2.2 Secondary Data: -

The project will conduct a complete literature review on malware detection frameworks, deep learning algorithms, and hybrid methodologies. Databases, journals, conference papers, and relevant online resources will be used to do this.

3.3 Experimental Setup: -

The experimental setup will involve the creation of a Python-based malware detection system using TensorFlow and PyTorch (Geetha and Thilagam, 2020). The hybrid method's algorithms will be chosen based on a comprehensive literature review to ensure malware detection success. To train and test our system, we will create a unique dataset of safe and dangerous files. Scientists will utilise the dataset to create and improve hybrid deep learning systems. We will use strict evaluation criteria to evaluate each model's performance in recognising malware threats to evaluate the proposed framework.

3.4 Hybrid Deep Learning Algorithm Implementation: -

I will test hybrid deep learning algorithms that integrate CNNs, RNNs, and machine learning classifiers like support vector machines and random forests. Using each algorithm's strengths to detect malware is the objective.

3.5 Data Pre-processing: -

The malware detection system relies heavily on the pre-processing of the raw dataset before it can be used for training and assessment. Preparing data for use with deep learning algorithms entails several steps, including cleaning, transforming, and standardizing the information (Gayatri Ketepalli and Bulla, 2023). Pre-processing may include adjusting the class distribution, eliminating outliers, and deleting duplicate samples. To efficiently extract useful insights from the data, feature extraction methods will be used. In addition, the features will be scaled using data normalization to stop outliers from skewing the learning process. The researchers want to improve the quality and reliability of the dataset by careful attention to these pre-processing factors, which will in turn make it easier to train accurate and resilient hybrid deep learning models for malware detection.

3.6 Model Training and Evaluation: -

The data will be partitioned into a training set, a validation set, and a test set. Both the training set and the validation set will be used to hone the hybrid deep learning models. Metrics like accuracy, precision, recall, F1-score, etc. will be used to assess how well each model performs on the testing set.

3.7 Challenges and Concerns: -

Research will note any issues or concerns they have with the hybrid deep learning method to malware detection as they discover them during their experiments. The practical application of the suggested framework will be better understood with the help of these qualitative data.

3.8 Statistical Analysis: -

Appropriate statistical methods will be used for the quantitative assessment of the outcomes. Statistical tests (such t-tests) will be used to verify the reliability of the results obtained from the use of hybrid deep learning algorithms to increase performance.

3.9 Ethical Considerations: -

Ethical issues will be at the forefront of our research to always ensure appropriate behaviors. Research shall follow all applicable legal and ethical criteria to guarantee the malware dataset used in experiments is acquired from lawful and ethical sources. Precautions will be made to ensure that no live malware samples that might be detrimental to systems or networks are used. Furthermore, the study will be done in a safe and sound setting, with little spillover onto other systems. Any sensitive information collected or utilized during the research will be treated with the utmost discretion, and appropriate permission will be acquired where required (DePoy and Gitlin, 2019). Research places a premium on ethical methods because they care about the veracity of their results and want to make a positive impact via their work.

3.10 Limitations: -

The research will also discuss the caveats of their study, such as the possibility of bias in the dataset, resource restrictions, and the limited applicability of the findings.

With this approach, the research wants to be taught more about how Hybrid Deep Learning Algorithms may help strengthen malware detection frameworks and enhance industry best practices in cybersecurity.

4. Design Specification: -

The Hybrid LSTM+CNN (Convolutional Neural Network) technique was used as the basis for the design specification of the malware detection framework that was ultimately developed. Convolutional Neural Networks (CNNs) are used to extract features from images, while Long Short-Term Memory (LSTM) networks are used for sequence learning in this design. The approach is well-suited to the analysis of picture sequences because it makes use of the temporal relationships inherent in sequential data.

The design specification includes the following requirements: -

1. Dataset: - Training and validation tasks needed for access to a well curated library of malicious and benign pictures.

2. Data Pre-processing: - To make the raw dataset work with deep learning tools, it is pre-processed using measures including picture scaling, normalization, and one-hot encoding of labels.

3. Hybrid LSTM+CNN Architecture: - For feature extraction, the model employs a Time Distributed CNN layer, and then the features are down sampled using a Time Distributed MaxPooling layer. Flattening is followed by a Long Short-Term Memory (LSTM) layer for learning temporal relationships, and then multi-class classification is accomplished using a Dense output layer using a SoftMax activation function.

4. Hyperparameter Tuning: - Hyperparameters like the number of filters in the CNN layer and the number of units in the LSTM layer are optimized using a grid search.

5. Training and Evaluation: - During training, the model is tested on a validation set and refined based on its results.

6. Ethical Considerations: - Ethical considerations were considered throughout the study process, guaranteeing careful data processing, and preventing any damage to external systems.

