Contents

Plenary Section.

UDK 004.75

Vyacheslav Chimshir, Olena Gavrylenko,
Grzegorz Nowakowski, Eduard Zharikov, Sergii Telenyk

APPROACH TO DESIGN AND DEVELOPMENT OF SERVICES IN INFORMATION SYSTEMS OF TELECOMMUNICATIONS PROVIDERS

Abstract. The problem of design and development of services in the information systems of telecommunications providers is considered, taking into account evaluations of the effectiveness of services, trends in the development of the IT industry and analysis of user needs from various sources. The problem is considered at the levels of service design, the development of the provider's service catalog, the provider's service package for the consumer company, and the development of the provider's services. When solving problems of design and development of services at each level, in addition to traditional parameters, new evaluations of service efficiency, development trends of the IT industry and analysis of user needs from various sources are used, including the use of consumer feedback on social media.

Keywords: service; telecom provider; social media; social media; service lifecycle; service design; catalog of services.

Introduction

Information technology (IT) is successfully used to improve the efficiency of enterprises in high-tech industries, such as the provision of telecommunication services. Competition forces companies and enterprises to look for new ways of using IT to gain competitive advantages. Today, the development of business is significantly influenced by the tools of cooperation between business units and IT services.

The most common approach to the implementation of IT to support the activities of enterprises is the service approach. This was initially facilitated by the effective use of services based on the SaaS, IaaS, and PaaS models. The experience gained allowed providers to provide services according to developed models, for example IPaaS [1], managed services (Managed Services) [2]. But services based on the E2E model, when design, implementation, delivery and provision of services to production, trade and other companies and enterprises are undertaken by IT companies, turned out to be especially

convenient for business support. First of all, this model is implemented through the support of information systems of production, trade and other companies and enterprises within the IT infrastructures of the providers [3]. New classes of providers have appeared, such as MSP, competition in classes is growing, and the range of services has expanded. Accordingly, the problem of effective support of the life cycle of services that providers provide to their clients has arisen [4]. Taking into account the differences of business consumers, the diversification of business processes of client companies that require support, it is advisable to build the appropriate classification of consumers, services and their providers and develop technologies to support the life cycle of services for classes of consumers, services and providers, taking into account the peculiarities of business, the state of IT development - industries, accumulated experience and important information from other sources.

This is especially true for providers in such high-tech industries as the provision of telecommunications services. They provide a wide range of services to wide classes of consumers, among whom medium and large business companies occupy an important position, since in addition to telecommunication services, it is already a matter of supporting the business processes of enterprises, software services, etc.

Creating tools to support the life cycle of services for such providers turns into a complex problem. To increase the effectiveness of service support tools at all stages of the life cycle, it is necessary to find and use additional knowledge, for example, evaluations of users of the provided services in social media.

In work [4], the concept of a platform for supporting the life cycle of services in the information systems of communication service providers is proposed, the architecture and implementation of the specified platform are described. As part of the creation of the components of the service life cycle support platform in the information systems of communication service providers, it is necessary to conduct research, develop models and methods for the implementation of individual processes of the stages of the service life cycle. Based on these models and methods, it is possible to create technologies for effective support of all stages of the service life cycle. The same applies to the processes of designing and developing services in the information systems of telecommunication service providers.

The report considers the approach to the design and development of services in the information systems of telecommunications providers, taking into account evaluations of the effectiveness of services, trends in the development of the IT industry, and analysis of user needs from various sources.

The problem of designing and developing services in the information systems of telecommunications providers

The importance of effective tools to support the service approach is determined by a number of factors. First of all, the economy of developed countries is oriented towards services. Second, many services are complex, and their delivery requires the integration of organizations, people, technology, and information to meet consumer needs. The third, but not the last, is related to the fact that each person is a consumer of many services.

Therefore, it is important to understand what the purpose of these complex service systems should be, how they should operate to achieve the purpose, and to define a systematic approach to the effective delivery of services according to customer requirements. This requires combining knowledge from the fields of business, human behavior sciences, decision-making, and information technology to formulate new concepts, views, and approaches that will contribute to the development of a service approach.

It is necessary to give answers to many new questions regarding the improvement of efficiency, productivity and reliability of service provision. New approaches based on a unique combination of skills, knowledge and experience should create a basis for coordination of available resources of all kinds in the provision of services, innovative development of service systems.

Philosophically, the aspect of transition from product-oriented thinking to service-oriented thinking is important for implementing a service approach. There are many commonalities and differences here. Thus, in relation to both products and services, the concept of the life cycle is used, which primarily includes the design, development, delivery, provision and development of the product or service. But the product actually realizes the possibilities that the consumer needs, and the service combines the provider and the consumer as a form of realization of these possibilities under certain conditions, primarily the price and requirements for the level of service.

There are several approaches to determining the stages of the life cycle of services, which embody different views on the nature of services, take into account the peculiarities of business processes of consumers, and other factors [5]. Accordingly, there are a number of options for selecting processes that are required for each of the stages of the service life cycle. The classification of tools is usually based on the selection of processes of life cycle stages, the support of which is provided by the

information systems of providers and client companies. The process selection option, which is the basis of the organization of the research carried out in the report, is given in the article [4]. The architecture of the platform, which provides comprehensive support for all processes of the life cycle of services in the information systems of information and communication service providers and the conceptual approach to its implementation, is given in the same article. The most important are the processes of design and development of services. The location of the problem in the list of service life cycle support processes given in [4] will be important for its understanding. The approach refers to Service Design and Change Management processes.

The purpose of this article is to develop models, methods and technologies for designing and developing services in the information systems of telecommunications providers. It is necessary to perform an analysis of critical factors, describe the structure of information communication services provided by providers, develop models, methods and technologies for designing and developing services. Such a choice of services in the IT industry is determined by the fact that in such a high-tech industry there is a place for the most important classes of services - technical, cloud, and software.

The peculiarity of the study is that it is necessary to take into account the influencing factors characteristic of the current state of development of the industry. Here, first of all, it is important to take into account all available sources of information useful from the point of view of effective implementation of processes, for example, it is advisable to use the feedback of users themselves in social media for the design and development of services.

Secondly, when designing and developing services, it is necessary to take into account the economic aspects of the service approach, ensure the competitiveness of services taking into account the experience of performing operations, managing the resources of providing services with monitoring and quick response, implementing the possibility of providing services by several organizations, etc.

Thirdly, since it is important for providers of information and communication services to transition to providing services based on the E2E model, it is necessary to describe in an integrated manner the scenarios of E2E operations related to the processes of design and development of services, to clearly define milestones, inputs and outputs, roles, scope objects of special attention in order to avoid loss of time and resources, disagreement and isolation of requirements for systems and services.

Fourthly, the developed tools, and accordingly the models, methods and approaches on the basis of which they will be created, should form a fully functional complex of integrated solutions. The developed solutions for the design and development stages of services, which are based on E2E work scenarios, should cover the basic volume of scenarios and templates of service design and development operations, provide the possibility of reviewing and clarifying scenarios and processes, manual and automated preparation of versions, provide advanced tools for analysis and automated determination of requirements and their impact on the design and development of services.

Fifth, service design and development tools provide an opportunity to take into account consumer requirements. For this, the context of the application of models, methods and technologies for supporting the stages of the life cycle of services in different conditions should be related to the financial aspect, the value created through services. The rationality of the design and development of services based on the minimization of the price/quality ratio will allow creating a complete catalog of services, choosing the appropriate package of services for each consumer, rationalizing services, making them available to the widest possible range of customers.

Analysis of existing approaches

In the research and technological spheres of the IT industry, the problem of designing and developing services in the information systems of information and communication service providers is positioned as a high-priority problem.

An IT service is a complex, knowledge-based activity aiming to provide customers with defined service [6]. Thus, IT services should be provided properly to support ongoing business processes. A set of processes and procedures that are performed to ensure the usage, improvement and design of IT services is defined as IT Service Management (ITSM). The recent ITSM literature has studied benefits, challenges, opportunities, and practices for ITSM implementation [7-10].

In [11], the authors conducted qualitative research study and developed a modular service design framework as a set of design principles for information-technology-enabled services (ITeS). The authors show how different ITeS design elements, and their combinations impact the outcome-driven design of service experience. The presented service design method adopts design principles to create effective modular ITeS designs by service designers and managers.

The authors of [12] proposed to adopt ITSM and ITIL best practices to work with a specific environment of enterprise. They compare challenges and advantages of organizations with different degrees of implementing IT service management frameworks executing defined processes at different levels. The proposed approach allows to reduce issues during operation processes.

The model of Fuzzy ITIL proposed in [13] allows to identify the maturity level of continual IT service improvement by measuring maturity level using best practice of ITIL v3 to condition before and after of improving process. For this purpose the authors propose a questionnaire for obtaining the value of the maturity level for each cycle within ITIL. The proposed approach results in increasing the level of maturity in each ITIL cycle which is confirmed by the case study.

In [14], the authors propose a step-by-step approach to model the guidance of different ITSM frameworks, such as ITIL, ISO/IEC 20000, MOF, and FitSM using standardized model types and model elements eliminating the differences in presenting guidance on particular processes. Consequently, the resulting models can then be compared with relative ease.

Service-dominant economic processes impact nearly all domains of human activity. This recent trend results in transition from a product-centered to a service-centered business model switching to servitization [15]. The term “servitization” is widely spread among both product companies and agencies that provide services [16]. The recent service-centered literature has studied servitization and digitalization that are two business model innovations allowing product companies to radically transform their business model [17, 18].

In [19], the authors incorporated professional input of scholars, practitioners, and reviews to identify service research priorities focusing on four service research priorities related to managing and delivering service. They revealed an opportunity for the interdisciplinary field of service to build service ecosystems and perform transformative service research addressing important macro-level problems that will influence critical societal and environmental outcomes.

The authors of [20] provide qualitative insight for exploring the possibilities of digital transformation for better understanding where opportunities for digital service innovation lie, how to create and capture value, and readiness of the enterprise to implement digital changes. A number of connected areas are highlighted: the challenges and opportunities of digital transformation at a strategic level, the customer

experience that affected by a digital service strategy, influence of data-driven business models on service transformation.

In the research [21], a systematic literature review of 89 published studies was conducted to analyze the mutual relation between digital transformation and sustainability at a firm-level. The authors propose a research framework that considers digital transformation as a driver and a predecessor of sustainability. The focus was made on digital capabilities enhancement while balancing economic, environmental, and social impacts. With the active participation of users, the integration both digital and sustainability transformations will successfully influence the improvement of business efficiency and productivity.

The hybrid approach in smart product-service system (PSS) development is presented in [22]. Main conclusion of the paper is that the core task in the development of smart PSS is continuous evolution of the modules/components in products and services for better user experience. The authors adopt the knowledge graph technique and concept-knowledge model to propose the evolutionary design approach. A case study of a smart nursing bed fulfilling multiple personalized requirements shows an insightful guidance to industrial organizations in their development of smart PSS.

Unfortunately, the entire spectrum of the listed problems has not yet been fully resolved. In particular, there is a need for models and methods of analyzing user messages in order to determine the characteristics, functions and elements of services that are important to them, methods of determining service evaluations based on information in social media, models and methods of restructuring services in order to meet the needs of users. At the same time, the integration of statistical methods, methods of applied linguistics, decision-making theory, artificial intelligence, first of all, neural networks with deep learning are promising.

Statement of the research problem

In the report, the general emphasis is placed on the problems of creating effective tools for such important stages of the life cycle of services as their design and development. In particular, the greatest attention is paid to the following tasks:

1. Designing services provided by telecommunications providers.
2. Maintenance of the catalog of telecommunications provider services.
3. Formation of a package of services of a telecommunications provider taking into account the specifics of the consumer.
4. Development of services provided by telecommunications providers.

We will begin the characterization of these problems with the first one, which, in fact, is one of the key problems of supporting the life cycle of services, as it is related to the design of the subject of attention of most other processes - the actual services provided by telecommunications providers to consumer companies. This problem consists in structuring services, determining their constituents - components and other services, using all the available capabilities of the IT infrastructure due to scaling, balancing, parallelization, replication to ensure a certain level of performance, efficiency, reliability and other characteristics of services. In general, it is advisable to divide the solution of this problem into two steps. First, it makes sense to build an architecture that is potentially capable of providing the above properties of the service and will provide the required functionality of the service. Recently, we are talking about a microservice architecture with advanced components for organizing their interaction, such as a service registry, a security component, a balancer, a logging component, and others. The second step is to add components to this architecture that use the resources of the IT infrastructure to provide the specified functionality while complying with the entire set of customer requirements. Here we can talk about the use of components, means and technologies of replication, consistent hashing, sharding, distributed systems, clusterization, synchronization, cache and others. Here we can talk about tools that allow the telecommunications provider to form services in such a way that in the event of failure of one of the components of the structure based on the principles of consensus, losses can be quickly compensated due to other components and their interaction. When solving this problem, it is important to combine solutions in the form of patterns, components capable of taking uncertainty into account, leaving room for non-standard solutions. In addition to tools aimed at professional architects of providers and IT companies, when solving this problem, it would be desirable to develop tools, perhaps somewhat simplified, mainly based on the use of patterns, for consumers to implement their own vision of the services that will be provided to them.

The second of these problems is related to the formation of the provider's place in the services market and consists in the constant updating of the service catalog, taking into account the traditional aspects of the analysis of trends in the development of the industry, analysis and forecasting of the market development, and taking into account modern concepts of the development of the industry, primarily evaluations in social media. Here we can talk about tools that allow a telecommunications provider to maintain and expand its market share, taking into account ratings in social media by adding new services, removing services, improving services or replacing services.

The third of these problems is related to the regular clarification of specific service packages that the provider provides to specific client companies in the service market. Here we can talk about tools that allow a telecommunications provider to analyze customer service packages and reformat them taking into account the update of the service catalog, the history of the customer's use of services, evaluations in social media and the features of the state and prospects of the customer's business.

The fourth of these problems also belongs to the key problems of supporting the life cycle of services and is related to the development of services provided by the telecommunications provider to client companies, and consists in the constant development of the services themselves, taking into account the traditional aspects of the analysis of trends in the development of the industry, analysis and forecasting of market development and taking into account modern concepts of industry development, primarily evaluations in social media. Here we can talk about tools that allow a telecommunications provider to develop the structure of services, for example, by adding new functions and components, restructuring services in order to improve their productivity, efficiency, reliability and other characteristics, taking into account ratings in social media.

When solving each of the problems, evaluations in social media are used. Obtaining ratings based on user feedback is a separate problem that, in turn, requires several sub-problems.

First, it is a selection of sources and messages useful from the point of view of forming evaluations of services in social media.

Secondly, determining the positive or negative color of messages.

Thirdly, calculation of ratings of services and their components based on user reviews in social media.

Here, possible estimates can be the number of new customers of the services provided by the telecommunications provider. In addition, it seems useful to use a feedback scale.

A general approach to solving the problem of designing and developing services in the information systems of telecommunication providers

This section of the report discusses the models and methods that will be used to solve the problem of designing and developing services in the information systems of telecommunications providers, taking into account the above trends, conditions, concepts and limitations.

Models and methods of development of the catalog of telecommunications provider services

Let's start by considering the models, methods and technologies of updating the catalog of services. As mentioned above, we will take into account both traditional aspects of the analysis of trends in the development of the industry, analysis and forecast of market development, and modern aspects (concepts of the development of the industry, evaluations in social media).

First, let's consider the features of the catalog, in which service providers define the services provided to customers. In general, the catalog for each service traditionally describes its attributes, SLA parameters, terms of service provision and service prices. Common parameters of services are the category and name of the service, description of the service, functions of the service, time frames for providing the service, etc.

The consumer can view the catalog of services to compare the available services, their functions and other parameters, the price, and choose the most beneficial services for the consumer. The catalog of services is the basis for ordering services in the order of self-service.

But for this, service providers must define, publish and manage the catalog of services using the appropriate tools.

Mathematical formulation of the problem:

Let $X=\{x_1, x_2,\ldots,x_n\}$ – the set of all services, n – the number of services, $Y=\{y_1,y_2,\ldots,y_m\}$ – the set of all providers, m – the number of providers. It is necessary to establish a mutually unambiguous mapping between these sets $X \to Y$. That is, we must match each service with the provider that provides it. In this way, we can combine all services provided by one provider.

Reasoning:

In fact, the formulated problem is reduced to a classification problem. We have n objects (services) and m classes (providers). To solve it, it is recommended to use neural networks or decision trees (Card algorithm, ID3, Random Forest). You can also use the Naive Bayesian classifier.

Classification accuracy can be monitored using the Recall and Precision metrics (Fig. 1).

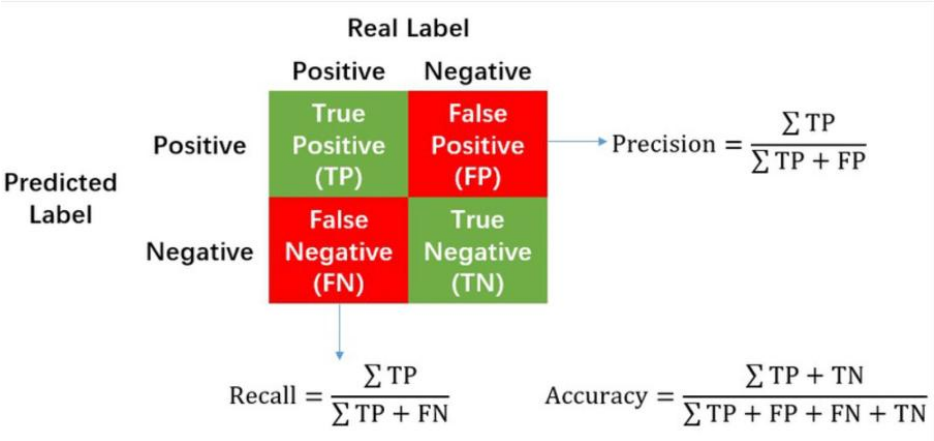


Fig. 1. Calculation of Precision, Recall and Accuracy in the confusion matrix [23].

Since the catalog of services can be represented as a set of services, the following operations apply to it:

1) adding an element to a set;

2) removing an element from the set.

Accordingly, we are talking about adding or removing a service (a group of services).

Therefore, the problem arises of determining the expediency of implementation or the expediency of service support.

If information is known about the number of users who ordered a certain service for a fixed period of time, then with the help of time series it is possible to forecast the number of service orders in the next period of time [24]. In addition, it is possible to apply the ATAPSN algorithm (algorithm taking into account posts in social networks) [25] for more accurate forecasting of the number of service orders. Thus, it is possible to determine the popularity of the service and make decisions about the feasibility of its support.

If information is known about the profit from the implementation of the service or the costs of its implementation for a certain period of time, then this service can be represented as a random variable. Thus, by calculating the mathematical expectation and variance of a given random variable, it is possible to determine the average expected profit (average cost) and the risk of not getting it (root mean square deviation). Based on these data, it is possible to draw a conclusion about the expediency of implementing a new service or the expediency of supporting an existing one.

If you conduct a survey of users regarding the quality of a particular service, you can calculate the ratio of favorable reviews to the total number of reviews. It is also advisable to

calculate the ratio of disapproving reviews to the total number of reviews. These parameters will allow you to monitor the opinion of users about the quality of the service.

It is also recommended to conduct a survey of potential users for the purpose of introducing a new service. In this case, you can also use similar parameters.

Based on such studies, decisions will be made about adding or removing services. Thus, the catalog of services will contain services that are popular among users and will help the provider to improve their quality.

Models and methods of development of a package of services of a telecommunications provider

Now let's move on to consider the models, methods and technologies of updating the catalog of services. As mentioned above, we will take into account both traditional aspects of the analysis of trends in the development of the industry, analysis and forecast of market development, and modern aspects (concepts of the development of the industry, evaluations in social media).

Mathematical formulation of the problem:

Let $X=\{x_{1j}, x_{2j},...,x_{ij}\}$ – the set of all services provided by the provider, i – the number of services, m – the number of providers. It is necessary to select some subsets of services X_q from the set X, q - the number of subsets. We will consider these subsets as packets.

Reasoning:

Subsets X_q are formed on the basis of the principles of establishing hidden regularities between services from the set X. The following principles can be considered:

1. Similarity of interests or needs of users.

The interests or needs of consumers can be determined by analyzing their profiles in social networks. In such a case, the problem is reduced to a clustering problem, where a cluster is a group based on interests (needs). For implementation, the K-means algorithm is recommended, in which the recommended number of clusters can be calculated using the "Elbow" method.

Analyzing the social graph, in each cluster it is possible to additionally determine a social network user who will influence the opinion of his friends or like-minded people (that is, if he uses a service, his friends will also want to use it). In this way, such a user can be chosen if there is a need for effective distribution of service advertisements. Moreover, if a certain user uses several services at once, then they can be recommended to form a package of services, which will be based on the common interests (needs) of a certain social group of users.

It should also be added that the service package may include services provided by different providers.

2. Similarity of service characteristics.

If a list of service characteristics is formulated, then each of them can be represented as a vector of 0 and 1 (1 – the service corresponds to this characteristic, 0 – it does not). Thus, we can calculate similarity coefficients (Jaccard, Sorensen, Simpson, Otiai, Brown-Blanquet, etc.) between pairs of services. It is possible to recommend adding such services to the package of services that have the value of the given coefficient above the indicated minimum threshold (preferably closer to unity).

3. Statistical dependence of services.

If information is known about how many users ordered this or that service during the specified time period (month, year, etc.), it is possible to determine the degree of dependence between pairs of services. For this, it is worth using correlation coefficients (Pearson, Spearman, Kendall). Statistical significance between them should be established using the statistical t-test.

As a result of the conducted statistical analysis of each service for the specified period of time, it is also possible to determine the feasibility of implementing this or that service. This will make it possible not to consider services that are not relevant for a certain target audience of users.

