Malware Detection Across Mobile and Pervasive Computing

1st Queenly Xie
Electrical and Computer Engineering
University of Central Florida
Orlando, United States of America
queenly.xie@ucf.edu

2nd Lakshmi Katravulapalli
Electrical and Computer Engineering
University of Central Florida
Orlando, United States of America
lakshmi.katravulapalli@ucf.edu

3rd Russell Ridley
Electrical and Computer Engineering
University of Central Florida
Orlando, United States of America
russell.ridley@ucf.edu

***Abstract*—Advanced technologies are integrated into mobile, desktop, and online platforms humans use daily. Cybersecurity measures are in place to combat malware threats, and these measures are improving in reliability over time. This literature survey will examine the breadth of research and existing malware detection and prevention solutions. The focus will primarily focus on mobile and online environments. Through a comprehensive analysis of peer-reviewed articles and papers from research scholars, we will explore malware detection techniques and strategies. This literature survey will highlight deep learning algorithms to evaluate their effectiveness, strengths, and limitations. Ethical and privacy concerns will also be assessed as companies and individuals may use these types of malware detections for safety measures.**

I. Introduction

Malware is still a prevalent and serious cybersecurity threat in today's digital environment. Malware infiltrates millions of devices, capable of executing various malicious actions such as leaking sensitive data, encrypting files, impairing system performance, and more [16]. Thus, detecting malware is vital for safeguarding our computers and mobile devices against malicious attacks. Our literature review will focus on Malware detection in Big Data. The goal of this project is to tackle various solutions to malware detection. Deep Learning (DL) and artificial intelligence have surfaced as a promising technology for the detection of malware in recent developments. Mobile malware is considered malware detection technique for Android and iOS, another specialized term includes advanced persistent threats and ransomewhere in which they target threats of higher sophistication. Detection strategies will be analyzed so that there is a protocol for ongoing and adaptive malware attack. We will examine the effectiveness of deep learning models, data mining techniques, community detection algorithms, and deep neural models as potential strategies for combating malware attacks.

II. Traditional Methods of Malware Detection

Malware detection is a critical line of defense against threats that target computers and online platforms. Modern malware detection algorithms use Deep Learning, which relies on a large sample of data to make decisions and are trained. There are conventional techniques, such as Signature-Based Detection, Behavioral-Based Detection, Static Analysis, Sandboxing, and File Integrity Sharing. The algorithms listed do not use any source of deep learning, so the decision-making is based purely on what the malware does in real-time. A problem with non-deep learning malware detection algorithms is that the malware has already caused catastrophic damage before being detected and flagged. Malware Detection that utilizes Deep Learning started rolling out later to resolve some of these issues. For example, if a person's financial information was already tampered with by malware and the detection algorithm reports later, this solution did not fully prevent the unfortunate event. It's important to understand older malware detection algorithms and limitations for developers to incorporate deep learning into existing algorithms.

A. Signature-Based Detection

Signature-Based Detection relies on a database, which is known as malware signatures. The signatures are a unique string of data or patterns that associates and identifies a particular malware[1]. The malware detection software scans all the computer's files and compares it to the database for any matches to malicious malware. This detection algorithm is simple to understand and has high accuracy. However, the detection only works for pre-existing malwares that are in the database. This implies that newer or modified versions of malware will not be detected and could cause damage to the computer system. The database has to be consistently updated to be more effective, but there will always be a type of malware that is not in the database. This algorithm is a cycle that repeats with loopholes, which Deep Learning algorithms fixes. The cycle is shown in Figure 1.

B. Heuristic-Based Detection

Heuristic-based detection is more flexible than Signature-Based Detection. The main difference between Signature-Based Detection and Heuristic-Based Detection is that – Heuristic-Based Detection does not use any source of

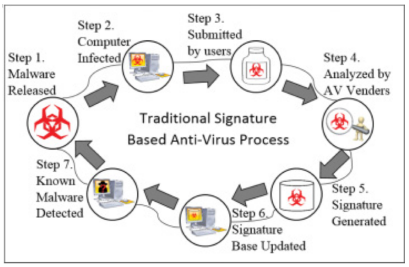

Fig. 1: Siganture-Based Detection [4]

databases for making decisions of identifying malware and detecting it [1,5]. Instead, Heuristic-Based Detection analyzes the characteristics and behavior of files on a computer system to identify suspicious activity, which may indicate malware may have made malicious modifications. Heuristic-Based detection will check system files, unusual start-up behavior, or any malicious executable files. Heuristic-Based does cover weak points from Signature-Based by flagging malware that may not have been seen before and rely on databases. However, a fault with the Heuristic approach is that there could be false positives by accidentally flagging regular software as malware. The Signature-Based approach makes a direct comparison, so there is a lower chance of false positives. Another downside of the Heuristic-Based approach is a delay in action when the malware enters the system. The algorithm takes a while to notice malicious behavior before flagging, which may be enough time for the malware to cause harm.

C. Behavioral-Based Detection

Behavioral-Based Detection observes the actions of a program while it's being executed. It does not rely on sample data or databases but does analyze in real-time of how the program interacts with the system. If the program starts to simulate malicious behavior, such as replication like viruses, using the internet to broadcast large amounts of data, or encrypting files, then it can be analyzed and processed. This detection algorithm is like Heuristic-Based Detection, but in different aspects. However, the same problem arises, such as there is a delay from recognizing the malware before it does damage.

D. Static Analysis

Static Analysis inspects the code of a program without execution. This detection method derives insights into the program's behavior from the structure, metadata, and dependencies, such as what it needs access to. This detection method looks through binary files for any malicious instruction sequences [5]. Static Analysis is significantly different from the other methods because it looks ahead of time at the possible behavior instead of analyzing the behavior after execution. An advantage of Static Analysis is that it prevents damage from malware once it's properly flagged as malware. However, cybercriminals are aware of these methods and engineer ways to evade being flagged from Static Analysis detection for malware. Another downside is that the malware is complex because it can take a lot of resources to analyze the code and take longer to decide whether or not to flag the program as malware.

