Deep Learning For Ransomware Detection Using Encrypted Network Traffic

Christopher Lamprecht*
LMPCHR002myuct.ac.za
University of Cape Town
Cape Town, Western Province, South Africa

Abstract

In recent years, ransomware has become more common, and destructive, affecting many organisations and individuals daily. On top of this, it has become more easily accessible to malicious attackers. Currently, there are many network intrusion detection techniques and tools available, however, these carry a large burden, requiring help from experts, cumbersome analysis techniques and heavy computing resource requirements. Along with this, many of these tools only offer methods for the detection and removal of malware once it has infected the system and lack in ability to detect it before it causes damage. Deep learning has recently become a tool for network intrusion detection, resolving many of the burdens mentioned above. Despite the large amounts of research available for network intrusion detection using deep learning, there are only a few methods targeted specifically at ransomware. In this paper, we review the various techniques that have been taken for network intrusion and ransomware detection, including deep learning techniques, along with the strengths and weaknesses of these approaches. In particular, we look at the techniques that can be used within community networks, which come with additional resource and analysis constraints.

Keywords: datasets, deep networks, ransomware, community networks

ACM Reference Format:
Christopher Lamprecht. 2023. Deep Learning For Ransomware Detection Using Encrypted Network Traffic. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Network traffic classification has vast applications. One of these applications is providing Quality of Service (QoS), in which different priorities are assigned to different applications, such as video traffic requiring high-speed traffic transmissions, as opposed to applications such as email. Another application is network intrusion detection - a mechanism for monitoring network activity, and identifying malicious traffic.

Previously, the tools created for network intrusion detection have relied on using rules and code signatures that human experts have created. These can take large amounts of time to make and can often only be made after a certain attack has happened at least once, making these tools vulnerable to zero days attacks - newly discovered vulnerabilities used to harm a system, that a vendor has 'zero days' to fix. Moreover, these tools struggle to cope with modern network traffic due to their inability to work with encrypted traffic. Additionally, these approaches commonly rely on port-based methods, such as deep packet inspection, and are ineffective due to port obfuscation and non-standard port numbers [26]. A newer approach to network intrusion detection is machine learning, which can work with encrypted traffic. Some examples of this are the random forest algorithm and the k-nearest-neighbour algorithm. However, the ML approach can be computationally expensive and often requires human-engineered features[17], making this approach unsuitable to be used when there are compute resource constraints.

DL - a branch of machine learning, in which the neural networks have three or more layers - has been shown to reduce the burdens of classical and machine learning approaches, being more lightweight, needing less human-engineered features and having greater learning abilities [25]. This makes deep learning more suitable for use in systems where there are computing resource constraints, such as community networks - large-scale, distributed and decentralized systems, often used in rural areas. Community networks often function on low-cost devices, providing challenges when working with heavier network traffic and flow [6].

This paper aims to review work that has been done using DL to detect ransomware in encrypted traffic. Ransomware can be broken down into two types: Locker Ransomware and Crypto Ransomware. Locker ransomware locks a user out of a system, preventing them from using the system. Crypto

ransomware locks files within a system, preventing the user from accessing these files. Most of the time, a ransom is demanded, upon which the system/files are unlocked once it has been paid.

To our knowledge, there is little research that has been published with regard to network intrusion detection on encrypted traffic. There is even less when it comes to ransomware intrusion detection with resource-constrained community networks. This is a gap in research that needs to be filled as ransomware poses the greatest cyber threat to digital infrastructure, making it the most dominant form of attack [16].

The paper is organized as follows. Section 2 gives an overview of the topics that will be discussed in and used in our research. Additionally, this section discusses the most common and favourable deep learning models used in network traffic classification and network intrusion detection. Section 3 discusses previous work done on network intrusion detection and traffic classification. In Section 4, the challenges of network classification are discussed. After this, Section 5 discusses what makes up a good dataset, and then Section 6 reviews datasets that contain encrypted ransomware network traffic. Section 7 provides a discussion and analyses of the methods used pertaining to ransomware detection and insights gained from the paper. Lastly, Section 8 summarizes the key conclusions of the paper.

