Training method for artificial intelligence system for robust and resource-efficient data block identification

Maksym Boiko[1,2] and Viacheslav Moskalenko[1]

[1] Sumy State University, 116, Kharkivska St., Sumy, 40007, Ukraine
[2] The National Anti-corruption Bureau of Ukraine, 3, Denysa Monastyrskogo St., Kyiv, 03035, Ukraine

Abstract

Identifying binary data blocks is a critical problem in recovering fragmented files during the digital forensics process. A promising area for solving this problem is using neural networks, which have shown high efficiency in classifying data blocks. An important point is that the amount of data usually analyzed during a typical case can be measured in hundreds of gigabytes. Therefore, in addition to improving the accuracy of data identification, the task of reducing the cost of resources involved comes into focus. In addition, real datasets may have certain features compared to test datasets. The goal is to develop a parameter-efficient tuning method for artificial intelligence systems. This method should increase the accuracy of identifying binary data blocks while reducing resource costs. The method proposed in this paper is to add blocks of parallel adapters to pre-trained frozen convolutional neural networks. The adapters mentioned above are trained on the same dataset and then tuned using the marginal entropy minimization with one test point method. It has been experimentally confirmed that the proposed method increases the robustness of the model and the efficiency of identifying data blocks while saving resource costs.

Keywords

digital forensics, data block identification, neural network, parameter efficient transfer learning, parallel adapters, marginal entropy.

1. Introduction

One of the stages of digital forensics is data recovery. For this purpose, a relatively wide range of specialized software tools are used, automatically searching for deleted information using combinations of different methods [1]. A typical situation is to perform signature analysis in the unallocated disk space without any other information from the filesystem (signature-based methods to identify the first and last blocks of files). As a rule, this method is quite suitable for recovering non-fragmented or bi-fragmented files. However, it is common for files to consist of three or more non-sequential blocks of data, which may also follow in the incorrect order [2]. Also, the first sectors can be overwritten by other data. As a result, file fragments without explicit signatures are not thoroughly analyzed in the later stages. For the above reasons, researchers identify data blocks at the initial phase of file recovery.

On the other hand, when conducting computer-assisted research, the researcher's ability to process large amounts of information in a short time, reduce the amount of data under study [3], isolate important data, establish the relationship between different artifacts, and draw correct conclusions from it, comes to the fore. Due to the complexity of human perception of binary data and its large volumes, recent tendencies are increasingly using artificial intelligence models and methods with automatic feature selection [4, 5]. This greatly facilitates the work of digital researchers since, otherwise, the number of correctly identified data blocks depends on the quality of the manual selection of classifier features.

In general, the stage of identifying data blocks is critical. The amount of information analyzed in a typical case can reach hundreds of gigabytes. If there are errors in identification, a large amount of information that may be significant will be ignored. In addition, most artificial intelligence models require improvement and significant resources for their training. It should also be noted that sensitive data is submitted for examination. Therefore, leaking or distributing this information is unacceptable. Taken together, all of this leads to the fact that the tasks of increasing the accuracy of

ICST-2024: Information Control Systems & Technologies, September 23-25, 2023, Odesa, Ukraine.
mboiko25@gmail.com (M. Boiko); v.moskalenko@cs.sumdu.edu.ua (V. Moskalenko)
0000-0003-0950-8399 (M. Boiko); 0000-0001-6275-9803 (V. Moskalenko)

data block identification without a significant increase in computational complexity take a higher priority.

The object of the research is the process of parameter-efficient tuning for artificial intelligence systems, which identify binary blocks of data.

The subjects of the research are parameter-efficient tuning methods of an artificial intelligence system that improve the performance of binary data block identification.

The goal is to develop a parameter-efficient tuning method for artificial intelligence systems, which will increase the accuracy of identification of binary data blocks while reducing resource costs.

2. Review of the literature

2.1. File fragment identification

Artificial intelligence's appearance and widespread use have transformed many areas of life. Currently, artificial intelligence models and methods capable of identifying binary data blocks are being actively implemented in the digital forensic community [6]. For this purpose, methods based on the internal structure of files [7, 8, 9, 10, 11], based on the contents of files [12], for file restoring [13], and calculating the entropy of data blocks [14] are used.

Techniques using n-grams have become widely used [15]. The researchers try to avoid the above disadvantages by using n-grams. However, it should be noted that the search for the specific features of files of different types has been replaced by the need to select a range of statistical measures that best satisfy the defined goals.