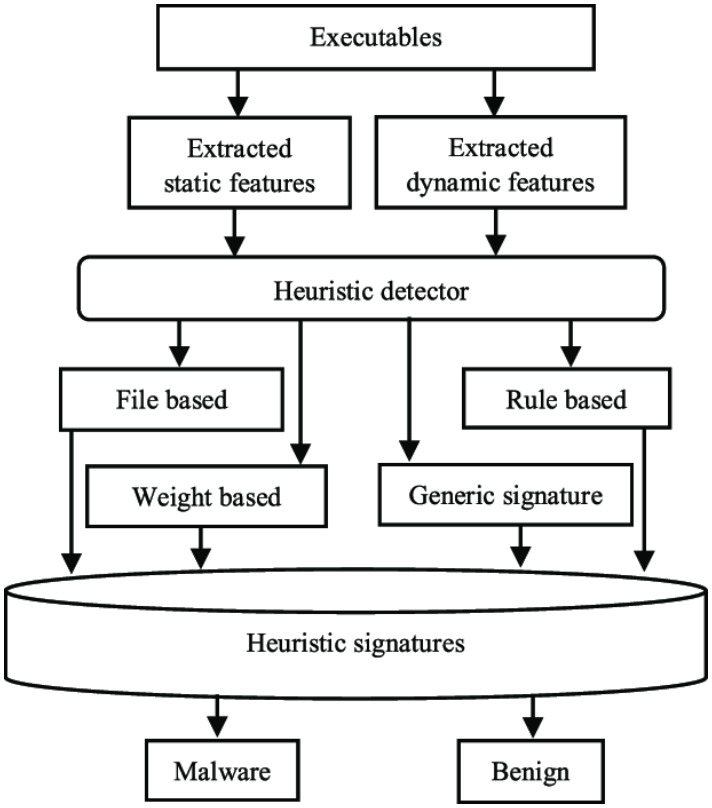

Fig. 2: Behavioral-Based Detection [6]

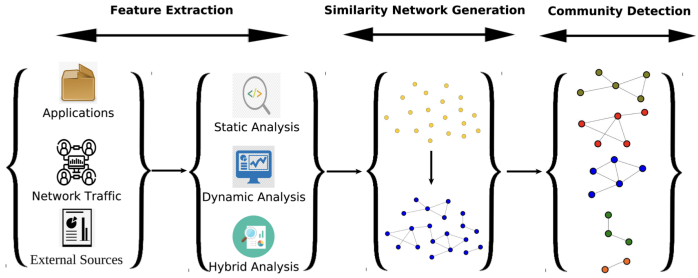

Fig. 3: Behavioral-Based Detection [1]

E. Sandboxing

Sandboxing is another popular technique for malware detection. Sandboxing allows malicious software to be executed is an isolated and monitored environment. The isolation prevents the malicious software from accessing other files or other permissions to cause harm [5]. If it's determined the program is performing harmful actions, then it can be flagged as malware. This is especially effective because if it evades the Static Analysis check, then it can be flagged by its actions and behavior. An advantage of sandboxing is that you can test if software behaves like malware without sacrificing your system. However, cybercriminals are making their algorithms sneakier by adding code that can detect if the malware is

running in an isolated or sandbox environment and will change its behavior to avoid being flagged as malware.

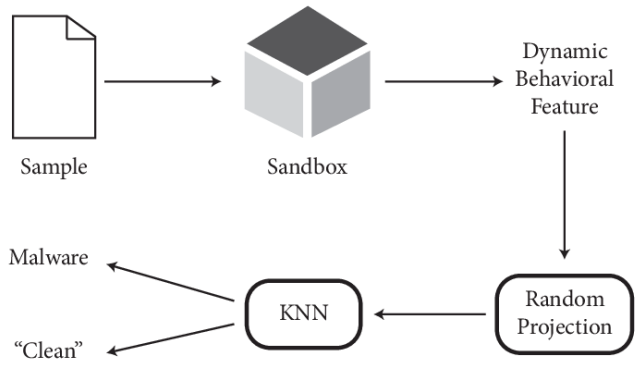

Fig. 4: Sandboxing Visualization [11]

F. File Checking Integrity

File Integrity Checking is a technique where it captures the state of the system periodically before the malware is ever executed. When the malware is executed, it will continuously compare the baseline to the current state to detect any malicious modifications caused by the malware [5]. This method just detects unauthorized changes. However, this may be a flaw for malwares that doesn't necessarily change any files and just sends information out through a network. Additionally, there are sometimes File Integrity can have false positives because it may not accurately tell the difference between malicious and non-malicious alterations. File Integrity does not prevent malware, but can only alert the user after an unauthorized change has occurred.

III. Community Detection Approaches for Advanced Malware Analysis

Malware families have groups of malware that have similar attack methods to accomplish their goals. These variants have similar codes, behaviors, and infrastructure. Community detection allows us to discover concealed patterns and structures within a specified system or network. We will be discussing different community detection approaches for malware detection. Instead of identifying malware individually, this method focuses on detecting clusters across a network. This allows the detection process to work more efficiently and allows us to identify threats faster.

The workflow of malware detection using this method includes 3 steps, which are feature extraction, similarity network generalization, and community detection [1]. Feature extraction involves examining inputs to identify their behaviors. Three different malware approaches that are used for feature extraction are static, dynamic, and hybrid. Static analysis is able to analyze the characteristics of a file and its features. Dynamic Analysis executes the file, and also monitors runtime activities. These features are critical because they give insight into the function and behavior of executable files which helps with the development of detection algorithms. Hybrid analysis combines both techniques to become more powerful, which is able to use multiple malware types at the same time. Similarity network generalization creates a graph/network based on similarities between data. It assigns feature vectors to a graph by grouping nodes with similar characteristics. Lastly, community detection uses similarity network graphs to combine common nodes into communities.

