Duygu Sinanc Terzi

GRAMIAN ANGULAR FIELD TRANSFORMATION-BASED INTRUSION DETECTION

Abstract *Cyber threats are increasing progressively in their frequency, scale, sophistication, and cost. The advancement of such threats has raised the need to enhance intelligent intrusion-detection systems. In this study, a different perspective has been developed for intrusion detection. Gramian angular fields were adapted to encode network traffic data as images. Hereby, a way to reveal bilateral feature relationships and benefit from the visual interpretation capability of deep-learning methods has been opened. Then, image-encoded intrusions were classified as binary and multi-class using convolutional neural networks. The obtained results were compared to both conventional machine-learning methods and related studies. According to the results, the proposed approach surpassed the success of traditional methods and produced success rates that were close to the related studies. Despite the use of complex mechanisms such as feature extraction, feature selection, class balancing, virtual data generation, or ensemble classifiers in related studies, the proposed approach is fairly plain – involving only data-image conversion and classification. This shows the power of simply changing the problem space.*

Keywords encoding intrusions as images, convolutional neural networks, Gramian angular fields, intrusion detection, network security

Citation Computer Science 23(4) 2022: 571–585

1. Introduction

Cyber-attacks, which are unauthorized actions against a computer or network that result in a security policy breach, constitute the fastest-growing crime. Cybersecurity Ventures estimated that cyber-attacks will cost the world \$6 trillion annually by 2021 [8]. Cyber-attacks can be evaluated in five classes according to their sources: network, host, software, physical, and human. Network threats rely on the manipulation of packet flows sent over a network; the most common forms are denial of service and distributed denial of service. Host attacks aim to compromise or disrupt the functions of a specific computer or system by running malware programs. Software threats are the executions of malicious code (as in code injection and cross-site scripting). Physical threats are attempts to network hardware or its configuration. Lastly, human threats include user-to-root, remote-to-local, or hijacking attacks.

Determining how to protect a network and finding possible security vulnerabilities are possible by the clear specification of a cyber-attack's life-cycle. Such a life-cycle is a sequence of events that an attacker goes through in order to infiltrate a computer or network; this generally consists of eight steps [26]:

1. Initial reconnaissance: an attacker conducts research on a target (system or people) in order to gather information and determines an attack methodology.
2. Initial compromise: the attacker executes malicious code on the system by exploiting its vulnerabilities.
3. Establish a foothold: the attacker maintains control over the system by installing a persistent backdoor or downloading additional utilities.
4. Escalate privileges: the attacker collects legitimate credentials with higher privileges to gain deeper access to the network.
5. Internal reconnaissance: the attacker explores the system and its environment to gain as much information as possible.
6. Move laterally: the attacker utilizes previously compromised accounts in order to move undetected around the network.
7. Maintain presence: the attacker sustains continuous and long-term control over the systems.
8. Complete mission: the attacker achieves the ultimate goal, which is information or data theft. After the mission is completed, most attackers do not leave the environment in case they they choose initiate a possible new mission.

Despite the dramatic increase in cyber-attacks for missions such as obtaining confidential information, deleting data, stealing credentials, or gaining higher privileges, cyber security technology promises new solutions to combat and reduce cyber-attacks. Cyber security constitutes various processes that are projected to protect data or networks from attacks. One of the most common cyber-security mechanisms is an intrusion-detection system (IDS).

IDS is a program that automates the processes of monitoring and analyzing traffic, defining abnormal activities, and generating alarms. An ordinary IDS is made up of sensors, an analysis engine, and a reporting system [16]. Sensors are placed in various network locations in order to collect information such as traffic statistics, service requests, packet headers, or file system changes. The analysis engine investigates the data that is collected by sensors and detects intrusions. Finally, the reporting system sends an alert to the network administrator when the analysis engine discovers a violation.

IDSs can be categorized into three categories according to their characteristics: the implementation method, detection method, and architecture.

Implementation methods are host-based, network-based, and hybrid [19]. Host-based IDS monitors activities in a system to detect local attacks. Network-based IDS audits network packets or flows to catch remote attacks. Packet-based data contains payload information, while flow-based data contains meta information about connections. A hybrid method is a combination of host-based and network-based IDS.