2 Background

2.1 Community Networks

Community networks are networks that are mostly distributed, decentralised, low-resource systems that rely on low-cost, wireless devices to connect the nodes within the network. Community networks strive to reduce the gap in digital access by offering more affordable internet connectivity in areas that are usually rural or underdeveloped, where people may otherwise face difficulties in accessing reliable and affordable internet services. [8] [24]. It is important to bare in mind the resource constraints when working with community networks. The largest of these constraints are bandwidth constraints, affecting the user's application usage. Another constraint is the security constraints, as the previous security techniques cannot be applied to due them being computationally resource-heavy.

2.2 Online Classification

Traffic classification can be done in two, online and offline classification.

With offline classification, the network traffic does not need to be classified in real-time. In offline classification, the analyses are done on pre-recorded network traffic that has been captured. It allows for deeper analyses of the network, usually over a longer period of time. Offline classification is used for monitoring and analyses of the traffic, such as troubleshooting a network or analysing the network for security flaws.

With online traffic classification, the traffic has to be analysed immediately, in real-time, using only the first couple of packets. This is often used in providing QoS since priorities need to be assigned while the user is using the network. In our case, online classification is needed as network intrusion needs to be detected in real-time, giving the network the ability to block malicious traffic, not allowing it to ever enter the system.

The analyses of online classification can be achieved through either flow-based or packet-based approaches. Flow-based traffic classification involves grouping packets by their flows, looking at destination IP addresses/port numbers and protocol types. Traffic can then be classified using the first few packets and determining which group of flow they belong to. Flow analyses can give a more holistic view of the network than what may be achieved with packet analyses. Packet-based traffic classification involves analysing the individual packets in the traffic, without associating them with a certain flow. This allows for a more detailed inspection of the network than flow-based analyses as each packet's contents are carefully examined. Both flow-based and packet-based classification have their advantages [3] and this paper will look into both techniques.

2.3 Deep Learning

Deep learning has the ability to identify higher-level features from the input provided to it. The lower layers in the deep network can identify the edge cases, and the higher lowers identify the more obvious features. There the system automatically learns features giving it the ability to learn complex patterns from the raw input, without needing human-designed features [15]. Additionally, deep learning has greater learning abilities compared to common machine learning algorithms, allowing them to obtain higher accuracies [25].

2.3.1 Model Evaluation.

In Section 2.4, are the deep learning models that will be used to detect ransomware using encrypted network traffic. How they work and why these models were chosen will be discussed, looking at the work of other research in this area. The different models that have been used for intrusion/ransomware detection will be evaluated. The main metrics that will be looked at are accuracy, precision, recall and F1 score. These metrics are calculated using the following definitions: *True positives* - the number of times each label is correctly identified; *False Positives* - the amount of times a label is incorrectly predicted as positive; *True negatives* - the amount of times a label is correctly predicted as negative and *False Negatives* - the number of times a label is incorrectly predicted as negative. The goal

of the deep learning models is to maximise the true positives and negatives whilst minimising the false positives and negatives.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precission = \frac{TP}{TP + FP}$$

$$F1score = \frac{Precision \times Recall}{Precision + Recall}$$

2.4 Deep Learning Approaches

2.4.1 Longest Short-Term Memory (LSTM). Longest short-term memory (LSTM) is a type of recurrent neural network (RNN), that a specifically aimed at analysing long-term dependencies in sequential data [17]. The LSTM does this by making use of a memory cell and controlling the flow of information in and out of the cell using three gates: forget gate, input gate and output gate. This allows the neurons within the network to retain temporal information. LSTMs are a commonly used branch and RNNs as they help solve the vanishing gradient problem - a problem that happens during backward propagation, when the gradients become very small, hindering the network's ability to update the parameters effectively. The LSTM helps solve this problem by selectively retaining and forgetting information over time [9]. In network classification and intrusion detection, traffic flow is made up of a sequence of packets, and the data of the current packet could be linked to the previous one. Additionally, the data within each packet's payload is sequential. This makes LSTMs a favourable model to use for network classification and intrusion detection.