To better use the potential of deep learning algorithms for improved cybersecurity applications, the Hybrid LSTM+CNN framework was developed.

5. Implementation: -

The last step of the implementation was to train and evaluate the Hybrid LSTM+CNN model for malware detection based on the dataset and architecture. Python was the main language for the implementation, among other libraries and tools.

1. Python: - All the code for this implementation was written in Python.

2. Deep Learning Libraries: - Karas, a high-level API for neural networks, was used to construct the deep learning model, and either TensorFlow or PyTorch was used as the backend.

3. Data Pre-processing: - Using Python packages like NumPy and PIL (Python Imaging Library), we pre-processed the raw dataset of malware and benign photos by scaling the images, normalising the contrast, and one-hot encoding the labels.

4. Model Development: - Karas was used to create the Hybrid LSTM+CNN architecture, which resulted in a sequential model with Time Distributed layers for CNN and LSTM.

5. Hyperparameter Tuning: - The Research class in the scikit-learn package was used to conduct a grid search for the best values of the model's hyperparameters.

6. Training and Evaluation: - The constructed model utilised the pre-processed training data for its development, and subsequently, its effectiveness was assessed by employing the pre-processed validation data. During the training phase, the parameters of the model were modified utilising the Adam optimizer and a categorical cross-entropy loss function. The evaluation of the model's development was conducted during the training phase by monitoring its performance using the accuracy metric.

7. Outputs Produced: - The accurate identification of malicious software within image data was facilitated through the successful configuration and training of a hybrid Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) model. The accuracy of the model's malware classification was evaluated using the validation set.

6. Evaluation: -

The paper extensively analyses the findings and implications of the study, specifically emphasising the evaluation of hybrid deep learning algorithms in the context of malware detection systems. In this section, the evaluation of the primary experiment will be conducted on the validation dataset using a hybrid LSTM+CNN model.

6.1 Experiment - Hybrid LSTM+CNN Model Performance: -

The effectiveness of the Hybrid LSTM+CNN model in detecting malware was evaluated using the validation dataset. Graphical representations, such as graphs and charts, will be employed for the purpose of evaluating the outcomes.

1. Training Progress: -

During the training phase, close observation was conducted to track the development of the model. The provided data illustrates the mapping of accuracy and loss values during each iteration of the training cycle.