Detection methods are misuse-based, anomaly-based, and hybrid [4]. Misuse-based IDS evaluates network traffic against a set of signatures of known attacks. Misuse-based IDS is outstanding for detecting known attacks but not reasonable enough to detect zero-day attacks or multi-step attacks. Anomaly-based IDS uses the baseline profile of normal network activities as a reference in order to distinguish abnormal activities. Anomaly-based IDS is pretty good at detecting new attacks but generates a high number of false positive alarms. A hybrid method is a combination of misuse-based and anomaly-based methods.

The architecture of an IDS can be grouped as centralized, decentralized, and distributed [21]. Centralized IDS monitors a network and analyses collected data in a central processing unit. Decentralized IDS pre-processes data in multiple processing units in a hierarchical structure before it reaches the main processing unit. Distributed architecture uses multiple autonomous agents to both collect and process data.

Once malicious security events are detected by a system, actions that are called security responses are generated as being passive or active [20]. A passive response is received by the network administrator when abnormal behavior is detected, and an alert is issued. Active response involves taking automatic and immediate actions when malicious activities are detected in accordance with a predefined script.

For an IDS to be considered effective, it must have low false-positive and high detection rates. Besides, relying solely on the detection rate for an IDS evaluation will not reflect its real performance. Other important evaluation factors to be regarded are ease of use, the security of a system, power consumption, memory requirements, throughput, interoperability, and transparency [3]. In order to develop an effective IDS and overcome issues in a dynamic network environment, deep-learning methods are increasingly preferred. Unlike traditional machine-learning methods, deep-learning methods do not require feature engineering and domain knowledge. Deep learning has stronger fitting and generalization abilities due to its deep structure that contains multiple hidden layers. Therefore, deep learning-based techniques are highly effective in predicting new and complex intrusions.

In this study, a different perspective has been developed for intrusion detection. Network traffic flows were considered to be time series; these flows were transformed into images, and these images were classified with a deep-learning architecture. Experiments were carried out for both binary and multi-class classification, and the obtained results were compared with conventional machine-learning methods as well as related studies. According to the results, the proposed approach produced success rates that were close to those that have been achieved with sophisticated approaches by using such additional methods as feature extraction, feature selection, class balancing, virtual data generation, or ensemble classifiers.

2. State of the art

Since the 1980s, researchers have been working on intrusion-detection techniques. Summarizing the IDS techniques in the literature five categories have emerged: statistical, information-theoretic, supervised, unsupervised, and semi-supervised. Statistical techniques have been developed using statistical theories; for instance, chi-square theory [33] or principal component analysis [25]. Information-theoretic techniques use several measures such as information gain, entropy, or information cost to extract correlations between network traffic features [2, 29]. Supervised techniques mostly rely on the knowledge of security experts and the use of classification approaches. In supervised techniques, combinations of several approaches [23] are used in addition to traditional classification approaches [34]. Unsupervised techniques refer to using clustering approaches that do not require pre-labeled data for detection [13,27]. Clustering approaches are often used to create high-quality signatures or to group similar intrusions. Finally, semi-supervised techniques combine supervised and unsupervised methods to enhance the performance of IDS in ways like co-training [22] or self-training [14].

In addition to these methods, studies that transform intrusion data into images and classify them using deep-learning architectures have emerged in recent years. Xia et al. converted raw malware files to gray-scale images and classified them with a support vector machine [32]. To detect DoS and DDoS attacks on the Internet of Things, Hussain et al. chunked network traffic data, transformed chunks into image matrices, mapped the matrices to the RGB channels of an image, and classified the images with a convolutional neural network [10]. In another study, Mao et al. transformed time-series data in structural health-monitoring systems to images and detected data anomalies with generative adversarial nets and autoencoders [17].

Despite the many different approaches and solutions, intrusion detection is still an open research issue; this is due to the difficulty of overcoming various problems. One of these problems is the higher false detection rate. Misclassified system activities that have been previously unseen increase false positive rates, while the high frequency of new attacks that are launched in cyberspace increases the false negative rate. The other problem is the huge growth in real-time network traffic; this makes it difficult to model and evaluate IDS. Last but not least, the presence of an imbalanced class distribution in intrusion data sets can lead to the biased classification of a class with a majority of records or ignoring classes with minority records. Therefore, it is essential to develop new perspectives for IDS.

