Privacy-Preserving Federated Learning for Phishing Detection

Amr I. Elkhawas
Department of Electrical & Electronic Engineering
City, University of London
London, UK
Amr.Elkhawas@city.ac.uk

Thomas M. Chen
Department of Engineering
City, University of London
London, UK
Tom.Chen.1@city.ac.uk

Ilir Gashi
Department of Computer Science
City, University of London
London, UK
Ilir.Gashi.1@city.ac.uk

***Abstract*—Machine learning is one of the most prominent technologies used to combat phishing detection, however, the vast amount of data required for training models for detection raises a privacy concern for end users. Gathering email or document data may very well contain private information and the machine learning models learn from the words and other attributes from these text based documents. Gathering this information in a centralized location and using them to train models could pose a security risk on all levels of data acquisition, from the transfer of the data to the storage. Federated learning is emerging as a promising alternative to traditionally centralized machine learning for phishing detection. The advantages of federated learning, mainly in privacy and scalability, are weighed against the issue of detection accuracy. Federated learning provides the ability to train models without the transfer of sensitive data, more or less no raw data from the device and allows the training to be done locally; this eliminates the privacy exposure accompanied with traditional machine learning models that operate in a centralized manner. However, this alone is not enough to comply to privacy regulations like GDPR, the EU AI act and privacy preserving technology must be used in conjunction to ensure federated learning's compliance to privacy regulations. This paper is a dedication to Professor Thomas Chen's aspirations in the field of Cyber Security. This paper is dedicated to his memory.**

I. INTRODUCTION

Phishing continues to pose a significant threat in the realm of cybersecurity. Statistics highlight

Published by the IEEE Computer Society

the persistent and evolving nature of phishing attacks. According to various reports, the volume of phishing attacks has shown an upward trend, with millions of phishing attempts detected globally each month. These attacks are not only increasing in number but also in sophistication, leveraging social engineering tactics and advanced techniques to bypass traditional security measures such as automated sandboxes and signature based detection. Phishing presents a major privacy leak as threat actors employ deception to trick individuals into willingly providing sensitive personal information such as login credentials, financial details, and other private data. Once obtained by malicious actors, this information can be exploited for identity theft, unauthorized access to accounts, or even sold on the dark web, compromising individuals' privacy and leading to potential financial and reputational harm. The primary delivery method for phishing and spam is emails and that is also the main entry point for the vast majority of malware infections. According to a recent survey around 85% of the world's email traffic are spam emails. Fraud emails amounting to 2.5% of all spam emails and identity theft, more commonly known as credential phishing, comprises about 73% of that number [4].

Phishing documents are highly effective as they are file types one is accustomed to receiving and interacting with on a daily basis. That is why threat actors increased the adoption of malicious documents and emails (such as Emotet, Trickbot, and Qakbot) as a delivery mechanism for several threats in the past few of years [15]. Phishing documents and emails with flashy headlines are used as social engineering lures to trick people into unknowingly downloading or executing malicious payloads either online or embedded within the documents. Phishing documents themselves are not exploiting a specific flaw in the various file types used to deliver them, they are benign documents with no distinct behavioural aspects other than that they illicit a user to click on a link within them. As usual, social engineering continues to successfully exploit human nature which is often the weakest link in the information security life cycle [7]. The embedded link may not present the phishing content or the malicious payload to users because of evasion techniques employed by several phishing threat actors e.g geographic based evasion, ip evasions, temporal evasions, etc [11]. This makes spotting the threats by automated systems which click on a link and mark links as a phishing attempt when they see an attempt to harvest credentials difficult. In that sense, security researchers have turned to machine learning including deep learning for phishing detection, but the technology comes with considerable costs related to data collection and computation. For accurate malware detection, vast amounts of malicious and clean samples must be collected from end user devices for training and testing.

Security providers are challenged to protect end users while simultaneously protecting their private data when it comes to phishing documents which raises an issue with data collection and user privacy, particularly due to the nature of phishing documents and the information contained within them, traditional data collection techniques used for executable files may not protect user privacy adequately. While other file types may also contain private information, these file types have a higher privacy risk than others because the file itself may contain confidential information written in the content body or a confidential image embedded within the document, personal details, financial information, contacts, correspondence, and so on.

Phishing detection platforms using machine learning rely on collecting vast amounts of end user data and should be revamped with stronger mechanisms for protecting user privacy. For instance, traditionally data is collected from user devices and uploaded to a centralized server where the data is featurized and used to train a machine learning model. This would not be the best approach because of the possible exposure of private data at the server. Data in transit through the network may also be at risk of exposure but can be protected by traditional cryptographic protocols such as TLS (Transport Layer Security) or IPSec.