Feature selection in Community detection means using community detection algorithms to organize groups of similar features and then choosing a smaller group of the most used features from each community. This process helps avoid unnecessary or irrelevant features that could hurt the classification system. Using strategies like meta-heuristics, node centrality, and association with the class variable are used to select the most significant set of features from communities [1]. The community detection approach has to analyze the type of detection algorithm used and its function in malware analysis, which narrows the process down to detection or feature selection.

Some well-known community detection algorithms in network analysis are Louvain, Infomap, Label Propagation, and Girvan-Newman [1]. Louvain algorithm is a clustering method that helps find communities within a larger network. The goal of this algorithm is to improve a quality function known as modularity, which can improve the structure of communities and their connections. Infomap is based on the information theory concept which focuses on dividing communities based on well-defined qualities and grouping nodes that work well with each other. Label propagation has the ability to handle the organization of community relationships inside a network based on the common characteristics and behaviors. In addition, it can assign nodes to a network based on common features seen in neighboring communities. This algorithm helps improve the functionality of community detection for larger-size networks. The Girvan-Newman method functions by removing edges that show the biggest number of paths with nodes crossing them [1]. This aids in creating a community with distinct similarities and high-level connections. Overall, the algorithms discussed above utilizes the idea of a hierarchical structure which reflects how a community is designed using different levels of importance within a network. This type of organization allows for higher levels of abstraction and analyzes networks across multiple levels of resolution.

IV. Discussion on Deep Learning Algorithms

Deep Learning algorithms are being used to fight malware detection with today's growing increase in cyber attacks and the use of AI. DL has helped make remarkable progress by adapting to the nature of cyber threats and applying advanced techniques. Some of the advantages of Deep Learning Models is the ability to handle larger volumes of data and perform automatic feature extraction. They can also understand complex patterns from large datasets and have better detection accuracy.

Recurrent Neural Networks (RNN) and Convolution Neural Networks (CNN) are both commonly used DL algorithms.

RNN is mainly used to analyze sequential data, which uses memory to keep track of previously inputted data and output the next sequence. RNN requires significantly less time to detect malware in data. In addition, it can prevent attacks by detecting them during execution time. CNN is mostly used for classifying images and detecting patterns [8]. CNN can extract features from data and show signs of malware behavior. Another widely used technique is General Adversarial Networks. This uses generative algorithms to find important patterns from input data. Generative algorithms can understand the structure of data at a high level to create new samples that might be similar to the original. These samples can be helpful in catching variations in real-world malware that could be purposely made by attackers to avoid detection. This algorithm is crucial in today's environment because of the increase in adversarial attacks and challenging situations.

Recent studies have been exploring how Deep Transfer Learning (DTL) and Deep Reinforcement Learning (DRL) are being used to help train DL models and make them more effective. Deep transfer learning utilizes the knowledge from one DL model and transfers it to another one to help them learn specific tasks [8]. This gives DTL models an opportunity to learn from each other and improve their performance and training. DRL uses deep learning and reinforcement learning to help training models make decisions about malware attacks. This helps deal with challenges presented by recognizing and identifying malware to aid in choosing an optimal solution.

One of the challenges being presented is the growth of mobile devices, which has led to an increase in cyber attacks. Although many detection models have been proposed, many of them are not compatible with Android devices. A new research motive is to find new models that can support the features and complexity of mobile devices. Another recent advancement is the use of parallelization mechanisms. Parallelization for procedures in DL models has led to higher efficiency and lower execution time [8]. Future algorithms are aiming to use advanced parallelization methods to create high-level DL models and improve implementation as well as resources for complex tasks.

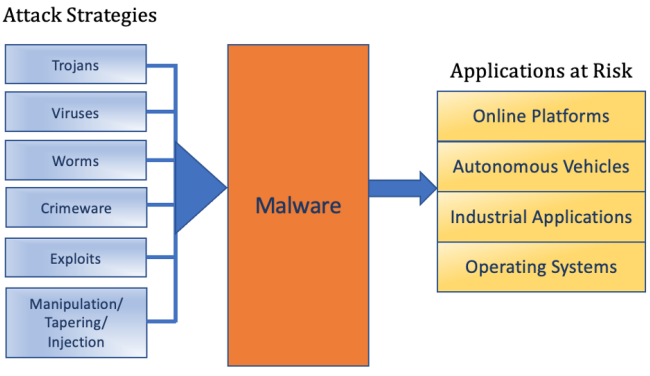

Fig. 5: Visualization of cross field malware attack types and variety.

V. Operating Systems

In the context of malware detection systems for operating systems, machine learning has been the commonplace solution. However, current malware detection systems for operating systems caters towards Linux and Microsoft, leaving MacOS geared detection, prevention, and remediation largely underdeveloped and under researched. MacOS has a 17percent-sign market share, in which the primary operating system is macOS [10]. The large market and userbase of Mac OS systems, would many users of MAC products vulnerable to hackers and attackers with malicious intent if the attacks were carried out successfully. Hence the importance of dealing with malware for macOS is supported. Since malware detection systems exists already for operating systems, similar approach can be used and a complete recreation of the system is unnecessary. A machine learning approach to malware detection systems typically follows feature extraction to allow the machine to acquire the important features that characterizes the malware attack. Typical characteristics of detection techniques include a signature based approach, in which a match is generated based on existing database of malware attacks. Sandboxing allows for operating systems to be removed from the attack, as the attack or unusual behavior is isolated to determine the origin. Limitations for malware detection which is similar across many fields include false positives and negatives, and detection influencing the performance. A highly complex multi modal interface like linux result in difficulty for setting a standard for detection. Windows have the history and statistics for high usage, making it a prime target and vulnerable for reasons similar to linux - diverse configuration. Potential solutions towards efforts in malware detection in operating system includes integrated security in which many techniques are used in tandem, layered upon the previous one. Training the users population on correct techniques is a way to improve the rates of malware detection and reporting, without referencing the software and hardware aspect and configuration.