3. Methodology

In this study, a novel approach was developed for intrusion detection. The proposed approach consists of two main stages: transforming intrusions into images with Gramian angular fields, and classifying the images with convolutional neural networks. Gramian angular fields present a way of both preserving the temporal dependency between values and enabling the implementation of convolutional neural networks. In this way, convolutional neural networks better analyze the complexity of an image (in our case, an intrusion).

3.1. Gramian angular fields

A Gramian angular field (GAF) is a data-transformation method that represents time-series data as images, thus allowing image-based deep-learning methods to be applied. To create GAF images, time-series vector $X = \{x_1, ..., x_n\}$ with n number of samples is first normalized to $[-1, 1]$ or $[0, 1]$ by the following equations:

$$\tilde{x}^i_{-1} = \frac{(x_i - max(X) + (x_i - min(X)}{max(X) - min(X)} \tag{1}$$

$$\tilde{x}^i_0 = \frac{(x_i - min(X)}{max(X) - min(X)} \tag{2}$$

Next, rescaled time series $\tilde{X}$ is represented in the polar coordinates that are given in (3) by encoding the value as the angular cosine and the time stamp as the radius, where N is a constant factor to regularize the span of the polar coordinate system:

$$\begin{cases} \phi = \arccos(\tilde{x}_i), -1 \leq \tilde{x}_i \leq 1, \tilde{x}_i \in \tilde{X} \\ \\ \qquad\qquad r = \frac{t_i}{N} t_i \in \mathbb{N} \end{cases} \tag{3}$$

Finally, GAF can be determined by defining the angular perspective as the trigonometric difference of each point in the interval as given below, where I is a unit row vector [31]:

$$GAF = [\sin(\phi_i - \phi_j)] \tag{4}$$

$$= \sqrt{I - \tilde{X}^2}' . \tilde{X} - \tilde{X}' . \sqrt{\tilde{X}^2} \tag{5}$$

The resulting matrix and generated images are bijective. The position moves from the top-left to the bottom-right as time increases. In this way, GAF provides temporal correlations.

3.2. Convolutional neural networks

A convolutional neural network (CNN) is a widely used deep architecture that demonstrates effective performance in computer vision with the advantage of equivalent representations, parameter sharing, and sparse interactions [7].

A typical CNN is made up of three main types of neural layers: convolutional layers, pooling layers, and fully connected layers [30]. The convolutional layers serve as feature extractors that use various kernels to learn the feature representations of input images, the pooling layers are responsible for reducing the spatial dimensions of the feature maps that are generated by the convolutional layers, and the fully connected layers that follow the several convolutional and pooling layers interpret the feature maps and perform the function of high-level reasoning.

The CNN architecture plays a critical role in improving the performance of applications. For this purpose, various modifications such as structural reformulation, regularization, and parameter optimizations can be achieved [1]. Selecting the suitable architectural features in terms of input size, depth, robustness, etc. will be the key to success in the target task.

4. Experimental setup

The evaluation data sets are vital for validating the ability of the proposed methods to detect intrusive behavior. However, there are a few publicly available and up-to-date intrusion-detection data sets. This study uses the CIC-IDS 2017 data set, which contains benign and intrusion flows [24]. The data consists of network traffic that lasts five days in which certain attack scenarios are carried out; such scenarios include Botnet, DoS, DDoS, port scan, brute force, web attacks, and infiltration.