4. If statistical information is known about which services a specific user orders at the same time, then associative rules can be applied to form a package of services: if the user ordered service X_k, he will also order service X_p.

A chain of services can be formed from such elementary rules, on the basis of which a package of services can be formed. APRIORI or ECLAT algorithm should be used to build associative rules. The accuracy of the obtained results can be monitored using metrics such as supp, conf, and lift.

It should be noted that the proposed principles of formation of service packages allow only to recommend services, on the basis of which service packages will be formed. The final decision on their formation must be taken by the provider(s).

It is also worth emphasizing that the choice of the principle of formation of service packages depends on the type of statistical data. The larger and more diverse the input datasets will be, the wider the range of principles for forming service packages can be offered. This will allow a more personalized approach to both the needs of the user and the needs of the provider.

Models and methods of development of a separate service of a telecom provider, taking into account social media

Support of the life cycle of services refers to services that are entered in the catalog of services. That is, the services must be designed, implemented, delivered, provided to consumers according to the parameters, conditions, prices specified in the catalog and, if necessary, developed. This involves, first, the implementation of common actions related to the construction of the service structure and the description of the work process of its provision, the implementation of the components of the structure and their interaction to provide the service defined by the user with the minimization of the provider's resources. Secondly, it also involves the implementation of certain processes specific to the services of certain industries, for example, service proposals, creating a contact for a service. For services that are specific to telecom providers, most often it is about defining a service template, creating an orchestration workflow.

In general, a service pattern is a set of interrelated hardware and/or software components that make up a service and interact during service deployment. Service templates are usually defined in the service directory. After receiving a service request, hardware and/or software resources are allocated, configured, and integrated according to a service template to create an instance of the service. Thus, a service pattern provides a standard for multiple creation of intended instances of a given service. From the perspective of consumers, the features of the service template help them understand the hardware configuration, software, and security mechanism for the service. From the provider's perspective, it provides guidance on creating workflows for service organization. Features of a service template typically include the following entities:

Service Structure: Defines the structure of the service, which includes the components of the service and their relationships. For example, a requirement for a SaaS service may be reflected in running a specific business application on a guest OS, deploying a database to support the application, and hosting the database and application on virtual machines.

Service Attribute: Defines configurations of service components. For example, the attributes of a virtual machine that is provided as a service component are: the number of processors of a certain computing power, the size of memory, and the number of attached disks of a certain size.

Service Operation: Defines management operations such as add, modify, start, and stop that can be performed in the service management interface.

Creating an orchestration workflow: After defining a service template, workflows for service orchestration are created in the orchestrator based on the template specification. These workflows enable the automated allocation, configuration, and integration of hardware and/or software resources for a service according to a service template.

Conclusion

An approach to the design and development of services in the information systems of telecommunications providers, taking into account social media, is proposed. Within the framework of the proposed approach, attention is paid to solving such tasks as designing services provided by telecommunications providers, maintaining a catalog of telecommunications provider services, forming a package of telecommunications provider services taking into account the specifics of the consumer, developing services provided by telecommunications providers.

The implementation of the approach is based on the creation of tools for the development of the catalog of services, the development of a package of services, the development of a separate service of a telecommunications provider, taking into account social media, based on models and methods within the framework of the service life cycle support platform in the information systems of telecommunications providers.

Based on the proposed approach, the report outlines models and methods for solving these problems, which integrate the apparatus of set theory, statistical theory, decision-making theory, and artificial intelligence. This allows users to make decisions in the processes of designing and developing services, maintaining a catalog of services and forming a package of services of a telecommunications provider, taking into account the accumulated experience, the state of development of the industry, and makes it expedient to use them in information systems aimed at increasing business competitiveness.

Further research is related to the development of models and methods of design and development of services in the information systems of telecommunications providers, oriented to use by professional architects of providers and consumers, which is very important for the realization by consumers of their own vision of services and the development of real services of telecommunications providers.

References

1. Top 7 Integration Platform as a Service (iPaaS) Software Companies in 2021 [Online] Available from: https://www.spiceworks.com/tech/cloud/articles/top-ipaas-companies/ Accessed 21.06.2023.

2. What Are Managed IT Services and Why Do You Need Them? [Online] Available from: https://www.spiceworks.com/tech/cloud/articles/managed-it-services/ Accessed 21.06.2023.

3. Maddern, H., Smart, P. A., Maull, R. S., & Childe, S. (2014). End-to-end process management: implications for theory and practice. Production Planning & Control, 25(16), 1303-1321.

4. V. Chymshyr, S. Telenyk, O. Rolik, E. Zharikov, "The platform for supporting the life cycle of services in the information systems of information and communication service providers", Adaptive systems of automatic control, Vol. 1 No. 42, pp. 205–226, 2023, https://doi.org/10.20535/1560-8956.42.2023.279172.

5. Service Lifecycle Management [Online] Available from: https://dokumen.tips/documents/service-lifecycle-management-semantic-scholar-sents-the-work-on-service-lifecycle.html?page=23 Accessed 21.06.2023.

6. Kowalkowski, C., Kindström, D., & Gebauer, H. (2013). ICT as a catalyst for service business orientation. Journal of Business & Industrial Marketing, 28(6), 506-513.

7. Serrano, J., Faustino, J., Adriano, D., Pereira, R., & da Silva, M. M. (2021). An IT service management literature review: challenges, benefits, opportunities and implementation practices. Information, 12(3), 111.

8. Mora, M., Gómez, J. M., Wang, F., & Díaz, E. O. (2021). A review of the IT service design process in agile ITSM frameworks. balancing agile and disciplined engineering and management approaches for IT services and software products, 248-270.

9. Richter, H., & Lantow, B. (2021, June). IT-service value modeling: a systematic literature analysis. In International Conference on Business Information Systems (pp. 267-278). Cham: Springer International Publishing.

10. Mora, M., Marx-Gomez, J., Wang, F., Diaz, O. (2021). Agile IT Service Management Frameworks and Standards: A Review. In: Arabnia, H.R., Deligiannidis, L., Tinetti, F.G., Tran, QN. (eds) Advances in Software Engineering, Education, and e-Learning. Transactions on Computational Science and Computational Intelligence. Springer, Cham.

11. Tuunanen, T., Salo, M., & Li, F. (2023). Modular service design of information technology-enabled services. Journal of Service Research, 26(2), 270-282.

12. Ilieva, R. Y., & Nikolov, Y. P. (2020, July). IT Service Management Framework to Improve Business Information Structures. In 2020 XI National Conference with International Participation (ELECTRONICA) (pp. 1-5). IEEE.

13. Yandri, R., Utama, D. N., & Zahra, A. (2019). Evaluation model for the implementation of information technology service management using fuzzy ITIL. Procedia computer science, 157, 290-297.

14. Michael, S., Michael, B., & Thomas, S. (2019, April). IT service management frameworks compared–simplifying service portfolio management. In 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM) (pp. 421-427). IEEE.

15. Kowalkowski, C., Gebauer, H., Kamp, B., & Parry, G. (2017). Servitization and deservitization: Overview, concepts, and definitions. Industrial Marketing Management, 60, 4–10.

16. Rabetino, R., Harmsen, W., Kohtam¨ aki, M., & Sihvonen, J. (2018). Structuring servitization-related research. International Journal of Operations & Production Management, 38(2), 350–371.

17. Favoretto, C., Mendes, G. H., Oliveira, M. G., Cauchick-Miguel, P. A., & Coreynen, W. (2022). From servitization to digital servitization: How digitalization transforms companies' transition towards services. Industrial Marketing Management, 102, 104-121.

18. Zheng, P., Wang, Z., Chen, C. H., & Khoo, L. P. (2019). A survey of smart product-service systems: Key aspects, challenges and future perspectives. Advanced engineering informatics, 42, 100973.

19. Field, J. M., Fotheringham, D., Subramony, M., Gustafsson, A., Ostrom, A. L., Lemon, K. N., ... & McColl-Kennedy, J. R. (2021). Service research priorities: designing sustainable service ecosystems. Journal of Service Research, 24(4), 462-479.

20. Zaki, M. (2019). Digital transformation: harnessing digital technologies for the next generation of services. Journal of Services Marketing.

21. Gomez-Trujillo, A. M., & Gonzalez-Perez, M. A. (2022). Digital transformation as a strategy to reach sustainability. Smart and Sustainable Built Environment, 11(4), 1137-1162.

22. Li, X., Chen, C. H., Zheng, P., Wang, Z., Jiang, Z., & Jiang, Z. (2020). A knowledge graph-aided concept–knowledge approach for evolutionary smart product–service system development. Journal of Mechanical Design, 142(10), 101403.

23. Ma, J., Ding, Y., Cheng, J. C., Tan, Y., Gan, V. J., & Zhang, J. (2019). Analyzing the leading causes of traffic fatalities using XGBoost and grid-based analysis: a city management perspective. IEEE Access, 7, 148059-148072.

24. Balaji, T. K., Annavarapu, C. S. R., & Bablani, A. (2021). Machine learning algorithms for social media analysis: A survey. Computer Science Review, 40, 100395.

25. Задача аналізу публікацій для побудови прогнозу відносно зміни курсу криптовалют/ О. Гавриленко, М. Мягкий, Я. Жураковський /Адаптивні системи автоматичного управління т.2, №41, 2022, С. 90-99, https://doi.org/10.20535/1560-8956.41.2022.271349

Parallel Section SEC. Security, Fault Tolerance

UDC 004

Artem Volokyta, Artem Dremov

METHODS AND SOLUTIONS FOR TRAFFIC MONITORING AND CONTROL IN SDN ARCHITECTURES

Abstract: The article aims to provide an analysis of modern computer network traffic monitoring solutions and technologies. In this article an overview of network traffic monitoring solution is provided, including general firewall designs and the use of firewalls in SDN networks. The article also touches on general architecture of SDN networks and SDN controllers. An example firewall solution application is presented in the article.

Keywords: firewall, cybersecurity, network security, traffic monitoring, SDN.

Introduction

While computer networks are an integral part of modern infrastructure, this has also made them susceptible to remote attacks, such as malware injections, DNS poisoning, DoS attacks, TCP hijacking, and other attacks. As such, the topics of network security and traffic monitoring remain some of the most crucial topics in cybersecurity. Additionally, emerging architectures and concepts, such as SDN networks, require individual solutions, that can provide the best results for a specific problem [1]. One of the most common techniques for traffic monitoring and control is a firewall, which allows to control incoming and outgoing traffic based on a table of security rules. These systems allow individual hosts or a controller host to block out malicious traffic while allowing communication between computers in the local network. However, traditional firewall solutions are proving hard to adapt in SDN environment. As such the topic of SDN network security remains one of the most forward-looking and important topics in the area of research [2].

Problem Definition

General problem explored in this article is of network security. In particular, the article explores subjects of defence against malware attacks, DoS attacks, DHCP, DNS attacks, etc. This is a problem prominent in modern day computer network infrastructure for organisations as well as networks designed for personal use. Coincidentally, the use of SDN networks is becoming more popular with businesses and for personal use. As such the problem of data security and traffic control in SDN networks is prominent and is the main topic of this article.

SDN network architecture

A typical SDN network consists of a number of hosts, switches (routers, other connection devices) and a controller device. SDN controller is software that typically runs on a unix based system. In a SDN network infrastructure we can define data plane, control plane and management plane (fig. 1).

Data plane is mainly responsible for the movement of data, in this case packets, in the network. The movement of packets is performed by a network device, such as a switch, that support openflow protocol. Unlike traditional networks, data plane devices do not implement logic on where to forward packets, as that is handled by the control plane.

Management plane is the part of the network that houses network applications. These applications can be running on a host device, or, occasionally, a controller. The applications allow host devices to communicate with the controller and can implement networking logic.

Control plane contains controller devices that implement the bulk of routing logic. A SDN controller is responsible for constructing and maintaining flow tables (similar to routing tables), ARP mapping, network devices discovery, STP protocols, etc. SDN controller is also responsible for communicating routing rules to the data plane, via southbound API (such as openflow protocol). The controller is also responsible for communication with management plane applications via northbound API, such as REST API, python API, programming languages such as frenetic and pyretic, and others [3]. The specific API's available depend on the controller software implementation. Popular SDN controller software includes opendaylight, floodlight, pox, nox, cisco ACI, vmware NSX, and others.

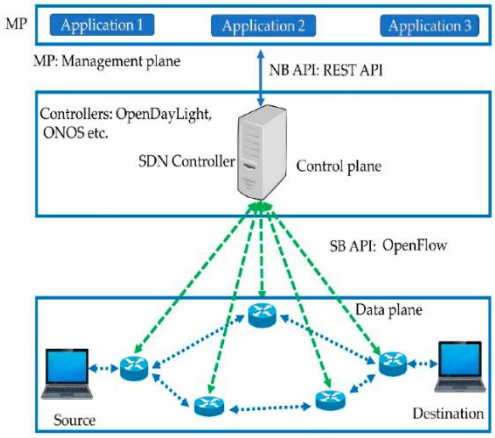


Fig. 1. Architecture of a SDN network

Additionally, the control plane can be centralised (with one controller), hierarchical (with multiple controllers organised in a hierarchy) or distributed (with multiple controllers that control different parts of the data plane).

For the topic of this article, we are mainly interested in management plane network applications for firewall implementation. The application itself may be hosted on the controller itself (fig. 2). As described in previous paragraphs, the SDN controller is responsible for constructing and maintaining the flow tables for the network. These flow tables are copied on to switches and are used to determine whether to drop or forward packets. The switch uses packet header data, such as destination ip, source ip, port, etc to match to flow table rules (fig. 3) [4]. Dropped packets, or packets that do not have a matching flow rule can be configured to be forwarded to the controller for inspection. As such it is possible to implement a variety of firewall designs, including flow-based firewalls, tcp firewalls. The controller can then update the flow tables for controllers.

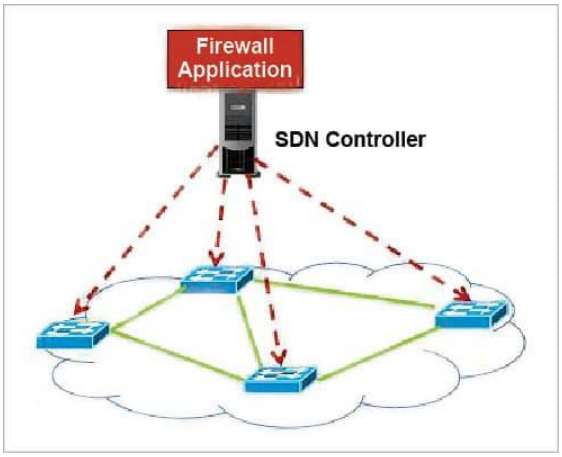


Fig. 2. A variant of integrating a firewall application in a SDN network

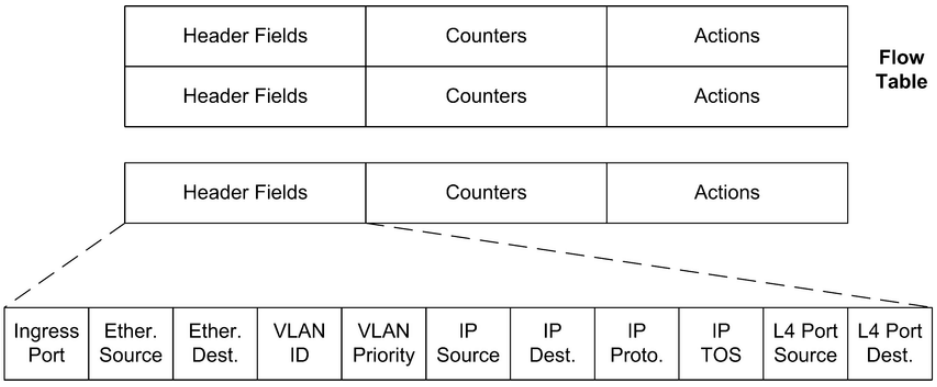


Fig. 3. Example of a flow table and packet header data used

Firewall design

A firewall is a system that is designed to monitor and control incoming traffic. In context of this article, we are primarily focusing on software-based firewalls. A software firewall as opposed to hardware firewall, can be deployed on any compatible device as an application. For example, in case of a SDN controller, a firewall can be a python-based application (in case of pox/ryu controllers) or Rest API and java application (odl, floodlight).

A simple implementation can operate as a packet filtering firewall. This design uses an access control list (ACL) table to match incoming traffic to the table rules. An example of an ACL table in figure 4. The controller can then construct a flow table based on incoming packets and their ACL table matches [5].

order	protocol	src_ip	src_port	dst_ip	dst_port	action
1:	tcp,	140.192.37.20,	any,	*.*.*.*,	80,	deny
2:	tcp,	140.192.37.*,	any,	*.*.*.*,	80,	accept
3:	tcp,	*.*.*.*,	any,	140.192.37.40,	80,	accept
4:	tcp,	140.192.37.*,	any,	140.192.37.40,	80,	deny
5:	tcp,	140.192.37.30,	any,	*.*.*.*,	21,	deny
6:	tcp,	140.192.37.*,	any,	*.*.*.*,	21,	accept
7:	tcp,	140.192.37.*,	any,	140.192.37.40,	21,	accept
8:	tcp,	*.*.*.*,	any,	140.192.37.40,	21,	accept
9:	tcp,	*.*.*.*,	any,	*.*.*.*,	any,	deny
10:	udp,	140.192.37.*,	any,	*.*.*.*,	53,	accept
11:	udp,	*.*.*.*,	any,	140.192.37.*,	53,	accept
12:	udp,	*.*.*.*,	any,	*.*.*.*,	any,	deny

Fig. 4. Example of an ACL table of a typical firewall

It is also possible to implement stateful packet inspection system. Stateful packet inspection is a technology that allows the firewall application to construct a state table of inspected packets. The state table allows the software to keep track of and log incoming connections (fig. 5). Incoming packets can be matched to the state table to be allowed through, according to the security policy. The state table entries can be deleted with appropriate packet for session end, or as a timeout. The benefits of this approach including traffic logging capabilities, more thorough packet inspection and finer control over incoming traffic. In context of SDN networks, it is possible to direct all incoming packets towards the controller for inspection. This, however, creates additional round trips from switch to controller back to switch. This can also be demanding of the controller’s device hardware. Alternatively, the Open vSwitch software has the capability to track incoming connections [4].

Source Address	Source Port	Destination Address	Destination Port	Connection State
192.168.1.100	1030	192.0.2.71	80	Initiated
192.168.1.102	1031	10.12.18.74	80	Established
192.168.1.101	1033	10.66.32.122	25	Established
192.168.1.106	1035	10.231.32.12	79	Established

Fig. 5. Example of state table of a stateful firewall design

Use cases and vulnerabilities

As discussed earlier, a SDN based firewall software can be used as a software-type firewall operating at layers 3 and 4 of OSI model, and can be implemented with features such as stateful inspection and tcp/openflow packet filtering. It is also possible to implement a solution with the usage of application firewall technology, which would allow the firewall software to perform deeper packet inspection at layer 7 of OSI model. Due to architecture of SDN controllers discussed earlier, it is also possible to implement additional applications, such as anti-malware software in order to inspect traffic payloads for possible malware, as well as load balancing and IPS/IDS solution. The firewall software can be used for centralized policy enforcement or as a distributed design [6][7][8]. A firewall can be implemented with one of northbound APIs, which allows it to be used on a variety of controllers, depending on API used. As such a SDN based firewall can be used in a wide variety of network architectures and topologies.

Overall, the main advantages of using firewalls in SDN networks come down to several factors. One of the main factors is the separation of control and data planes via SDN controllers, which allows to use a single firewall solution running alongside the controller to monitor the traffic a part of the network. This also allows the usage of simpler switches and other connection devices, that may be less susceptible to DoS attacks. Additionally, by running several controllers in a network, it is possible to divide the network into segments, where traffic between segments goes directly through the controller. This means that if a single segment gets compromised, the other segments can still be operational and can adjust security policies. Lastly, the use of software-based solution and network virtualisation, as well as programmable connection devices, allows for a highly scalable architecture, in particular in terms of firewalls designs.

Common attack vectors on SDN networks include switch device DoS attacks, spoofing attacks, for example, by imitating network switches, as well as man-in-the-middle attacks. Another vulnerability lies in the management plane applications, which, if compromised (for example with malware) can allow the insertion of forged rules. Another

big vulnerability of SDN networks is the importance of SDN controllers to the network's operation, since the controller implements most of the networking and packet forwarding logic [2]. This means that SDN controllers can be vulnerable to DoS attacks in particular, and a compromised controller can make the entire network inoperable. This, however, can be mitigated with hierarchical and distributed controller designs.

Example solution

For an example implementation, a mininet environment will be used. We will also be using POX controller and provided python API. To store the rules it is possible to use a python Dictionary, using values from packet header fields, such as destination IP address and source IP address as keys (fig. 3), and security policy, such as drop and forward as values. Alternatively, it is possible to store firewall rules in a separate configuration file instead. To check the rule, we can match one or several values of the packet header to the rules and determine the policy from there. It is then possible to employ openflow API to instruct the switch to forward or drop the packets as needed. Alternatively, it is possible to instruct the openflow API to forward packets to the controller for inspection by firewall software, or other network applications. For the topology a simple linear topology with 4 switches and 4 hosts was used (fig. 6). The rule inserted specifies that no traffic can be transmitted between hosts 1 and 2(fig. 7). Additionally, it is worth nothing that the implementation of a firewall is tied to controller's available northbound API. Since there is no common API for all available controller software, the firewall application cannot be deployed on certain controllers, for example, this implementation cannot be deployed on java/REST based controllers such as opendaylight or floodlight.