A. Android Operating Systems

As the increase in reliance of mobile devices occur every year for the general human population, android operating systems are left vulnerable to attackers because of the fruitful outcome if successful attack occurs. Similarly to computer operating systems, signature based and behavior based detection are used to counteract malware attacks. In addition, there is also the method of static, dynamic, and hybrid analysis, in which the analysis of the code occurs in various time points before execution. This concept can be applied to the operating systems as discussed before. Furthermore, a combination of the techniques can be used to encourage effectiveness, comprehension, and adaptability [12]. A host of limitations similar to operating systems across the board includes intensive research use, false positives and false negatives. A way to intercept the false positives and false negative rates is targeting better heuristic methodologies while also simultaneously improving the adaptability of the methods to newer malware attacks.

Name	Objective	Features	Limitations
Feature Extraction	Feature extraction uses the process of changing data into a group of features that are more informative for a task or application	Uses dynamic, static, and hybrid analysis	-Loss of information -Domain Dependence -Human Bias
Multilayer Perceptron	Uses neural networks to classify files as either safe or harmful based on their features	-Input Layer -Hidden Layer -Forward Propagation Activation Functions	-Overfitting -Local Minima -Lack in Interpretability - Extra Resources
Recurrent Neural Networks	Uses memory to store information and determine the next output in a sequence, which helps catch patterns of malicious behavior.	-Recurrent Connections -Shared Parameters -Temporal dynamics	-Sensitivity to Hyperparameters -Limited Parallelism -Requires large amounts of Data
Artificial Neural Network	This algorithm is capable of understanding patterns and similarities in data, which allows it to be useful for a number of different tasks. ANN is used to examine features that are extracted from files and data, determine if they are benign or malicious based on pattern recognition.	-Feature Learning -Scalability -Real-Time detection -Adversarial Robustness -Deep Learning Architectures	-Data dependency -Privacy Concerns - Extra Resources -Overfitting
Generative Adversarial Networks	Uses generative algorithms to find new patterns from inputted data. GAN's have the ability to generate artificial malware samples that mimic real-world threats.	-Enhancing training data -Feature Extraction -Adversarial Training -Anomaly Detection	-Training instability -Overfitting -Challenges when evaluating bigger sets of data
Feature Selection	Feature selection is important in classification tasks and data with more features/complexity. Feature selection uses 3 different methods which are filter, wrapper, and embedded. These techniques are important to increasing the readability and functioning of models, and improving the accuracy of classification.	-Generalization -Adaptability -Easily integrated into other detection models -Dimensionality Reduction	-Depends on Feature Representation -Sensitive to characteristics of training data -Human Bias
Heuristic-Based	Heuristic-based was a highly used algorithm used in the past. This is a traditional signature based model. It also focuses on identifying suspicious behavior and characteristics that present malware.	-Behavior Monitoring -Pattern Recognition -Reduces False Positives	-Limited Visibility -Less effective with new and evolved threats
Signature-Based	Detects if a file is malicious by comparing its signature to a database of malware signatures. This process involves extracting specific strings from file codes and checking to see if they match signatures found in the database.	-Effective protection -Easily integration with other detection models -String Matching -Signature Database -Automatically compares signatures to determine unsafe files	-Updating signature databases to adapt to new malware variants -time-consuming -Labor-intensive -can produce false negatives
Cloud-Based	Classifiers analyze the unknown file samples and check if they are safe and harmful from the cloud server. During this process, clients and servers discuss operations and outcomes. Using feedback from the server, clients are able to assess new security solutions.	-Scalability -Centralized analysis -Cost effective -Combined with data mining techniques -Uses Centralized cloud servers	-Relies on the internet -Less Protection Offline -Requires network latency -Privacy concerns
Community detection	The objective of malware detection is to group together samples that exhibit similar behaviors and characteristics. This can help detect common attacks against malware and collaborate to find a solution.	-Malware classification -Group together samples with similar behavior -Clustering -Graph and Nodal based techniques -Feature extraction -Graph representation	-Overfitting -Scalability -Privacy concerns -Lack information from outside resources
DBN-based Android malware detection	DBN is an artificial neural network used to detect Android Malware detection.	-Automated feature learning –Real-Time detection -Deep Learning techniques	-Data dependency - Prone to adversarial attacks -Privacy concerns

Fig. 6: Comparison of Deep Learning Algorithms [1, 7, 8, 9]

VI. Internet of Things (IoT)

IoT malware exists through its various forms and applications within the home and commercial company settings. In malware detection, identification is the first step in which for IoT settings graphing and non-graphing techniques are used with varying efficacy. CNN and LSTM combined is a common malware detection solution that combines the advantages so that spatial and sequence prediction is achieved [13]. There are computational limitations becuase IoT systems are not designed with complex architectures and hardwares and oftentimes a requirement includes being contained and ending up in a small space. The intentions of the models are clear when deep learning can detect anomoly, however they encounter the issue of black box and low explainability is a drawback. Feature selection reduces computational complexity and is a malware detection technique which derives the important peaks from the dataset.