The paper [16] proposes an approach to determining file types by 512-byte fragments, where the basic classifier is support vector machines (SVM). The authors break down the data blocks into unigrams and bigrams and then actually calculate ten statistical values for each fragment: the mean of unigrams and bigrams, their standard deviation, Hamming weights, and so on. As a result, the average accuracy was 67.78% when analyzing a dataset of 14 different file types. PPT, PDF, and DOC files had the lowest percentage of truly positive cases.

Several experiments to identify 31 file types by their fragments using the n-gram technique and the support vector method are described in [17]. The authors achieved 74.9 % and 87.3 % accuracy rates in classifying data blocks with sizes of 512 bytes and 4096 bytes, respectively. It should be noted that the proposed method achieves high performance in identifying text file types such as TXT, XML, CSV, and LOG. Instead, a large number of errors are observed among compound file types, such as PPT, AVI, PDF, DOCX, GZ, PPT, and ZIP.

In [18], various techniques for identifying file types by 512-byte fragments were also investigated. The application of two models was compared: 1) based on a feed-forward neural network (FNN) using unigrams and bigrams; 2) based on a three-layer one-dimensional convolutional neural network (1D-CNN). The first approach, used to identify 18 file types, provided an F1-score of 79.93% to 81.38% against 61.55% in the second case.

In contrast to previous works, a number of papers propose another promising approach, the key feature of which is the automatic feature selection and the use of convolutional neural networks.

Thus, in the work [19], 4096-byte data blocks are transformed into images with dimensions of 64x64 pixels in grayscale. Subsequently, an artificial intelligence model was used to identify these objects, which included several two-dimensional convolutional neural networks (2D-CNN). It achieved an accuracy of 70.9% in the case of the analysis of 16 file types.

The paper [20] investigates the use of recurrent (RNN), convolutional (CNN), and feed-forward neural networks (FNN) as classifiers. 512-byte data blocks were transformed into 8192 features by representing them as bits, each of which was assigned two features. Thus, when analyzing four file types, the highest accuracy was 98.04% with recurrent neural networks and was not less than 73% in other cases. Although the author conducts rather limited experiments, the results demonstrate the possibility of applying the presented models to classify file types by their fragments.

A series of experiments on the classification of 512-byte and 4096-byte data fragments was carried out and described in [21]. The proposed model used one-dimensional convolutional neural networks in different cases, where all byte values from a data block were input. Depending on the number of classes to be evaluated (75, 11, 25, 5, and 2), the identification accuracy of identifying 512-byte fragments was 65.6%, 78.9%, 87.9%, 90.2%, and at least 99.0%, respectively. However, many false positives were noted when analyzing fragments belonging to a number of compound file types, such as MOV, 7Z, EXE, DJVU, PDF, PPT, PPTX, and DOC. As in other cases, additional factors affecting accuracy could be similar internal structure of files and the existence of embeddings, such as different kinds of media embeddings in Word, Excel, and PowerPoint files.

2.2. Parameter efficient tuning

In order to improve the developed artificial intelligence models, researchers use various methods. For example, it can be a simple increase in data, varying the neural network's architecture, iterating over its hyperparameters, and using more advanced activation functions. Each of the approaches has its pros and cons. However, the main disadvantage of the majority of methods is their resource consumption. For instance, full fine-tuning needs to update the entire model when a new task appears. Therefore, applying various parameter-efficient tuning methods for pre-trained models can reduce the resources required for work, increase productivity and learning speed, etc. In particular, parameter-efficient tuning methods have performed quite well with NLP models [22, 23, 24].

The paper [22] presents a strategy for tuning a large language model. The main advantage is the need to add a small number of parameters for the task. This is made possible by using an adapter module. The authors achieve results that are close to full fine-tuning cases. At the same time, only 3.6% of parameters were involved.

A unified parameter-efficient tuning framework for multitasks is presented in [24]. This framework uses prefix-tuning and adapter-tuning modules to solve different NLP and Vision-and-Language tasks. In general, the proposed approach achieves better results using a much smaller number of parameters.

The paper [23] discusses various variants of parameter-efficient tuning methods and proposes a unified framework that establishes connections between methods. In addition, the authors conducted experiments using parallel and sequential adapters. In all cases, the best results are obtained through the use of parallel adapters.

In contrast to previous works, where parameter-efficient tuning methods were applied to NLP tasks, parameter-efficient tuning modules for convolutional networks are proposed in [25]. In particular, the authors developed the adapter architecture as a bottleneck structure and considered four adapting schemes to ResNet50. This method can achieve results comparable to full fine-tuning with much fewer parameters. However, the method needs to show better results on tasks with significant domain shifts and depends on the quality of the features of the pre-trained model.