Zeng et al. proposed an LSTM that utilises time-related information [25] contained within the network traffic. The input to this LSTM was a graph. This is different to previous work in which the input was traffic-byte data transformed into character-data [21]. The LSTM produced by Zeng et al. achieved the highest accuracy of 99.41% with regard to network intrusion detection. This was compared to a CNN and an SAE which attained lower accuracies. On top of this, the LSTM had much lower storage requirements, compared to previously used ML models. This makes the LSTM more suitable to be used within a community network environment.

Wang et al. [22] proposed a system denoted hierarchical spatial-temporal feature-based intrusion detection system (HAST-IDAS) for intrusion detection of malware. This model used a combination of a CNN and LSTM that work together. CNN is used to learn the low-level spatial features and LSTM to learn the high-level temporal features. The model achieved an impressive accuracy averaging above 99.6% on 9 different types of malicious traffic.

2.4.2 Convolutional Neural Networks (CNN). CNN's primary use is in image pattern recognition. They comprise of three different layers. These layers are the convolutional layers, pooling layers and fully-connected layers [14]. The convolution layer employs image kernels containing a limited set of trainable parameters to capture significant spatial features from the input. The pooling layer reduces the spatial dimensionality of the data and the fully-connected layer consists of neurons that are fully connected to each of the neurons in the layers next to it. These three types of layers would be grouped under the 'hidden layer' of the CNN. The overall structure of a CNN is an input layer, a hidden layer, and an output layer. Although CNNs have mostly been used in image recognition, they are one the most employed deep learning models for network intrusion detection.

Mohammad et al. proposed that 1 Dimensional CNNs are the optimal choice when working with network traffic classification due to their ability to capture spatial dependencies between adjacent bytes within the network packets (1-dimensional data), which helps find patterns for each protocol or application [12]. This is opposed to 2-dimensional CNNs which recognize patterns in 2D data, such as images. Aceto et al. [1] found that 1D and 2D filters obtain the same results and hence state that network traffic can be considered as 1 dimensional, mitigating the need for 2D filters. Furthermore, it was shown in a past study that a 1D CNN is able to achieve higher accuracy when classifying network traffic than a 2D CNN [23]

Zeng et al. [25] demonstrated that a 1D CNN performed with the best results when classifying encrypted network traffic, obtaining an accuracy of 99.85%. This was compared to LSTM and SAE deep learning models which obtained accuracies of 99.22% and 98.74%. The CNN used consisted of 2 convolutional layers, 2 max-pooling layers, 2 local response layers and a densely connected layer which used a softmax classifier.

Although Multi-layer perceptrons (MLP) are similar to CNNs - both consisting of many layers with learnable parameters they struggle to work with high dimensional data. Rezaei and Liu [17] state that in order to overcome this problem, CNNs can be used. This is an essential criterion, due to the high dimensionality of modern encrypted network traffic.

Chen et al. [7] employed a CNN model consisting of 2 convolutional layers, 2 pooling layers, and 3 fully connected layers to classify protocols and applications. They transformed the initial time series data into 2-D images using reproducing kernel Hilbert space (RKHS) embedding. Their CNN model demonstrated superior performance, obtaining higher accuracy when compared to traditional machine learning methods and MLP in the task of protocol and application classification.

2.4.3 Stacked Auto Encoder (SAE). An autoencoder refers to a type of artificial neural network that undergoes training to compress input data into a more concise and compact representation, with a reduced number of dimensions. Subsequently, it is designed to reconstruct the original input data from this compressed representation. SAEs consist of multiple autoencoders stacked on top of each other. The deep learning models mentioned in sections 2.4.1 and 2.4.2 used supervised learning - training data is labelled - to perform their tasks. SAEs are used for unsupervised learning tasks - training data is unlabelled - such as dimensionality reduction, feature extraction and data compression.