VII. Autonomous Vehicles and Malware Detection

Autonomous vehicles are the new age type of transportation in which human intervention is not required for the transportation vehicle. As electric vehicles and regular gasoline vehicles adopt autonomous modes, and their functionality is becoming the status quo, malware detection is required for analysis. The autonomous vehicles are connected to data transfer directions and are vulnerable because of their need to transfer information and receive real time updates from the web. Such ways include remote access attacks, ransomware, data theft, and more. Unusual activity which deviates from the normal functionality is detected conventionally using machine learning techniques. Personal information is stored to support customizable driving functionality like the telematics control units, user identification system, etc. Hence encryption is used to authenticate the user. Limitations of autonomous vehicle malware detection is that the interface between the parts are more complex than mobile, which can introduce storage and increased latency for the detection. Cyber attacks also evolve at a faster rate than car updates [11], a change in the system update center, automated updates, or real time connection to current vehicles from the security standpoint would be viable solutions. For future malware detection solutions, blockchain's capability in decentralizing transactions can be leveraged in the interconnections between the automobile functionalities.

VIII. Ethical Concerns of Deep Learning

Deep Learning is a branch of Artificial Intelligence where companies use machines to analyze large data values with minimal human intervention, which is used in Malware Detection, for example. As these intelligence services are a part of our daily lives, there is a rise of ethical concerns and challenges. Concerns that arise with ethical concerns of Artificial Intelligence and Deep Learning, such as bias, fairness, transparency, and privacy concerns.

A. Transparency

Transparency is an essential factor when it comes to humans using Artificial Intelligence and Deep Learning [14]. Transparency involves trust, fairness, and accountability. The challenges involved with trust include interpretability and performance trade-offs. Simpler algorithms make it easier for humans to read and understand how the algorithm is making decisions. However, simpler algorithms can be less powerful and less reliable, such as decision trees. Deep Learning is often complex because it requires a lot of data as input but has higher accuracy. This may pose a concern because humans may not understand the rationale behind a decision a Deep Learning made. Currently, there are no laws or regulations for transparency of Artificial Intelligence and Deep Learning. No regulations can lead to variations of transparency of Artificial Intelligence and Deep Learning systems, allowing systems to vary their standards, which could cause more harm or concern than other systems. Possible solutions for transparency include implementing explainable Artificial Intelligence, which explains the decision process. This can also be tweaked to explain the decision process while making the decisions. Regulations or standards that require transparency can also be implemented. For instance, the European Union's General Data Protection Regulation includes the right for an explanation of an algorithmic decision [2]. These solutions can help us understand why Malware Detection Algorithms may flag a software as malicious and can learn the types of malware that computers or mobile devices may be targeted with.

B. Bias and Fairness

As the number of cyber threats increases, the usage of deep learning in malware detection has become more popular. Malware detection algorithms have great advantages in identifying malicious software and minimizing the risks. The bias in malware detection comes from the data that is used to train these types of systems. The systems are only good from the data they learn from. This may pose a problem to malware detection algorithms because the data may not be fully representative of all real-world scenarios. There will always be a risk of biased outcomes based off the data. For example, the data can reflect from specific geographical regions, which means the model can fail to detect newer malware types from geographical regions. This can lead to false negatives, which can pose a risk to companies or individuals using malware detection algorithms. The bias can come from skews in the data, where the data does not reflect newer malware types; sampling bias, where some malware types are overrepresented or underrepresented; and labeling bias, where there may be a possibility that some software is incorrectly flagged and decreases accuracy. To mitigate these issues, Deep Learning algorithms can include diverse and updated training data so the model can be exposed to a broad spectrum of malware threats. Rigorous testing to verify the validity of the model can be a solution as well because the model's performance can be evaluated across a variety of scenarios to ensure high accuracy and unbiased. Developers and cybersecurity

	Function	Advantage	Limitation
Convoluted neural network – Long Short Term Memory [CNN-LSTM] Hybrid Model	Leverages the spatial pattern detection and ability to reference material from over long durations of time learn the typical patterns of IoT systems.	Combined spatial feature and temporal dependency analysis and capture results in increased learning of 95.5%. [Malware]	Using both components increase complexity and resource requirement.
Static Analysis	Inspection of contents before execution.	Applications in antivirus software and security classification.	Some key characteristics are demonstrated after execution, which would be overlooked.
Dynamic Analysis	Analyzes during execution time, looks for characteristics of malware attacks.	Defense against malware targets launching at execution time.	Resource intensive behavior analysis.
Graph-based	Inspection and analysis of control flow and operating dynamics and relationships.	Targets malware that uses high complexity.	Increased complexity results in increased processing and increased latency.
Non graph-based	Implements a more basic heuristic/signature based method of malware detection.	Explainable and has speed advantage.	Loses efficacy for use against new malware.
Machine Learning	Deploys the use of algorithms for pattern detection and learning for the model.	Not all threats needs to be known beforehand, and adaptability is able to be done.	Large and representative datasets required initially for training.

Fig. 7: Machine learning malware detection outline

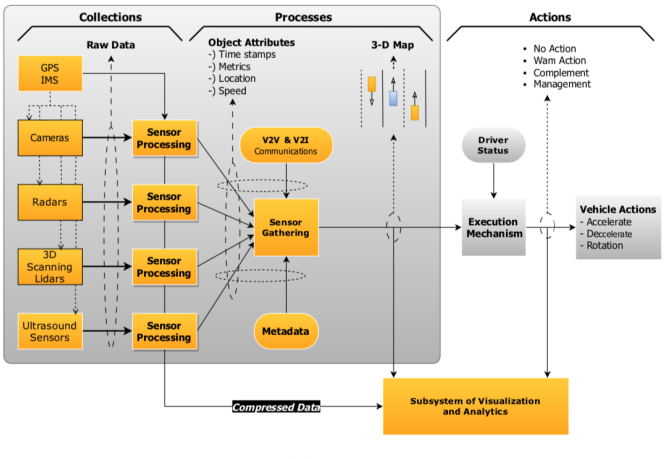

Fig. 8: Autonomous vehicles overview [11]

Fig. 9: Internet of Things [14]

professionals of these detection algorithms can improve the reliability and fairness of these systems by acknowledging these solutions provided to maintain the security of mobile and online platforms.