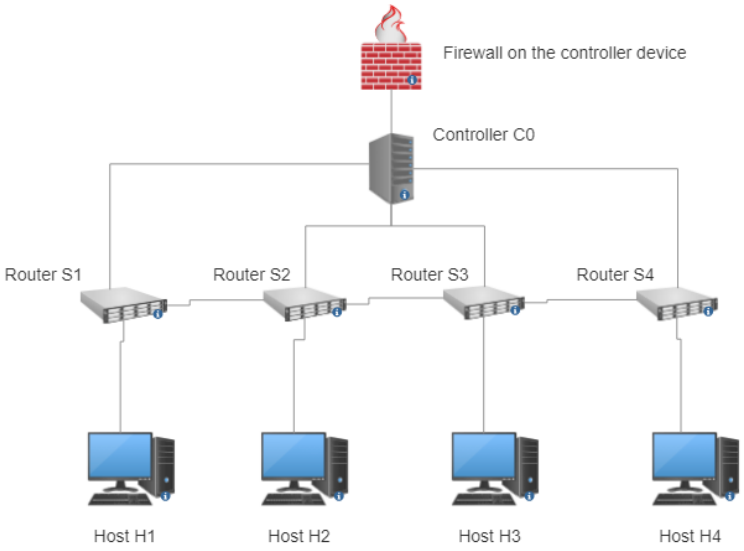


Fig. 6. Topology used for testing the example firewall

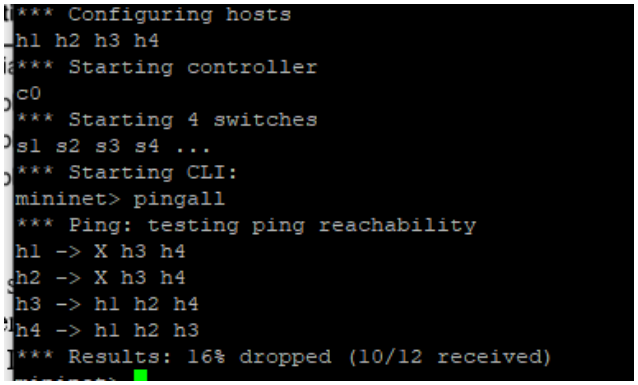


Fig. 7. Demonstration of packets being dropped via ping command using example implementation

Conclusions

In this article we have reviewed the problem of traffic monitoring and control and how it applies to SDN networks. We have reviewed the architecture of modern SDN networks. We have provided an analysis of firewall's role in a SDN network. SDN networks are becoming more and more widely used and this facilitates the adaptation of conventional firewall designs to SDN specific architectures. Additionally, SDN networks and controllers provide a wide range of tools, that can allow to implement additional security features, such as stateful packet inspection, distributed firewalls and application firewalls. SDN architecture also provides a number of potential security and network management advantage. We have also highlighted possible security vulnerabilities in SDN networks, that can be covered with usage of firewalls. We have reviewed the design of a typical software-based firewall. We have pointed out common firewall technologies that can be implemented with the use of modern SDN controllers.

Additionally, we have identified potential benefits and drawbacks of using SDN-based firewalls. We have identified potential use cases, including centralized and distributed network designs. We have also discussed potential benefits and vulnerabilities of reviewed firewall designs.

We have presented a simple implementation of a firewall on POX controller with Python API. The implementation can be configured to apply drop rule to packets between specified devices, or to apply forward to specific packets and drop all other packets. The example is used to demonstrate a way that a firewall can be implemented as a network application and to demonstrate the workings of a SDN-based firewall. It is possible to further extend the solution to implement additional features, such as stateful inspection and application layer design.

References

1. Mousa M., Bahaa-Eldin A., Sobh M. Software Defined Networking concepts and challenges // 2016 11th International Conference on Computer Engineering & Systems (ICCES). Cairo, Egypt, 2016. C.79-90. URL: https://ieeexplore.ieee.org/ document/7821979

2. Security Issues in Software Defined Networking (SDN): Risks, Challenges and Potential Solutions / M. Iqbal та ін. // International Journal of Advanced Computer Science and Applications. 2019. № 10 (10). C.298-302 URL: https://thesai.org/Downloads/Volume10No10/Paper_42-Security_Issues_in_Software_Defined_Networking.pdf

3. Paliwal M., Shrimankar D., Tembhurne O. Controllers in SDN: A Review Report // IEEE Access. 2018. № 6 (1). C.36256-36270 URL: https://ieeexplore. ieee.org/document/8379403

4. Braun W., Menth M. Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices // Future Internet. 2014. № 6 (2). C.302-336 URL: https://www.researchgate.net/publication/284696928_Software-Defined_Networking_Using_OpenFlow_Protocols_Applications_and_Architectural_Design_Choices

5. EFFICIENT ALGORITHMS AND ABSTRACT DATA TYPES FOR LOCAL INCONSISTENCY ISOLATION IN FIREWALL ACLS / S. Pozo та ін. // The International Joint Conference on e-Business and Telecommunications / Italy. Milan, 2009. C.42-53. URL: https://www.researchgate.net/publication/221436666_Efficient_Algorithms_and_Abstract_Data_Types_for_Local_Inconsistency_Isolation_in_Firewall_ACLs

6. Pena J., Yu W. Development of a distributed firewall using software defined networking technology // International Conference on Information Science and Technology (ICIST). N/A, 2014. C.449-452. URL: https://www.researchgate.net/ publication/286573159_Development_of_a_distributed_firewall_using_software_defined_networking_technology

7. Bannour F., Souihi S., Mellouk A. Distributed SDN Control: Survey, Taxonomy, and Challenges // IEEE Communications Surveys & Tutorials. 2017. № 20 (1). C.333-354 URL: https://ieeexplore.ieee.org/document/8187644

8. Obadia, M., Bouet, M., Leguay, J., & et al. (2014). Failover mechanisms for distributed SDN controllers. *2014 International Conference and Workshop on the Network of the Future (NOF)*, (1-6). Paris: IEEE. Retrieved from https://ieeexplore.ieee.org/ document/7119795

AUTHORS

Volokyta Artem – Ph. D., Associate Professor, Department of Computer Engineering NTUU “KPI”.

Scientific interests: real time systems, information security, distributed computations

Dremov Artem – student of the Department of Computer Engineering NTUU “KPI”.

Scientific interests: real time systems, cybersecurity, machine learning.

UDC 004.9

Roman Serebriakov, Artem Volokyta

ENHANCING SYSTEM FAULT TOLERANCE THROUGH THE USE OF BLOCKCHAIN TECHNOLOGY

The paper deals with the issues of the profitability of using blockchain technology to create more enhanced fault-tolerant systems. The advantages and disadvantages of using public and private blockchain are considered, as well as the impact of this technology on distributed and centralized systems. Examples of the use of blockchain technology to increase system productivity in various spheres of human activity are given.

Keywords: blockchain, fault-tolerant systems

Relevance of the research topic. Blockchain technology, which forms the foundation of decentralized digital currencies, is becoming very popular in today's world due to qualities such as transparency, immutability, reliability, speed, and efficiency. Therefore, it's necessary to consider the use of this technology to increase the reliability and fault tolerance of systems in various spheres of life.

Target setting. The opportunity to find better ways to increase the reliability and fault tolerance of computer systems.

Actual scientific researches and issues analysis. Recent studies show that blockchain technology can be very useful for the task of increasing system reliability and fault tolerance due to decentralization, transparency in task execution, and security features such as immutability. However, there is also an opinion that for large computations, the question of energy efficiency arises, especially when performing consensus algorithms to confirm the result.

Uninvestigated parts of general matters defining. This article is dedicated to studying and analyzing approaches to optimizing the use of blockchain technology in high computational requirements and in distributed systems.

The research objective. Analyze the appropriateness of using blockchain technology to increase system fault tolerance. Consider the advantages and disadvantages of using this technology in centralized and distributed systems.

The statement of basic materials. Blockchain technology, which underlies most decentralized digital currencies such as Bitcoin, Ethereum, Binance Coin, and many others, has great potential for creating reliable and fault tolerant systems [1].

Thanks to features like decentralization, transparency, and immutability, it's possible to enhance fault tolerance and improve the security of computer systems.

Blockchain for fault tolerance. Fault tolerance is the system's ability to continue performing actions programmed into it even after the occurrence of malfunctions or the complete breakdown of some of its components [2]. Blockchain ensures fault tolerance through decentralization and consensus mechanism. As each node contains a copy of the entire blockchain, the system will continue to work even if one node fails. Thanks to the consensus mechanism, a single node failure won't disrupt the system's operation, as each node's state is agreed upon with others, and nodes with deviations will be ignored by the system. These consensus mechanisms come in various forms, including Proof of Work (PoW), Proof of Stake (PoS), and Byzantine Fault Tolerance (BFT). PoW requires significant computational effort to validate transactions, whereas PoS requires participants to have a certain number of coins. BFT, on the other hand, is designed to provide reliable consensus in systems with potentially hostile nodes.

Public and private blockchain. However, a system based on blockchain technology is not perfect and has many different usage aspects. For example, to create a fault tolerant system, we can use both a public and a private blockchain [3]. In the case of using a public blockchain, you need to pay "gas" when performing operations. Using this approach can have a very negative impact on your bank account, especially under high system load. But in this case, the fault tolerance of the system will be at its maximum, as anyone can join the computation process and for a certain fee, above-mentioned "gas", provide their computing capabilities to perform operations. Thus, we have a strongly decentralized topology in the network created around the public blockchain, in which the correctness of the program is checked using public consensus. However, this approach can have weak performance due to the same strong decentralization of the system. Another approach to creating a fault tolerant system is the use of a private blockchain. In this case, it is possible to control the entire network and set your own rules for its operation. Such a system is not decentralized, so each node must be serviced and maintained for stable task execution. This solution leads to a reduction in fault tolerance but does not require transaction fees and can be useful for organizations that want to use blockchain properties without making their network externally accessible.

Blockchain in distributed systems. The next issue may be the use of blockchain technology in distributed systems, as they are designed to maximize the division of program code into parts and parallel execution of a given task, while blockchain implies performing the same task on many nodes simultaneously. Indeed,

blockchain technology and distributed computing serve two different purposes and they solve two different types of problems. Because each blockchain transaction is verified by the entire network, operations performed with blockchain provide a high level of security and increase fault tolerance. However, this technology will perform very poorly in large computations, as each node will perform the same operation. Distributed systems are used for maximum optimization of the use of computational resources. Large tasks are divided into parts that can be executed in parallel on many nodes. This approach provides much higher execution speed, but the system is much less fault tolerant. Therefore, while it is possible to use blockchain technology for distributed computing to ensure transparency and security, such an approach will significantly reduce the execution time of a task and heavily load the system.

Other approaches to enhance fault tolerance. Now let's consider other ways to increase system fault tolerance, separately for distributed and centralized ones. For centralized systems, the following approaches can be used:

1. Redundancy - duplicating hardware, data storages, or software to ensure continuous system operation in case of one component failure.
2. Regular backup - copying system operating data to a remote storage to restore the state of system operation after a failure.
3. Disaster recovery mode - introducing some solutions into the software that allow the system to quickly recover after a failure.

For distributed systems, the following approaches can be used:

1. Replication - creating copies of data or services in different parts of the system so that if one node fails, another node can continue processing.
2. Sharding - distributing data among different nodes, allowing the system to continue working, even if one node fails.
3. Applying consensus algorithms - manage the state of the system in a distributed environment using consensus algorithms to ensure data consistency between nodes.

So, we see that in addition to blockchain technology, there are many ways to increase the fault tolerance of computer systems depending on the type and requirements. Depending on the task, you can choose which one fits best.

Application areas. Let's talk about the exact areas where blockchain technology would be useful in creating a fault tolerant system that also has a high level of security and transparency:

1. Financial sector: Blockchain technology is already widely used in cryptocurrencies to ensure fault tolerance in conditions where there is no centralized authority that would control transactions. The same solution can be useful for ensuring security in banking systems and other financial institutions.

2. Logistics sector: Blockchain can be used to create a fault tolerant supply tracking system where each stage in the supply chain can be tracked and verified, ensuring transparency, and reducing the possibility of fraud.

3. Healthcare sector: Blockchain can help create fault tolerant systems for recording and storing patient data that guarantee the integrity and privacy of patient data.

4. Legal sector: Blockchain technology can ensure fault tolerance and transparency in electronic voting systems, reducing the possibility of manipulation and fraud in elections. Also, the immutability of the blockchain would help to avoid the "disappearance" of evidence and other investigative materials from court cases, which would help eradicate corruption in the judicial system.

Implementation problems. With widespread implementation of blockchain technology, you may face the following problems:

1. Scalability: Blockchains may have problems processing many transactions per unit of time, making them less suitable for large systems.

2. Energy efficiency: Some consensus algorithms, such as Proof of Work, use a large amount of energy to confirm the correctness of execution results.

3. Implementation complexity: Blockchain is a relatively new technology, and its implementation will require specialists who will create new and adapt existing systems using blockchain technology.

4. Legal and regulatory issues: Blockchain, especially in financial sectors and those related to personal data, may face issues of compliance with existing laws and regulatory norms.

I would like to note that blockchain technology isn't always the best way to improve the fault tolerance of computer systems. However, the advantages that this technology provides can be widely used in various areas of human activity and make blockchain a standard tool for creating fault tolerant and secure systems. It's important to note that some countries are already making the first steps in using blockchain technology at the state level. For example, Estonia successfully uses the e-Estonia electronic citizenship system, which operates on the blockchain [4]. Through this system, elections are held, taxes are paid, and it helps the country save up to 2% of

GDP annually. But it's important to remember that the example of a country with a population of less than two million people cannot guarantee that a similar model will take root in other countries. However, it's impossible not to recognize that distributed digital registers are becoming increasingly popular and blockchain has the potential to radically change the world around us.

Conclusions. The potential of blockchain technology to enhance the fault tolerance of both distributed and centralized systems, including in high computational requirements, has been analyzed. Approaches and methods of using this technology in those systems where it can be most effective have been identified. Areas where the implementation of fault tolerant systems based on blockchain technology is most profitable have been identified.

References

1. Blockchain – Wikipedia [Electronic resource] – Access mode: https://en.wikipedia.org/wiki/Blockchain – Access date: June 2023

2. Fault tolerance – Wikipedia [Electronic resource] – Access mode: https://en.wikipedia.org/wiki/Fault_tolerance – Access date: June 2023

3. Private, Public, and Consortium Blockchains - What's the Difference? – Binance Academy [Electronic resource] – Access mode: https://academy.binance.com/en/articles/private-public-and-consortium-blockchains-whats-the-difference – Access date: June 2023

4. Blockchain: principle of operation, areas of application and prospects – mc.today [Electronic resource] – Access mode: https://mc.today/uk/blokchejn-printsip-roboti-sferi-zastosuvannya-ta-perspektivi/ – Access date: June 2023

AUTHORS

Volokyta Artem – associate professor, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: artem.volokita@kpi.ua

Serebriakov Roman – master, Department of Computer Engineering, NationalTechnical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: rserebryakov1902@gmail.com

РОЗШИРЕНА АНОТАЦІЯ

Р. О. Серебряков, А. М. Волокита

ПІДВИЩЕННЯ ВІДМОВОСТІЙКОСТІ СИСТЕМ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЇ БЛОКЧЕЙН

Актуальність теми дослідження. Технологія блокчейн, яка є основою децентралізованих цифрових валют, стає дуже популярною в наш час завдяки таким якостям як прозорість, незмінність, надійність, швидкість та ефективність. Тож вважаю за потрібне розглянути використання даної технології для підвищення надійності та відмовостійкості систем у різних сферах життєдіяльності.

Постановка проблеми. Можливість відшукати кращі способи підвищення надійності та відмовостійкості комп'ютерних систем.

Аналіз останніх досліджень та публікацій. Останні дослідження показують, що технологія блокчейн може бути дуже корисною для виконання задачі підвищення надійності і відмовостійкості систем завдяки децентралізації, прозорості у виконанні задач та безпековим характеристикам, таким як незмінність. Однак також існує думка, що для великих обчислень постає питання енергоефективності, особливо при виконанні консенсусних алгоритмів для підтвердження результату.

Виділення недосліджених частин загальної проблеми. Дана стаття присвячена вивченню та аналізу підходів до оптимізації використання технології блокчейну при великих обчислювальних вимогах, а також у розподіленого системах.

Постановка завдання. Проаналізувати доцільність використання технології блокчейн для підвищення відмовостійкості систем. Розглянути переваги та недоліки використання даної технології у централізованих та розподілених системах.

Викладення основного матеріалу. Проведено аналіз різних аспектів використання технології блокчейну в розподілених та централізованих системах, включаючи безпеку, відмовостійкість, прозорість, масштабованість та вартість використання, включаючи витрати на gas при виконанні програмного коду.

Висновки. Проаналізовано потенціал технології блокчейну для підвищення відмовостійкості як розподілених так і централізованих систем, в тому числі при великих обчислювальних вимогах. Визначено підходи та способи використання даної технології у тих системах, де це може бути найбільш ефективним. Визначено напрями, де впровадження відмовостійких системи заснованих на технології блокчейн є найбільш вигідним.

Ключові слова: блокчейн, відмовостійкі системи

UDK 004.8:004.9

Olena Savchuk, Artem Martyniuk

USAGE DIAGNOSTICS TO IMPROVE THE FAULT TOLERANCE OF THE IT STRUCTURE ELEMENT BASE

The article considers the idea of obtaining information about the technical conditions of the element base with the decomposition of the IT structure. Preliminary diagnosis of the components of the element base organizes neural network technologies to increase the fault tolerance of IT structures. Classification and sorting of components according to their technical condition was carried out in the MATLAB environment.

Key words: fault tolerance, neural network technologies, diagnosis, element base of the IT structure.

Fig.: 3. Tabl.:1. Bibl.: 13.

Target setting. The process of constant improvement and updating of the modern IT structure and its element base is stochastic in nature, therefore, when diagnosing the technical condition of the studied IT structure and/or its component, greater reliability is required, by which some data about the properties of the studied object can be determined.

Actual scientific researches and issues analysis. In connection with the development of neural network technology, new high-tech directions were developed in the theory of reliability and diagnostics of IT structures components [1], which contributed to increasing the fault tolerance of the latter.

Uninvestigated parts of general matters defining. Despite the significant number works devoted to the application of neural networks for the study of IT structures, the problem of increasing the fault tolerance of the element base of IT structures remains understudies. Moreover, it is necessary to conduct a separate study and a separate selection parameters for each type of element base.

The research objective. The purpose of the article is to choose the type of neural networks, methods of processing accumulated data to improve the quality and reliability of diagnostic results and their research on examples of the implementation of the process of EB diagnostics of the IT structure, including integrated microcircuits (ICs). For probabilistic reasons, predict processes in it that will lead to undesirable changes in the technical state of an element of the IT structure, and remove this element.

The statement of basic materials.

I. METODOLOGY FOR OBTAINING A PRIORY DIAGNOSTIC INFORMATION

For local diagnostic methods, instrumentally measurable information for the most part has the character of an image of the topographic distribution of certain properties of external environments. This allows not only to detect the presence of defects in the EB, but also to localize the location of the defects, indicating their size and orientation in space. The disadvantage of these methods is to increase the operational duration and complexity of the diagnostic process.

The emergence of non-linearity, inertia, fluctuations is related to the work processes of the investigated element to be diagnosed (IED), and has the same origin of activation. This means that the same sources of energy activation that ensure the operation of the IED during its operation are required for diagnostics. The short duration of operational diagnostics reduces its effectiveness. Diagnostic digital information often has various analog signatures [2]. Thus, it is possible to detect unfavorable malfunctions in the IED and determine its faulty operating state from the signature.

Observation of nonlinearity has specific measurement of ambiguous, since cannot a single. These are necessary functions, up the defining degrees, derivatives of higher ones. As for inertial, the corresponding inertial IEDs carried out the activation their input in the energy conversion and begin to operate. Transient integral characteristics reflect local macro-characteristics of the manifest and hidden impulse changes. The form of $\boldsymbol{g}(.)$ physical obtain during electrophysical transformation of electrical components, is an 1st a non-linear [3,4].

General model structure.

II. CONVERSION OF AN INPUT VECTORS SET FOR MICROCIRCUITS DIAGNOSTICS

The transformation of the input set of vectors into eigenvectors was carried out simultaneously by Karhunen and Loeve [5-7]. Let X be an n-dimensional random vector, then X can be exactly represented by a distribution

$$X = \sum_{l=1}^{n} y_i \Phi_i = \Phi Y_i \tag{1}$$

$$\Phi = [\Phi_1 \ldots \Phi_n] \tag{2}$$

$$Y = [y_1 \ldots y_n]^{\mathrm{T}} \tag{3}$$

The matrix Φ is deterministic and consists of n linearly independent column vectors:

$$\left|\Phi\right| \neq 0 . \tag{4}$$

Accordingly, linear combinations of the columns of the matrix Φ form an n-dimensional space that contains X. The columns of the matrix Φ are called basis vectors. These columns must be orthonormal, i.e

$$\Phi_i^T \Phi_j = \begin{cases} 1, i = j, \\ 0, i \neq j. \end{cases} \quad (5)$$

If the condition of orthonormality is fulfilled, then as follows:

$$yi = \Phi_i^T X, i = 1, \cdots, n. \quad (6)$$

If m ($m<n$), then following formula can be used to estimate the vector X (at the same time, the unknown components of Y are replaced by preselected constants):

$$\hat{X}(m) = \sum_{i=1}^{m} y_i \Phi_i + \sum_{i=m+1}^{n} b_i \Phi_i. \quad (7)$$

Without restriction of commonality, we can assume that only the first m components.

Each set of base vectors and values of constants corresponds to some value $\varepsilon 2(m)$. It is necessary to choose them in such a way as to minimize $\varepsilon^2(m)$. The optimal selection of bi constants is performed as follows:

$$b_i = E\{y_i\} = \Phi_i^T E\{X\}. \quad (8)$$

$$\Sigma_x \Phi_i = \lambda_i \Phi_i. \quad (9)$$

That is, the optimal basis vectors are the eigenvectors of the covariance matrix Σ_x. Thus, the minimum root mean square error is equal to

$$\varepsilon^{-2}(m)_{opt} = \sum_{i=m+1}^{n} \lambda_i \quad (10)$$

In pattern recognition tasks, the coefficients $y1,\ldots,yn$ of this distribution are considered as features representing the observed vector X.