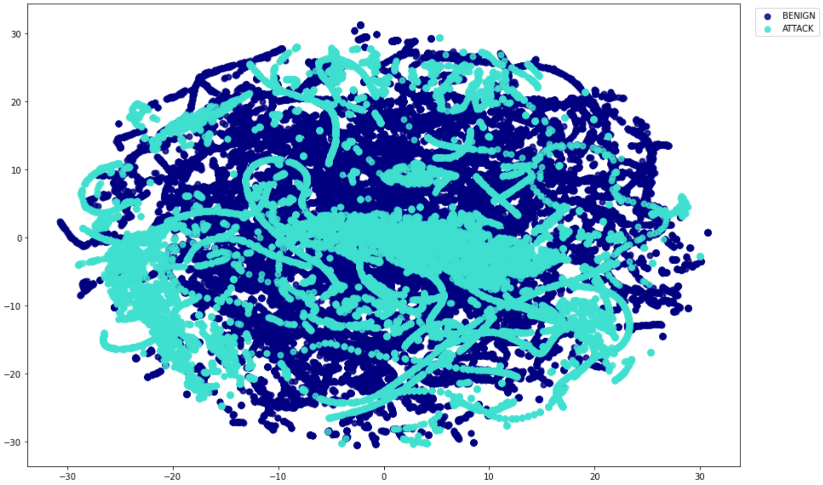

Figure 1. Visualizations of benign and attack classes

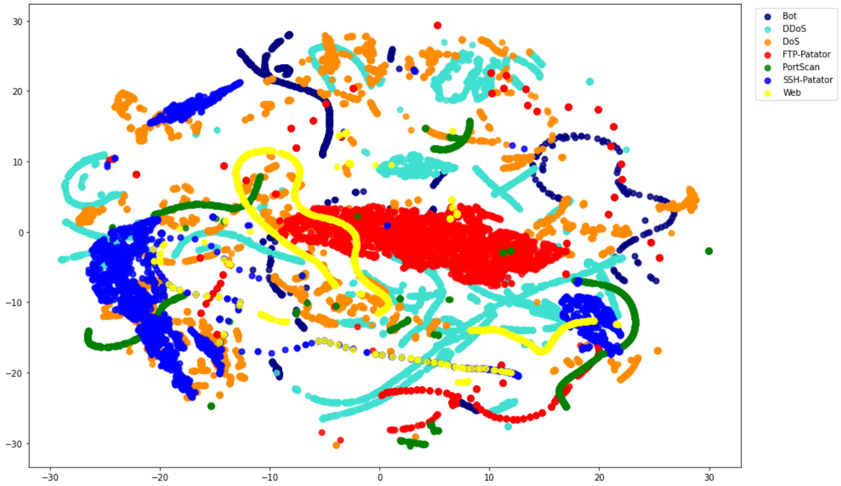

Figure 2. Visualizations of attack types

To explore high-dimensional intrusion data with a faithful representation, a training set was visualized with t-SNE [15]. While Figure 1 shows the benign and attack classes, Figure 2 details the attack types. As seen from the distributional characteristic of the data set, the classes are not linearly separated from each other.

Table 1
Distribution of training and test sets

Class	Training Set	Test Set
Benign	60,000	20,000
Botnet	1000	437
DoS	6000	2000
DDoS	6000	2000
FTP-Patator	5000	931
Port Scan	1500	456
SSH-Patator	2500	719
Web Attacks	1500	643
Total	83,500	27,186

In the data-pre-processing stage, the destination port column was dropped in order to avoid over-fitted training toward the socket information. In addition, any duplicate rows were dropped in order to avoid bias. Then, the remaining data was randomly split into a training set and a test set (as shown in Table 1). Lastly, the training set was fitted with a standard scaler; then, the scaler was used to transform both sets. In this way, the data was prepared for image encoding.

The generated GAF images for a sample of each class in the CIC-IDS 2017 data set are given in Figure 3. It can be seen that the characteristics of each class are transferred to two-dimensional images with different color-intensity attributes (such as lines and points). Since there were 77 features in the data, the generated image dimensions were 77 × 77.

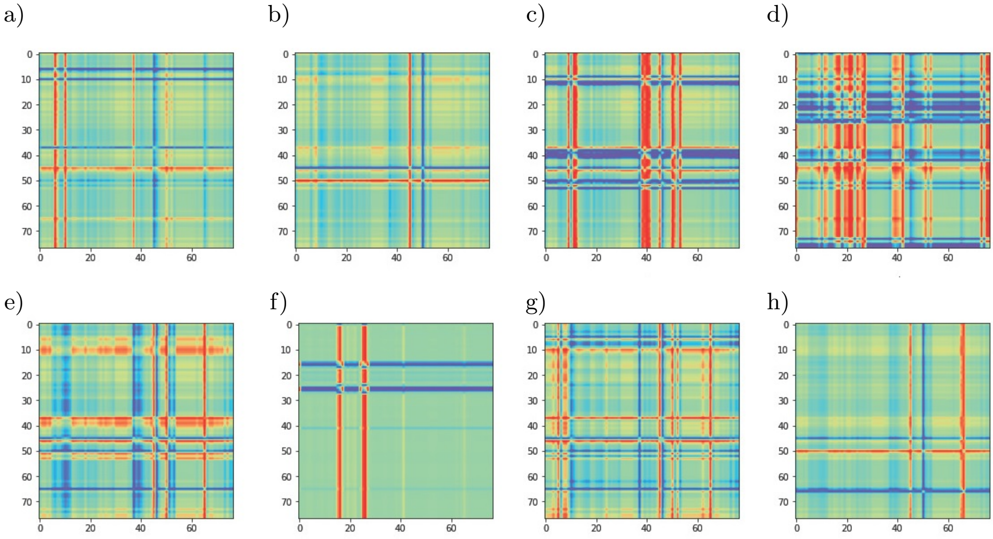

Figure 3. GAF images for benign (a), botnet (b), DDoS (c), DoS (d), FTP-patator (e), port scan (f), SSH-patator (g), and web attack classes (h)