Why would users voluntarily allow their data to be collected? This has always been standard practice for mutual benefit to both the security provider and end user. More data means that malware detection will be more accurate, obviously beneficial to users. However, security providers are cognizant of new privacy preserving regulations such as GDPR (General Data Protection Regulation), CCPA (California Consumer Privacy Act), and COPPA (Children's Online Privacy Protection Rule) that address the collection of private identifiable information (PII) [20]. The European Union (EU) recently introduced a new Artificial Intelligence Act (AI Act) which is a regulatory framework that significantly impacts the development of AI systems. The implications revolve around the following points: data governance and privacy; stakeholder responsibilities; robustness and quality management; energy efficiency; and bias and fairness [24].

This paper gives an overview of an alternative approach, federated learning which can alleviate privacy issues. In federated learning, local machine learning models are trained at each user endpoint, and only the model parameters are shared with the other endpoints to form the final global model. In principle, this arrangement offers a more private solution but some are sceptical because of presumed performance degradation compared to traditional machine learning and deep learning. Experiments have been conducted by a number of researchers in the security domain to measure the model performance degradation [3][10][13][22][23]. They found that in many cases federated learning does decrease the accuracy of the trained models, though the accuracy degradation could be mitigated to some extent through tuning of the nodes. Federated learning is designed with security and privacy in mind, with both the client and server operators sharing the responsibility of keeping the data secure; and it may help mitigate bias in training, since models are trained on diverse datasets. These aspects of federated learning can also help in addressing the requirements and implications of the EU AI act [24].

The remainder of this paper is organized as follows. Section II reviews the current approach to phishing detection using machine learning and deep learning and its effectiveness. Section III presents an overview of the concept of federated learning and its potential benefits. Section IV describes the application of federated learning to phishing detection and reviews previous work evaluating privacy. Finally, Section V presents conclusions and discusses open issues for research.

II. MACHINE LEARNING FORPHISHING DETECTION

Training machine learning algorithms to detect various malicious files has been a major focus area for security researchers in the past decade. Improving accuracy in the face of constantly evolving threats is the main goal. Collecting sufficient training data is one of the main challenges faced by the security industry. Figure 1 shows the current practice of collecting samples from user endpoints to a central server. The server pre-processes the data, extracts useful features, and trains a model that is then propagated to the antivirus nodes. The data collection is often mentioned in the terms of service of antivirus software for the purpose of improving the security service. In principle, there are no limits on the amount of files that can be uploaded.

E. Toch et al. in a recent survey highlighted the privacy implications of employing cyber security defence systems and proposed a methodology to classify and analyze these privacy implications [19]. They take a deep dive into numerous cyber security defense systems from different angles namely:

- Architecture of system.
- Type of Detection.
- Ecosystem.
- Type of Data.

They identify the privacy exposure tied to each of the previous angles and classify them based on the privacy leak. Taking into consideration malicious document file types or emails in an antivirus scenario, where it's a client-server

architecture that employ both signature and anomaly based detections that can span across all the aforementioned ecosystems (mobile devices, IoT devices and enterprise ecosystems) and using the methodology proposed, the following privacy implications are at risk. The client-server architecture has a privacy risk of data exposure due to the nature of the architecture due it is high network centrism. Antiviruses deliver both anomaly and signature based detections, they both pose a privacy risk of exposing PII and sensitive information due to the data accessed during the training of anomaly based detection models and during monitoring.

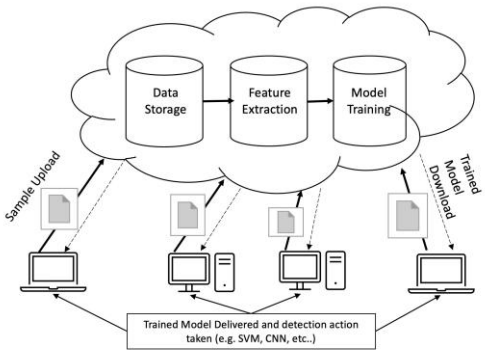

Figure 1. Current machine learning architecture for malware detection

Machine learning has been found to be fairly successful for malware detection. Selamat and Ali tested numerous machine learning algorithms that are popularly used in malware classification: Knearest neighbor, decision tree, and support vector machine (SVM) [17]. Their features were based on static PE (portable executable) information along with other statically extracted features such as byte sequences, n-grams, and DLLs (dynamically loaded libraries). PE files have always been the de facto file type used in malware classification problems. Their experimental results showed high accuracy of 99% with decision tree, 94% with K-nearest neighbor, and 91% with SVM. Although their dataset was relatively small in size and hence difficult to extrapolate the same effectiveness all malware, it is clear that machine learning can be a practical means to detect malware if adequately trained with known samples.