These features have the following useful properties [7].

In other words, Σz and λ'_i are normalized eigenvalues. However, the transformation [7] must be justified from a physical point of view, since the statistical properties of the vector Z, including the covariance matrix, are completely different from the statistical properties of the vector X.

The study of nonlinearity was carried out for the integrated microcircuit (IMs) of the TDA2593 synchroprocessor. Dependences of nonlinearity on module and phase for IMs obtained by the method [3]. Dependencies on the components of cosine Fc [a2 (Uo)] and sine Fs [a2 (Uo)] are calculated according to the Karhunen-Loeve distribution (KLD) [4].

The number of required basic vectors m allows to determine the dimension of the space of features with a given decomposition error $\varepsilon^2(m)$. The base vectors themselves are the statistical characteristics of defective and suitable microcircuits, and the expansion coefficients are spatial coordinates.

For greater clarity, the Hilbert space is replaced by the Euclidean space. In the two-dimensional Euclidean orthonormal space (m = 2, $\varepsilon^2(m)$=8%, n = 100), intersections of the images of many suitable and defective chips, selected a priori based on expert data, were observed. At the same time, the KLD coefficients of suitable microcircuits were mainly located near a circle of unit radius, and defective ones shifted to the middle of the circle (Fig. 1, a).

When adding the third basis vector (m = 3, $\varepsilon^2(m)$ = 2%, n = 100), the orthonormal space of cosine vectors is transformed into a hemisphere on the left, and sine vectors on the right (Fig. 1, b). The points of the reflected suitable chips lie in the near-surface layer of a sphere of unit radius, and the reflections of different defective ones almost do not intersect.

When adding the third basis vector (m = 3, $\varepsilon^2(m)$ = 2%, n = 100), the orthonormal space of cosine vectors is transformed into a hemisphere on the left, and sine vectors on the right (Fig. 1, b). The points of the reflected suitable chips lie in the near-surface layer of a sphere of unit radius, and the reflections of different defective ones almost do not intersect. The decomposition of volumetric hemispheres made it possible to create a simple algorithm for recognizing serviceable and faulty IMs (Fig.1, c-e).

In reality, the space of non-linear basis vectors is an eight-dimensional ellipsoid. Displaying it in Euclidean space does not interfere with the natural classification of faulty and potentially unreliable microcircuits as small spheres in the middle of the main hemispheres (Fig. 1, b).

Experiments.

USAGE OF NEURAL NETWORKS FOR THE MICROSERCUITS CLASSIFICATION IN MATLAB

Processing of the complex of biharmonic influence was carried out on 164 IMs [3,4]. The analog signatures were transformed into discrete vectors, then processed by the KLD method. The following algorithms were selected for training MLP in the MATLAB software package with its library Neural Network Toolbox [8-10]: Bayesian regularization or learning a function based on backpropagation of the error using Bayesian regularization [12]; gradient descent backpropagation method; back propagation; combined iterations; elastic backpropagation or inverse elastic

distribution. The results of learning neural networks: MLP, self-organized map, RBF networks are given below. International types errors - MSE, MAE, MSEREG, MAE - were used as a criterion for assessing the accuracy of training [11]

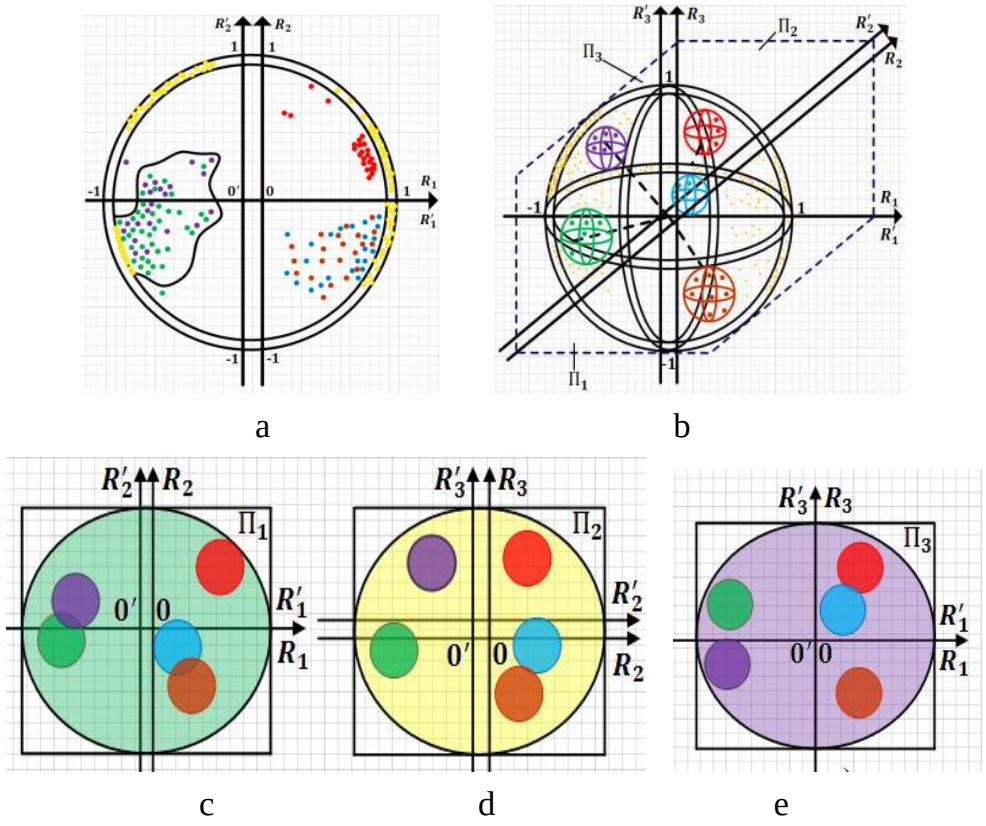
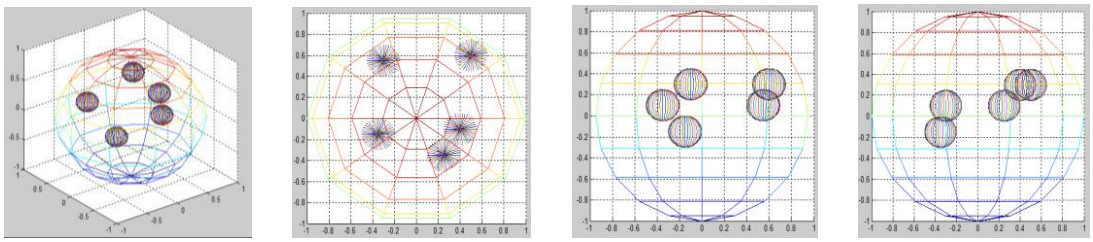


Fig. 1. Geometric interpretation of KLD for microcircuits [4]: a) in two-dimensional orthonormal space of vectors; b) in three-dimensional orthonormal space of vectors; c) the projection of this space onto P1 in the two-dimensional Euclidean orthonormal space (on the left – for the cosine component, on the right – for the sine component); d) projection on P2 (with overlapping axes for cosine and sine components), e) projection of this space on P3.

Table1. **Training a Multiplayer Perceptron for IMs**

Number of classes	**Educational function trainbr**					
	logsig- logsig- logsig					
	10E-5				***Epochs, unit***	***Learning time, s***
	MSE	***SSE***	***MSEREG***	***MAE***		
2	1.4	4200	1.8	1100	52	24
3	3	8000	3	2100	62	33

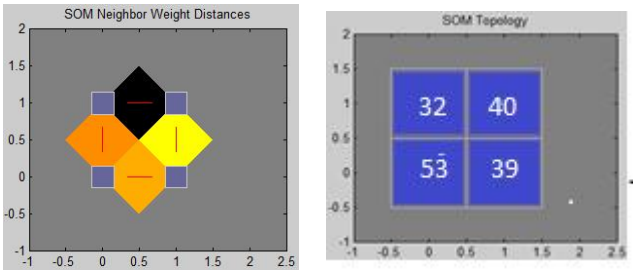
The most important advantages of probabilistic networks are that they are faster and easier to train than backpropagation networks. When training a PNN network, time is spent almost at the input. The disadvantage of the network is its size, since it actually contains all the training data and requires a lot of memory, which can reduce speed of operations [11-13].



a) b) c) d)

Fig. 2. The results of RBF network training: a) spatial image of the elements of the obtained KLD matrices for microcircuits; b) projection of the received KLD matrices for microcircuits onto the XY plane; c) onto the YZ plane; d) onto the XZ plane.

When using the Kohonen map, the neurons of the active sphere have a regular structure. Such maps are often used for clustering graphic images and audio signals, as well as for rich information processing. The map is implemented as an announcement (Fig. 3).The best reliability of the microcircuits' classification (83,7%) can be achieved for gridtop topology with mandist distances and steps=30. And the best result of chip rejection is provided by MLP and RBF neural networks (pnn), when trained with a teacher (> 90%).



a) d)

Fig. 3. Topology gridtop: a) the principle of forming distances between adjacent clusters: b) classification of samples with mandist distances.

Conclusions. The task of fault tolerance increasing using improving of electro physical methods of diagnostics with neural technologies was solved.

The Karhunen-Loeve distribution for information transformation provides a simple algorithm for learning and recognizing the technical and physical state of IT structure components using neural networks in the MATLAB environment.

The observed information signs increase the diagnostic capabilities of technical methods for diagnosing the element base of an IT structure in identifying hidden defects, potential instability and unplanned degradation processes.

References.

1. Rolik, O.I., Telenyk, C.F., Yasochka, M.V. Corporate IT – infrastructure management // K.: Naukova Dumka, 2018. – 576.

2. Volodarskyi, E.T., Kuharchuk, V.V. , Podzharenko, V.O., Serdjuk, G.B. Metrological Support for Measuring and Checking / Textbook. – Vinnytsia, Veles, 2001. – 219.

3. Usatenko, V.G. Serdjuk, G.B., Sazonov, S.M. Unrealible and Unstable Microcircuits' regection method. Patent N1056 for an invention (Ukraine). – G 01 R 31/27. – 1993

4. Telenyk, S., Savchuk, O., Pocrovskyi, E., Morgal, O., Krivenko, K., Latash, I. On Searching of Type and Optimizing of Neural Network in Electric Component Diagnosis // Proceedings of the IV Congress on «Automatic Control and Information Technology», (ICACIT 2017), Cracow, Poland, 14 – 16 of December, 2017, pp.147 – 159.

5. Castrillon-Candas, J., Amaratunga, K. – Fast estimation of continuous Karhunen-Loeve eingenfunctions using wavelets. – IEEE: IEEE Signal Processing Society. – 2002. – .78-86.

6. Castrillon-Candas, J.E., Jun Li, Eijkhout, V. A discrete adapted hierarchical basis solver for radial basis function interpolation, BIT Numerical Mathematics, Springer, July 2012.

7. Fucunaga, K. Introdaction to Pattern Recognition Statistical Theory. M.: The Science, 1979. – 368.

8. Dyakonov, V., Kruglov, V. MATLAB Math Expantion Packs. Special Handbook. – Spb: Peter, 2001. – 480.

9. Yakimenko, O. Engineering computations and modeling in MATLAB/Simulink. – Monterey, California: American Institute of Aeronautics and Astronautics, INC., 2011. – 938.

10. Haikin, S. Neural Networks: Full Course.2nd edition. –M.: Williams, 2006. – 104.

11. Savchuk, Olena, Morgal, Oleg. On the Neural Network Technologies Application in Diagnosing of the IT-Structure Element Base. In: *2022 IEEE 4th International Conference on Advanced Trends in Information Theory (ATIT)*. IEEE, 2022. p. 234-237, doi: 10.1109/ATIT58178.2022.10024227.

12. Pysarenko, A., Tyshenko, D. The Model of Vehicle Diagnostic Subsystem Based of Bayesian Network / IV International Scientifical and Practical Conference Proceedings on Information Systems and Technologies, Summer Infocom'2017. Kyiv, June 1-2, 2017. – K.: Engineering, 2017, 62 – 65.

13. Shymkovych, V., Telenyk, S., Kravets, P. Hardware implementation of radial neural networks with Gaussian activation, functions on FPGA/ Neural Computing and Applications, 33, 2021, 9467 – 9479. https://doi.org/10.1007/s00521-021-05706-3.

AUTHORS

Savchuk Olena – associate professor, Department of Information Systems and Technologies, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". E-mail: savchuk_l1@ukr.net, elenasavchuk0@gmail.com

Martyniuk Artem – student, Department of Information Systems and Technologies National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" E –mail: martuniukartem1123@gmail.com

РОЗШИРЕНА АНОТАЦІЯ

Олена Савчук, Артем Мартинюк

ВИКОРИСТАННЯ ДІАГНОСТИКИ ДЛЯ ПОКРАЩЕННЯ ВІДМОВОСТІЙКОСТІ ЕЛЕМЕНТНОЇ БАЗИ ІТ-СТРУКТУРИ

Актуальність теми дослідження. Процес постійного вдосконалення та оновлення сучасної ІТ-структури та її елементної бази (ЕБ) має стохастичний характер, тому при діагностиці технічного стану досліджуваної ІТ-структури та/або її компонента необхідна більша достовірність, за допомогою якої деякі дані про властивості досліджуваного об'єкта можна визначити.

Актуальні наукові дослідження та аналіз проблем. У зв’язку з розвитком нейромережевих технологій отримали розвиток нові високотехнологічні напрямки в теорії надійності та діагностики компонентів ІТ-структури, що сприяло підвищенню відмовостійкості останніх.

Постановка проблеми. 1. Обумовлювати в даний час і ідентифікувати внутрішній стан техніки під час виконання. 2. Відстеження місцезнаходження для несправного стану. 3. Виконати діагностику базового елемента ІТ-структури. Використати нейронні мережі для класифікації дослідженого елемента. 4. З ймовірнісних міркувань передбачити процеси в елементі ІТ-структури, які призведуть до небажаних змін технічного стану.

Виділення недосліджених частин загальної проблеми. Незважаючи на значну кількість робіт, присвячених застосуванню нейронних мереж для дослідження ІТ-структур, проблема підвищення відмовостійкості елементної бази ІТ-структур залишається недостатньо вивченою. Причому для кожного типу елементної бази необхідно проводити окреме дослідження і окремий підбір параметрів.

Постановка завдання. Вибір типу нейронних мереж, методів обробки накопичених даних для підвищення якості та достовірності результатів діагностики та їх дослідження на прикладах реалізації процесу діагностики елементної бази ІТ-структури, в тому числі інтегральних мікросхем (ІМС). З ймовірнісних міркувань спрогнозувати в ньому процеси, які призведуть до небажаних змін технічного стану елемента ІТ-структури, і видалити цей елемент, що сприяє підвищенню відмовостійкості елементної бази і всієї ІТ–структури.

Викладання основного матеріалу. Вирішено завдання підвищення відмовостійкості шляхом вдосконалення електрофізичних методів діагностики

нейротехнологіями. Пропоновано розподіл Карунена-Лоева для перетворення та стиснення діагностичної інформації , що забезпечує простий алгоритм для вивчення та розпізнавання технічного та фізичного стану компонентів ІТ-структури за допомогою нейронних мереж у середовищі MATLAB.

Спостережувані інформаційні ознаки підвищують діагностичні можливості технічних методів діагностики елементної бази ІТ-структури щодо виявлення прихованих дефектів, потенційної нестабільності та незапланованих процесів деградації.

Висновки. Розглядається ідея отримання діагностичної інформації елементної бази ІТ-структури. Для підвищення надійності елементів ІТ-структури пропонується попередньо сортувати компоненти елементної бази за технічним станом. Проведено класифікацію та сортування компонентів за технічним станом у середовищі MATLAB з використанням нейромережевої технології, що сприяє подальшому підвищенню відмовостійкості ІТ структури в цілому.

Ключові слова: відмовостійкість, нейромережеві технології, діагностування, елементна база ІТ-структури.

Рис.:3. Табл.:1. Бібл.:13.

Parallel Section RT. IoT, Real Time Systems.

UDC 004.383

Anatoly Sergiyenko, Anastasia Molchanova, Ivan Mozghoviy

METHOD OF MAPPING CYCLO-DYNAMIC DATA FLOWS INTO HARDWARE

The article focuses on the relevance of high-level synthesis (HLS) systems used for designing pipelined datapaths. The goal is to explore methods of mapping algorithms to the pipelined datapaths implementing the cyclic data flow graphs with dynamic schedules. The proposed method involves creating and optimizing cyclo-dynamic data flow graphs, describing them in VHDL. The method demonstrates its effectiveness through examples like run-length encoding decompression and can be implemented in HLS tools.

Keywords: data flow graph, field programable gate array, VHDL, pipeline, dynamic schedule.

Fig.: 3. Tabl.: 1. Bibl.:12.

Introduction. The high-level synthesis (HLS) systems are increasingly distributed by companies producing the CAD tools for integral circuits and FPGAs. They are intended for the computer-aided design of hardware devices that execute algorithms described in a high-level programming language such as C in parallel. Their use makes it possible to speed up the design process tenfold. But HLS still do not provide a decent minimization of hardware costs of synthesized pipelined datapaths in comparison with the manual design. Therefore, it is necessary to search for new methods of mapping algorithms into hardware computing devices. Existing methods of mapping the data flow graphs of various types have found application in the program compilation, but they are rarely used in hardware design.

In this work, the known methods of the pipelined datapath design are considered. These methods analysis make it possible to select the approach to create a method of designing the pipelined datapaths, which is focused on the execution of algorithms based on the cyclic data flow graphs with a dynamic schedule. The proposed method consists in mapping a cyclo-dynamic data flow graph into the datapath with finite state machine (FSMD) which is described in the VHDL language.

Methods of pipelined datapath design and dataflow graphs. A typical method of the datapath design consists in describing the algorithm with a dataflow graph (DFG) and control flow graph, and mapping them into hardware. Such a mapping consists in the sequential execution of three stages: resource selection, operation scheduling, and operation assignment. The synthesis is finished by the

deriving the interconnection scheme and the finite state machine (FSM) design [1]. Since this method synthesizes both the datapath and FSM controlling it, it is often called the FSMD method [2]. But the quality of the resulting device depends significantly on the performance of the stages of synthesis, each of which has a different goal. Although the method is used to implement a large set of algorithms, it is not directly adapted to the synthesis of pipelined datapaths with high throughput.

For the synthesis of pipelined computers, the method of mapping the synchronous data flow graphs (SDF) has become widespread. Such a graph consists of operator nodes and directed edges connecting them. It is considered that the operator in the node is executed immediately (fired for execution) as soon as data (tokens) appear at its inputs, and it outputs the results in the output edges. An edge serves as a dataflow and has a buffer to store the data. Most often, this is a FIFO buffer. It is represented by thick dashes across an edge that correspond to register delays. DFG is synchronous, i.e., SDF, if there is a one-to-one correspondence between the data in the dataflows, for example, they have indices, which depend on the iteration number. Therefore, the execution of the algorithm on SDF has a constant period during which each node consumes and generates the same number of tokens [3]. Because of this, SDF is classified as a statically scheduled dataflow graph (SSDF)[4].

In a single-rate SDF, inputs and outputs of nodes consume and produce the same number of tokens during the calculation period. Therefore, it is not difficult to map a a single-rate SDF into a pipelined datapath that executes a given algorithm with a period of one cycle. By such a mapping, the nodes correspond to the logical circuits that calculate operators and edges do the communication lines, their register delays do the pipeline registers [5]. The representation of SDF in a multidimensional space makes it possible to formally design the pipelined datapaths with a given period of algorithm execution [6]. The SDF use is limited by the set of algorithms which nodes execute the same operations in each period.

The cyclo-static dataflow (CSDF) considers that the actors can have different number of executed tokens in different firings, but the amount of these tokens in a single cycle is stable. Therefore, such a model provides the static schedule [7].

The parametrized SDF (PSDF) is more general, sofisticated, and impressive model of the cyclic algorithms. It considers that the nodes can perform a set of different operations, which can be switched depending on the configuration of tokens in the node inputs. Moreover, the graph can be hierarchical one. But CSDF, SSDF and PSDF are practically used only in the automatic programming but not in the hardware design [4,8].

There is a wide class of cyclic algorithms which could not be represented by SDF, CSDF or SSDF. They distinguished in that the algorithm period depends on the data which it executes. For example it is the compression algorithm, each output code calculation period is variable and strongly depends on the data. Therefore, such an algorithm has the dynamic schedule and the respective computation model is named as cyclo-dynamic dataflow (CDDF) [9]. The hardware implementation of such an algorithm is performed usually by the FSMD method, and therefore, it is complex and often ineffective. Consider the design of a new method of mapping CDDF into the pipelined datapath.

Prerequisites for the method creation. The method is intended for the design of the pipelined datapaths performed in FPGA. So, it must take into consideration the FPGA architecture features like the operation hardware execution and the dataflow performing. On the other hand, the CDDF model arrangement must provide such algorithm shedule which assures both correct hardware implementation and deadlock absence.

The hypothesis is that CDDF can be mapped into the pipelined datapath as the homogeneous SDF can. Such a mapping is possible when a set of conditions is satisfied. Firstly, the necessary conditions of CDDF to be deterministic and free of deadlocks must be satisfied [9]. They are the following.

1) A set of values of the control token, which infer the dynamic behavior, must be limited, this token must be present in the same phase (iteration) in which it is used to determine which phase should be executed, the executed phase must not depend from the value or index of the input datum [7].

2) The graph should not have cycles of dependencies without any delay in the edges [7]. It should be noted that by this condition, there are no dependency cycles in the corresponding data dependency graph and, accordingly, such a graph gives structural solutions without blocking. This is also a condition for the absence of a loop in the corresponding combination scheme, which results the latch [9].