SAE are can be used alongside other deep learning models, such as a CNN [17]. The SAE can extract meaningful features, that are passed to the CNN as inputs. Along with this, the SAE can perform dimensionality reduction, which can make training the CNN more efficient and improve its performance. This comes in handy due to network data being highly dimensional. Ferreira et al. [10] highlight the large volume of data network traffic and the need for its dimensions to be reduced. An autoencoder is used to do this.

Zeng et al. propose an SAE applied with a sigmoid activation function [25]. Although the SAE produced the least favourable results, the results produced are still commendable, achieving an accuracy of 98.74% for network traffic classification. Despite the SAE not producing the best results, it is important to evaluate their functionality, and how they can be combined with other deep learning models to produce more desirable results.

3 Related Work

Zeng et al. [25] present a framework for a deep learning-based network encrypted traffic classification and intrusion detection framework. In the paper, three deep-learning algorithms are tested and employed. These are namely, a Convolutional Neural Network (CNN), a Long Short-Term Memory (LSTM) and a Stacked Auto-Encoder (SAE). The authors also address storage and computational expense factors by providing a lightweight framework. The paper accurately classifies encrypted traffic, with the CNN obtaining an accuracy of 99.85%. While LSTM obtains an accuracy of 99.41% for intrusion detection. The paper does not specifically target ransomware and hence may come short when detecting the vast amounts of ransomware families.

Modi et al. [13] performed ransomware detection using machine learning on encrypted network traffic. The paper outlines related research in this area and highlights the weaknesses of each of those papers, mostly focussing on the limited research when dealing with encrypted traffic and ransomware. The authors mention the lack of datasets containing encrypted ransomware network traffic and hence solve this by collecting a dataset that was made public [18]. Three machine learning classifiers were used, support vector machine (SVM), random forest and logistic regression. In the results, it is shown that the random forest performs best, obtaining an accuracy of 99.9% and a false-positive rate of 0%. These results are exceptional, however, these machine-learning approaches are computationally intensive and require large storage, making them less ideal for resource-constrained nodes.

It is often the case that ransomware detection tools report an accuracy of 100%, but are only tested with one ransomware sample. Additionally, it is hard to test the results among these different tools as they use different samples and evaluation metrics [4]. Berrueta et al. discuss a public repository, containing ransomware traffic which was obtained during the attack on a large network. This dataset contains updated ransomware binaries since 2015 and can be used to test the majority of ransomware detection tools. Berrueta et al. provide a dataset that will be used to train our deep-learning models.

A common approach to traffic classification and network intrusion is payload inspection. This involves inspecting the payload of each packet (discussed more in Section 2.2) [11]. Previously, methods involve using regular expressions as signatures for the different protocols. These methods struggle when the regular expressions need to be updated for each new protocol released. Moreover, these methods become less efficient, or unusable when dealing with encrypted traffic. Sherry et al. proposed a system to perform deep packet inspection with encrypted data [19]. Although the paper only deals with the HTTPS protocol and does not use deep learning, the importance of working with encrypted traffic is highlighted.

4 Network Classification Challenges

In order to successfully detect malware/ransomware within a network, the network traffic needs to be successfully classified into its sub-categories. The traffic can roughly be separated into two sub-categories: benign traffic - normal network traffic that is considered 'safe' - and malicious traffic - network traffic that is considered harmful. However, recently, network classification has become more challenging.

Many different protocols are being applied in order to make network traffic more secure. Some of these protocols are HTTPS, SSL and SSH. This means that the payloads of different traffic have a large variety of encryption techniques, making the payloads harder to classify.