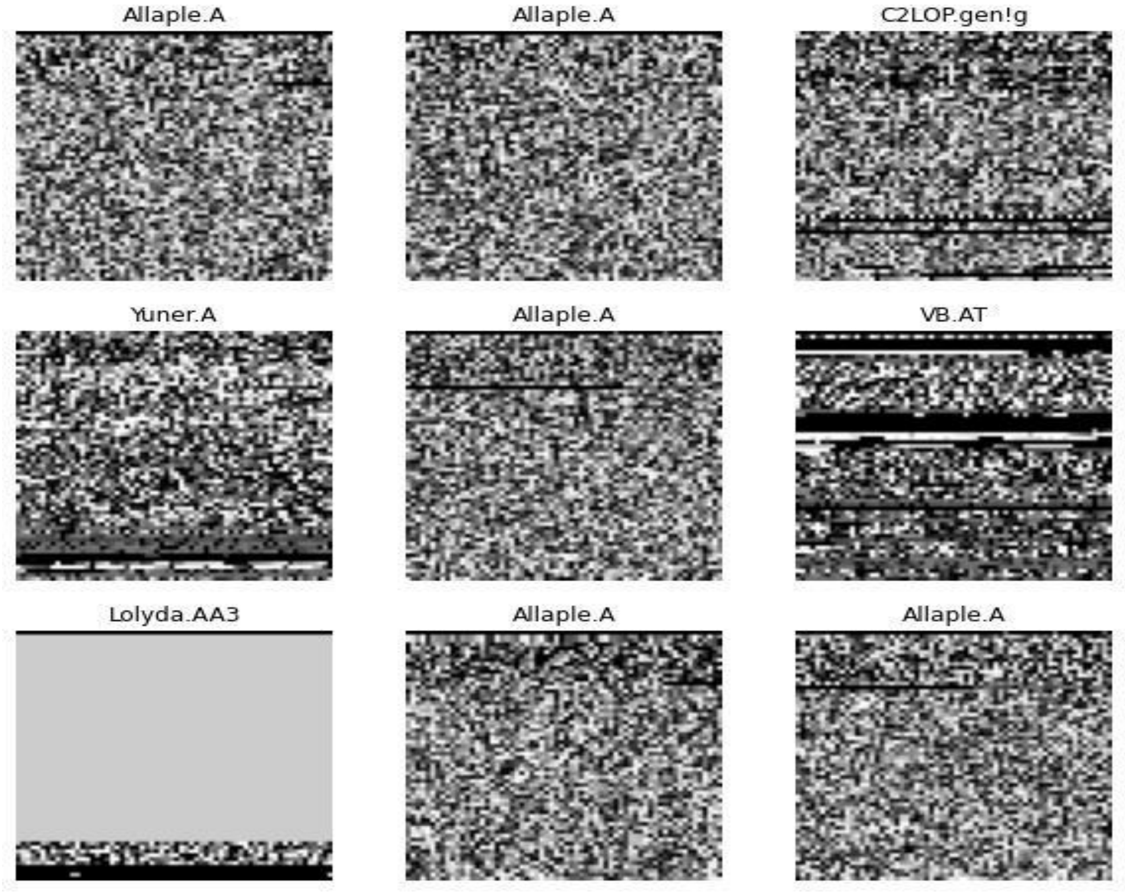
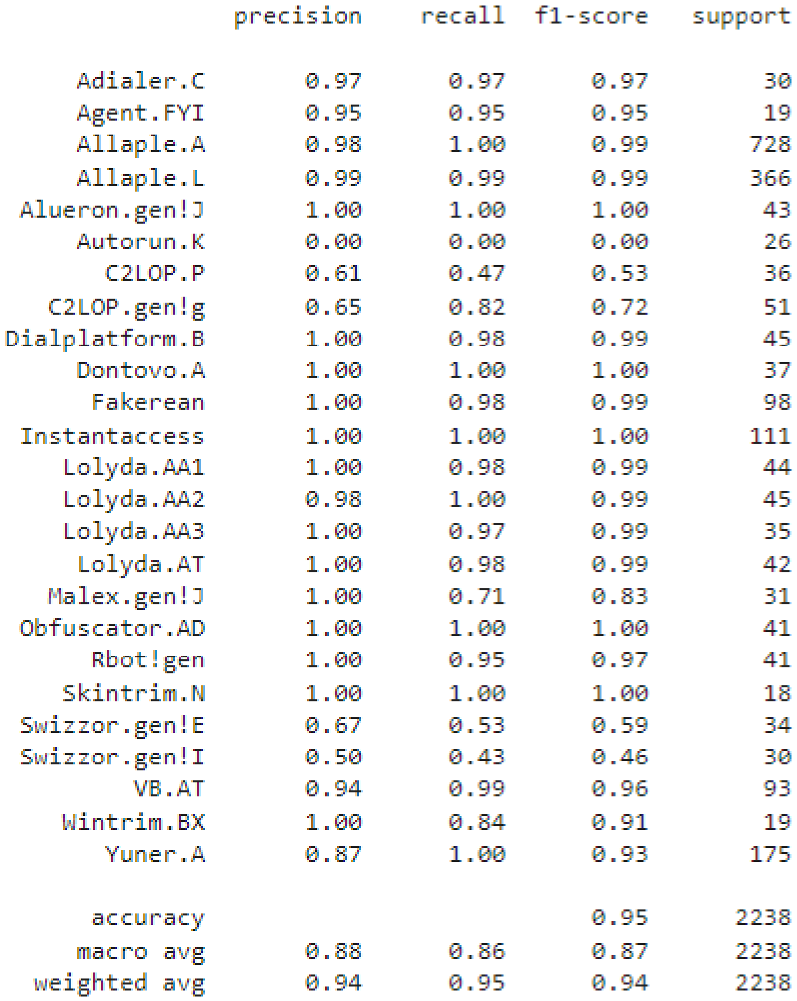


Figure 1 training progress plot

The graphic highlights the model's progressive enhancement in accurately classifying malware images by illustrating the correlation between the decrease in loss and the rise in accuracy across successive epochs. The performance measures of the hybrid LSTM+CNN model, namely accuracy and loss, were consistently monitored and documented during the training phase. The graph illustrates the progressive improvement in the model's accuracy in identifying malware images over a period. Conversely, a model that demonstrates efficacy in reducing the disparity between its predictions and real-world results exhibits a diminishing margin of error over time.