3) All the delays which load the edges must have the initial data or tokens, and this condition is named as the live cycle condition [10].

4) CDDF operates in cycles, each of them executes different number of iterations. The number of consumed and generated tokens by any node in each iteration must be stable as in the homogeneous SDF [7, 10].

5) After execution of a single cycle, CDDF must return in the initial state of this cycle. This assures that CDDF is period safe [10].

6) FSM, which generates the control tokens, must not be deadlocked.

Each CDDF node is mapped into respective logic scheme. And the dataflow represented by an edge weighted by a FIFO delay is mapped into respective register chain or FIFO buffer. When the resuling structure is described by some hardware description language like VHDL the following conditions must be satisfied.

1) The logic scheme which is described in the VHDL language using the process operator, must use the IF-THEN-ELSE and CASE logical operators. And condition that deadlocks do not occur is that the ELSE alternative and all alternative branches of the CASE statement are enabled. The similar features have the WHEN-ELSE and WITH-SELECT operators as well.

2) The dataflow is described in the VHDL language as a process that is triggered by the edge of a common clock signal, in which assignments are made to the signals that mark the FIFO registers.

To describe CDDF and for its convenient perception, each node together with the edges coming out of are described by one process operator. However, a set of such process operators can be combined in a single process operator [11].

For the convenience of compiling the algorithm, the CDDF elements have a symbolic representation, the examples of which are shown in Fig. 1. So, the counter is set using the node of incrementing and register delay as in Fig. 1,a. At the node inputs the enable, initialization control data as well the data stored in the register delay are inputted. The similar counter but with the separate output edge without a delay is shown in Fig. 1, b.

a) b)

Fig. 1. Counter model with output from the register (a) and from the node (b)

If the random access memory (RAM) is the mapping target then the specific subgraph of CDDF is related to it. The Block RAM (BRAM), which is used in FPGA, has a storage as the register set drive, writing data register, write and read address registers, and respective write address decoder and read data multiplexor. The corresponding subgraph has a node for data writing (MW), a storage and a reading data node (MR) (Fig. 2). Nodes MW, MR have the data D and addresses A inputs. The storage itself is

depicted by a long bar. The edge that passes through the storage is also thickened because it characterizes the *n* data buses that are attached to the *n* registers of the storage.

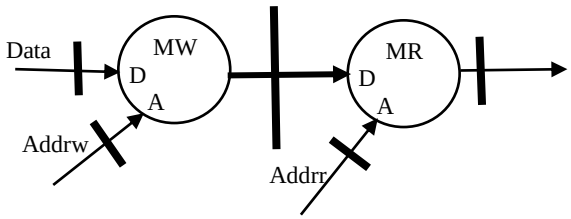


Fig. 2. Subgraph of BRAM

The subgraph of FSM consists of the node which forms the next state (NS) based on the input signal X, the output state (OS) node, which forms the output data Y, and the state register (ST) which loads the edge connecting both nodes (Fig.3).

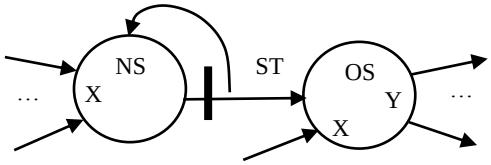


Fig. 3. Subgraph of FSM

CDDF Optimization. When synthesizing pipeline datapath, SDF is optimized by shortening the critical path in the corresponding datapath circuit, and then, optimized SDF is mapped to the datapath. The retiming is recognized as an universal method of optimizing SDF, and it consists in such a permutation of delays in edges that does not disrupt the general execution of the algorithm. The pipelining method is most often used for SDFs, according to which the same number of delays is inserted into the edges, which are directed in the same direction relative to the graph intersection. The pipelining is similar to the retiming in that it shortens the critical path. But at the same time, the latent delay of the algorithm increases [12].

Likewise, CDDF can be optimized using the retiming and pipelining methods.

Method of mapping CDDF to the pipelined datapath. The method is intended for the design of pipelined datapaths, which are configured in FPGAs with a period of one cycle.

Initial data for design are:

— a dataflow algorithm that is represented as CDDF and that can use access to single or multi-port memory, which has the control part like FSM;

— optimality criterion t_C as the minimum period of the clock interval;

— a library of FPGA elements of a certain series, which includes registers, adders, BRAM with specific delays.

Design results: the device description in VHDL or Verilog, which is ready for synthesis and further configuration in FPGA.

1. Representation of the algorithm in the form of CDDF. The functions performed by the logic circuits are represented by the corresponding nodes. The data transfer between the nodes along with the corresponding delays for the required number of clock cycles are represented by edges loaded by the respective delays. The functions of storing data into BRAM and reading these data are represented by the subgraph like one in Fig. 2. The control FSM is represented by the subraph like one in Fig. 3. The derived CDDF must satisfy the necessary conditions to be deterministic and free of deadlocks mentioned above.

2. Optimization of CDDF using pipelining and retiming. The goal of optimization is to minimize the value of t_C. It is equal to the critical path in the CDDF.

3. Mapping the optimized CDDF to the hardware. At the same time, nodes with outputted edges incident to them are described by VHDL language process operators or Verilog language Always constructs. The FSM is described as the separate process. The resulting VHDL or Verilog program is a description of the functional scheme at the level of register transfers of a pipeline computer that executes a given algorithm with a period of one cycle, which is minimized in terms of duration.

The following should be taken into account during the optimization of CDDF. The critical path t_C in CDDF is the maximum path delay. It is determined as a sum of delays in the logic circutits which are relevant to the nodes that belong to the path between two edges loaded by registered delays:

$$t_C = \max \sum_i t_{Pi}, \quad (1)$$

where t_{Pi} is the delay of the i-th node of the P-th type which belongs to the considered path. It should also be noted that in modern FPGAs, the share of delay in the interconnection lines reaches 60-90%, and the delay in logic circuits accounts for 10-40% of the total delay, respectively. Therefore, the final decision to obtain an optimized project should be made after processing the VHDL file with a compiler-synthesizer, placer and router of the FPGA CAD tool.⇒

Synthesis example. The run-length encoding (RLE) decompressor is a simle example of the module which executes the CDDF algorithm. Consider a bit sequence 11100001100. Then, it is encoded by the RLE algorithm as a code sequence 3422. The decompression algorithm must be performed cyclically, i.e., for each input symbol, a sequence of zeros or ones of the appropriate length is generated. Thus, each

cycle has a variable number of iterations that depends on the data. So, the algorithm can be represented by CDDF. The algorithm is written in C language as follows.

```
if (start){
pw = 0;    //pointer to write in a buffer
while (~eos){ //  main cycle finishes by the end-of-sequence signal
                    while (pw ≠ N) { // cycle of writing  codes in a buffer
                                B(pw) = yi;
                                pw++;
                    }
                    pr = 0;  //pointer to read from a buffer
                    fl = 0;   // flag of the bit value
          while (pr ≠ N) { cycle of reading codes from a buffer
                                c = B(pr);
                               while (c ≠ 0){  // iterations of the bit sequence generation
                                          c--;
                                          xj = fl;
                                }
                                fl = ~ fl;
                                pr++;
                    }
                    pr = 0;
}
```

The input codes yi are written to the buffer memory B(pw) at the pointer pw address in the write cycle. The resulting sequence xj is generated in the cycle of reading from the buffer memory codes yi at the pointer pr address. Both cycles are relatively independent and can run in parallel until the eos signal arrives.

When designing the pipelined datapath for computing the sequences of the undefined length, the buffer memory B must be organized as FIFO implemented as a circular buffer. Then, the FIFO depth and code yi tracking period must be such that the buffer memory does not overflow. When FIFO is implemented in BRAM, then its volume is selected as $M = 2^{nb}$, where *nb* is the memory address bit width. Then, the writing pointer state pw must outrun the reading pointer state pr at least to the value of *N*, i.e., $pw - pr \geq N$. Note, that both pointers are incremented modulo *M*.

The corresponding CDDF is shown in Fig.4. The FSM diagram of the decompressor is shown in Fig. 5.

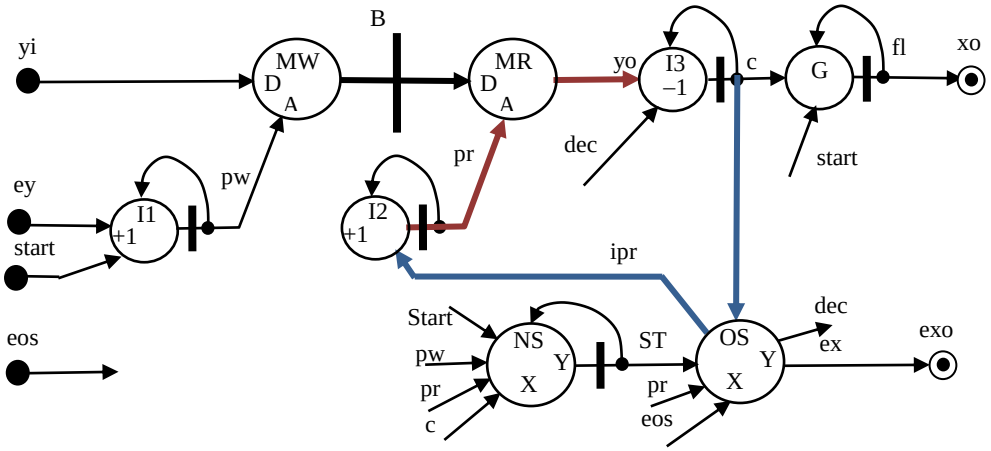


Fig. 4. CDDF of the decompressor

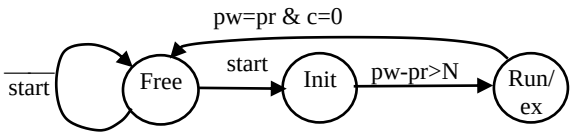


Fig. 5. FSM diagram

Here, the bold points and marked bold points represent the input and output nodes, nodes MW, MR with the storage B form the buffer memory, I1,I2 are the incrementer nodes of the pointers pw and pr, node I3 is the incrementer for deriving code *c* of the running length, node G generates the output sequence xo, nodes NL, OL with the state register ST form FSM. One can see that all conditions of the CDDF correctness are satisfied.

The alternative critical paths in Fig. 4 are shown in bold color lines. So, according to (1) the minimum clock period is equal to

$$t_C = \max((t_{MR} + t_{I3}),\ (t_{OS} + t_{I2})), \qquad (2)$$

where t_{MR}, t_{I3}, t_{OS}, t_{I2} are delays of the logic circuits implemented in the respective nodes.

The FSM diagram (Fig.5) has the idle state Free, initialization state Init and the operation state Run. In the state Init the buffer memory loads N input data. In the state Run along with loading next input data the codes yi are read from the buffer and loaded to the register c. After that the code c is decremented yi times which derives the length of the output sequence of equal bits. The flag trigger fl is inverted each time when the code c is zeroed.

The optimization of CDDF consists in using the methods of retiming and pipelining. The goal is to add the pipelined registers at the inputs and outputs of CDDF and exchange the delay position for getting the subraph as in Fig. 2 which is mapped into BRAM and trying to minimize the critical path.

The resulting optimized CDDF is shown in Fig. 6. Its description in VHDL is the following.

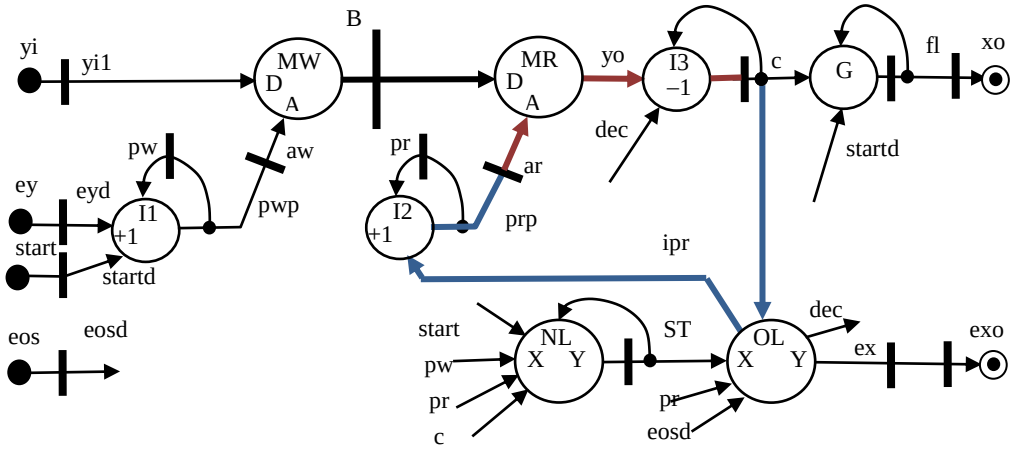


Fig.6. Optimized CDDF

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.Numeric_STD.all;
entity RLDecompressor is generic(nb: natural:= 9; -- address bit width
                N:natural:=256);      -- FIFO deep
        port(
                CLK : in STD_LOGIC;
                RST : in STD_LOGIC;
                EY : in STD_LOGIC;                -- input data enable
                START : in STD_LOGIC; -- start of the input sequence
                EOS : in STD_LOGIC;               -- end of input sequence
                YI : in STD_LOGIC_VECTOR(7 downto 0); -- input sequence
                XO : out STD_LOGIC;               -- output sequence
                EXO : out STD_LOGIC               -- enable of output sequence
                );
end RLDecompressor;
architecture synt of RLDecompressor is
        type ram_type is array (0 to 2**nb-1) of unsigned(7 downto 0);
        signal B : ram_type:=(others=>x"01");
        type state is (Free,Init, Run);
        signal ST: state;
```

```
        signal yi1,yo,c:unsigned(7 downto 0);
        signal pw,aw,pr,ar,pwp,prp:unsigned(nb-1 downto 0);
        signal eosd,startd,ipr,eyd,fl,dec,decd,ex: std_logic;
begin
        REGS:process(CLK,RST) begin  -- register set
                if RST='1' then
                        eyd<= '0';      startd<='0';    eosd<='0';
                        pw<=(others=>'0');  pr<=(others=>'0');
                        fl<='0';        decd<='0';      c<=x"01";  EXO<='0';
                elsif rising_edge(CLK) then
                        pw<=pwp;    pr<=prp;        decd<=dec;
                        eyd<=ey;        eosd<=eos;    EXO<=ex;
                        if dec ='1' and ST=Run then
                                c<= c - 1;
                        elsif c = 0 and ipr='1'  then
                                c<=yo;
                        end if;
                        if STARTd = '1' then
                                fl<='0';
                        elsif c = 0 then
                                fl<= not fl;
                        end if;
                        XO<=fl;
                end if;
        end process;
        I1:     pwp <=(others=>'0') when startd ='1' else     -- node I1
                        pw + 1 when eyd ='1' else pw;
        I2:     prp <=  pr + 1 when ipr ='1' else pr;                -- node I2

        FSM:process(CLK,RST,pw,pr,eosd,c) begin  -- finite state machine
                if RST='1' then
                        ST<=Free;                       -- state register
ex<='0';
                elsif rising_edge(CLK) then
```

```
                if ipr='1' then
                        ex<='1';
                end if;
                case ST is
                        when Free => if START='1' then
                                        ST<=Init;
                                end if;
                        when Init => if pw-pr>N     then
                                        ST<= Run;
                                end if;
                        when Run => if  pr = pw    and c=0 then
                                        ST<=Free;
                                        ex<='0';
                                end if;
                        when others => null;
                end case;
        end if;
        dec<= '0';      ipr<= '0';
        if ST = Run    then
                if (( pr(pr'high) = '1' and pw(pw'high) ='0' and pw + not pr >N) or
                        (pr(pr'high) = '0' and pw-pr>N) or eosd = '1')
                        and c=0  then
                        ipr<= '1';
                elsif c/=0 then
                        dec<='1';
                end if;
        end if;
end process;

RAMB:process (CLK)  begin                       -- circular buffer
        if clk'event and clk = '1' then
                aw<=  pwp;   ar <=   prp;    yi1<= unsigned(YI);
                if eyd ='1' then
                        B(to_integer(aw)) <= yi1;    -- writng data
```