Another challenge is the use of adversarial machine learning in network intrusion detection [2]. Machine learning and deep learning can cope better with zero-day attacks due to their knowledge of the difference between normal traffic flow and malicious traffic flow, making them more appealing

than previous tools made by human experts. However, these learning models are susceptible to adversarial machine learning - a model that creates inputs for a learning model that are purposely made to trick the model, or that train the model incorrectly. In our case, these inputs would be malicious traffic that is disguised as benign traffic that could bypass are ransomware intrusion detection model, decreasing its accuracy and confidence.

The third challenge, and arguably the most impactful one, is the diversity of network traffic. With the vast number of applications, the move to cloud computing and the Internet of Things (IoT), network traffic has become more diverse. Additionally, networks are being created with the goal of making them as diverse as possible to achieve resilience. This means that extremely diverse datasets are needed to train the deep learning models, as a model that can accurately classify only a small range of network traffic, is not a useful one.

5 Data Collection And Selection

What makes a good deep learning model is the dataset, and what makes a good dataset is the quantity of data within it, and the quality of that data. Many datasets exist for traffic classification/network intrusion, however, there is no agreed-upon dataset that can be used to train each deep learning model. [17] states that this is due to three reasons: the impossibility of one dataset containing all possible traffic classes, no common data collection/labelling methods and the many different collection methods and scenarios. Because of this, datasets are usually chosen according to what the researcher wishes to classify/detect. Furthermore, the data can be collected in different locations within a network, eg. client/server/network edge etc.

For the dataset to have quality, it needs to be labelled reliably and needs to be representative. There are tools available for labelling data, called Deep Packet Inspection (DPI) modules. The deep learning model's accuracy depends on how well these tools label the data. These tools often fall short when working with encrypted traffic. Additionally, even in fully controlled environments, removing background traffic is a challenge as 70% of smartphone traffic is background, and only 30% is linked to user interactions [20]. Despite these limitations, capturing each category in a controlled environment is the standard approach.

For a dataset to be representative, it needs to contain diverse samples from different classes. Accuracy can drop dramatically when the Operating system used in the test set is different to that in the training set [17]. Many users need to be used when collecting the dataset, or else the dataset can overfit to user-specific features, rather than the traffics features.

6 Dataset

The dataset needed to train, test and validate the deep learning models should have the criteria of containing ransomware traffic that is encrypted. This section reviews the work done using datasets with these criteria. Berrueta et al. [5] presents a ransomware PCAP Repository. The dataset was obtained by capturing the network traffic whilst ransomware binaries are encrypting a set of files shared from an SMB server. In total, there are 94 samples from 32 different ransomware families. The main goal of the authors is to provide a dataset for testing both new and old ransomware detection tools.

Modi et al [13] provides a dataset including network activity from ransomware and benign applications. The dataset consists of 20 ransomware families, with a total of 666 samples. This is a low number of samples for the purpose of training deep learning models, however, due to the limited amount of datasets available, this dataset provides a step in filling this gap.

7 Discussion

From the literature reviewed in this paper, it is clear that there is a lot of research that has been done on network traffic classification and network intrusion detection. This section will discuss which of the approaches/constraints reviewed above are most applicable to our research on ransomware detection.

With respect to ransomware intrusion detection, there is little research that has been done, and even less so when it comes to encrypted traffic. Ransomware is, however, a type of malware, and a deep learning model that can accurately detect malware should be able to detect ransomware, so long as there is a well-maintained and suitable dataset. Therefore, the research done in this literature review can be applied to our research topic. Furthermore, there are suitable ransomware datasets, as shown in Section 6. However, compared to other available datasets for network intrusion detection, the datasets containing encrypted ransomware traffic are less dense, containing fewer samples. This could affect the results of the deep learning models, making them prone to overfitting.