C. Privacy Concerns

The usage of deep learning algorithms for malware detection has increased due to the numerous varieties of malware that exist in the world. Deep Learning algorithms often require large volumes of data to accurately recognize scenarios of detecting malwares. However, this may rise of privacy concerns because this uses data from human's computers or mobile devices when reporting the malware. The data that can be included in the Deep Learning algorithm are behavior logs, network traffic, and files that may contain sensitive information. Data breaches are possible to happen in the cybersecurity world, which could pose a risk to users that have their data collected by the Deep Learning algorithms. Attackers could possibly sell the data for money or use the data for malicious reasoning, such as fraud and identity theft. The misuse of the data can also go against individuals and companies because attackers can use the data to analyze backdoors or loopholes to trick the algorithm to cause harm. Balancing between security and privacy can be a huge dilemma because algorithms can use less user data for their algorithms, but this results in lower accuracy of detecting malware. If developers want to

	Function	Advantage	Drawback
Deep Transfer Learning	Pre-trained deep learning models employed on pattern recognition of non-malware situations, to be later employed onto malware situations.	• Applications on cloud server/platform • Lightweight models on resource-constraint mobile devices • Pre-train models • Low computational/memory overhead	• Domains for pre-trained model and executed domain must be similar • Recognition of new malware attacks • Black box interpretability
Mobile Edge Computing	Improves mobile networks' efficiency and specifically for mobile devices, improve security and defense.	• Intrusion Detection Systems • Intrusion Prevention Systems • Real-time detection	• Reliance on network • Integration challenges
Adversarial Machine Learning for Intrusive Detection Systems (IDS)	Machine learning based security mechanisms that addresses malware attacks that evade detection.	• Online intrusion detection systems • Proactive defense • Adaptability	• Computational/resource overhead • Needs to account for false malware inputs
Convolutional Neural Networks (CNNs)	Utilizes layered neural networks to analyze binary data and extract spatial hierarchies of features.	• High accuracy in image-based malware detection • Effective in feature extraction from static content	• High requirement for computational resources • Decreased efficiency with encrypted malware
Recurrent Neural Networks (RNNs)	Processes sequences of data. Increasing suitability for analyzing temporal or sequence-based data.	• Efficient pattern detection for sequential data • Adaptable to time based malware changes	• Overfitting • Not ideal for long sequence dependencies
Long Short-Term Memory Networks (LSTMs)	Type of RNN, mitigates long-term dependency problem. Effective for analyzing long sequences of processes.	• Excellent at learning from sequences with long intervals; robust against the vanishing gradient problem.	• High complexity • Increased training times • High memory requirement during training
Autoencoders	Unsupervised neural networks by replicating the input data at the output.	• Useful for anomaly detection by learning normal behavior and detecting deviations; efficient in unsupervised scenarios.	• Potential for failure to capture complex patterns • Malware mimicking normal behavior
Deep Belief Networks (DBNs)	Stacked neural networks consisting of multiple layers of latent variables. Trained firstly unsupervised. Finally, supervised fine-tuning for the trained dataset.	• Feature extraction • Dimensionality reduction • Robust pretraining helps in effective learning	• Less common • Effective and less complicated alternatives

Fig. 10: Key deep learning Malware techniques

improve the accuracy of malware detection, they need a large volume of data at the expense of privacy concerns. This ties back into transparency, as previously discussed, because the only solution is to have clear communication of how the data collected will be used and the time the data will be retained for. Developers can also have consent forms for users to agree of the data collection. Mitigation strategies include data anonymization, where the data collected should not be able to identify the user. Data minimization can also be an option because it may be discovered that not all the data collected is used or needed for the decision-making of malware detection algorithms [3]. Encryption of the data can also be used to prevent unauthorized access in case of an algorithm data breach [3]. Balancing security and privacy concerns includes a lot of measures that need to be put into place for ethical considerations. Since it is impossible to favor both, the best approach is to have balance when developers are developing detection algorithms.

IX. Data Mining

Data Mining is a popular technique within the cybersecurity industry when it comes to fighting malware. Data Mining utilizes advanced algorithms to analyze and detect malicious software, such as software. These algorithms utilize machine learning to adapt to the changing behavior of malware. Machine Learning algorithms often rely on statistical techniques, such as clustering algorithms to find patterns and similarities in different types of malware, which can help in predicting and identifying new threats [17]. Additionally, detection techniques are used to spot unusual activities that deviate from standard network behavior, often signaling a potential disruption to the computer system. For example, association rule mining helps in discovering relationships between various characteristics of malware, aiding in the refinement of detection rules. By integrating these diverse data mining methods, cybersecurity experts can enhance their defenses, making it tougher for malware to penetrate and damage systems.

A. Association Mining

Association rule mining is particularly effective in malware detection by uncovering hidden patterns and relationships within large datasets of network activity and malware samples [17]. In cybersecurity, it involves identifying frequent combinations of behaviors that are commonly associated with

malware, such as specific code sequences, network signatures, or system changes. These associations are then used to develop rules that help predict whether a new software behavior or network traffic indicates a malicious threat. For example, if a particular sequence of system calls and network requests frequently occurs in a dataset of known malware, a rule can be established that triggers an alert when this sequence is detected, and the malware is flagged. This approach enhances the detection capabilities of cybersecurity systems by enabling them to recognize and respond to potential threats based on learned patterns, rather than relying solely on known malware signatures as discussed with traditional malware detection algorithms.