2.3. Robustness

Deep neural networks can achieve acceptable results on different test samples. However, training methods will not always be robust to perturbations in the input, changes in the domain, etc. Therefore, before applying these methods to actual cases, they usually need to improve their robustness.

The paper [26] conducts a study of the robustness of vision transformers in relation to adversarial examples, common corruptions, and distribution shifts. The authors also present the method that consistently achieves outstanding performance on ImageNet and six robustness benchmarks.

Another technique to improve the robustness is proposed in [27]. For training, the authors use pre-prepared images. For these purposes, augmentation operations are performed with each input image. Then, the resulting images are combined by mixing. As a result, the authors achieve fewer errors with CIFAR-10/100-C, ImageNet-C, CIFAR-10/100-P, and ImageNet-P than the other presented techniques.

The problems of resistance to perturbation are also solved in [28]. The authors propose a method of marginal entropy minimization with one test point. This method allows to improve the performance of ResNet and vision transformer models. It also leads to improved performance on ImageNet-A, ImageNet-C, and ImageNet-R (by 1-8%).

3. Neural model and training method

To achieve the stated goal, it is evident that the task should be divided into separate blocks. Since the goal is to develop a parameter-efficient tuning method of an artificial intelligence system, which will increase the accuracy of identifying binary data blocks while reducing resource costs, these stages should be (Fig. 1):

1. Building or selecting a model that can identify data blocks at an acceptable level.
2. Building, selecting, or adapting a parameter-efficient tuning method that can achieve acceptable results while reducing the resources involved.
3. Application of the method to improve the efficiency of the model and its robustness to disturbances.

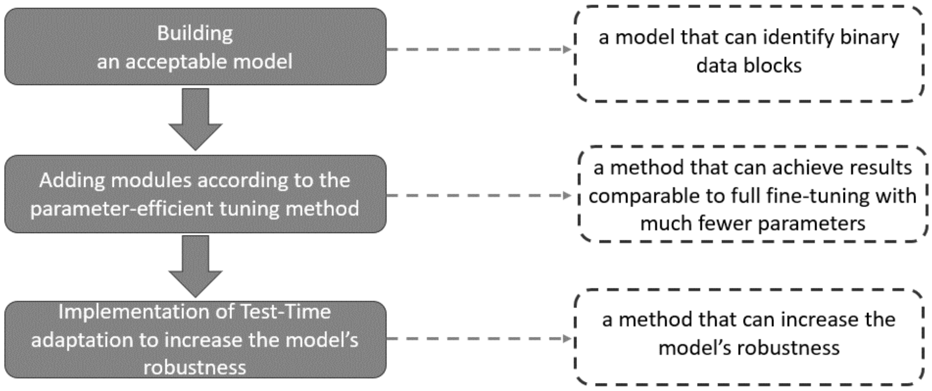


Figure 1: Proposed approach to solving the task

3.1. Neural model

When choosing the model to be proposed as a basis, we focused on the results demonstrated in various studies on the identification of binary data blocks [6]. In general, all studies solve the problem of finding a function *f* that classifies *n*-bytes data blocks *B* by file type labels *T* [21]:

$$f : B \rightarrow T, \tag{1}$$

where $B \in \mathbb{Z}_{255}^{n}$, $T \in \{PDF, DOCX, \dots, PNG\}$, $\mathbb{Z}_{255} = [0, \dots, 255]$, and *n* is typically 512 or 4096 bytes.

At the same time, attention was paid to the need to reduce the influence of the human factor in the manual feature selection. As a result, the model chosen for identifying binary data blocks was based on convolutional neural networks with automatic feature extraction, such as those described in [19, 20, 21, 29]. In addition, as the analysis of the works has shown, such models are a perspective direction in data identification.

Such models are usually developed using a set of several convolutional and max-pooling layers with appropriately configured hyperparameters. Globally, only the initial layers of the model differ depending on the type of input data representation chosen. Schematically, typical models are shown in Fig. 2 [21, 19].

3.2. Parameter-efficient tuning method

At the next stage, the model should be improved with a reduction in the resources involved. Since the adapters are to be used with convolutional neural networks, the method described in [25] is suitable for this purpose. The concept of the parameter-efficient approach is to introduce adapters with a small number of parameters into a pre-trained model with frozen weights. Only these added parameters are then trained. The tuning method steps are summarized in Table 1.