When it comes to deep learning, many insights were gained. First, it is shown that although more traditional ML models can be used to detect/classify certain network traffic, it is clear that DL models are the better choice, obtaining higher accuracies and consuming fewer compute resources. On top of this, DL models are more attractive to our field of study, as they are less computationally intense than ML models. This is an important criterion as our research is aimed at community networks, which function on low-resource, low-cost devices. The next insight is that CNNs perform the best when it comes to network traffic classification. Additionally, 1D CNNs are preferred over 2D CNNs for network

classification/intrusion detection, due to the sequential nature of network traffic. However, Chen et al. [7] showed that a 2D CNN can accurately classify protocols and applications when the network traffic is converted into a 2D image. It is important to note that this conversion takes additional computational resources than needed.

When it comes to classification, it was highlighted that online classification is the suitable approach for our research. This is due to the fact that to detect an intrusion, the network traffic will have to be analysed in real-time. The packet-based approach adheres to real-time requirements more effectively because packets can be sorted without being recognized as part of a data stream beforehand. Nonetheless, analysing packets using the flow-based approach carries the advantage of consuming less computational resources as it only requires specific characteristics extracted from these packets, instead of analyzing the entirety of the packet's contents.

Most existing network intrusion and classification tools do not deal with encrypted traffic. However, from the papers reviewed, it is clear that there are ways of dealing with encrypted traffic, especially when it comes to deep learning. Two techniques were found and documented: flow analyses - where features like packet length and inter-arrival times are used - which is independent of encryption, and packet inspection - where patterns are learnt in the network traffic without decoding the encrypted traffic.

The use of encryption techniques, diverse protocols, and adversarial machine learning all pose significant obstacles to accurately detecting malicious traffic. Additionally, the increasing diversity of network traffic due to the proliferation of applications, cloud computing, and the Internet of Things (IoT) means that deep learning models must be trained on extremely diverse datasets to be effective.

8 Conclusions

Network intrusion detection has been extensively studied and clearly has a significant role in network management and security. Furthermore, businesses and individuals are affected daily by malicious traffic, with ransomware having one of the highest negative impacts. Hence it is of utmost importance that this ransomware can be detected before it infects the network users.

In this literature review, we provided an overview of what ransomware detection is, the constraints of modern-day network traffic, network traffic classification and community networks. Following this, a deep analysis was performed on the research that has been done on these topics. From the analysis, it was clear that network intrusion detection using DL should be performed using either flow-based analysis or packet-based analysis.

In terms of the datasets, two that contained encrypted ransomware network traffic were examined. Out of these two, one had relatively low samples. This should be kept in mind when building intrusion detection models, as a small dataset can lead to overfitting.

The DL learning models that achieved the greatest performance in the literature reviewed, were CNNs and LSTMs. However, combining an SAE with these models can also produce desirable performance and results. This serves as a guide when selecting which deep learning models we should use in our ransomware detection system. Furthermore, it was found that deep learning models are less computationally intense and are hence more desirable than traditional ML methods when working with low-resource community networks.

References

[1] Aceto, G., Ciuonzo, D., Montieri, A., and Pescapé, A. Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges. *IEEE Transactions on Network and Service Management 16*, 2 (2019), 445–458.

[2] Alhajjar, E., Maxwell, P., and Bastian, N. Adversarial machine learning in network intrusion detection systems. *Expert Systems with Applications 186* (2021), 115782.

[3] Andreas, B., Dilruksha, J., and McCandless, E. Flow-based and packet-based intrusion detection using blstm. *SMU Data Science Review 3*, 3 (2020), 8.

[4] Berrueta, E., Morato, D., Magaña, E., and Izal, M. Open repository for the evaluation of ransomware detection tools. *IEEE Access 8* (2020), 65658–65669.

[5] Berrueta, E., Morato, D., Magaña, E., and Izal, M. Open repository for the evaluation of ransomware detection tools. *IEEE Access 8* (2020), 65658–65669.

[6] Braem, B., Blondia, C., Barz, C., Rogge, H., Freitag, F., Navarro, L., Bonicioli, J., Papathanasiou, S., Escrich, P., Baig Vinas, R., et al. A case for research with and on community networks, 2013.