```
                    end if;
                end if;
        end process;
        yo<= B(to_integer(ar));                    -- reading data from the buffer
end synt;
```

The CDDF before the optimization was described in VHDL as well. Both projects were compiled by the Xilinx ISE tool. The results of compiling, placing and routing are shown in Table 1.

Table 1. **Features of the RLE decompressor configured in Xilinx Virtex-7 FPGA**

Decompressor	**Hardware volume**				**Maximum clock frequency, MHz**
	LUTs	**Triggers**	**CLBS**	**18k BRAMs**	
Before optimization	219	37	78	0	358
After optimization	132	53	53	1	332

These results show the following. The decompressor before the optimization has the FIFO buffer implemented on the base of look-up tables (LUTs) which are configured as the distributed RAM. This is the cause of increased hardware volume of the unoptimized decompressor. Such a RAM has less delay in the MR node which provides in 8% higher maximum clock frequency. The optimized decompressor contains one BRAM module, which provides the hardware minimization in 70% in LUTs and in 47% in the configurable logic block slices (CLBS).

So, the use of the proposed method simplifies the design process of the pipelined datapaths, comparing to the most popular method of FSMD design. It optimizes the project both in clock frequency and in hardware volume, providing the use of the specific blocks of FPGA like BRAM.

Conclusions

The graph models of data flow processing algorithms are analyzed and a class of cyclo-dynamic data flow graphs is selected. A method of designing the pipelined datapath that perform cyclic algorithms with a dynamic schedule is proposed. The

results of designing a lossless decompression device using this method are given. The method can be used manually as well as be implemented in the HLS systems.

References

1. Gajski D. D., Abdi S., Gerstlauer A., Schirner G. (2009). *Embedded System Design. Modeling, Synthesis and Verification.* Springer. 352 p.

2. Schaumont P. (2011). *A Practical Introduction to Hardware/Software Codesign*. Springer. 396 p.

3. Lee E. A., Messerschmitt D. G. (1987). *Synchronous data flow*. in Proceedings of the IEEE, vol. 75, no. 9, pp. 1235-1245, Sept. 1987, doi: 10.1109/PROC.1987.13876.

4. Lee E. A, Neuendorffer S. (2005). *Concurrent models of computation for embedded software.* IEE-INST ELEC ENG. IEE Proceedings – Computers and Digital Techniques, Vol. 152. No. 2, pp. 239-250.

5. Khan S. A. (2011). *Digital Design of Signal Processing Systems. A Practical Approach*. UK: Wiley.

6. Sergiyenko A., Serhienko A., Simonenko A. (2017). *A method for synchronous dataflow retiming.* 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kyiv, Ukraine, april 2017, pp. 1015-1018, doi: 10.1109/UKRCON.2017.8100404.

7. Parks T. M., Pino J. L., Lee E. A. (1995). *A comparison of synchronous and cycle-static dataflow*. Conference Record of The Twenty-Ninth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, pp. 204-210 vol.1, doi: 10.1109/ACSSC.1995.540541.

8. Bhattacharya B., Bhattacharyya S. (2001). *Parameterized dataflow modeling for DSP systems*. IEEE Transactions on Signal Processing. V. 49. No. 10, pp. 2408–2421.

9. Wauters P., Engels M., Lauwereins R., Peperstraete J. A. (1996). *Cyclo-dynamic dataflow*. Proceedings of 4th Euromicro Workshop on Parallel and Distributed Processing, Braga, Portugal, 1996, pp. 319-326, doi: 10.1109/EMPDP.1996.500603.

10. Fradet P., Girault A., Poplavko P. (2012). SPDF: *A schedulable parametric data-flow MoC*. Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 2012, pp. 769-774, doi: 10.1109/DATE.2012.6176572.

11. Сергиенко А. М. (2004). *VHDL для проектирования вычислительных устройств.* Киев: Диасофт. 205 с.

12. Woods R., McAllister J., Lightbody G., Yi Y. FPGA-based Implementation of Signal Processing Systems. Wiley, 2d Ed. 2017, 447 p.

AUTHORS

Anatoly Mykhailovych Sergiyenko, Dr. Tech. Sciences, Senior Scientist, professor of the Computer Engineering Department of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: asera@comsys.kpi.ua

Molchanova Anastasia Anatoliivna, assistant lecturer of the Computer Engineering Department of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

Ivan Mozghoviy, postgraduate of Computer Engineering Department of National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: mozg.v34@gmail.com

РОЗШИРЕНА АНОТАЦІЯ

А. М. Сергієнко, А. А. Молчанова, І. В. Мозговий

МЕТОД ВІДОБРАЖЕННЯ В АПАРАТУРУ АЛГОРИТМІВ ОБРОБЛЕННЯ ЦИКЛІЧНИХ ПОТОКІВ ДАНИХ З ДИНАМІЧНИМ РОЗКЛАДОМ

Актуальність теми дослідження. Системи високорівневого синтезу (СВС) все більше поширюються фірмами-виробниками засобів САПР мікросхем та програмовних логічних інтегральних схем (ПЛІС). Вони призначені для автоматизованого проєктування апаратних пристроїв, які паралельно виконують алгоритми, що описані високорівневою мовою програмування такою як С. Їх використання дає змогу у десятки разів прискорити процес проєктування.

Постановка проблеми. СВС досі не забезпечують гідної мінімізації апаратних витрат синтезованих конвеєрних обчислювачів у порівнянні з ручним проєктуванням. Тому необхідний пошук нових методів відображення алгоритмів у апаратні обчислювальні засоби. Існуючі методи відображення графів потоків даних різних видів знайшли застосування у автоматизованому програмуванні, але вони рідко використовуються у проєктуванні апаратури.

Аналіз останніх досліджень і публікацій. Найбільш поширеним тепер методом проєктування апаратних пристроїв для обробки потоків даних є метод синтезу блоку обробки даних з керуючим автоматом, який має труднощі з автоматизацією через низький рівень формалізації. Існуючі формалізовані методи синтезу конвеєрних обчислювачів орієнтовані на вузький клас алгоритмів обробки синхронних потоків даних зі статичним розкладом.

Виділення недосліджених частин загальної проблеми. Є широкий клас алгоритмів, які представляються графами циклічних потоків даних з динамічним розкладом, які ще недостатньо досліджені з боку удосконалення їх відображення у конвеєрні апаратні засоби і впровадження у СВС.

Постановка завдання. Завданням є створити метод проєктування конвеєрних обчислювачів, які орієнтовані на виконання алгоритмів оброблення циклічних потоків даних з динамічним розкладом.

Викладення основного матеріалу. Новий метод полягає у створенні графу циклічних потоків даних з динамічним розкладом, його оптимізації та описі мовою VHDL стилем для синтезу. Приклад синтезу декомпресора

кодування довжини рядків повторюваних символів показує дієвість та ефективність методу. Також показані результати синтезу декомпресора за алгоритмом LZW та їх порівняння з іншими пристроями.

Висновки. Проаналізовані існуючі графові моделі представлення алгоритмів оброблення потоків даних і виділено клас графів циклічних потоків даних з динамічним розкладом. Запропоновано метод проєктування конвеєрних обчислювачів, що виконують циклічні алгоритми з динамічним розкладом. Наведені результати проєктування за допомогою цього метода пристрою для безвтратної декомпресії. Метод може бути впроваджений у СВС.

Ключові слова: граф потоків даних, програмовна логічна інтегральна схема, VHDL, конвеєр, динамічний розклад.

Relevance of the research topic. Systems of high-level synthesis (HLS) are increasingly distributed by companies producing CAD tools for integral circuits and FPGAs. They are intended for the computer-aided design of hardware devices that execute algorithms described in a high-level programming language such as C in parallel. Their use makes it possible to speed up the design process tenfold.

Formulation of the problem. HLS still do not provide a decent minimization of hardware costs of synthesized pipeline computers in comparison with manual design. Therefore, it is necessary to search for new methods of mapping algorithms to hardware computing devices. Existing methods of displaying data flow graphs of various types have found application in automated programming, but they are rarely used in hardware design.

Analysis of recent research and publications. The most common method of designing hardware devices for data flow processing is the method of synthesizing a datapath with FSM, which has difficulties with its implementation due to a low level of its formalization. The existing formalized methods of synthesis of pipelined datapaths are focused on a narrow class of algorithms for processing synchronous data flows with a static schedule.

Highlighting unexplored parts of the general problem. There is a wide class of algorithms, which are represented by the cyclo-dynamic data flow graphs, which have not yet been sufficiently explored in terms of improving their mapping in the pipelined datapathse and implementation them in HLS.

Setting objectives. The task is to create a method of designing the pipelined datapaths, which are focused on the execution of algorithms for processing the cyclic data flow graphs with a dynamic schedule.

Presentation of the main material. The new method consists in creating a cyclo-dynamic data flow graph, optimizing it and describing it in the VHDL language in a synthesis style. An example of the synthesis of a run-length encoding decompressor shows the effectiveness of the method. The results of decompressor synthesis using the LZW algorithm and their comparison with other devices are also shown.

Conclusions. The graph models of data flow processing algorithms are analyzed and a class of cyclo-dynamic data flow graphs is selected. A method of designing the pipelined datapath that perform cyclic algorithms with a dynamic schedule is proposed. The results of designing a lossless decompression device using this method are given. The method can be implemented in the HLS tools.

Keywords: data flow graph, field programable gate array, VHDL, pipeline, dynamic schedule.

UDC 004.94

Vitalii Omelchenko, Oleksandr Rolik

FORECASTING-AT-SCALE ALGORITHMS FOR PREDICTION CLUSTER WORKLOAD

The paper deals with the issue of predicting workloads in a cluster to use in proactive scaling of computing resources. Forecasting-at-scale algorithms Prophet and Greykite for forecasting time series are considered, their accuracy and universality are evaluated.

Keywords: resource management, proactive scaling, Kubernetes, automatic scaling, Prophet, Greykite. Fig.: 4. Tab.: 1. Bibl.: 10.

Problem statement. When managing the quality of services provided by clouds or enterprise-level IT infrastructures, it is important to maintain the quality of services at an agreed level [1]. In general, maintaining the quality of services at the agreed level is ensured by providing additional computing, communication, and other resources to those applications that provide the corresponding service. Effective management of the quality of services can be ensured mostly by automatic adding or reducing the number of resources. For this purpose, data center computing resource management systems of cloud service providers or corporate IT infrastructures contain modules that use automatic scaling methods and tools, taking into account service delivery technologies. The autoscaling methods that use the results of load forecasting will be more effective from the point of view of maintaining the quality of services at the agreed level. Modern paradigms of service provision are built on the use of microservices, containers, etc. The popularity of microservices in the provision of services entailed the creation of various technologies for the implementation of microservice architecture, including those based on clusters. At the same time, it is advisable to solve the problems of maintaining the quality of services at the agreed level by scaling all components of the cluster. It is necessary to take into account the fact that each component of the microservice architecture has its unique features of work and functionality, as well as the fact that the load pattern is individual for each component, and there can be a large number of such components. Therefore, when determining automatic scaling methods, taking into account load forecasting, it is advisable to process historical metrics, as well as adjust model parameters to predict a load of each group of components with the same properties separately.

Actual scientific researches and issues analysis. The topic of using both statistical prediction approaches [2] and artificial intelligence-based approaches for

automating the scaling of computing resources of deployed applications is quite well studied [3]. But these works do not consider the issue of implementing the developed solutions into existing systems, in particular the need to prepare and process historical data, adjust algorithms for existing load patterns, and validate the obtained results.

Uninvestigated parts of general matters defining. A large number of scientific works are devoted to the topic of proactive scaling both by statistical methods and based on artificial intelligence, but the vast majority only evaluate the accuracy of the obtained models and such a characteristic as universality – the ability of the model to work with a large number of various load patterns without manual adjustment – remains secondary [4].

The research objective. The purpose of this work is to investigate the practicability and possibility of applying forecasting-at-scale algorithms for predicting loads in a Kubernetes cluster. Through experiments, test the ability of Prophet and Greykite algorithms to predict typical load patterns of cluster components.

Overview of algorithms. Let us consider two main representatives of forecasting-at-scale algorithms [5], namely Prophet and Greykite. But first, let us define what forecasting-at-scale algorithms are. The name «at-scale» in this context has two meanings. First, it is a simple and powerful tool that does not require the user to have deep knowledge of prediction algorithms. It lets us scale prediction pipeline in terms of reducing time for creating more or less accurate model. Secondly, these algorithms allow solving a large number of various forecasting problems, including reliable practical forecasting of time series.

Prophet [6] is a time series forecasting library that was developed at Facebook. The main goal of the development was to create a simple, transparent and understandable model generation algorithm that would allow to quickly obtain reliable predictions.

This algorithm is based on an additive regression model, which has several components.

$$y(t) = g(t) + s(t) + h(t) + e(t), \quad (1)$$

where $g(t)$ – a trend component, $s(t)$ – a seasonal component, $h(t)$ – anomalies, $e(t)$ – an error function. In addition to the additive regression model, Prophet also uses a Fourier transform.

Among the advantages of this model it has the possibility of working with various time series, the ability to work effectively with large data sets and missing data, as well as flexibility in the setting.

The Greykite Library [7] is a powerful time series forecasting library developed by LinkedIn. Its main task is to provide a flexible, fast, and scalable solution for generating forecasts based on a large amount of data.

Greykite uses the Silverkite model, which, like Prophet, is an additive regression model with several components [8]:

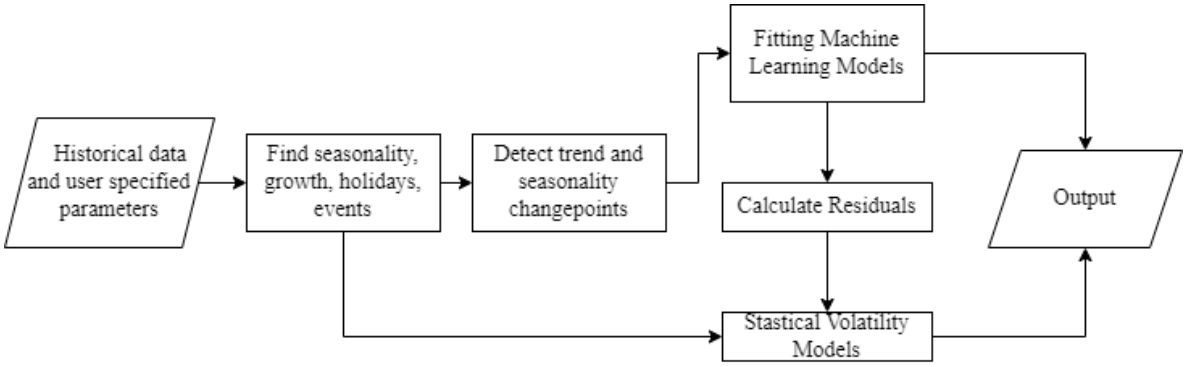


Fig. 1. Greykite's work algorithm

One of the unique features of Greykite is the ability to flexibly configure trending and seasonal components to adapt to different patterns in the data. In addition, Greykite uses different machine learning algorithms to process different components of the model, which makes this solution versatile.

Experiments. To check the accuracy and universality of the models, two typical load patterns were chosen [9]. The first pattern has weekly and daily load fluctuations, the second one has weekly on/off seasonality. Weekly seasonality is chosen because this period of time reflects the repeatability of people’s actions throughout the day, including in the business environment: working hours, time for rest, sleep, and so on. In general, any other seasonality can be chosen, and the goal is to test the performance of the models on complex seasonalities. It is important to clarify that the obtained models should not be adjusted to a specific time series in any way, since it is important to check the universality of the approaches.

To assess the accuracy of time series forecasting models, it is appropriate to use two accuracy metrics – RMSE (root mean square deviation) and MAPE (average absolute error in percentage).

RMSE allows you to compare the deviations of the original values and helps to assess the overall accuracy of the prediction. RMSE is calculated using the following formula:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n} (\hat{x}_t - x_t)^2}{n}}, \quad (2)$$

where $x(t)$ – an actual value at the moment of time t, $\hat{x}_t$ – prediction at the moment of time t, and n – a number of datapoints in the dataset.

The mean absolute percentage error (MAPE) makes it possible to compare the predictions of different models at different scales or data. MAPE is calculated as follows

$$MAPE = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{\hat{x}_t - x_t}{x_t} \right|, \quad (3)$$

where $x(t)$ – an actual value at the moment of time t, $\hat{x}_t$ – prediction at the moment of time t, and n – a number of datapoints in the dataset.

The assessment of the universality of the method is determined on the basis of the assessment of accuracy without additional adjustment of the models. It is important to determine how the model can adapt to various load patterns.

In the first experiment, the selected models are compared on the example of the above-described time series with two periodicities of different lengths – daily and weekly. The data has not been pre-processed. The purpose of this experiment is to investigate the prediction capabilities of the selected models on complex load patterns without any data distortion, and also to investigate the effect of the size of the historical data during training on the prediction accuracy.

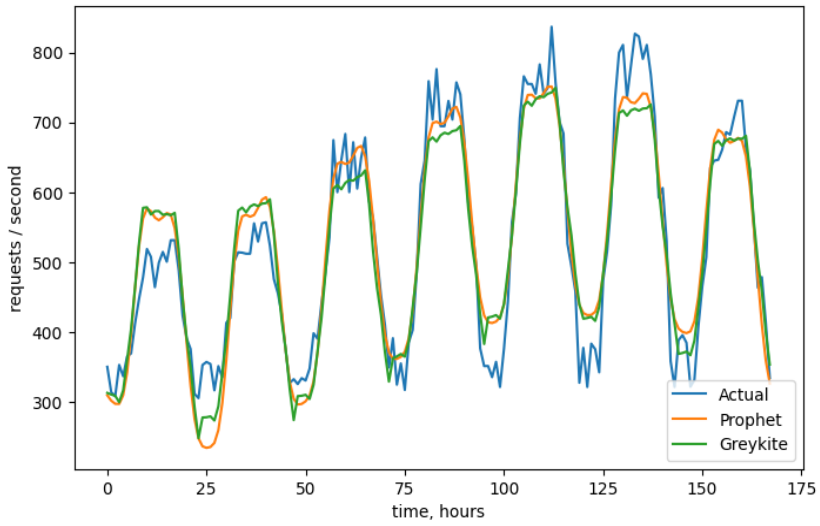


Fig 2. Model predictions based on three-week historical data

In fig. 2 shows the performance results of the selected algorithms after training for three weekly periods. The MAPE indicators are approximately at the same level – 0.085, which indicates high accuracy of the results. It is worth noting that the minimum length of historical data should be at least two target periods for pattern detection. It is appropriate to check the accuracy on the minimum permissible period.

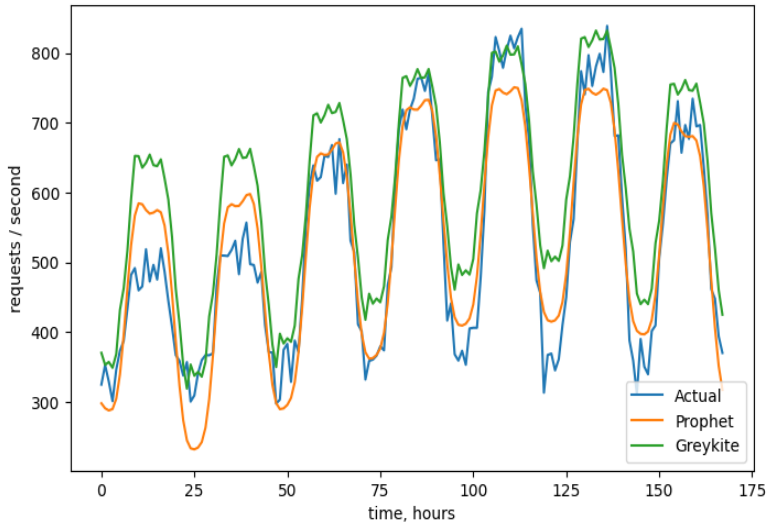


Fig 3. Model predictions based on two-week historical data

In this case, Greykite's accuracy dropped significantly to 0.16, while Prophet remained at the same level of 0.085.

The last experiment is conducted for another pattern, namely weekly on/off, where the load appears quickly and disappears quickly.

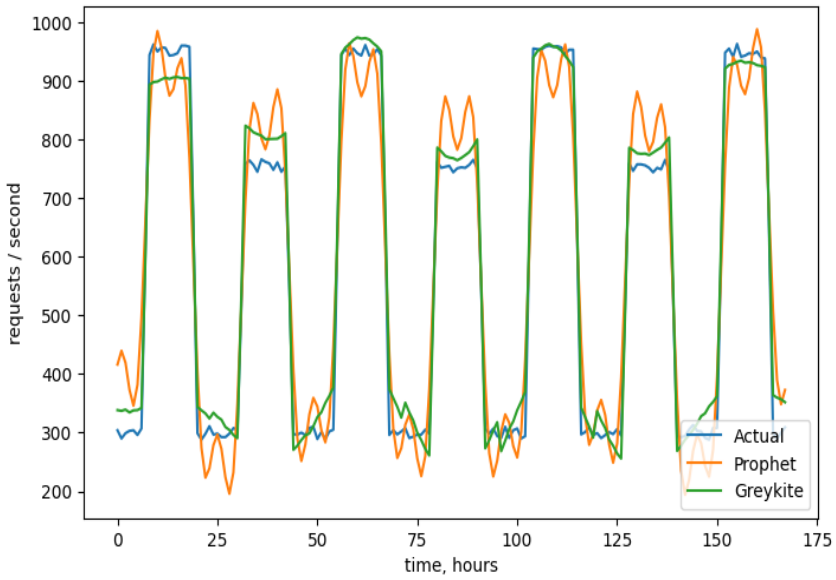


Fig 4. Prediction of on/off pattern patterns

In this experiment, Greykite is extremely accurate and has a MAPE of 0.06. In comparison, MAPE for Prophet is 0.12. All final accuracy indicators are shown in Table 1.

Table 1. **Prophet and Greykite performance results**

Approach	RMSE 3 weeks	MAPE 3 weeks	RMSE 2 weeks	MAPE 2 weeks	RMSE on/off	MAPE on/off
Prophet	49.2857	0.08723	51.1198	0.08929	72.7108	0.12620
Greykite	49.0042	0.08400	88.4236	0.16385	34.1663	0.06612

Conclusions. The experiments conducted in this work prove that the forecasting-at-scale algorithms for predicting time series Prophet and Greykite are appropriate to use when developing solutions for managing the autoscaling of computing resources in a cluster. Both algorithms showed high accuracy and relative versatility. Prophet is more accurate on shorter historical data lengths, and Greykite did much better with the on/off load pattern.

In future works, it is advisable to integrate these algorithms and reactive management approaches [10] into the resource distribution subsystem and conduct similar experiments in real conditions.

References

1. Rolik, A. I., Telenyk, S. F., Yasochka, M. V. (2018). Upravlenye korporatyvnoi ynfrastrukturoi. Kyiv: Naukova Dumka. 576 p.

2. De Livera, M., Hyndman, R. J., & Snyder, R. D. (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. In Journal of the American Statistical Association (Vol. 106, No. 496, pp. 1513-1527).

3. Rolik, O., Kolesnik, V., & Halushko, D. (2015). Neural network approach for resource allocation in IT infrastructure Management System. In Proc. Of the Congress on Information Technology, Computational and Experimental Physics 2013 (CITCEP'15) (pp. 176-179). 18–20 December, Cracow, Poland.

4. Lorido-Botrán, T., Miguel-Alonso, J., & Lozano, J. (2014). A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments. In Journal of Grid Computing (Vol. 12).

5. Taylor, S. J., & Letham, B. (2018). Forecasting at scale. In The American Statistician (Vol. 72, No. 1, pp. 37-45).

6. Facebook. Prophet: Automatic Forecasting Procedures. Available: https://github.com/facebook/prophet [Accessed: June 21, 2023].

7. LinkedIn. Greykite. Available: https://github.com/linkedin/greykite [Accessed: June 21, 2023].

8. Hosseini, R., Chen, A., Yang, K., Patra, S., Su, Y., Al Orjany, S. E., Tang, S., & Ahammad, P. (2022). Greykite: Deploying Flexible Forecasting at Scale at LinkedIn. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 3007-3017).

9. Kim, I. K., Wang, W., Qi, Y., & Humphrey, M. (2016). Empirical Evaluation of Workload Forecasting Techniques for Predictive Cloud Resource Scaling. In 2016 IEEE 9th International Conference on Cloud Computing (CLOUD) (pp. 1-10). San Francisco, CA, USA.

10. Omelchenko, V., & Rolik, O. (2022). Automation of resource management in information systems based on reactive vertical scaling. In "Adaptive Systems of Automatic Control" Interdepartmental scientific and technical collection (No. 2 (41), pp. 65-78).

AUTHORS

Rolik Oleksandr Ivanovych – professor, Doctor of Technology, head of the Department of Information Systems and Technologies, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: arolick@gmail.com

Omelchenko Vitalii – PhD student, Department of Information Systems and Technologies, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: vitaly.om25@gmail.com

РОЗШИРЕНА АНОТАЦІЯ

Віталій Омельченко, Олександр Ролік

FORECASTING-AT-SCALE АЛГОРИТМИ ДЛЯ ПЕРЕДБАЧЕННЯ НАВАНТАЖЕНЬ В КЛАСТЕРІ

Постановка проблеми. Прі управлінні якістю послуг, які надають хмари або ІТ-інфраструктури корпоративного рівня, важливо підтримувати якість послуг на узгодженому рівні [1]. Здебільшого підтримання якості послуг на узгодженому рівні забезпечується шляхом надання додаткових обчислювальних, комунікаційних та інших ресурсів тім застосункам, які забезпечують відповідний сервіс. Ефективне управління якістю послуг може бети забезпечено лише автоматичним додаванням або зменшенням обсягів ресурсів. Для цього системи управління обчислювальними ресурсами ЦОД хмарних провайдерів або корпоративною ІТ-інфраструктурою містять модулі, які використовують методи та засоби автоматичного масштабування з врахуванням технологій надання сервісів. Зрозуміло, що більш ефективними з точки зору підтримання якості сервісів на узгодженому рівні будуть ті методи автомасштабування, які використовують результати прогнозування навантаження. Сучасні парадигми надання сервісів побудовані на використанні мікросервісів, контейнерів та ін. Популярність використання мікросервісів при наданні послуг потягнула за собою створення різноманітних технологій реалізації мікросервісної архітектури, у тому числі і на основі кластерів. При цьому вирішення задач підтримання якості сервісів на узгодженому рівні доцільно здійснювати шляхом масштабування всіх компонентів кластеру. При цьому необхідно зважати на те, що кожен компонент мікросервісної архітектури має свої унікальні особливості роботи та функціонал, а також на те, що шаблон навантаження є індивідуальним для кожного компоненту, а таких компонентів може бути велика кількість. Тому при визначенні методів автоматичного масштабування з врахування прогнозування навантаження доцільно здійснювати обробку історичних метрик, а також налаштування параметрів моделі для передбачення навантаження кожної групи компонентів з однаковими властивостями окремо.

Аналіз останніх досліджень і публікацій. Тема застосування як статистичних підходів передбачення [2], так і на основі штучного інтелекту для автоматизації масштабування обчислювальних ресурсів кластеру розгорнутих

додатків є досить добре вивченою [3]. Але в цих роботах не розглядається Але в цих роботах не розглядається питання впровадження розроблених рішень в існуючі системи, зокрема необхідність підготовки та обробки історичних даних, налаштування алгоритмів під існуючі шаблони навантаження та валідацію отриманих результатів.

Виділення недосліджених частин загальної проблеми. Темі проактивного масштабування як статистичними методами, так і на основа штучного інтелекту, присвячена велика кількість наукових праць, проте переважна більшість проводить лише оцінку точності отриманих моделей, а така характеристика, як універсальність – здатність моделі працювати з великим числом різноманітних шаблонів навантаження без ручного налаштування залишається другорядною [4].

Мета дослідження. Метою даної роботи є дослідження доцільності та можливості застосування forecasting-at-scale алгоритмів для передбачення навантажень в Kubernetes кластері. Шляхом експериментів перевірити здатність алгоритмів Prophet та Greykite передбачати типові шаблони навантаження компонентів кластеру.

Викладення основного матеріалу. В роботі розглядається архітектура роботи двох застосування forecasting-at-scale алгоритмів – Prophet та Greykite. Описуються умови проведення експериментів та дані для них. Проводяться 3 експеримента з різною довжиною історичних даних та шаблонами навантаження.

Висновки. Проаналізовано доцільність застосування forecasting-at-scale алгоритмів для передбачення робочих навантажень кластеру. Дані алгоритми показали здатність точно передбачати робочі навантаження з різними шаблонами даних і комплексною сезонністю.

Ключові слова: управління ресурсами, проактивне масштабування, Kubernetes, автоматичне масштабування, Prophet, Greykite. Рис.: 4. Табл.: 1. Бібл.: 10.

UDC 004.8

Ponochovnyy Pavlo, Oliinyk Volodymyr

CLASSIFICATION MODEL OF MILITARY AVIATION BASED ON NEURAL NETWORK ENSEMBLE

The paper presents a model for classifying military aviation based on images using an ensemble of neural networks. Popular actual convolutional neural networks (CNN) served as the base models and were fined-tuned on relevant data. An ensemble of models was utilized to improve performance achieving high accuracy with reasonable compute requirements.

Key words: image classification, military aviation, transfer learning, ensemble of models.

Fig.: 2. Tabl.:1. Bibl.: 6.

Relevance of the research topic. The recognition of military vehicles has become increasingly relevant due to the rapid development and creation of numerous new prototypes, which are difficult to distinguish at first glance. Therefore, there is a need for systems that can provide reliable and fast information about the specific type of military equipment. This information can be crucial for intelligence agencies, military organizations, and other entities. This work aims to contribute to the development of more efficient and accurate models for the recognition of military aviation based on modern machine learning methods such as transfer learning and ensemble modeling.

Problem statement. The problem of creating and developing classification models in the military domain is associated with the limited availability of information in open sources. Building such classifiers requires a high quality datasets with enough data for training the models to make predictions. Despite possible applications in real-time scenarios where special detectors [1] or even trackers [2] are required, we focus on classical image recognition and applications in non-military OSINT systems.

Analysis of recent research and publications. In recent years, there have been emerging approaches to classifying various military equipment, including the application of machine learning techniques [3]. Other approaches combines detection and classification [4] that lowers accuracy and is redundant for OSINT-like systems.

Identification of unexplored aspects of the general problem. This article focuses on studying and analyzing the proposed approach for constructing a model for

the classification of military aviation based on an ensemble of neural networks. The research is concentrated on studying transfer learning for complex datasets and combining the results of trained models in an ensemble for joint prediction.

Target setting. The task is to train base models to recognize aircraft of different classes in the input images and combine their results in an ensemble that provides the final classification outcome.

The statement of basic materials. Solving the proposed task can be divided into three main stages[5]. The first stage involves loading pre-trained base models, adding additional layers, and training these models on a custom dataset. In the second stage, the trained weights are used to make predictions on the input image by extracting aircraft features and determining the probability of similarity to a specific class. The final stage includes processing the obtained results from the previous stage, specifically combining them into a single combined prediction, which serves as the final classification result.

Model training. In order for the models to classify specifically military aircraft, they should be trained a dataset containing classes of aircraft to be classified. The dataset with categorized images of aircraft was obtained from an open-source Kaggle dataset. ResNet-50, VGG-16, EfficientNetB3, and EfficientNetB7 were used as base models. All pre-trained weights of the base models were frozen after loading to prevent them from being retrained because we had limited amount of training data. Subsequently, the model architecture for training was defined, consisting of the following layers:

1. Base model.

2. Flatten layer– to transform the input data from a multi-dimensional form to a one-dimensional vector before feeding it to the next layer of the network.

3. Dense - to model complex non-linear dependencies between input and output data. It has the following attributes: units - 512 (a positive integer representing the dimensionality of the output space); activation='relu' - which returns the element-wise maximum of 0 and the input tensor.

4. Dense - with the following attributes: units - nclass (number of classes); activation='softmax' - meaning it will activate the softmax function, which transforms the vector of values into a probability distribution.

After defining the architecture, the model was trained with the following hyperparameters: optimizer - Adam optimizer; loss - 'sparse_categorical_crossentropy'; metrics - 'accuracy'; and the number of epochs - 10.

General model structure. In our case, the proposed solution is precisely the combination of model results, which is referred to as an ensemble. Its operating principle is as follows: the trained models make predictions on the loaded image, and then these results are combined using the averaging method (adding up all prediction results and distributing them among the total number of models). The highest result is then selected as the final classification outcome. The overall structure of the neural network ensemble and its functioning are depicted in Fig. 1.

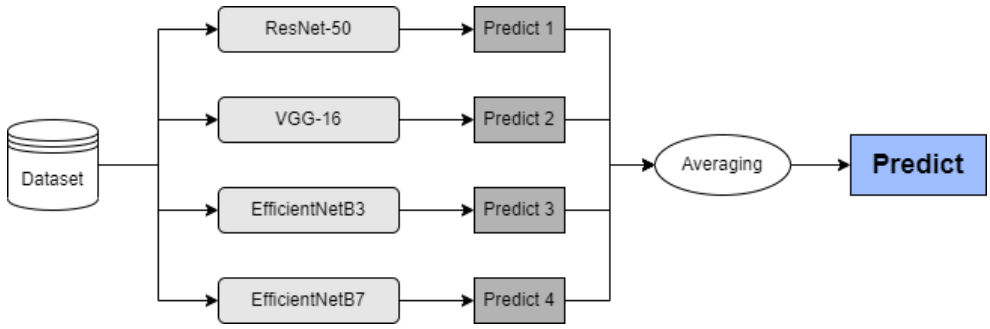


Fig. 1. General model structure

Results. The obtained results of the military aviation classification model based on the ensemble of neural networks are presented in Table 1. This table displays the accuracy of each individual model, ranging from 81% to 87%, as well as the accuracy of the ensemble model, which is 91.91%.

Table 1. **Model accuracy results**

Model	*Accuracy*	*Validation accuracy*
ResNet-50	1.0000	0.8569
VGG-16	0.9993	0.8165
EfficientNetB3	0.9780	0.8611
EfficientNetB7	0.9822	0.8660
Prediction result of the model ensemble		*0.9191*

Experiments. A simple interface was created for testing the developed ensemble neural network model for military aviation classification. Serious of experiment were conducted to analyze model performance. For instance, two images of the same aircraft model (A-10) obtained from open sources with high resolution but taken from different angles were tested. In the first experiment, the image clearly showed all the key elements and features of the aircraft, and the models correctly classified it with accuracy

results of 99% and above. The ensemble model achieved a final accuracy of 99.89%. In the second example, an image was used where the aircraft was not fully visible. In this case, all the models produced different classification results with lower accuracy scores, and the ensemble model correctly identified the aircraft model but with a low accuracy score of 33.8%. The test results are displayed in Fig. 2.

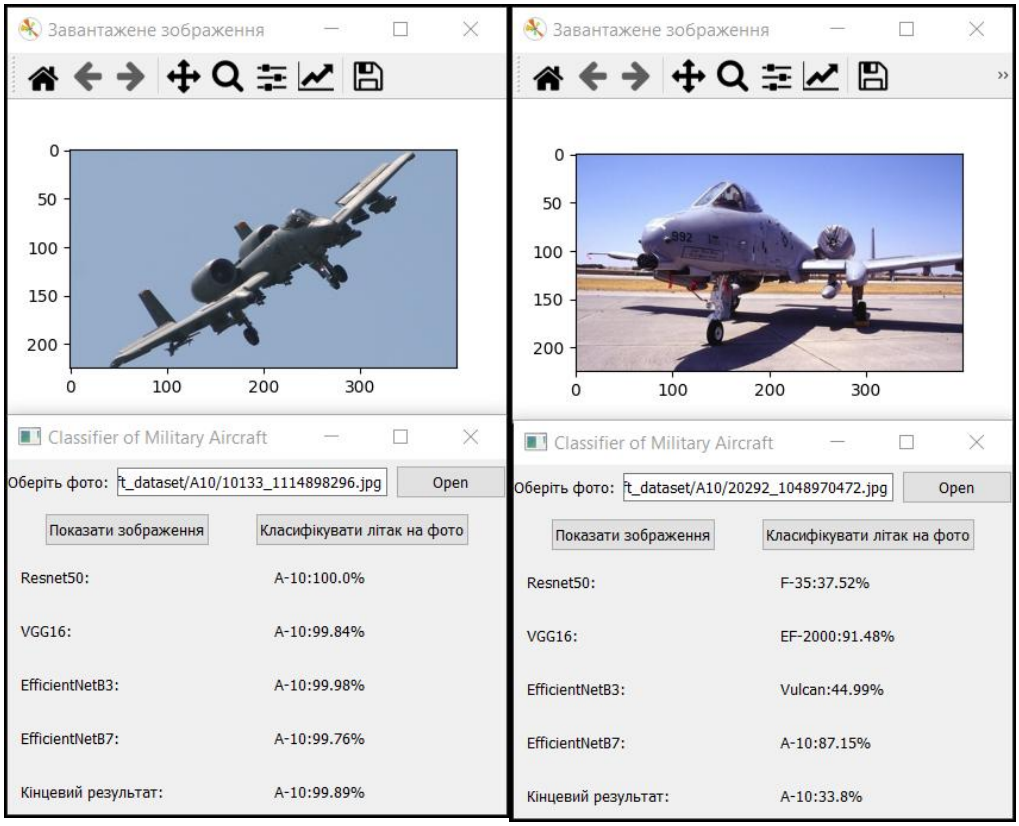


Fig. 2. Test Results

Conclusions. The paper proposes an approach to solve the task of military aviation classification on loaded images. This approach involves the application of modern machine learning methods, such as transfer learning and ensemble modeling. The utilization of a neural network ensemble allowed for an increase in classification accuracy of validation data from 81-87% to 91.91%, approximately a 5% improvement. This is a high performance considering the complexity of the task and wide variety of aircraft photos in the dataset. Dataset volume and quality is crucial for achieving state-of-the-art result. In our case of using proposed classification model in OSINT-systems gathering more training data from open sources with automatic class label proposal based on low-resource text classification techniques[5, 6] looks promising and we plan to experiment with it in future work.

References

1. Oliinyk V. An efficient facemask detection model for real-time applications / Oliinyk V., Ryzhiy A. // Adaptive systems of automatic control, 2022. Vol. 1, №40. – P. 54-64.

2. Panteleev A.S. Metod vizualnogo multitrekinga v realnom vremeni na osnove korreljacionnyh filtrov / Panteleev A.S., Olejnik V.V. // Mizhvidomchij naukovo-tehnichnij zbirnik "Adaptivni sistemi Avtomatichnogo Upravlinnja", K: Politehnika – 2018. – Vol.1, №32. – P. 97-106.

3. Sampath P. Revolutionizing Military Surveillance: Advanced Deep Learning Techniques for Aircraft Detection / P. Sampath, S. Haseeb, A. Jaideep, J. P. Kiran, S. A. Devi and P. V. V. S. Srinivas // *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, Coimbatore, India, 2023, pp. 12-19, doi: 10.1109/ICSCSS57650.2023.10169655.

4. Ajay Kumar Goud P. et al. Military Aircraft Detection Using YOLOv5 //Intelligent Communication Technologies and Virtual Mobile Networks. – Singapore : Springer Nature Singapore, 2023. – C. 865-878.

5. Oliinyk V. Data augmentation with foreign language content in text classification using machine learning / Oliinyk V., Osadcha K. // Adaptive systems of automatic control, 2020. Vol. 1, №36. – P. 51-59.

6. Oliinyk V. Low-resource text classification using cross-lingual models for bullying detection in the Ukrainian language / Oliinyk V., Matviichuk I. // Adaptive systems of automatic control, 2023. Vol. 1, №42. – P. 87-100.

AUTHORS

Oliynyk Volodymyr – Ph.D., associate professor, Department of Information Systems and Technologies, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: oliinyk.volodymyr@gmail.com

Ponochovnyy Pavlo – BS, Department of Information Systems and Technologies, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: pauluayt@gmail.com

РОЗШИРЕНА АНОТАЦІЯ

П.С.Поночовний, В.В.Олійник

МОДЕЛЬ КЛАСИФІКАЦІЇ ВІЙСЬКОВОЇ АВІАЦІЇ НА ОСНОВІ АНСАМБЛЮ НЕЙРОННИХ МЕРЕЖ

Актуальність теми дослідження. Розпізнавання військової техніки стає більш актуальним в останні дні у зв'язку із все більшим її розвитком та створенням великої кількості нових прототипів, які важко відрізнити на перший погляд. Таким чином, виникає необхідність систем, що надійно та швидко будуть надавати інформацію, про даний тип техніки, оскільки це може мати важливе значення для розвідувальних служб, армійських структур та інших організацій. Дана робота присвячена сприянню розвитку більш ефективних та точних моделей розпізнавання військової авіації на основі сучасних методів машинного навчання, таких як трансферне навчання та ансамбль моделей.

Постановка проблеми. Проблема створення та розвитку моделей класифікації на військову тематику пов’язана з обмеженістю інформації про це у відкритих джерелах. Для побудови такого класифікатора необхідно мати набір даних, завдяки якому відбувається навчання моделей, що будуть робити передбачення. Для того аби точність моделей була достатньою та мала високі показники набір даних повинен містити в собі велику кількість різних зображень.

Аналіз останніх досліджень і публікацій. Протягом останніх років з’являється все більше статей присвячених класифікації різноманітних об’єктів, зокрема, завдяки стрімкому розвитку різноманітних способів застосування машинного навчання. Проте побудова моделей саме для об’єктів високої складності, такої як військова техніка, зустрічається вкрай рідко.

Виділення недосліджених частин загальної проблеми. Дана стаття присвячена вивченню та аналізу запропонованого підходу для побудови моделі класифікації військової авіації на зображенні на основі ансамблю нейронних мереж. Дослідження сфокусовано на вивченні трансферного навчання для складних наборів даних та поєднанням результатів навчених моделей у ансамбль для спільного передбачення.

Постановка завдання. Завданням є навчити базові моделі розпізнавати літаки різних класів на зображеннях завантажених у функцію передбачення, та

поєднати їхні результати у ансамбль, що буде давати кінцевий результат класифікації.

Викладення основного матеріалу. Вирішення поставленої задачі можна розділити на три основні етапи. На першому відбувається побудова моделей та їх навчання на власному наборі даних. На другому етапі здійснюється передбачення навчених ваг на завантаженому зображенні. Заключний етап включає в себе обробку отриманих результатів попереднього етапу – об’єднання у одне кінцеве комбіноване передбачення, що і буде фінальним результатом класифікації.

Висновки. В роботі запропоновано підхід до вирішення задачі класифікації військової авіації на завантаженому зображенні. Використання ансамблю нейронних мереж дало змогу підвищити точність класифікації перевірочних даних. Під час виконання всі поставлені задачі були виконані успішно та було досягнуто високих показників точності, що доводить ефективність застосування обраних підходів.

Ключові слова: класифікація, військова авіація, трансферне навчання, ансамбль моделей.

UDC 004

A. Dremov, A. Volokyta

METHOD FOR MALICIOUS NETWORK TRAFFIC CATEGORISATION

Abstract. This paper aims to provide a solution for malicious network traffic detection and categorisation. In this paper we propose a semi-supervised GAN to train a discriminator model to categorise malicious traffic, as well as identify malicious and non-malicious traffic. The main goal is to achieve accurate categorisation of malicious traffic with few labelled examples.

Keywords: cybersecurity, network security, malicious traffic identification, machine learning, generational adversarial networks, semi-supervised learning.

Introduction/Relevance of the research topic

Computer networks are a key part of modern digital communications. However, these networks can be susceptible to malicious network traffic and various attacks. These attacks can be categorised by specific packet information used in these attacks. As such, network intrusion and attack detection play an important part in identifying an attack and counteracting it and are a relevant area of research.

Additionally, modern machine learning methods and algorithms can be used to categorise data or objects with great precision, provided a large enough training sample. However, rapid developments in security penetration create a problem, where new penetration methods appear frequently and gathering enough packet samples for model training becomes a difficult task. Therefore, the problem of training models with few initial samples remains relevant today. A combination of these areas is the main research area of this paper.

Problem Definition

The core problem that the research focuses on is the problem of malicious traffic identification and categorisation. First part of the problem is the identification of whether or not traffic is malicious in nature. Malicious traffic is one that can be used to attack the computer network and individual devices in the network and include malware, DoS attacks, network scanning, data exfiltration, R2L etc. Second part of this problem is categorisation of malicious traffic.

Actual scientific research and issue analysis

A number of researchers have tackled the problem of network attack classification [1][2] and the effect of malicious traffic on computer networks [3]. Of particular interest to this paper is the general approach to performing a network attack described in [1], as well as classification and effects described in [2] and [3] respectively.

Additionally, research into the intrusion detection and, more importantly, an analysis of malicious traffic packet contents [4][5][6] help connect network attacks to packet contents. This allows to define features used by the machine learning algorithm.

Lastly, research in the area of applying machine learning to solve network intrusion detection problem was performed [7], where a variety of models and algorithms are used. The research describes the architecture of semi-supervised GAN networks [8].

Uninvestigated parts of general matters defining

In author's opinion, the problem of intrusion detection using machine learning algorithms when there is insufficient data remains understudied. Additionally, proposed solutions may encounter difficulty with generalisation when being applied in different scenarios. A GAN based model could be used to achieve greater degree of generalisation.

Research objective

The purpose of this work is to research methods and models of malicious network traffic detection and categorisation with the usage of artificial intelligence models. Additionally, the purpose of the work is to create an AI model that can be used to detect classify malicious traffic with packet information.

Presentation of the main material

The dataset used in this research is NSL-KDD (https://www.kaggle.com/datasets/hassan06/nslkdd), which contains 125000 examples of network traffic packet data, as well as 22 categories based on attack type. Packets labelled "normal" indicate no attack. The features used in the classification include internet protocol used, service used, login status, login attempts, attempts to take root status, file and script creation, error rate, and other, for a total of 41 features. A total of 67000 records are labelled as non-malicious traffic and 58000 are labelled as malicious (fig. 1), (fig. 2), (fig. 3).

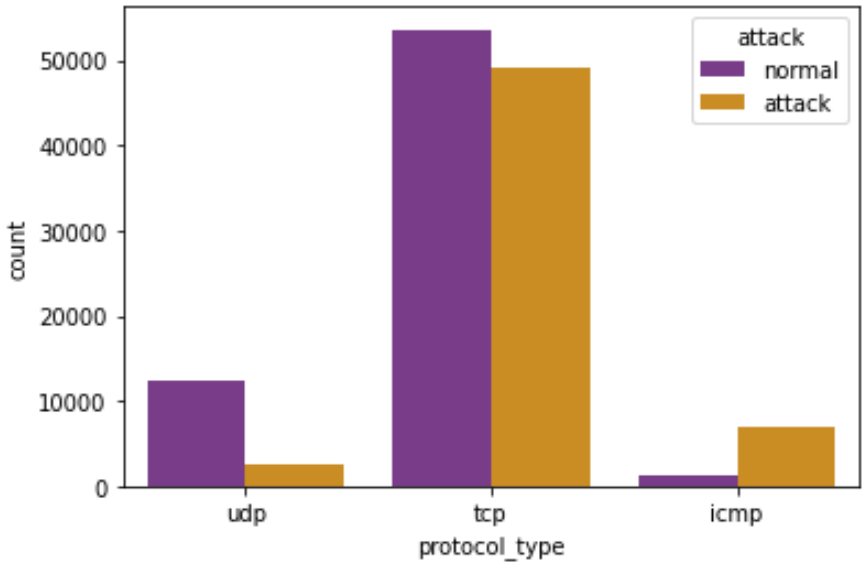


Fig. 1. Distribution of malicious and non-malicious traffic with regards to protocol used