Table 1
Tuning method

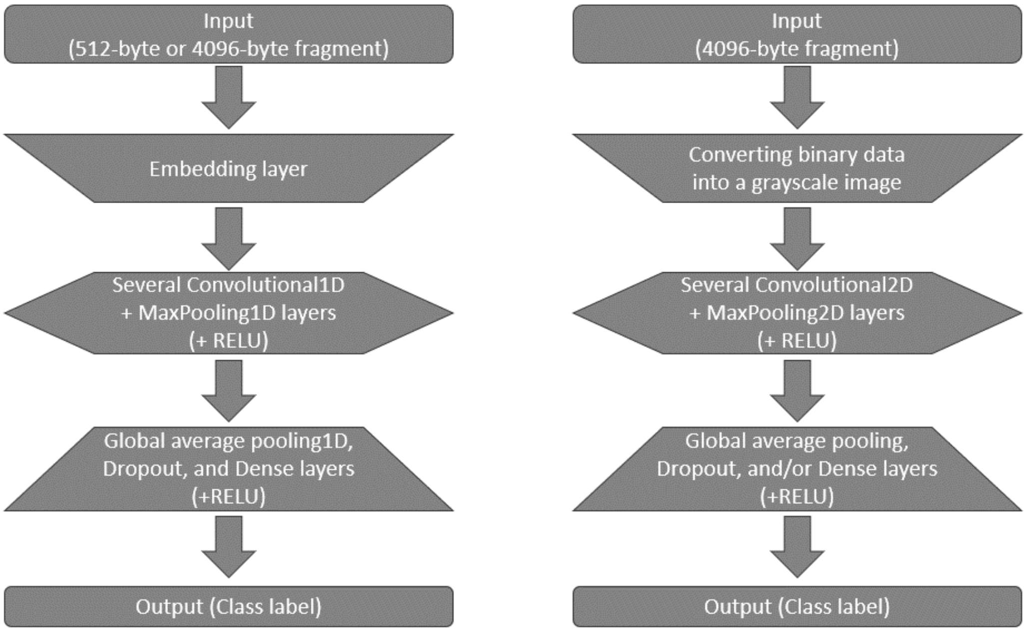
#	Steps
1	Fix model weights after training
2	Add parallel adapters to the model
3	Train parallel adapters on the same training dataset

In general, adding parallel adapters is the most convenient and universal approach [30]. The results obtained in [23] give additional reasons to suggest that parallel tuners are the optimal option for solving the problem. A schematic illustration of this approach and adapter architecture are shown in Fig. 3, where C_{in} is the input channel dimension, γ is a hyperparameter to regulate channel compression. The adapter starts with the depth-wise convolutional layer. A non-linear activation function is then applied. The last adapter layer is the point-wise convolutional layer [25]. The dimensions of the convolutional layers are the same as the frozen blocks.

In this case, the output x' of the final model's block should be calculated as follows:

$$x' = OP(x) + Adapter(x), \tag{2}$$

where x is the input, *OP* and *Adapter* perform operations over input tokens with frozen parameters of the original model's block and with adapter's parameters, respectively.



a b

Figure 2: Proposed model architecture with the input as a set of bytes (a) and as a grayscale image that is converted from all bytes of the data block (b)

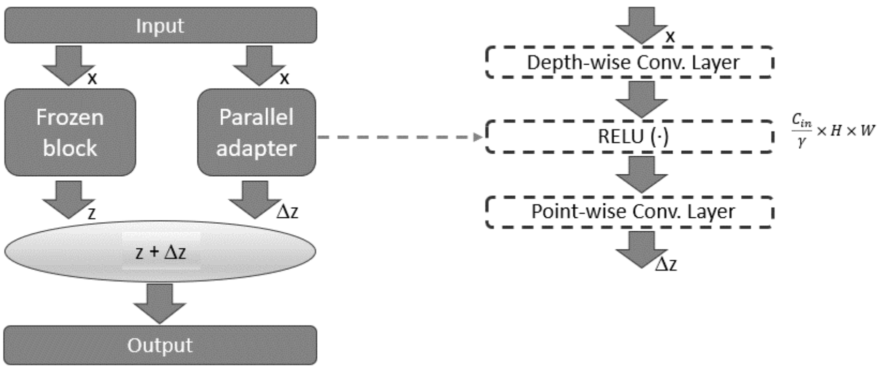


Figure 3: Proposed parameter-efficient tuning approach and adapter architecture

3.3. Method of increasing the model robustness

After training the model, adding a block of adapters to it, and then training them on the test dataset, the task of improving the robustness of the obtained artificial intelligence system to disturbances remains unsolved. To achieve an increase in the robustness of the model, we propose at the final stage to apply the method of marginal entropy minimization with one test point (MEMO), described in [28]. It is important to note that the whole network is not tuned, but only the adapter parameters. The key stages and the schematic approach of the method are shown in Fig.4 and Table 2.