The CNN architecture that was used in this study is given in Figure 4. The proposed architecture consists of three blocks, and each block is comprised of convolution, batch normalization, activation (ReLU), convolution, batch normalization, activation (ReLU), and pooling (max) layers, respectively. After the feature extraction was completed with these blocks, the classification was performed with four fully connected layers. A dropout is applied after the first fully connected layer.

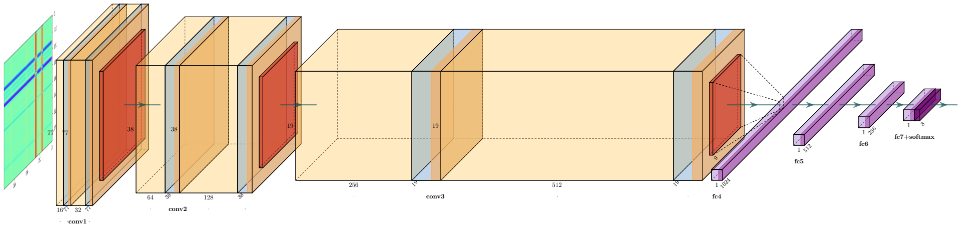

Figure 4. Proposed CNN architecture

5. Experimental results

This section elaborates on two experiments to evaluate the classification success of the proposed approach. The first experiment was designed to compare the results of the proposed approach to conventional machine-learning methods. The second experiment was organized to compare the results of the proposed approach to related studies. Both the binary and multi-class classification results were evaluated. The weighted average of the evaluation metrics was used in the comparisons.

Table 2

Comparison of binary classification results for proposed approach and conventional machine-learning methods

Class	Evaluation Metric	Proposed Approach	kNN	LR	GNB	SVM
Benign	Precision	99.32	99.00	96.60	98.95	96.85
	Recall	99.77	99.54	96.51	21.26	97.05
	F1-score	99.54	99.27	96.55	35.00	96.95
Attack	Precision	99.34	98.71	90.31	31.20	91.73
	Recall	98.11	97.19	90.54	99.37	91.22
	F1-score	98.72	97.95	90.42	47.49	91.47

In the first experiment, well-known conventional machine-learning classification methods such as k-nearest neighbor (kNN), logistic regression (LR), Gaussian naive Bayes (GNB), and support vector machine were adopted. All of the methods used the default parameters that were provided by the Scikit-learn module.

As can be seen in Tables 2 and 3, the proposed approach created a good balance between recall (which demonstrated its success in classifying the intrusions) and precision (which demonstrated the success in those cases that were classified as intrusions). This showed that the proposed approach is better than conventional machine-learning methods in many cases. Unlike conventional methods, the proposed approach makes it possible to model the positive or negative effects of the bilateral relationships between the features on the label.

In the second experiment, those studies that used the CIC-IDS 2017 data set (or a part thereof) were evaluated. Marir et al. developed a distributed approach using a combination of non-linear dimensionality reduction and multi-layer ensembles [18]. Chiba et al. created a model for the cloud environment, which was a combination of various machine-learning algorithms [5]. Lee et al. proposed a system that used event profiling and artificial neural networks [11]. Elmasry et al. used a pre-training phase before the deep-learning models to cope with any redundant and irrelevant features [6]. Tama et al. analyzed the use of a stacked ensemble architecture [28]. Lee and Park focused on solving the data imbalance by using a deep-learning method that generated virtual data [12]. Zhou et al. proposed a framework that was based on feature selection and ensemble learning [36]. Zhang et al. designed a new class imbalance-processing technology and integrated it with a convolutional neural network [35].

Huang and Lei developed a model that consisted of three modules: feature extraction, a generative adversarial network with an imbalanced data filter, and a deep neural network [9].