```
RangeIndex: 125973 entries, 0 to 125972
Data columns (total 43 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   duration                     125973 non-null  int64
 1   protocol_type                125973 non-null  int64
 2   service                      125973 non-null  int64
 3   flag                         125973 non-null  int64
 4   src_bytes                    125973 non-null  int64
 5   dst_bytes                    125973 non-null  int64
 6   land                         125973 non-null  int64
 7   wrong_fragment               125973 non-null  int64
 8   urgent                       125973 non-null  int64
 9   hot                          125973 non-null  int64
 10  num_failed_logins            125973 non-null  int64
 11  logged_in                    125973 non-null  int64
 12  num_compromised              125973 non-null  int64
 13  root_shell                   125973 non-null  int64
 14  su_attempted                 125973 non-null  int64
 15  num_root                     125973 non-null  int64
 16  num_file_creations           125973 non-null  int64
 17  num_shells                   125973 non-null  int64
 18  num_access_files             125973 non-null  int64
 19  num_outbound_cmds            125973 non-null  int64
 20  is_host_login                125973 non-null  int64
 21  is_guest_login               125973 non-null  int64
 22  count                        125973 non-null  int64
 23  srv_count                    125973 non-null  int64
 24  serror_rate                  125973 non-null  float64
 25  srv_serror_rate              125973 non-null  float64
 26  rerror_rate                  125973 non-null  float64
 27  srv_rerror_rate              125973 non-null  float64
 28  same_srv_rate                125973 non-null  float64
 29  diff_srv_rate                125973 non-null  float64
 30  srv_diff_host_rate           125973 non-null  float64
 31  dst_host_count               125973 non-null  int64
 32  dst_host_srv_count           125973 non-null  int64
 33  dst_host_same_srv_rate       125973 non-null  float64
 34  dst_host_diff_srv_rate       125973 non-null  float64
 35  dst_host_same_src_port_rate  125973 non-null  float64
 36  dst_host_srv_diff_host_rate  125973 non-null  float64
 37  dst_host_serror_rate         125973 non-null  float64
 38  dst_host_srv_serror_rate     125973 non-null  float64
 39  dst_host_rerror_rate         125973 non-null  float64
 40  dst_host_srv_rerror_rate     125973 non-null  float64
 41  attack                       125973 non-null  int64
 42  level                        125973 non-null  int64
dtypes: float64(15), int64(28)
memory usage: 41.3 MB
```

Fig. 2. Dataset information

:

	duration	protocol_type	service	flag	src_bytes	dst_bytes
0	0	1	20	9	491	0
1	0	2	44	9	146	0
2	0	1	49	5	0	0
3	0	1	24	9	232	8153
4	0	1	24	9	199	420
...	...	...	...	...	...	...
125968	0	1	49	5	0	0
125969	8	2	49	9	105	145
125970	0	1	54	9	2231	384
125971	0	1	30	5	0	0
125972	0	1	20	9	151	0

Fig. 3. Example of values in dataset

The following data pre-processing was performed. The categorical values were converted to numerical values. The dataset was scaled using standard scaling, equation (1).

$$X' = \frac{x - \bar{x}}{\sigma} \tag{1}$$

Where x is the original feature vector, $\bar{x}$ is the mean of the feature vector, $\sigma$is standard deviation.

Lastly the labels were one-hot encoded for categorical classification.

For training we make use of a 70:30 split of training to test data.

As a baseline classifier, a simple deep network was implemented using tensorflow keras with two fully connected layers with 32 and 16 neurons, activation function is “relu”, batch normalisation layers and dropout layers to prevent overfitting (fig. 4). Final layer is a dense layer with “softmax” activation for categorical classification. Model metrics are “categorical_crossentropy” for loss function and “categorical_accuracy” for accuracy. The model was trained for 50 epochs on the dataset and achieved 99% accuracy, indicating possible overfitting (fig. 5). This classifier will be used to evaluate performance of the GAN-based classifier.

Second model is based on a generative adversarial network (GAN). These networks consist of a generator model and a classifier model. The generator uses gaussian distribution noise to generate fake information, equation (2).

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \quad (2)$$

Where μ is mean of the distribution, σ is standard deviation.

The classifier model of GAN is used to classify generator output as real or fake. For this a DNN with sigmoid activation is used. The result of the classification is used to calculate generator loss and discriminator loss (fig. 6). This allows to train the generator to create more believable fake data.

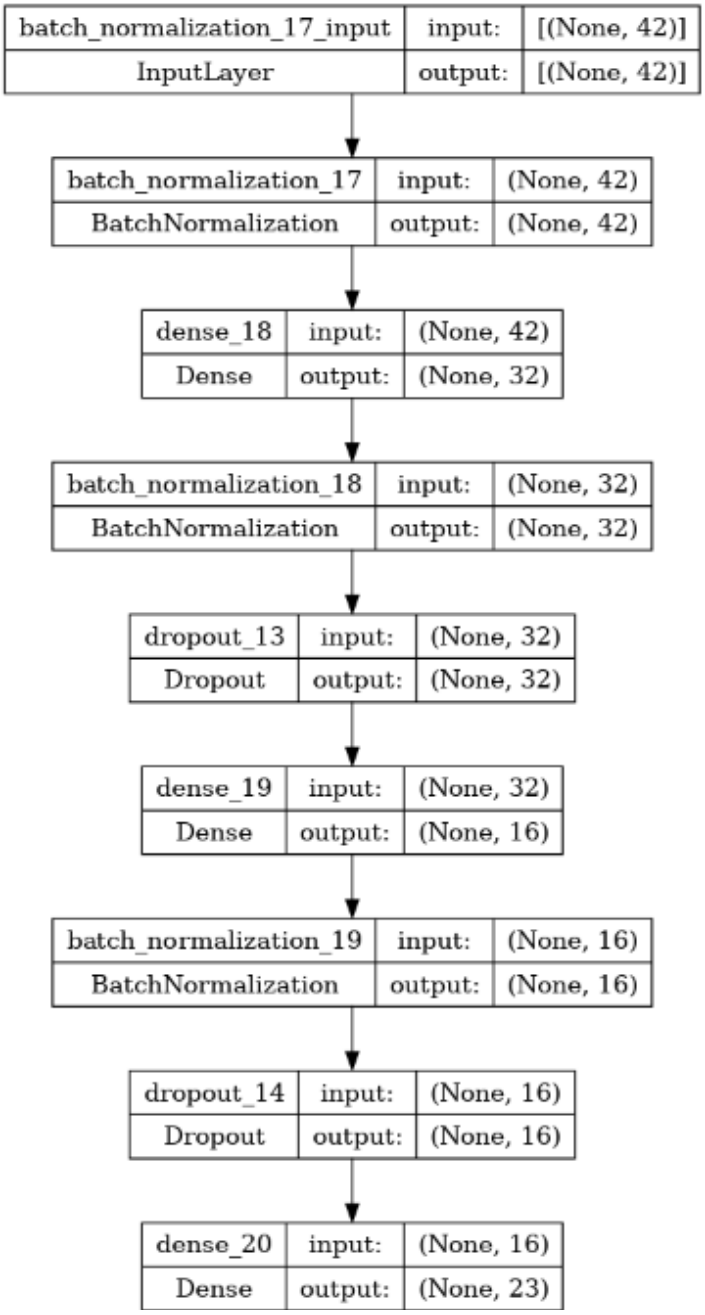


Fig. 4. Baseline classifier DNN

A subtype of GAN networks is a semi-supervised GAN. These are often used when trying to create a generator with little real samples available. In this case the discriminator predicts N+1 classes, with additional label being used for fake data classification. Of particular interest to this research is the efficiency of the categorical discriminator, not the generator model.

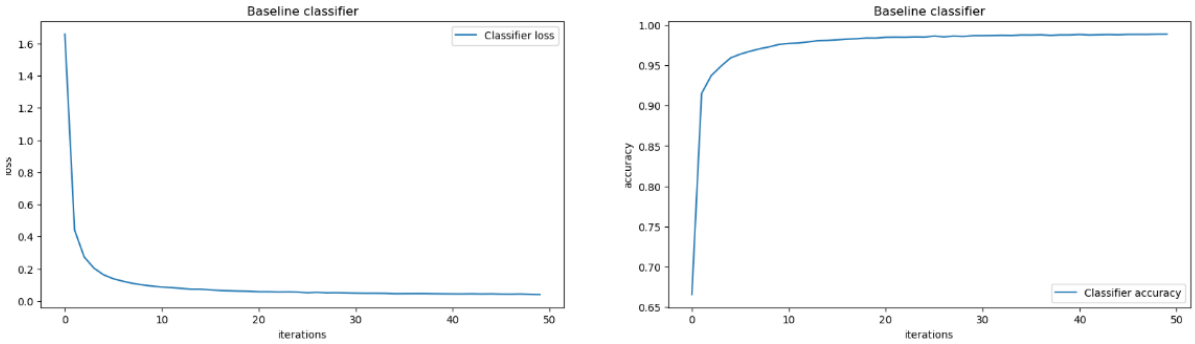


Fig. 5. Baseline classifier metrics

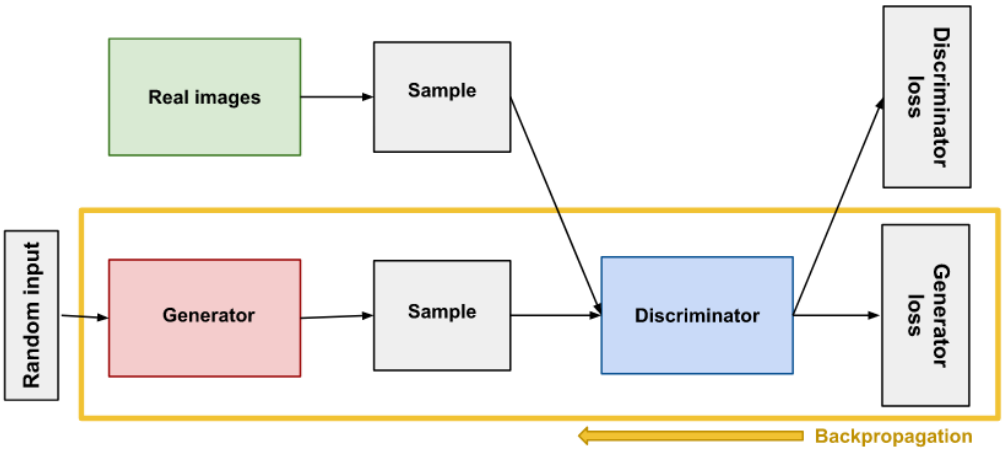


Fig. 6. General GAN architecture

In our implementation, we use two discriminator models, one for real/fake categorisation and another for attack categorisation. Target of the research is the attack categorisation model. The models share weights to ensure correct categorisation for real/fake as well as attack class. We use two dense layers size 256 and “relu” activation, as well as batch normalisation and dropout layers. Output layers are “softmax” for categorical classification model and “sigmoid” for binary classification. Loss functions and metrics are “categorical_crossentropy”, “binary_crossentropy”, “categorical_accuracy”, “binary_accuracy” for categorical discriminator and binary discriminator respectfully (fig. 7) (fig. 8). Since all of our input data is labelled, we only use a small sample of labelled entries, between 100 and 500 samples, as initial

input for categorical classifier model. For generator a model with three dense layers was used with 128, 256 and 512 nodes and “relu” activation. Additionally, batch normalisation and dropout layers were used. Output layer is dense layer with nodes equal to number of features and “tanh” activation (fig. 9). For model training 10 epochs were used. With final training categorical accuracy around 99% and binary accuracy around 78%. Final validation categorical accuracy around 89%. This indicates possible model overfitting (fig. 10-14).

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 42)]              0

 dense_3 (Dense)             (None, 256)               11008

 batch_normalization_3 (Batc  (None, 256)              1024
 hNormalization)

 dropout_2 (Dropout)         (None, 256)               0

 dense_4 (Dense)             (None, 256)               65792

 batch_normalization_4 (Batc  (None, 256)              1024
 hNormalization)

 dropout_3 (Dropout)         (None, 256)               0

 output_cat (Dense)          (None, 23)                5911

=================================================================
Total params: 84,759
Trainable params: 5,911
Non-trainable params: 78,848
```