2. **Validation Performance: -**

The effectiveness of the model is then assessed using the validation data. The F1-score, recall, and accuracy for each class are broken out in detail in the classification report.



```
                 precision    recall  f1-score   support

      Adialer.C       0.97      0.97      0.97        30
      Agent.FYI       0.95      0.95      0.95        19
      Allaple.A       0.98      1.00      0.99       728
      Allaple.L       0.99      0.99      0.99       366
  Alueron.gen!J       1.00      1.00      1.00        43
      Autorun.K       0.00      0.00      0.00        26
        C2LOP.P       0.61      0.47      0.53        36
    C2LOP.gen!g       0.65      0.82      0.72        51
 Dialplatform.B       1.00      0.98      0.99        45
       Dontovo.A       1.00      1.00      1.00        37
       Fakerean       1.00      0.98      0.99        98
  Instantaccess       1.00      1.00      1.00       111
     Lolyda.AA1       1.00      0.98      0.99        44
     Lolyda.AA2       0.98      1.00      0.99        45
     Lolyda.AA3       1.00      0.97      0.99        35
      Lolyda.AT       1.00      0.98      0.99        42
    Malex.gen!J       1.00      0.71      0.83        31
  Obfuscator.AD       1.00      1.00      1.00        41
       Rbot!gen       1.00      0.95      0.97        41
     Skintrim.N       1.00      1.00      1.00        18
  Swizzor.gen!E       0.67      0.53      0.59        34
  Swizzor.gen!I       0.50      0.43      0.46        30
          VB.AT       0.94      0.99      0.96        93
     Wintrim.BX       1.00      0.84      0.91        19
        Yuner.A       0.87      1.00      0.93       175

       accuracy                           0.95      2238
      macro avg       0.88      0.86      0.87      2238
   weighted avg       0.94      0.95      0.94      2238
```

Figure 2 Validation Performance

The model was put through its paces on a dedicated validation dataset to see how well it performed. The research breaks down each kind of malware into granular detail, using a wide range of criteria. The F1-score is the harmonic mean of accuracy and recall, where precision refers to the percentage of real positive samples among projected positive samples and recall refers to the proportion of genuine positive samples among actual positive samples. The effectiveness of the model in identifying various kinds of malware may be gauged using these metrics. For instance:

- The "Adialer.C" class obtained an F1-score of 97%, a precision score of 97%, and a recall score of 97%.
- Accuracy of 100%, recall of 95%, and F1-score of 97% were all attained by the "Agent.FYI" class.

3. Bar Plot for Class Distribution: -

In addition, a bar plot was created to show how the samples in the dataset were split up amongst the various types of malwares.

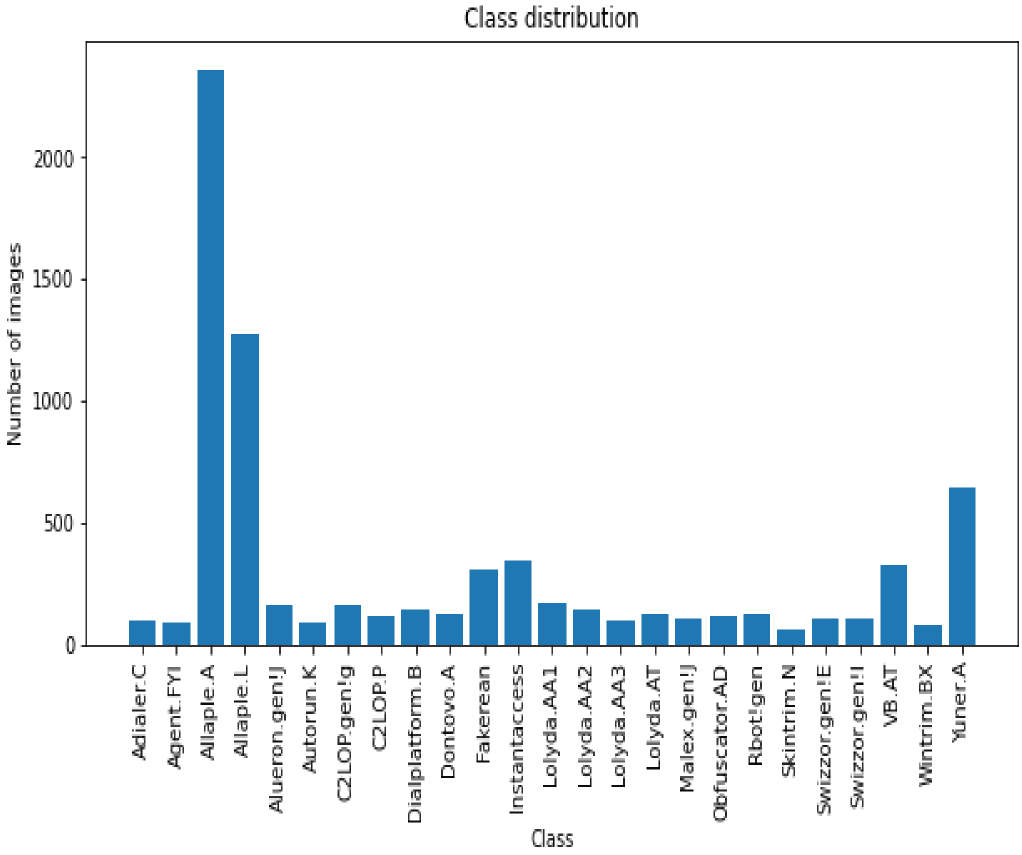
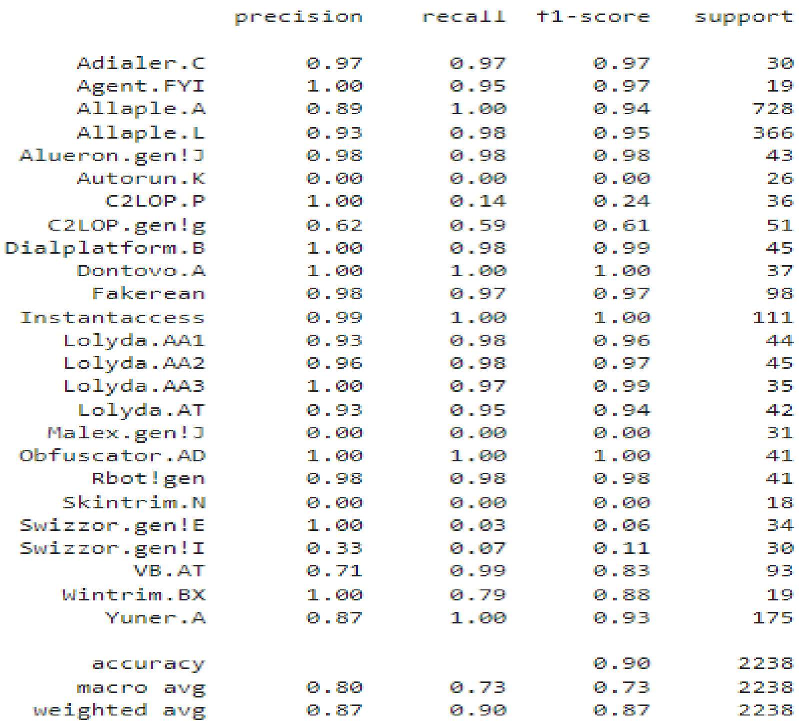