Table 3

Comparison of multi-class classification results for proposed approach and conventional machine-learning methods

Class	Evaluation Metric	Proposed Approach	kNN	LR	GNB	SVM
Benign	Precision	99.00	98.98	97.67	99.65	97.17
	Recall	99.50	99.62	98.65	59.71	98.35
	F1-score	99.24	99.30	98.16	74.68	97.76
Bot	Precision	99.77	96.67	0.00	14.88	0.00
	Recall	99.54	99.54	0.00	94.51	0.00
	F1-score	99.66	98.08	0.00	25.71	0.00
DDoS	Precision	99.85	99.60	99.39	84.87	97.59
	Recall	99.70	99.45	98.30	96.75	93.05
	F1-score	99.77	99.52	98.84	90.42	95.26
DoS	Precision	96.25	98.29	98.49	43.72	97.16
	Recall	97.65	97.90	87.95	93.35	88.85
	F1-score	96.95	98.10	92.92	59.55	92.82
FTP-Patator	Precision	100	99.78	97.14	86.89	88.73
	Recall	98.50	97.96	98.60	98.28	98.07
	F1-score	99.24	98.86	97.87	92.24	93.16
Port Scan	Precision	98.60	97.79	41.64	24.67	34.99
	Recall	77.19	77.63	76.97	36.40	58.55
	F1-score	86.59	86.55	54.04	29.41	43.81
SSH-Patator	Precision	98.72	96.81	99.13	69.17	99.85
	Recall	96.80	96.94	94.71	94.85	94.85
	F1-score	97.75	96.87	96.87	80.00	97.29
Web Attacks	Precision	92.86	96.46	80.19	19.57	83.50
	Recall	93.00	93.16	91.91	96.27	92.85
	F1-score	92.93	94.78	85.65	32.53	87.92

To summarize the previous works, various improvements and developments have been made to increase the detection rates in high-volume and class-imbalanced intrusion data. Unlike these works, the proposed approach does not require any method such as feature extraction, feature selection, class balancing, virtual data generation, or ensemble classifiers. As can be seen in Tables 4 and 5, only converting the intrusions to images produced success rates that were close to those that were achieved by the sophisticated approaches.

Table 4

Comparison of binary classification results for proposed approach and previous studies

Method	Accuracy	Precision	Recall	F1-score
Proposed Approach	99.33	99.33	99.33	99.33
[18]	∗	90.40	95.65	92.95
MLIDS [5]	99.93	99.95	∗	99.00
EP-FCNN [11]	99.5	∗	98.2	98.7
EP-CNN [11]	98.8	∗	98.5	97.1
EP-LSTM [11]	98.6	∗	97.8	96.7
DNN [6]	97.85	99.96	97.58	98.76
LSTM-RNN [6]	98.83	99.98	98.68	99.33
DBN [6]	99.91	99.99	99.92	99.95
[28]	99.98	∗	∗	99.53

∗ unspecified

Table 5

Comparison of multi-class classification results for proposed approach and previous studies

Method	Accuracy	Precision	Recall	F1-score
Proposed Approach	98.74	98.74	98.72	98.74
GAN-RF [12]	99.83	98.68	92.76	95.04
DNN [6]	97.01	88.08	88.04	88.06
LSTM-RNN [6]	98.10	92.44	92.41	92.43
DBN [6]	98.95	95.82	95.81	95.81
[36]	99.89	∗	∗	∗
SGM-CNN [35]	99.85	99.88	∗	99.86
IGAN-IDS [9]	99.79	∗	∗	99.79

∗ unspecified

6. Conclusion

In recent years, the rise in internet traffic and network communication has increased the importance of ensuring the security and privacy of data. In this context, many IDSs have been developed to monitor network traffic and alert network administrators when to counter cyber-attacks. Despite the significant advances in IDS technology, most of the solutions are still insufficiently robust and effective. There are several reasons for this situation, including the increase in the volume of data (both stored and passing through networks), the dynamic nature of network behavior, the presence of low-frequency attacks, the difficulty of gathering reliable training data, and the diversity in the protocols that are used in networks.

Moreover, recent years have also experienced the progress of artificial-intelligence techniques; specifically, deep-learning solutions eliminate the need for feature engineering and show great performance in pattern retrieval.