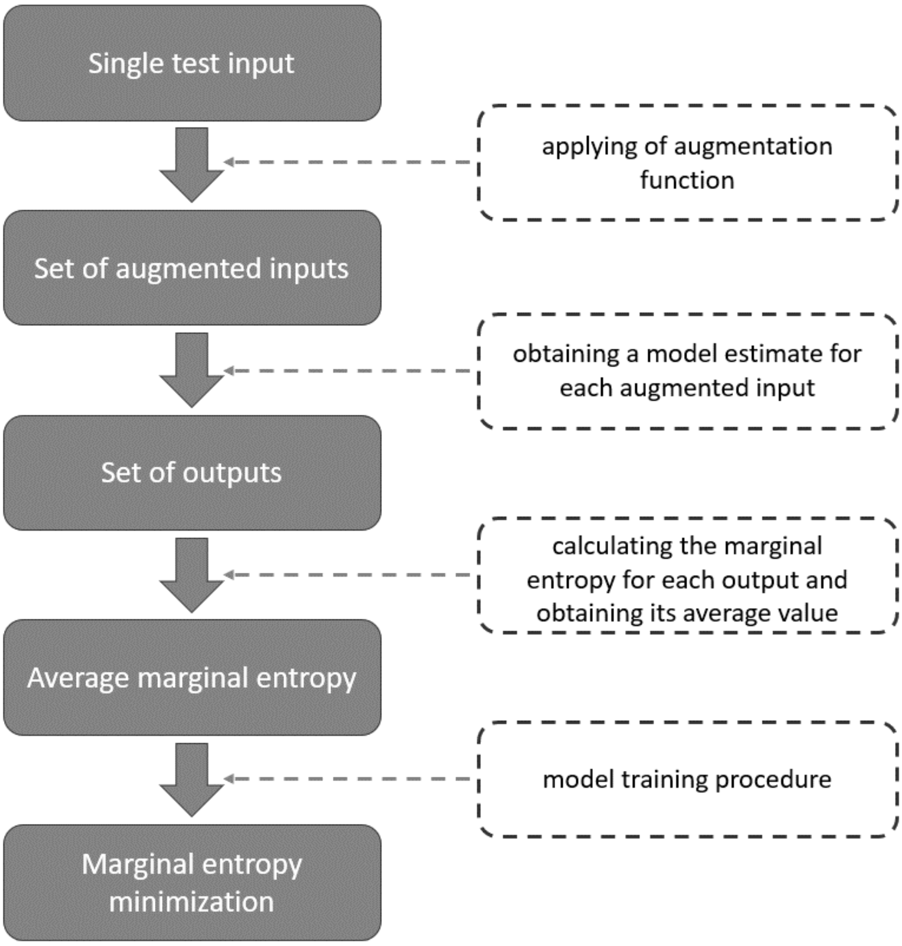


Figure 4: Proposed method scheme

Table 2
The key stages of the method

#	Stage
1	Applying augmentation function on single test input
2	Getting a model prediction from augmented inputs
3	Calculation of marginal entropy for each result obtained on augmented input
4	Calculation of the marginal output distribution averaged over augmentations
5	Training of the model with marginal entropy minimization

This method does not require a particular training procedure and does not apply any restrictions to deep neural network models. An additional advantage of the marginal entropy minimization with one test point method is that it can be applied to pre-trained models. This method does not require access to or modification of the model training process. In fact, this method adapts the model parameters, generating augmented data for each single test input and analyzing the results obtained on them.

To do this, the marginal entropy is computed for each input sample. The following formula will calculate the loss function:

$$l\,(\theta, x) \approx H\big(\bar{p}_\theta(\cdot\,|x)\big) = \sum_{y \epsilon Y} \bar{p}_\theta(y|x)\, log\bar{p}_\theta(y|x), \tag{3}$$

where $\bar{p}_\theta(y|x)$ is the model's marginal output distribution determined as follows:

$$\bar{p}_\theta(y|x) = \frac{1}{B}\sum_{i=1}^{B} \bar{p}_\theta(y|\widetilde{x_i}), \tag{4}$$

where *B* is the number of augmentations, $\widetilde{x_i}$is an *i*-th sample from the augmented data batch.