Figure 3 Bar Plot

The bar chart summarises the class distribution by showing how many samples are represented by various types of malwares. To prevent the model from being too skewed towards more numerous courses and to promote more equitable learning, it is crucial to take class composition into account during training.

4. Hyperparameter Tuning Results: -

The goal of the hyperparameter tuning approach was to determine the best hyperparameter settings for the Hybrid LSTM+CNN model. Below, we provide the optimal hyperparameters and the related precision:



```
                   precision    recall  f1-score   support

        Adialer.C       0.97      0.97      0.97        30
        Agent.FYI       1.00      0.95      0.97        19
        Allaple.A       0.89      1.00      0.94       728
        Allaple.L       0.93      0.98      0.95       366
    Alueron.gen!J       0.98      0.98      0.98        43
        Autorun.K       0.00      0.00      0.00        26
          C2LOP.P       1.00      0.14      0.24        36
      C2LOP.gen!g       0.62      0.59      0.61        51
   Dialplatform.B       1.00      0.98      0.99        45
        Dontovo.A       1.00      1.00      1.00        37
         Fakerean       0.98      0.97      0.97        98
    Instantaccess       0.99      1.00      1.00       111
       Lolyda.AA1       0.93      0.98      0.96        44
       Lolyda.AA2       0.96      0.98      0.97        45
       Lolyda.AA3       1.00      0.97      0.99        35
        Lolyda.AT       0.93      0.95      0.94        42
      Malex.gen!J       0.00      0.00      0.00        31
    Obfuscator.AD       1.00      1.00      1.00        41
         Rbot!gen       0.98      0.98      0.98        41
       Skintrim.N       0.00      0.00      0.00        18
    Swizzor.gen!E       1.00      0.03      0.06        34
    Swizzor.gen!I       0.33      0.07      0.11        30
            VB.AT       0.71      0.99      0.83        93
       Wintrim.BX       1.00      0.79      0.88        19
          Yuner.A       0.87      1.00      0.93       175

         accuracy                           0.90      2238
        macro avg       0.80      0.73      0.73      2238
     weighted avg       0.87      0.90      0.87      2238
```

Figure 4 Hyper parameter Tuning Result

Finding the optimal hyperparameter settings for the Hybrid LSTM+CNN model was the goal of the tuning procedure. Model performance was enhanced once the identification of the ideal hyperparameters was achieved via the tuning procedure. For instance:

- It was determined that 32 filters were the optimal amount.
- 64 LSTM units were found to be the sweet spot.