[7] Chen, Z., He, K., Li, J., and Geng, Y. Seq2img: A sequence-to-image based approach towards ip traffic classification using convolutional neural networks. In *2017 IEEE International conference on big data (big data)* (2017), IEEE, pp. 1271–1276.

[8] Dicks, M., and Chavula, J. Deep learning traffic classification in resource-constrained community networks. 7.

[9] Dicks, M., Tooke, J., and Weisz, S. A comparative evaluation of deep learning approaches to online network traffic classification for community networks.

[10] Ferreira, D. C., Vázquez, F. I., and Zseby, T. Extreme dimensionality reduction for network attack visualization with autoencoders. In *2019 International Joint Conference on Neural Networks (IJCNN)* (2019), IEEE, pp. 1–10.

[11] Khalife, J., Hajjar, A., and Diaz-Verdejo, J. A multilevel taxonomy and requirements for an optimal traffic-classification model. *International Journal of Network Management 24*, 2 (2014), 101–120.

[12] Lotfollahi, M., Jafari Siavoshani, M., Shirali Hossein Zade, R., and Saberian, M. Deep packet: A novel approach for encrypted traffic classification using deep learning. *Soft Computing 24*, 3 (2020), 1999–2012.

[13] Modi, J., Traore, I., Ghaleb, A., Ganame, K., and Ahmed, S. Detecting ransomware in encrypted web traffic. In *Foundations and Practice of Security: 12th International Symposium, FPS 2019, Toulouse, France, November 5–7, 2019, Revised Selected Papers 12* (2020), Springer, pp. 345–353.

[14] O'Shea, K., and Nash, R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458* (2015).

[15] Picon Ruiz, A., Alvarez Gila, A., Irusta, U., Echazarra Huguet,

J., et al. Why deep learning performs better than classical machine learning? *Dyna Ingenieria E Industria* (2020).

[16] Reshmi, T. Information security breaches due to ransomware attacks-a systematic literature review. *International Journal of Information Management Data Insights 1*, 2 (2021), 100013.

[17] Rezaei, S., and Liu, X. Deep learning for encrypted traffic classification: An overview. *IEEE communications magazine 57*, 5 (2019), 76–81.

[18] Sherif Saad, Issa Traore, A. A. G. B. S. D. Z. W. L. J. F. P. H. Detecting p2p botnets through network behavior analysis and machine learning. 100013.

[19] Sherry, J., Lan, C., Popa, R. A., and Ratnasamy, S. Blindbox: Deep packet inspection over encrypted traffic. In *Proceedings of the 2015 ACM conference on special interest group on data communication* (2015), pp. 213–226.

[20] Stöber, T., Frank, M., Schmitt, J., and Martinovic, I. Who do you sync you are? smartphone fingerprinting via application behaviour. In *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks* (2013), pp. 7–12.

[21] Torres, P., Catania, C., Garcia, S., and Garino, C. G. An analysis of recurrent neural networks for botnet detection behavior. In *2016 IEEE biennial congress of Argentina (ARGENCON)* (2016), IEEE, pp. 1–6.

[22] Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., and Zhu, M. Hast-ids: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE access 6* (2017), 1792–1806.

[23] Wang, W., Zhu, M., Wang, J., Zeng, X., and Yang, Z. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In *2017 IEEE international conference on intelligence and security informatics (ISI)* (2017), IEEE, pp. 43–48.

[24] Weisz, S., and Chavula, J. Deep learning traffic classification in resource-constrained community networks. 22.

[25] Zeng, Y., Gu, H., Wei, W., and Guo, Y. $deep - full - range$: a deep learning based network encrypted traffic classification and intrusion detection framework. *IEEE Access 7* (2019), 45182–45190.

[26] Zhang, J., Chen, X., Xiang, Y., Zhou, W., and Wu, J. Robust network traffic classification. *IEEE/ACM transactions on networking 23*, 4 (2014), 1257–1270.