While PE files are the most notorious form of malicious files seen on computer systems, malicious documents are increasing in prominence. Singh et al. studied the dangers of malicious documents found in the wild namely PDF files and Microsoft Office documents [18]. The authors point out the different detection methodologies used to detect malicious PDF files and Microsoft documents along with several features based on the metadata and file structure of these specific file types.

For PDF files, they used JavaScript features extracted from the embedded code; structural features like types of PDF actions present and their order; and metadata features including page count, word count, author name (and much more). Microsoft office documents are split into two categories: OLE compound document format and open office XML format. The structural differences between them have to be taken into consideration before extracting features for machine learning algorithms. Besides structural features, embedded macros were also used as features along with metadata similar to the ones used for PDF files. The use of these features has been widely adopted for malware classification without considering the privacy leakage that might result.

In efforts to reduce the PII leakage in electronic documents, Auro et al. proposed a tool to identify and remove PII from a file including [1]:

- username, user's real name, security ID;
- computer's NetBIOS name, DNS name and suffix;
- names and addresses of domain controllers, email servers, file servers, and such.

Office documents may also risk similar PII leakage. The common locations where PII is hidden and could cause a privacy leak were found as:

- human readable metadata usually including author information, document title, and keywords;
- hyperlinks possibly exposing websites within the author's organization;

- metadata in embedded objects, e.g., metadata in photos containing the creation date of the photo and the device used to take that photo;
- names and paths of embedded or linked objects which often link to their source document (similar to hyperlinks except that they point to a local file);
- macros in Office documents used for automation, potentially exposing an internal server;
- scripts usually associated with PDF files for retrieving and updating the file with information from an internal server.

The identification of these metadata as PII can be an obstacle to efforts to combat malicious actors. The use of macro enabled malicious documents or phishing PDF documents are the most notorious first stages of malware delivery, and these features are a crucial part of the machine learning model used for classification. The ethical issue is to balance the usefulness of uploading these files to extract useful features for malware detection against the ethical need to protect the privacy of users.

According to a recent survey conducted by Salloum et al. a select number of features are commonly used in the models used to train the phishing detection models [16]:

- Email body-based characteristics that are extracted from the email documents such as specific phrases, email HTML content, text based phrases and URL links.
- Email subject.
- URL-based attributes, e.g. the presence of an email address within the link, the general attributes like length, top level domains, etc..
- Script-based features, e.g. javascript code that change the email's behaviour/appearance based on user interaction.
- Sender-based features, e.g. sender's email address, information and location as an example.

All the above items can be classified as private information that may be used to identify both the sender and recipient of an email. This poses as a privacy violation even though it might aid in the detection of phishing emails. The use of these features that would not normally be available for a centralized machine learning model could be made possible in a federated learning setup.

III. PRIVACY IN FEDERATEDLEARNING FOR PHISHING DETECTION

Traditional machine learning requires centralized data storage where all the training data is collected and stored. The centralized data is used for feature extraction and training, and a model is produced and distributed to endpoints. Distributed learning is another approach where data collected from endpoints and shared among a pool of servers in a master-slave architecture. One server is the controller that coordinates them all to produce a unified trained model.

Federated learning however is different from distributed learning in that the raw data is never shared with a centralized server or any nodes within the network. Every node in the network participates in the training of the model and not a centralized server or even a pool of centralized server, hence the name of its acronym "collaborative learning".

In federated learning shown in Figure 2, the locally trained model is the only element uploaded by endpoints, and a global model is pushed to clients from the centralized server. The local model is a miniature representation of the global model using the same algorithm; all the data manipulation, featurization, and model training is done on the client side. This local model or effectively the weights, biases and other parameters that are produced as an outcome of the local model training, are the only data that are transmitted. These weights, biases and parameters are averaged out creating a new collective or global model that incorporates all the local models collected from the various end users in the network.