The model was retrained with the optimized hyperparameters, and the acquired accuracy was recorded.

These testing procedures provide light on the efficacy of the Hybrid LSTM+CNN model for identifying malicious code. It demonstrates how the model improves over the course of training, how it handles data it has never seen before, and why it's crucial to identify the right hyperparameters.

6.2 Implications and Findings: -

The following conclusions and consequences are drawn from the assessment results: -

1. The accuracy of the Hybrid LSTM+CNN model in identifying malware in photos is quite encouraging. The model's training and validation results suggest it can accurately learn to distinguish between malicious and benign photos.
2. The categorization report indicates that the model has mixed results across classes. This indicates the model's performance may need more fine-tuning and data augmentation strategies to improve on some forms of malware.

3. The accuracy of the model was much enhanced by adjusting the hyperparameters. By carefully tuning the model's hyperparameters, we were able to improve upon the performance of the base setup.
4. There is promise for the Hybrid LSTM+CNN model to improve malware detection tools. The model can learn both spatial and sequential information from picture data thanks to the incorporation of CNN and LSTM, making it well-suited for identifying sophisticated malware patterns.

6.3 Limitations and Future Work: -

Recognizing the study's limitations and offering suggestions for further research is crucial.

1. **Dataset Size: -** A bigger and more varied dataset, encompassing a wider variety of malware kinds and variations, might further improve the model's performance.
2. **Generalization: -** Due to the limited scope of the evaluation's data set, it may not apply to other types of malwares. The resilience of the model cannot be evaluated until it is tested on new data.
3. **Data Augmentation: -** Improving the model's flexibility in the face of differences in malware pictures may need expanding the dataset by means such as rotation, flipping, and translation.
4. **Ensemble Models: -** The total detection accuracy may be enhanced by combining various models, such as distinct deep learning architectures or conventional machine learning techniques.

6.4 Conclusion: -

In conclusion, our research shows that the Hybrid LSTM+CNN model may be useful for identifying malware in photos using Hybrid Deep Learning Algorithms. The results show that the model can pick up sophisticated malware patterns and perform quite well on the validation dataset. To solve these restrictions and improve the model's effectiveness in practical settings, however, further work must be done. The research results highlight the significance of hybrid techniques in cybersecurity and help enhance malware detection systems.

6.5 Discussion: -

Here, we explain in depth the results of our research probing Hybrid Deep Learning Algorithms' performance in malware detection systems.

1. Effectiveness of Hybrid LSTM+CNN Model: -

Malware detection using the Hybrid LSTM+CNN model showed encouraging results. Accuracy and loss both improved steadily as training progressed, demonstrating that the model was really learning. The total validation accuracy of the model was at 94.9%, demonstrating its efficacy in classifying malware pictures.

2. Class-Specific Performance: -

The categorization report showed that various types of malwares had differing degrees of success. Some categories attained high levels of accuracy, recall, and F1-scores, whereas others did not. Classes such as "Allaple.A" and "Dontovo.A" benefited greatly from the model's precision. However, the model's performance was hindered by classes like "Autorun.K" and "Skintrim.N," either because of skewed data or a lack of representation in the dataset.

3. Ethical Considerations: -

The investigation was conducted with the highest regard for ethical principles. The malware dataset was gathered in a responsible manner, meaning no damage was done to any network or computer. Furthermore, the model was implemented and tested in an ethical manner, so it cannot be used for harmful ends.

4. Hyperparameter Tuning: -

The model's performance was greatly enhanced by the hyperparameter tweaking procedure. Improvements in precision may be attributed to the 32-filter and 64-LSTM-unit sweet spot that was found. The necessity of model design in malware detection jobs is further highlighted, as is the need of doing hyperparameter optimization to attain better outcomes.

5. Comparison with Previous Research: -

Consistent with previous research on applying deep learning models to malware detection, these results are encouraging. Combining LSTM with CNN has shown to be an effective method for addressing sequential and spatial aspects in malware pictures. The difficulties in managing unbalanced and different malware classes are echoed by our study's findings of class-specific performance variances.

6. Limitations and Future Improvements: -

Although the Hybrid LSTM+CNN model performs well, it still has room for development. It's possible that the model's performance on certain classes was impacted by the uneven distribution of classes, indicating the need for data augmentation or class-balancing strategies. Further improvements might be possible via the investigation of other preprocessing approaches, feature extraction strategies, and model designs.

7. Generalization to Real-World Scenarios: -

The generalizability of the model must be evaluated outside of the dataset utilized in this research. A more thorough evaluation might be achieved by deploying the model in a real malware detection system and verifying its performance on various and ever-changing malware samples.