4. Results

The proposed artificial intelligence system is quite flexible and universal. During the development of this system, the convolutional neural network is selected as the base network in the first stage. This is mainly because this type of network is currently the most promising in terms of identifying data blocks [6]. It is necessary to note that the neural network can have an arbitrary architecture and hyperparameters since the subsequent parts of the system are independent of it.

At the next stage, we chose parallel adapters due to their better performance [23], lower resource consumption, and the possibility of applying them to convolutional neural networks [25]. It is worth noting that in all cases, after adding a block of parallel adapters, the increase in the number of parameters was only about 3.5%.

Otherwise, researchers are generally free to choose the architecture and parameters of these adapters necessary to achieve the projects' goals.

Finally, to achieve higher robustness of the artificial intelligence system, it is proposed to use the method of marginal entropy minimization with one test point [28]. This method is one of many that are possible. However, it is quite versatile and suitable for deep neural networks. This method has also shown itself well in experimental cases.

The study compared the performance of pre-trained models and the developed artificial intelligence system. For this purpose, the results obtained during the experiments in [25] and [28] were applied to the results obtained in [21] and [19].

In three of the four cases of their use (FGVC, VTAB-1k Natural, and VTAB-1k Specialized benchmarks [25]), the accuracy rates were in the range of -1.92% to 0.57% relative to the accuracy rates for the full fine-tuning method. The result for one of the datasets was 15% worse. This case is not considered in this study since such a result is obviously unacceptable. In such circumstances, the parallel adapter block would not be used, at least in that configuration.

However, the method of marginal entropy minimization with one test point increases accuracy from 1% to 8% in all experimental tests. As for the performance of the entire proposed artificial intelligence system, this increase, in most cases, contributes to the overall improvement of accuracy values. In Table 3, the "Pre-trained model" column shows the accuracy of selected pre-trained models from[25], [21] and [19].

The column "Model with pre-trained adapters" shows the results obtained in [25], as well as the estimated accuracy values for the cases of classifying data blocks (75 classes [21] and 16 classes [19]) when parallel adapters are added to the mentioned models.

The last columns show the estimated accuracy rates in the case of using the method of marginal entropy minimization with one test point and the proposed artificial intelligence system in all scenarios.

Table 3
Comparison of models

Dataset	Pre-trained model	Model with pre-trained adapters	Model with marginal entropy minimization	Model with adapters and marginal entropy minimization
FGVC [25]	83.46	83.77	84.29 – 90.14	84.61 – 90.47
VTAB-1k Natural [25]	72.19	72.60	72.91 – 77.97	73.33 – 78.41
VTAB-1k Specialized [25]	85.86	84.21	86.72 – 92.73	85.05 – 90.95
Fifty-75 (512-byte blocks) [21]	65.6	64.34 – 65.97	66.26 – 70.85	64.98 – 71.25
Grayscale image (4096-byte blocks) [19]	70.9	69.54 – 71.30	71.61 – 76.57	70.23 – 77.01

The results in Table 3 show that adding parallel adapters does not significantly affect the accuracy rates.

However, in combination with the method of marginal entropy minimization with one test point [28], it is possible to improve the performance on the test set despite the training data's limited representativeness.

5. Discussion

The artificial intelligence system proposed in this paper consists of separate blocks. That is, in general, the model, the method of its tuning, and the method of increasing robustness can be chosen depending on the actual needs. The proposed method allows to improve accuracy on test data from different datasets, which indicates that the approach is quite universal and can be used in other tasks. For some datasets, the result is worse than for others. For example, the reason may be that the test sample overlaps more with the training distribution and contains more minor novelty elements. Thus, in [25], significantly worse results were obtained on the VTAB-1k Structured dataset.

By contrast to the standard method of marginal entropy minimization with one test point, the proposed artificial intelligence system does not tune the entire network but only a small number of parameters (about 3.5%). A considerable speedup of test-time adaptation is expected. In addition, test-time adaptation is not used for all samples but only for those where the marginal entropy is less than a certain threshold. This allows tuning to be applied only to a small amount of data. Although the parallel adapter reduces the network speed by up to 96%, the overall accuracy increased by 6-9%.

This approach may be considered as a kind of controlled degradation mechanism when computational costs increase slightly for complex samples. At the same time, other properties of resilient systems are observed because tuning on the new data allows the system to improve and adapt to the novelty [31].