Konecnˇ y et al. were the first to propose theˋ federated learning model [9]. It was proposed in order to address the increasing use of machine learning on distributed nodes and a need to optimize the underlying algorithms to adapt to this new challenge. Their motivation was the

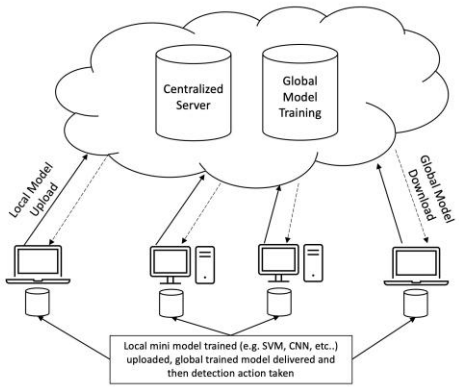

Figure 2. Federated learning

vast adoption of mobile phones and tablets which are rich data sources. Mobile devices are the most popular means of interacting online. Consequently they are at greatest risk of exposing private data and require extra protection.

Federated learning was introduced to enable data scientists to make use of private or otherwise unattainable data to produce more accurate models. A prominent use case of federated learning is the diagnosis of health conditions based on medical records of previous patients. On one hand, medical records are obviously considered to be sensitive information but on the other hand, the value of knowledge gained from medical records to improve future patient diagnosis is equally clear [21]. Federated learning was introduced for such cases to make use of data in a private and secure way. A similar situation is found in cyber security: the benefit of sharing data from endpoints for the mutual benefit of everyone is clear, but endpoint data can obviously include sensitive information.

Papernot et al. investigated attacks on confidentiality and privacy in machine learning [12]. They highlighted the possibility that malicious actors are able to extract information from the model itself, especially if the model represents confidential information as in financial or medical applications, for example. Membership tests can be used by malicious actors to prove the presence of data in a dataset from the model's output parameters without having access to the training data. In phishing scenarios, the screenshot of a electronic document maybe used as an indicator to classify the nature of this document. Models are trained to match on similar looking screenshots containing the phishing lures used by threat actors and other screenshots of benign documents. Machine learning models always perform better on their training data and will get higher confidence scores when they run through these models. Malicious actors may take advantage to reveal the training data used to train that model and learn what data was used in the training set. Imagine a malicious actor gets access to the facial recognition model used to match on certain people and runs a membership inference attack on this model. They could discover the identity of the people whose faces were used to train that model.

Among the common attacks on machine learning models are ”model inversion attacks”, a form of model inference where access to a machine learning model can lead to revealing sensitive information about the user. Fredrickson et al. described a model inversion attack to recover recognizable images of the user by using the name and access to the machine learning model [6]. In the attack, access to the machine learning model and the name of the victim were sufficient to reproduce a clear image of the victim, highlighting the severity of this attack in terms of leaking personal information. Since the attack requires access to the model, the attacker would need to obtain it somehow, possibly by compromising the device (not discussed here) or through the model update mechanism.

To address this vulnerability, secure aggregation protocols were introduced but they suffer from high communication and computation overhead as well as require many communication rounds. Fereidooni et al. proposed SAFELearn, a secure aggregation protocol that protects against the inference attacks in an efficient way [5]. Their implementation requires no more than two communication rounds and does not rely on any expensive cryptographic algorithms. Out of several experiments, one for a network intrusion detection system took 0.5 seconds to aggregate 500 models with more than 300,000 parameters.

In another effort to protect the privacy of users, homomorphic encryption for deep learning models was proposed by Gilad et al. [8]. Homomorphic encryption allows training neural networks with data in encrypted form to preserve the privacy of the user's PII. Homomorphism by definition is a mathematical structure preserving transformation. In this context, it allows mathematical operations on the data while it is encrypted. This seemed like a good solution to the privacy leakage until Bae et al. found that homomorphic encryption produces a trained model with less accuracy and efficiency, due to the mathematical alteration of the data in order to encrypt it [2]. The approaches above can be used to augment the current implementations of federated learning to not only address the privacy preserving aspect when transferring data from endpoints to the centralized servers but also protect the model against any privacy compromising attack on the endpoints.

Comparing the data leakage in the case where a malicious actor successfully compromises both the federated learning workflow against the centralized learning workflow:

- Centralized learning - the raw data PII data is compromised.
- Federated learning - model/model parameters are compromised.

IV. ACCURACY IN FEDERATEDLEARNING

One of the greatest concerns about the adoption of federated learning is possible degradation in the overall accuracy compared to centralized machine learning. Research so far in federated learning for cyber security, for example to detect malicious activities, indicates that detection accuracy can be maintained in general.