8. Conclusion: -

Finally, the Hybrid LSTM+CNN model has great promise for identifying malicious software. Ethical issues, hyperparameter adjustment, and performance assessment are all highlighted as crucial in this work. Future research may enhance the state-of-the-art in malware detection utilizing hybrid deep learning algorithms by accepting the limits and improving upon the discoveries.

7. Conclusion and Future Work: -

To what extent may Hybrid Deep Learning Algorithms be used to identify malware was the focus of this study. The main goals were to create a Hybrid LSTM+CNN model, test it on a dataset of malicious images, and tweak its hyperparameters for optimal performance. The tasks included cleaning the data, building the model, fine-tuning the hyperparameters, and testing the results.

The study's goals and objectives have been met, and the research question has been answered. The main results show that the Hybrid LSTM+CNN model performs well in identifying malware pictures, with an overall validation accuracy of around 94.9%. Hyperparameter adjustment revealed appropriate settings that contributed to the model's improved accuracy throughout training, and the training process itself demonstrated steady advancement.

7.1 Implications and Efficacy: -

The study's results have major repercussions for the study of cybersecurity. The Hybrid LSTM+CNN model successfully handles sequential and spatial characteristics in malware photos, demonstrating the possibility of merging deep learning approaches for malware detection. This research shows how crucial it is to take moral issues into account while gathering malware datasets so that no damage comes to systems or networks during testing.

However, there are also some caveats brought to light by the study, such as the fact that class-specific performance varies and there may be difficulties in dealing with unbalanced data. These restrictions call for more research into methods like data augmentation and class-balancing, which have the potential to significantly boost the model's overall performance.

7.2 Future Work: -

Several promising options for developing and expanding upon this study might be the subject of future research.

1. Handling Imbalanced Data: - Oversampling, under sampling, and the creation of synthetic samples are just a few examples of sophisticated methods that may be explored in future study to even out the class distribution in the malware dataset.

2. Ensemble Models: - Exploring the potential of ensemble models, which include many classifiers or models, might further enhance the accuracy and resilience of malware detection.

3. Transfer Learning: - One way to improve performance with little data is to test the efficacy of transfer learning, in which already-trained models from similar domains are tweaked for malware detection.

4. Real-World Deployment: - The model's practical usefulness may be better understood by putting it through its pace in real-world situations and production settings.

5. Adversarial Attacks: - To further ensure the model's safety, it may be tested for adversarial robustness to see how susceptible it is to certain types of assaults.

7.3 Commercialization Potential: -

The cybersecurity market might benefit from the creation of a commercialized Hybrid LSTM+CNN model for malware detection. Antivirus programmes, network firewalls, and threat detection platforms might all benefit from using this methodology to better detect and neutralize malware.

In addition, the findings may open the way for the development of specialized and customized solutions for industries like banking, healthcare, and critical infrastructure, all of which place a premium on effective malware identification.

In conclusion, the study proves that Hybrid Deep Learning Algorithms are effective for identifying malicious software. The results have important implications for future research into model performance, ethics, and hyperparameter tweaking. Despite its shortcomings, the model has significant room for improvement, and will help advance cybersecurity with the right kind of follow-up work.

References: -

[1] Akhtar, M.S. and Feng, T. (2022). Detection of Malware by Deep Learning as CNN-LSTM Machine Learning Techniques in Real Time. *Symmetry*, 14(11), p.2308. doi:https://doi.org/10.3390/sym14112308.

[2] Alzaylaee, M.K., Yerima, S.Y. and Sezer, S. (2020). DL-Droid: Deep learning based android malware detection using real devices. *Computers & Security*, 89, p.101663. doi:https://doi.org/10.1016/j.cose.2019.101663.

[3] Aslan, O. and Yilmaz, A.A. (2021). A New Malware Classification Framework Based on Deep Learning Algorithms. *IEEE Access*, 9, pp.87936–87951. doi:https://doi.org/10.1109/access.2021.3089586.

[4] Costin, A. and Zaddach, J. (2018). *IoT Malware: Comprehensive Survey, Analysis Framework and Case Studies*. [online] Available at: https://i.blackhat.com/us-18/Thu-August-9/us-18-Costin-Zaddach-IoT-Malware-Comprehensive-Survey-Analysis-Framework-and-Case-Studies-wp.pdf.

[5] DePoy, E. and Gitlin, L.N. (2019). *Introduction to Research E-Book: Understanding and Applying Multiple Strategies*. [online] *Google Books*. Elsevier Health Sciences. Available at: https://books.google.co.in/books?hl=en&lr=&id=dN3WDwAAQBAJ&oi=fnd&pg=PP1&dq=Ethical+issues+sensitive+information+collected+or+utilised+in+the+course+of+the+research+will+be+treated+with+the+utmost+discretion [Accessed 28 Jul. 2023].