6. Conclusions

This paper proposes a parameter-efficient tuning method for an artificial intelligence system for the first time that will be able to increase the accuracy of binary data block identification while reducing resource costs. The proposed approach is based on pre-trained convolutional neural networks that identify binary data blocks. After that, the model selected as the base one is tuned by adding a block of parallel adapters. Only the adapters mentioned above are trained on the same test dataset during this stage. Finally, in order to increase the robustness of the obtained model, the method of marginal entropy minimization with one test point is used.

De facto, the novelty consists of the combined use of parallel adapters to reduce the resources involved and the method of marginal entropy minimization with one test point that improves the robustness of the resulting artificial intelligence system. Implementing the above combination requires fewer resources than full fine-tuning methods and improves accuracy in the task of identifying binary data blocks.

Limitations. This paper focuses exclusively on convolutional neural networks as the base model for the proposed method. It is also restricted to comparisons of a limited number of existing approaches.

Future research should be focused on applying the developed methodology to other forensic analysis tasks.

Contribution of authors: conceptualization of the problem, supervision and editing of work – V. V. Moskalenko; original draft preparation, analysis of the results, and visualization – M. V. Boiko. All authors have read and agreed with the published version of the manuscript.

Acknowledgements

The research was concluded in the Intellectual Systems Laboratory of Computer Science Department at Sumy State University with the financial support of the Ministry of Education and Science of Ukraine in the framework of state budget scientific and research work of DR No. 0122U000782 “Information technology for providing resilience of artificial intelligence systems to protect cyber-physical systems”.

References

[1] N. U. A. Ali, W. Iqbal, and N. Shafqat, “Analysis of windows OS’s fragmented file carving techniques: A systematic literature review,” in: Advances in Intelligent Systems and Computing, 2019. doi: 10.1007/978-3-030-14070-0_10.

[2] V. van der Meer, H. Jonker, and J. van den Bos, "A contemporary investigation of NTFS file fragmentation," Forensic Science International: Digital Investigation, vol. 38, 2021, doi: 10.1016/j.fsidi.2021.301125.

[3] X. Du, Q. Le, and M. Scanlon, "Automated Artefact Relevancy Determination from Artefact Metadata and Associated Timeline Events," in: International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2020, 2020. doi: 10.1109/CyberSecurity49315.2020.9138874.

[4] D. O. Ukwen and M. Karabatak, "Review of NLP-based Systems in Digital Forensics and Cybersecurity," in: 9th International Symposium on Digital Forensics and Security, ISDFS 2021, 2021. doi: 10.1109/ISDFS52919.2021.9486354.

[5] Dunsin D., Ghanem M.C., Ouazzane K., and Vassilev V., "A Comprehensive Analysis of the Role of Artificial Intelligence and Machine Learning in Modern Digital Forensics and Incident Response," Forensic Science International: Digital Investigation, 2024. doi: 10.1016/j.fsidi.2023.301675.

[6] M. Boiko, V. Moskalenko, and O. Shovkoplias, "Advanced file carving: ontology, models and methods," Radioelectronic and Computer Systems, 3 107 (2023) 204–216. doi: 10.32620/reks.2023.3.16.

[7] M. Xu et al., "A novel file carving algorithm for EVTX logs," in: Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, 2018. doi: 10.1007/978-3-319-73697-6_7.

[8] L. Zhang, S. Hao, and Q. Zhang, "Recovering SQLite data from fragmented flash pages," Annales des Telecommunications/Annals of Telecommunications, 74 7–8 (2019). doi: 10.1007/s12243-019-00707-9.

[9] J. N. Hilgert, M. Lambertz, M. Rybalka, and R. Schell, "Syntactical Carving of PNGs and Automated Generation of Reproducible Datasets," Digit Investig, 29 (2019). doi: 10.1016/j.diin.2019.04.014.

[10] R. R. Ali and K. M. Mohamad, "RX_myKarve carving framework for reassembling complex fragmentations of JPEG images," Journal of King Saud University - Computer and Information Sciences 33 1 (2021) 21–32. doi: 10.1016/j.jksuci.2018.12.007.

[11] Z. A. Al-Sharif, A. Y. Al-Khalee, M. I. Al-Saleh, and M. Al-Ayyoub, "CARVING AND CLUSTERING FILES IN RAM FOR MEMORY FORENSICS," Far East Journal of Electronics and Communications 18 5 (2018). doi: 10.17654/ec018050695.