In this context, this paper presents a novel approach for intrusion detection by employing the conversion of network traffic into images and classifying them with convolutional neural networks. As compared to conventional machine-learning methods, it can be seen that the proposed model produced better results. As compared to the related studies, the obtained results were close to the sophisticated approaches. This demonstrates the positive effect on any results by simply changing the problem space rather than establishing complex mechanisms.

A future proposal would be to apply the proposed approach to different attack cases and to evaluate the resulting images in more detail. When the relationships between any attack images are revealed, it can be possible to detect new attacks. Another issue could be to address the need for the explainability of deep-learning models. Thanks to explainability, the weak spots of a model can be identified and reduced, leading to more reliable and accurate outcomes. The explainability and interpretability of such models will undoubtedly become a significant research area on IDSs in the future.

References

[1] Alzubaidi L., Zhang J., Humaidi A.J., Al-Dujaili A., Duan Y., Al-Shamma O., Santamaría J., Fadhel M.A., Al-Amidie M., Farhan L.: Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions, *Journal of Big Data*, vol. 8(1), pp. 1–74, 2021.

[2] Ambusaidi M.A., Tan Z., He X., Nanda P., Lu L.F., Jamdagni A.: Intrusion detection method based on nonlinear correlation measure, *International Journal of Internet Protocol Technology*, vol. 8(2–3), pp. 77–86, 2014.

[3] Axelsson S.: The base-rate fallacy and the difficulty of intrusion detection, *ACM Transactions on Information and System Security*, vol. 3(3), pp. 186–205, 2000.

[4] Buczak A.L., Guven E.: A survey of data mining and machine learning methods for cyber security intrusion detection, *IEEE Communications Surveys & Tutorials*, vol. 18(2), pp. 1153–1176, 2015.

[5] Chiba Z., Abghour N., Moussaid K., El omri A., Rida M.: Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms, *Computers & Security*, vol. 86, pp. 291–317, 2019.

[6] Elmasry W., Akbulut A., Zaim A.H.: Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic, *Computer Networks*, vol. 168, 2020.

[7] Goodfellow I., Bengio Y., Courville A.: *Deep learning*, MIT Press, 2016.

[8] HerjavecGroup: *Official Annual Cybercrime Report. Cybercriminal activity is one of the biggest challenges that humanity will face in the next two decades*, Steve Morgan, Editor-in-Chief Cybersecurity Ventures, 2019.

[9] Huang S., Lei K.: IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks, *Ad Hoc Networks*, vol. 105, 2020. doi: 10.1016/j.adhoc.2020.102177.

[10] Hussain F., Abbas S.G., Husnain M., Fayyaz U.U., Shahzad F., Shah G.A.: IoT DoS and DDoS Attack Detection using ResNet. In: *2020 IEEE 23rd International Multitopic Conference (INMIC)*, pp. 1–6, IEEE, 2020.

[11] Lee J., Kim J., Kim I., Han K.: Cyber threat detection based on artificial neural networks using event profiles, *IEEE Access*, vol. 7, pp. 165607–165626, 2019.

[12] Lee J., Park K.: GAN-based imbalanced data intrusion detection system, *Personal and Ubiquitous Computing*, vol. 25, p. 121–128, 2021. doi: 10.1007/s00779-019-01332-y.

[13] Liu L., Xu B., Zhang X., Wu X.: An intrusion detection method for internet of things based on suppressed fuzzy clustering, *EURASIP Journal on Wireless Communications and Networking*, vol. 2018(1), pp. 1–7, 2018.

[14] Lyngdoh J., Hussain M.I., Majaw S., Kalita H.K.: An intrusion detection method using artificial immune system approach. In: *International Conference on Advanced Informatics for Computing Research*, pp. 379–387, Springer, 2018.

[15] Maaten van der L., Hinton G.: Visualizing Data using t-SNE, *Journal of Machine Learning Research*, vol. 9(11), pp. 2579–2605, 2008.

[16] Manimurugan S., Majdi A., Mohmmed M., Narmatha C., Varatharajan R.: Intrusion detection in networks using crow search optimization algorithm with adaptive neuro-fuzzy inference system, *Microprocessors and Microsystems*, vol. 79, 2020.

[17] Mao J., Wang H., Spencer Jr B.F.: Toward data anomaly detection for automated structural health monitoring: Exploiting generative adversarial nets and autoencoders, *Structural Health Monitoring*, vol. 20(4), pp. 1609–1626, 2021.