Challenges and Future directions

Data mining techniques like classification and clustering may present challenges when detecting results and challenging difficulties in real-life applications. They are able to detect malware from file samples or data, but being able to confirm the threat provides new challenges. This often requires need for domain specialists and manual inspection, which may be time-consuming. With the increase in new types of malware and evolving techniques, it is difficult for older classification systems to keep up. These detection systems need to learn to evolve with the changing trends/threats in order to properly maintain its functionality. In addition, adversarial learning is a significant concern with these data mining approaches. This means that attackers might be able to trick classifiers by changing data distribution or feature importance. Researchers hope to create stronger and more effective techniques that are able to fight adversarial situations.

Conclusion

Malware detection algorithms have come a long way in improving the functionality and accuracy of malware detection. Before Deep Learning versions of malware detection algorithms existed, traditional algorithms relied on databases and the CPU to make the decisions instead of utilizing a deep learning algorithm. We have discussed the limitations of traditional malware detection and the significant advancements in deep learning detection algorithms. Traditional Malware Detection Algorithms often tried matching the signature of the malware for malware detection. However, traditional methods are struggling to keep up with the rapid development of malicious malware because traditional methods often require frequent updating and were found to be unpractical for individuals and companies to use.

In contrast, Deep Learning algorithms use large data samples to predict patterns for new types of malware that may have properties similar to those of existing malware types. Deep Learning is introduced as a proactive and dynamic solution to malware detection. The future of technology is using Deep Learning to enhance the security of cyber threats against computer systems. Deep Learning algorithms are still developing and integrated for cybersecurity strategies to transition from rule-based algorithms to adaptive algorithms.

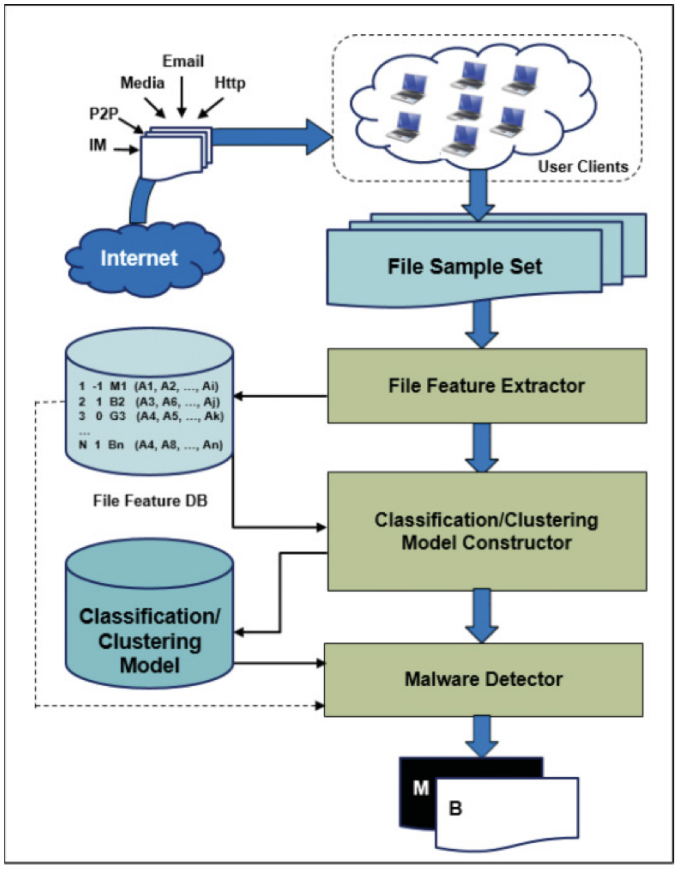

Fig. 11: Data Mining Visualization [4]

Deep Learning has provided a promising development for malware detection with its powerful techniques for larger datasets. We have discussed various DL algorithms that have shown the ability to adapt to changing variants and challenging situations. In addition, they have shown different approaches and features to help detect malicious files and recognize signs of malware. In the future, researchers hope to advance the use of deep learning with platforms for MacOS and Windows.

Privacy and ethical concerns arise in using Deep Learning Algorithms. Ethical concerns must be addressed to protect individuals and maintain trust when using these algorithms. As discussed, Deep Learning algorithms consume large amounts of data, which can contain personal information. Transparency about the algorithms is essential to maintain trust with the public. Regulations or laws for the usage of Deep Learning Algorithms don't exist but can be a beneficial solution. There are sacrifices of the performance of deep learning algorithms if regulations are implemented, but a balance would be an optimal solution.

The overview of the specific characteristics of malware detection from across various disciplines and fields allows us to discern the key characteristics, similarities, and differences. Through the analysis, themes can be disseminated across fields which are less established malware detection fields.

A suggestion for a future direction of malware detection re-

search includes a hardware based malware detection which can be applied to mobile computing. A different path for machine learning-based detection encompasses applications for online platforms and operating systems like MacOS and autonomous vehicles. Similar challenges and limitations are faced across the various fields, a deeper dive into new hardware accelerators in combination with software neural network solutions can reveal a more efficient, storage minimizer type of solution can pave the way for new malware detection techniques. Machine learning specifically has both supervised and unsupervised training, in which the concept can be applied to malware detection and using gpt type model baselines in the form of a neural network to tell the general form of application so that the solution can be curated and implemented on it's own. That type of solution would require a dataset of previous malware detection techniques, hence the compilation of the sources, limitations and advantages.

X. Individual Contributions

A. Queenly Xie

Contribution with the analysis of malware detection techniques for online platforms, operating systems, android operating systems, Internet of Things (IoT), autonomous vehicles, MacOS, Linux, and Windows. discussion and intellectual contributions with the malware techniques, and the background for those malware detection techniques. Updated the abstract and introduction with the specified malware detection techniques. Analyzed and looked into subtopics, drawbacks, limitations, and advantages. Looked into the cross usage of techniques from one field to another, strategize advantages and potential application from one field to another.