Nguyen et al. examined the case of detecting the Mirai botnet from network data using federated learning [10]. They extracted TCP packet features and used GRU (gated recurrent unit), a gating mechanism used in RNNs (recurrent neural networks) similar to LSTM (long short term memory) that is widely used for detecting anomalies in time series. They were able to achieve 95.6% accuracy in 256 msec in detecting the Mirai malware.

Bakopoulou et al. used federated learning with features extracted from HTTP (hypertext transfer protocol) packets, splitting the contents into words using a heuristic approach and some of the HTTP keys such as uri and cookies [3]. They used the SVM (support vector machine) algorithm to detect advertisement requests and PII exposure. They compared the accuracy achieved by different modes: federated mode, centralized mode, and local mode. They found that federated learning achieved significantly higher accuracy than the local mode and comparable accuracy to the centralized mode.

Preuveneers et al. experimented with federated learning for anomaly-based intrusion detection using network flow information [13]. They extracted 78 features about network flows from a real world intrusion dataset (CICIDS2017 dataset) which were then reduced to 50 features. At first, a centralized deep neural network with three hidden layers was used to detect anomalies achieving an accuracy of 97%. The same data using federated learning achieved similar results, although it required significant tuning on each of the nodes in the federated learning network.

Zhao et al. used the same CICIDS2017 dataset along with ISCXVPN2016 and ISCXTor2016 [23]. They were able to detect intrusions, VPN and TOR traffic anomalies. They used a multi task deep neural network in federated learning, extracting 23 features from the data and achieving an accuracy of around 97%, better than other centralized algorithms tested with deep neural networks, logistic regression, K nearest neighbor, decision tree, and random forest.

Although most of the literature available about the applications of federated learning in detecting malicious activity are based on network traffic anomalies, the same analogy can be used for detecting endpoint anomalies. The use of artificial intelligence in the detection of malicious execution is not new and has been used in both academic and commercial applications.

Zhao et al. [22] conducted an experiment using the shell block arguments. They used both a federated model and a centralized model applying

LSTM. The results are very similar, achieving 99.21% and 99.51% accuracy respectively, hence showing that federated learning and centralized learning with the same algorithm and dataset can produce very similar results with a very slight difference of less than 1% accuracy among all the above experiments. The slight drop in accuracy can be overlooked due to the benefit of reducing the privacy exposure drastically, the sacrifice of trading private information for protection is no longer necessary.

V. CONCLUSIONS AND FUTUREWORK

The importance of privacy preserving machine learning has been enhanced by wide-ranging regulations such as the UK General Data Protection Regulation (GDPR), California Consumer Privacy Act (CCPA), Children's Online Privacy Protection Rule (COPPA) and the EU AI act. The adoption of federated learning will allow the use of data present on all nodes in the network while eliminating the need for this data to be moved centrally. The adoption of this technology would also address scalability issues associated with centralized machine learning for malware detection due to the need to move and store massive amounts of data uploaded by endpoints.

There may be other benefits from reducing the demands on network bandwidth to upload all this data as well as the the computation and processing power to train machine learning models in a centralized location. Federated learning would make it easier for more users to join these networks and would enable more data to be analyzed, which otherwise would be inaccessible because of privacy concerns. Hence federated learning may enable the construction of more accurate models by facilitating running privacy preserving, scalable and cost effective malware detection systems. Traditionally federated learning has had a lower accuracy rate than its counterpart of centralized machine learning when training a model based on the same dataset. The goal of our research is to study the advantages of the adoption of this new technology against its potential deficiencies.

Security researchers have so focused on the accuracy of the trained models and the have not tested the limits of the privacy preservation aspects. There has been little study on the other benefits that come with the adoption of federated learning, such as scalability of the system and the resource utilization or optimization. Our research goals are to study the impact of adoption of this technology from several of these different perspectives. We plan to run experiments to measure the performance aspects of federated learning and test scalability with a large number of endpoints to quantify the privacy leakage risk. Common attacks on federated learning will also be conducted to measure the privacy implications on several implementations of federated learning leveraging open source frameworks like FADO [14].

Last but not least, the accuracy of federated learning will be evaluated in two scenarios: testing using the same dataset in a federated learning environment and centralized tradition machine learning environment; assuming that more users would opt in to sharing their data in this privacy preserving manner and have a larger number of samples in the dataset used to test federated learning verifying whether the increase in the quantity of the data would result in higher accuracy.