[18] Marir N., Wang H., Feng G., Li B., Jia M.: Distributed abnormal behavior detection approach based on deep belief network and ensemble SVM using spark, *IEEE Access*, vol. 6, pp. 59657–59671, 2018.

[19] Milenkoski A., Vieira M., Kounev S., Avritzer A., Payne B.D.: Evaluating computer intrusion detection systems: A survey of common practices, *ACM Computing Surveys (CSUR)*, vol. 48(1), pp. 1–41, 2015.

[20] Moustafa N., Hu J., Slay J.: A holistic review of network anomaly detection systems: A comprehensive survey, *Journal of Network and Computer Applications*, vol. 128, pp. 33–55, 2019.

[21] Nisioti A., Mylonas A., Yoo P.D., Katos V.: From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods, *IEEE Communications Surveys & Tutorials*, vol. 20(4), pp. 3369–3388, 2018.

[22] Rath P.S., Barpanda N.K., Singh R., Panda S.: A prototype Multiview approach for reduction of false alarm rate in network intrusion detection system, *International Journal of Computer Networks and Communications Security*, vol. 5(3), pp. 49–59, 2017.

[23] Rawashdeh A., Alkasassbeh M., Al-Hawawreh M.: An anomaly-based approach for DDoS attack detection in cloud environment, *International Journal of Computer Applications in Technology*, vol. 57(4), pp. 312–324, 2018.

[24] Sharafaldin I., Lashkari A.H., Ghorbani A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization, *ICISSp*, vol. 1, pp. 108–116, 2018.

[25] Shyu M.L., Chen S.C., Sarinnapakorn K., Chang L.: A Novel Anomaly Detection Scheme Based on Principal Component Classifier. In: *Proceedings of Conference: IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM'03)*, pp. 353–365, 2003.

[26] Stojanović B., Hofer-Schmitz K., Kleb U.: APT datasets and attack modeling for automated detection methods: A review, *Computers & Security*, vol. 92, 2020.

[27] Syarif I., Prugel-Bennett A., Wills G.: Unsupervised clustering approach for network anomaly detection. In: *International Conference on Networked Digital Technologies*, pp. 135–145, Springer, 2012.

[28] Tama B.A., Nkenyereye L., Islam S.R., Kwak K.S.: An enhanced anomaly detection in web traffic using a stack of classifier ensemble, *IEEE Access*, vol. 8, pp. 24120–24134, 2020.

[29] Tan Z., Jamdagni A., He X., Nanda P., Liu R.P.: A system for denial-of-service attack detection based on multivariate correlation analysis, *IEEE Transactions on Parallel and Distributed Systems*, vol. 25(2), pp. 447–456, 2013.

[30] Voulodimos A., Doulamis N., Doulamis A., Protopapadakis E.: Deep learning for computer vision: A brief review, *Computational Intelligence and Neuroscience*, vol. 2018, 2018.

[31] Wang Z., Oates T.: Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In: *Workshops At the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[32] Xia S., Pan Z., Chen Z., Bai W., Yang H.: Malware Classification with Markov Transition Field Encoded Images. In: *2018 Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*, pp. 1–5, IEEE, 2018.

[33] Ye N., Chen Q.: An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems, *Quality and Reliability Engineering International*, vol. 17(2), pp. 105–112, 2001.

[34] Yilmaz M., Catak F.O., Gul E.: Sensor based cyber attack detections in critical infrastructures using deep learning algorithms, *Computer Science*, vol. 20(2), pp. 213–243, 2019. doi: 10.7494/csci.2019.20.2.3191.

[35] Zhang H., Huang L., Wu C.Q., Li Z.: An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset, *Computer Networks*, vol. 177, 2020. doi: 10.1016/j.comnet.2020.107315.

[36] Zhou Y., Cheng G., Jiang S., Dai M.: Building an efficient intrusion detection system based on feature selection and ensemble classifier, *Computer Networks*, vol. 174, 2020. doi: 10.1016/j.comnet.2020.107247.

Affiliations

Duygu Sinanc Terzi
Amasya University, Department of Computer Engineering, Amasya, Turkey,
duygu.terzi@amasya.edu.tr, ORCID ID: https://orcid.org/0000-0002-3332-9414

Received: 23.07.2021
Revised: 07.07.2022
Accepted: 07.07.2022