B. Russell Ridley

Contributed to research on the ethical concerns of Deep Learning, such as transparency, biases, and privacy. Contributed to the research of Malware Detection algorithms that do not utilize Deep Learning to show how Deep Learning improves detection algorithms, such as Signature-Based Detection, Heuristic-Based Detection, Behavior-Based Detection, Static Analysis, Sandboxing, and File Checking Integrity. Researched the limitations of non-deep learning algorithms, highlighting the need for Deep Learning. Researched data mining techniques that malware detection algorithms use to detect similarities of behavior of malware. Additionally, researched types of data mining techniques, such as Association Mining, which is a popular data mining technique in malware detection algorithms.

C. Lakshmi Katravulapalli

Contribution with the analysis on Community Detection Approaches for Advanced Malware Analysis. Contributed to the analysis on Discussion for Deep Learning Algorithms and Techniques. Researched malware detection models in different fields. Contributed to the table comparing different algorithms and models present which highlights malware detection across multiple published papers. Furthermore, researched and discussed the challenges and future trends of malware detection. Contributed to the introduction and Background of this paper.

References

[1] Amira, Abdelouahab, et al. "A Survey of Malware Analysis Using Community Detection Algorithms." ACM Computing Surveys, vol. 56, no. 2, Sept. 2023, pp. 1–29. https://doi.org/10.1145/3610223

[2] Z. Ayub and M. T. Banday, "Ethics in Artificial Intelligence: An Analysis of Ethical Issues and Possible Solutions," 2023 Third International Conference on Smart Technologies, Communication and Robotics (STCR), Sathyamangalam, India, 2023, pp. 1-6, doi: 10.1109/STCR59085.2023.10396966.

[3] A. Golda et al., "Privacy and Security Concerns in Generative AI: A Comprehensive Survey," in IEEE Access, vol. 12, pp. 48126-48144, 2024, doi: 10.1109/ACCESS.2024.3381611.

[4] Y. Ye, T. Li, D. Adjeroh, and S. S. Iyengar, "A Survey on Malware Detection Using Data Mining Techniques," ACM Computing Surveys, vol. 50, no. 3, pp. 1–40, Jun. 2017, doi: https://doi.org/10.1145/3073559.

[5] B. Alsulami, A. Srinivasan, H. Dong and S. Mancoridis, "Lightweight behavioral malware detection for windows platforms," 2017 12th International Conference on Malicious and Unwanted Software (MALWARE), Fajardo, PR, USA, 2017, pp. 75-81, doi: 10.1109/MALWARE.2017.8323959.

[6] Ö. Aslan, Heuristic-based Malware Detection schema. 2020. Accessed: Apr. 25, 2024. [Online Image]. Available: https://www.researchgate.net/figure/Heuristic-based-malware-detection-schema.

[7] "A Survey on Malware Detection Using Data Mining Techniques." ACM Computing Surveys, vol. 50, no. 3, June 2017, pp. 1–40. https://doi.org/10.1145/3073559

[8] Maniriho, Pascal, et al. "A Survey of Recent Advances in Deep Learning Models for Detecting Malware in Desktop and Mobile Platforms." ACM Computing Surveys, Dec. 2023, https://doi.org/10.1145/3638240

[9] Qiu, J., Zhang, J., Luo, W., Pan, L., Nepal, S., Xiang, Y. (2020). A Survey of Android Malware Detection with Deep Neural Models. ACM Computing Surveys, 53(6), 1–36. https://doi.org/10.1145/3417978

[10] C. S. Htwe, M. M. Su Thwin and Y. M. Thant, "Malware Attack Detection using Machine Learning Methods for IoT Smart Devices," 2023 IEEE Conference on Computer Applications (ICCA), Yangon, Myanmar, 2023, pp. 329-333, doi: 10.1109/ICCA51723.2023.10181535.

[11] Giannaros, A.; Karras, A.; Theodorakopoulos, L.; Karras, C.; Kranias, P.; Schizas, N.; Kalogeratos, G.; Tsolis, D. Autonomous Vehicles: Sophisticated Attacks, Safety Issues, Challenges, Open Topics, Blockchain, and Future Directions. J. Cybersecur. Priv. 2023, 3, 493-543. https://doi.org/10.3390/jcp3030025

[12] Z. D. Patel, "Malware Detection in Android Operating System," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2018, pp. 366-370, doi: 10.1109/ICACCCN.2018.8748512.

[13] Riaz, S.; Latif, S.; Usman, S.M.; Ullah, S.S.; Algarni, A.D.; Yasin, A.; Anwar, A.; Elmannai, H.; Hussain, S. Malware Detection in Internet of Things (IoT) Devices Using Deep Learning. Sensors 2022, 22, 9305. https://doi.org/10.3390/s22239305

[14] Larsson, S. Heintz, F. (2020). Transparency in artificial intelligence. Internet Policy Review, 9(2). https://doi.org/10.14763/2020.2.1469

[15] Gopinath M. and Sibi Chakkaravarthy Sethuraman. 2023. A comprehensive survey on deep learning based malware detection techniques. Comput. Sci. Rev. 47, C (Feb 2023). https://doi.org/10.1016/j.cosrev.2022.100529

[16] L. Miche, Figure 3. 2011. Accessed: Apr. 25, 2024. [Online Image]. Available: www.researchgate.net/figure/Global-schematic-of-the-methodology-a-sample-is-run-through-the-sandbox-to-obtain-a-set

[17] R. Josphineleela, S. Kaliappan, L. Natrayan and A. Garg, "Big Data Security through Privacy – Preserving Data Mining (PPDM): A Decentralization Approach," 2023 Second International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2023, pp. 718-721, doi: 10.1109/ICEARS56392.2023.10085646.