REFERENCES

1. Tuomas Aura, Thomas A Kuhn, and Michael Roe. Scanning electronic documents for personally identifiable information. In *Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 41–50, 2006.
2. Ho Bae, Jaehee Jang, Dahuin Jung, Hyemi Jang, Heonseok Ha, and Sungroh Yoon. Security and privacy issues in deep learning. *arXiv preprint arXiv:1807.11655*, 2018.
3. Evita Bakopoulou, Balint Tillman, and Athina Markopoulou. Fedpacket: A federated learning approach to mobile packet classification. *IEEE Transactions on Mobile Computing*, 2021.
4. DataProt. What's On the Other Side of Your Inbox - 20 SPAM Statistics for 2023. https://dataprot.net/statistics/spam-statistics/.
5. Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Helen Mollering, Thien Duc¨ Nguyen, Phillip Rieger, Ahmad-Reza Sadeghi, Thomas Schneider, Hossein Yalame, et al. Safelearn: Secure aggregation for private federated learning. *IACR Cryptol. ePrint Arch.*, 2021:386, 2021.

6. Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.
7. Luigi Gallo, Danilo Gentile, Saverio Ruggiero, Alessio Botta, and Giorgio Ventre. The human factor in phishing: Collecting and analyzing user behavior when reading emails. *Computers Security*, 139:103671, 2024.
8. Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*, pages 201–210. PMLR, 2016.
9. Jakub Konecnˇ y, H Brendan McMahan, Daniel Ramage,` and Peter Richtarik. Federated optimization: Distributed´ machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.
10. Thien Duc Nguyen, Samuel Marchal, Markus Miettinen, Hossein Fereidooni, N Asokan, and Ahmad-Reza Sadeghi. D¨ıot: A federated self-learning anomaly detection system for iot. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 756–767. IEEE, 2019.
11. Adam Oest, Yeganeh Safaei, Adam Doupe, Gail-Joon´ Ahn, Brad Wardman, and Kevin Tyers. Phishfarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1344–1361, 2019.
12. Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P Wellman. Sok: Security and privacy in machine learning. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 399–414. IEEE, 2018.
13. Davy Preuveneers, Vera Rimmer, Ilias Tsingenopoulos, Jan Spooren, Wouter Joosen, and Elisabeth IlieZudor. Chained anomaly detection models for federated learning: An intrusion detection case study. *Applied Sciences*, 8(12):2663, 2018.
14. Filipe Rodrigues, Rodrigo Simoes, and Nuno Neves.˜ Fado: A federated learning attack and defense orchestrator. In *2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 141–148. IEEE, 2023.
15. Aakanksha Saha, Jorge Blasco, and Martina Lindorfer. Exploring the malicious document threat landscape: Towards a systematic approach to detection and analysis. In *2024 IEEE European Symposium on Security and Privacy Workshops (EuroSPW)*, pages 533–544, 2024.
16. Said Salloum, Tarek Gaber, Sunil Vadera, and Khaled Shaalan. Phishing email detection using natural language processing techniques: a literature survey. *Procedia Computer Science*, 189:19–28, 2021.
17. N Selamat and F Ali. Comparison of malware detection techniques using machine learning algorithm. *Indones. J. Electr. Eng. Comput. Sci*, 16:435, 2019.
18. Priyansh Singh, Shashikala Tapaswi, and Sanchit Gupta. Malware detection in pdf and office documents: A survey. *Information Security Journal: A Global Perspective*, 29(3):134–153, 2020.
19. Eran Toch, Claudio Bettini, Erez Shmueli, Laura Radaelli, Andrea Lanzi, Daniele Riboni, and Bruno Lepri. The privacy implications of cyber security systems: A technological survey. *ACM Computing Surveys (CSUR)*, 51(2):1–27, 2018.
20. Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017.
21. Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5(1):1–19, 2021.
22. Ruijie Zhao, Yue Yin, Yong Shi, and Zhi Xue. Intelligent intrusion detection based on federated learning aided long short-term memory. *Physical Communication*, page 101157, 2020.
23. Ying Zhao, Junjun Chen, Di Wu, Jian Teng, and Shui Yu. Multi-task network anomaly detection using federated learning. In *Proceedings of the Tenth International Symposium on Information and Communication Technology*, pages 273–279, 2019.
24. Artificial Intelligence Act, Regulation (EU) 2024/1689. (2024). Official Journal of the European Union, L 309, 1–60. http://data.europa.eu/eli/reg/2024/1689/oj