[12] A. Ravi, T. R. Kumar, and A. R. Mathew, "A method for carving fragmented document and image files," in: 2016 International Conference on Advances in Human Machine Interaction, HMI 2016, 2016. doi: 10.1109/HMI.2016.7449170.

[13] K. Alghafli and T. Martin, "Identification and recovery of video fragments for forensics file carving," in: 2016 11th International Conference for Internet Technology and Secured Transactions, ICITST 2016, 2017. doi: 10.1109/ICITST.2016.7856710.

[14] H.-W. Lee and H. Lee, "Block based Smart Carving System for Forgery Analysis and Fragmented File Identification," Journal of Internet Computing and Services 21 3 (2020). doi: 10.7472/jksii.2020.21.3.93.

[15] J. Sester, D. Hayes, M. Scanlon, and N. A. Le-Khac, "A comparative study of support vector machine and neural networks for file type identification using n-gram analysis," Forensic Science International: Digital Investigation 36 (2021). doi: 10.1016/j.fsidi.2021.301121.

[16] M. Bhatt et al., "Hierarchy-Based File Fragment Classification," Mach Learn Knowl Extr 2 3 (2020) 216–232. doi: 10.3390/make2030012.

[17] M. Erfan and S. Jalili, "File Fragment Type Classification by Bag-Of-Visual-Words," ISeCure, 13 2 (2021). doi: 10.22042/isecure.2021.243876.570.

[18] K. Vulinovic, L. Ivkovic, J. Petrovic, K. Skracic, and P. Pale, "Neural networks for file fragment classification," in: 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2019 - Proceedings, 2019. doi: 10.23919/MIPRO.2019.8756878.

[19] Q. Chen et al., "File fragment classification using grayscale image conversion and deep learning in digital forensics," in: Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018, 2018. doi: 10.1109/SPW.2018.00029.

[20] L. Hiester, "File Fragment Classification Using Neural Networks with Lossless Representations Networks with Lossless Representations," Undergraduate Honors Theses, 2018. URL: https://dc.etsu.edu/honors/454/.
[21] G. Mittal, P. Korus, and N. Memon, "FiFTy: Large-Scale File Fragment Type Identification Using Convolutional Neural Networks," in: IEEE Transactions on Information Forensics and Security 16, 2021, pp. 28–41. doi: 10.1109/TIFS.2020.3004266.
[22] N. Houlsby et al., "Parameter-efficient transfer learning for NLP," in: 36th International Conference on Machine Learning, ICML 2019, 2019. URL: https://proceedings.mlr.press/v97/houlsby19a.html.
[23] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, "Towards a unified view of parameter-efficient transfer learning," in: ICLR 2022 - 10th International Conference on Learning Representations, 2022. doi: 10.48550/arXiv.2110.04366.
[24] Z. Zhang et al., "HyperPELT: Unified Parameter-Efficient Language Model Tuning for Both Language and Vision-and-Language Tasks," in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.725.
[25] Chen H. et al., "Conv-Adapter: Exploring Parameter Efficient Transfer Learning for ConvNets," arXiv (Cornell University), 2022. doi: 10.48550/arXiv.2208.07463.
[26] X. Mao et al., "Towards Robust Vision Transformer," in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2022. doi: 10.1109/CVPR52688.2022.01173.
[27] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: a simple data processing method to improve robustness and uncertainty," in: 8th International Conference on Learning Representations, ICLR 2020, 2020. doi: 10.48550/arXiv.1912.02781.
[28] M. Zhang, S. Levine, and C. Finn, "MEMO: Test Time Robustness via Adaptation and Augmentation", Advances in Neural Information Processing Systems, 2022. doi: 10.48550/arXiv.2110.09506.
[29] K. M. Saaim, M. Felemban, S. Alsaleh, and A. Almulhem, "Light-Weight File Fragments Classification Using Depthwise Separable Convolutions," in: IFIP Advances in Information and Communication Technology, 2022. doi: 10.1007/978-3-031-06975-8_12.
[30] T. Bansal, S. Alzubi, T. Wang, J. Y. Lee, and A. McCallum, "Meta-Adapters: Parameter Efficient Few-shot Fine-tuning through Meta-Learning," in: Proceedings of Machine Learning Research, 2022. URL: https://proceedings.mlr.press/v188/bansal22a.html.
[31] V. Moskalenko, V. Kharchenko, A. Moskalenko, and B. Kuzikov, "Resilience and Resilient Systems of Artificial Intelligence: Taxonomy, Models and Methods," Algorithms 16 3 (2023). doi: 10.3390/a16030165.