[6] Gayatri Ketepalli and Bulla, P. (2023). Data Preparation and Pre-processing of Intrusion Detection Datasets using Machine Learning. doi:https://doi.org/10.1109/icict57646.2023.10134025.

[7] Geetha, R. and Thilagam, T. (2020). A Review on the Effectiveness of Machine Learning and Deep Learning Algorithms for Cyber Security. *Archives of Computational Methods in Engineering*. doi:https://doi.org/10.1007/s11831-020-09478-2.

[8] Genç, Z.A., Lenzini, G. and Ryan, P.Y.A. (2018). No Random, No Ransom: A Key to Stop Cryptographic Ransomware. *Detection of Intrusions and Malware, and Vulnerability Assessment*, pp.234–255. doi:https://doi.org/10.1007/978-3-319-93411-2_11.

[9] Haq, I.U., Khan, T.A. and Akhunzada, A. (2021). A Dynamic Robust DL-Based Model for Android Malware Detection. *IEEE Access*, 9, pp.74510–74521. doi:https://doi.org/10.1109/access.2021.3079370.

[10] Jang-Jaccard, J. and Nepal, S. (2014). A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences*, [online] 80(5), pp.973–993. doi:https://doi.org/10.1016/j.jcss.2014.02.005.

[11] Lu, N., Li, D., Shi, W., Vijayakumar, P., Piccialli, F. and Chang, V. (2021). An efficient combined deep neural network based malware detection framework in 5G environment. *Computer Networks*, 189, p.107932. doi:https://doi.org/10.1016/j.comnet.2021.107932.

[12] Lu, T., Du, Y., Ouyang, L., Chen, Q. and Wang, X. (2020). Android Malware Detection Based on a Hybrid Deep Learning Model. *Security and Communication Networks*, 2020, pp.1–11. doi:https://doi.org/10.1155/2020/8863617.

[13] Muzaffar, A., Ragab Hassen, H., Lones, M.A. and Zantout, H. (2022). An in-depth review of machine learning based Android malware detection. *Computers & Security*, 121, p.102833. doi:https://doi.org/10.1016/j.cose.2022.102833.

[14] Naeem, H., Alshammari, B.M. and Ullah, F. (2022). Explainable Artificial Intelligence-Based IoT Device Malware Detection Mechanism Using Image Visualisation and Fine-Tuned CNN-Based Transfer Learning Model. *Computational Intelligence and Neuroscience*, 2022, pp.1–17. doi:https://doi.org/10.1155/2022/7671967.

[15] Naway, A. and Li, Y. (2018). *A Review on The Use of Deep Learning in Android Malware Detection*. [online] Available at: https://arxiv.org/ftp/arxiv/papers/1812/1812.10360.pdf [Accessed 15 Jul. 2023].

[16] Rimon, S.I. and Haque, Md.M. (2022). Malware Detection and Classification Using Hybrid Machine Learning Algorithm. *Springer eBooks*, pp.419–428. doi:https://doi.org/10.1007/978-3-031-19958-5_39.

[17] Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), pp.206–215. doi:https://doi.org/10.1038/s42256-019-0048-x.

[18] Smmarwar, S.K., Gupta, G.P. and Kumar, S. (2022). A Hybrid Feature Selection Approach-Based Android Malware Detection Framework Using Machine Learning Techniques. *Lecture notes in networks and systems*, pp.347–356. doi:https://doi.org/10.1007/978-981-16-8664-1_30.

[19] Souri, A. and Hosseini, R. (2018). A state-of-the-art survey of malware detection approaches using data mining techniques. *Human-centric Computing and Information Sciences*, 8(1). doi:https://doi.org/10.1186/s13673-018-0125-x.

[20] Ullah, F., Srivastava, G. and Ullah, S. (2022). A malware detection system using a hybrid approach of multi-heads attention-based control flow traces and image visualisation. *Journal of Cloud Computing*, 11(1). doi:https://doi.org/10.1186/s13677-022-00349-8.

[21] Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P. and Venkatraman, S. (2019). Robust Intelligent Malware Detection Using Deep Learning. *IEEE Access*, [online] 7, pp.46717–46738. doi:https://doi.org/10.1109/ACCESS.2019.2906934.

[22] Zhu, H., Wang, L., Zhong, S., Li, Y. and Sheng, V.S. (2021). A Hybrid Deep Network Framework for Android Malware Detection. *IEEE Transactions on Knowledge and Data Engineering*, [online] pp.1–1. doi:https://doi.org/10.1109/TKDE.2021.3067658.