Fig. 7. Categorical discriminator model

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 42)]              0

 dense_3 (Dense)             (None, 256)               11008

 batch_normalization_3 (Batc  (None, 256)              1024
 hNormalization)

 dropout_2 (Dropout)         (None, 256)               0

 dense_4 (Dense)             (None, 256)               65792

 batch_normalization_4 (Batc  (None, 256)              1024
 hNormalization)

 dropout_3 (Dropout)         (None, 256)               0

 output_bin (Dense)          (None, 1)                 257

=================================================================
Total params: 79,105
Trainable params: 0
Non-trainable params: 79,105
_________________________________________________________________
```

Fig. 8. Binary discriminator model

```
Model: "model_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 64)]              0

 dense_5 (Dense)             (None, 128)               8320

 dropout_4 (Dropout)         (None, 128)               0

 batch_normalization_5 (Batc  (None, 128)              512
 hNormalization)

 dense_6 (Dense)             (None, 256)               33024

 dropout_5 (Dropout)         (None, 256)               0

 batch_normalization_6 (Batc  (None, 256)              1024
 hNormalization)

 dense_7 (Dense)             (None, 512)               131584

 dropout_6 (Dropout)         (None, 512)               0

 batch_normalization_7 (Batc  (None, 512)              2048
 hNormalization)

 dense_8 (Dense)             (None, 42)                21546

 model_1 (Functional)        (None, 1)                 79105

=================================================================
Total params: 277,163
Trainable params: 196,266
Non-trainable params: 80,897
_________________________________________________________________
```

Fig. 9. GAN model (generator and discriminator model)

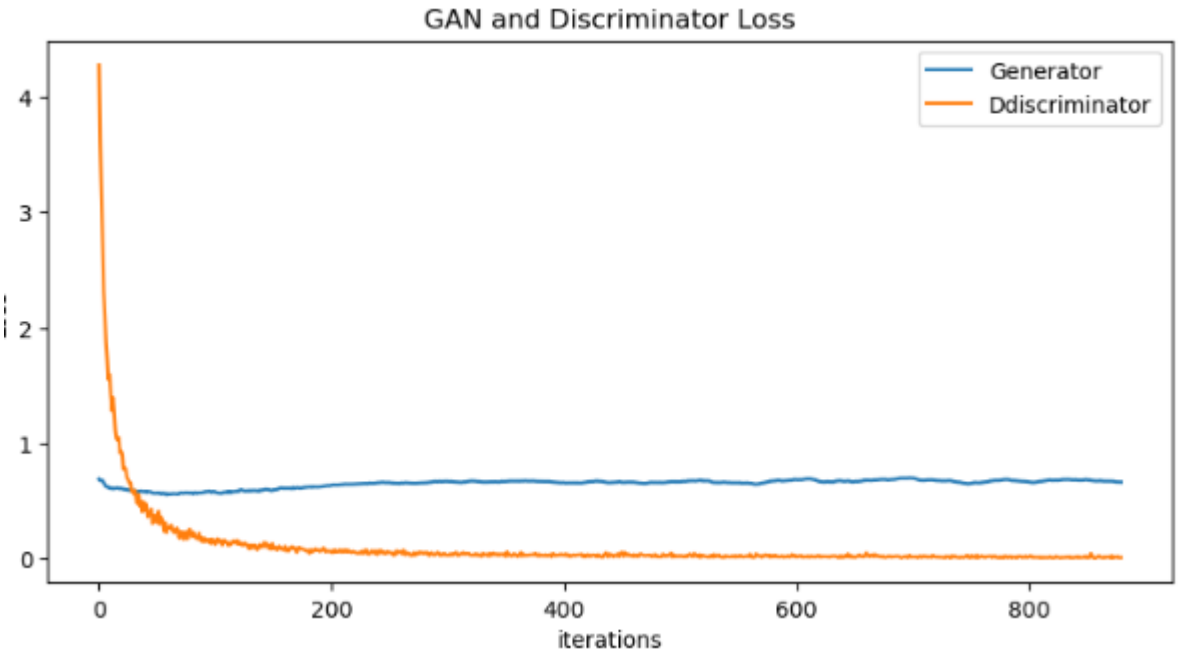


Fig. 10. GAN and discriminator(categorical) losses

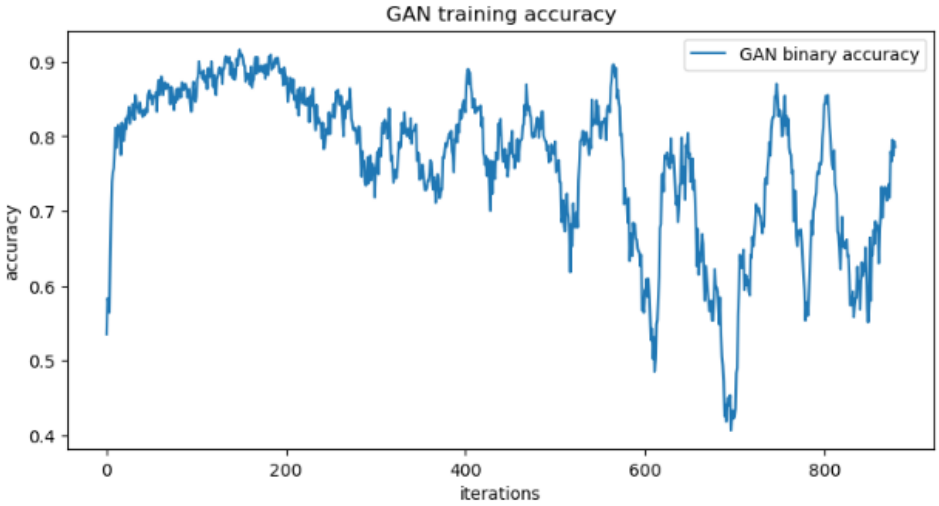


Fig. 11. GAN training accuracy

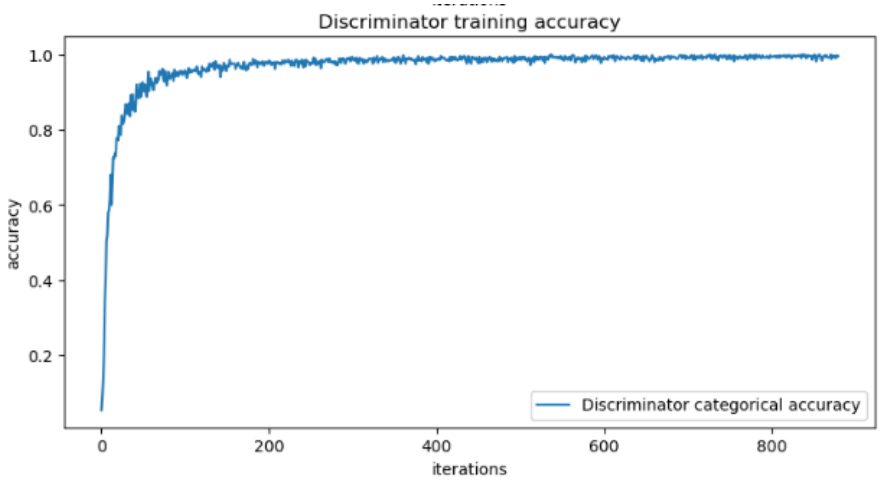


Fig. 12. Discriminator training accuracy

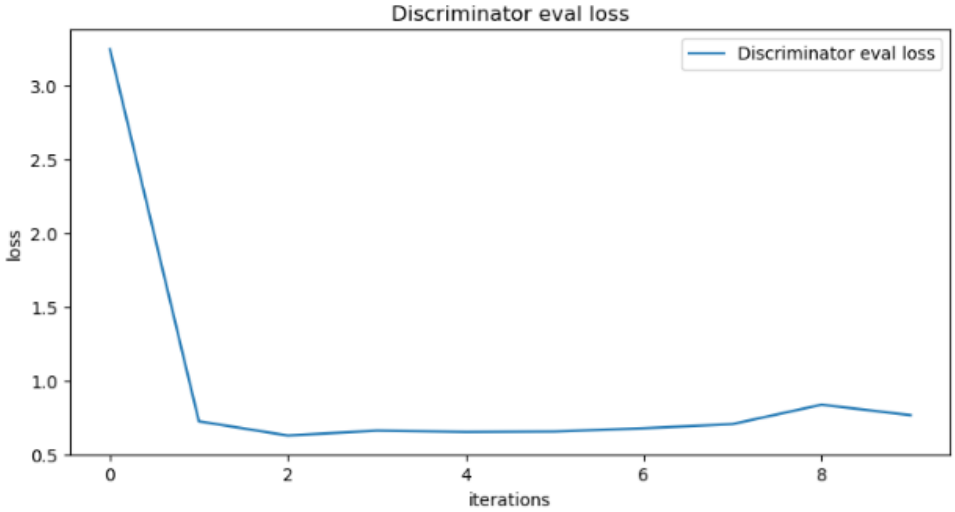


Fig. 13. Discriminator validation data loss (every epoch)

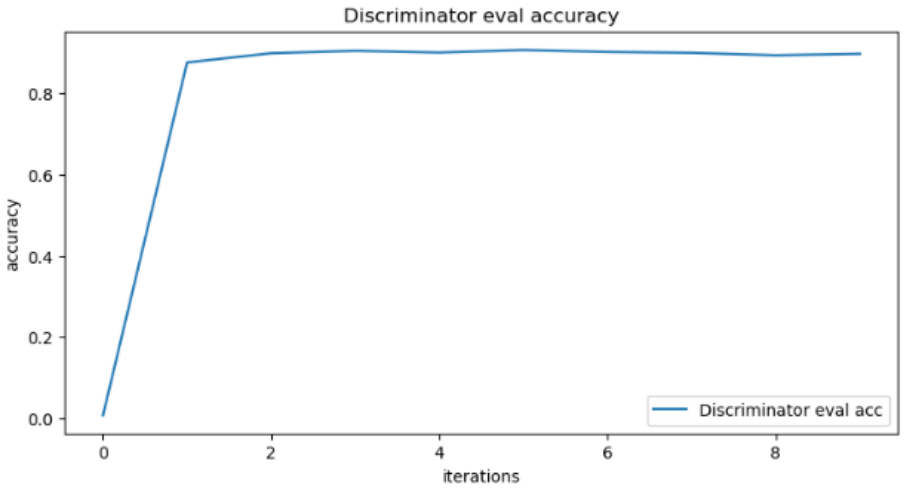


Fig. 14. Discriminator validation data accuracy (every epoch)

Conclusions

This research proposes the use of semi-supervised GAN model to train a classifier network for categorising malicious network traffic with limited number of labelled entries. For comparison we also used a baseline classifier DNN with a full dataset. The baseline classifier managed to achieve a validation accuracy of 99%, whereas SGAN discriminator only achieved 88%. The SGAN discriminator shows signs of overfitting with training accuracy of 99%. While the results are subpar compared to a full dataset classifier, it is worth noting that SGAN model only received a small portion of the dataset labels, between 100 to 500 samples, in different tests, while still achieving a relatively high accuracy score. It should also be pointed out, that GAN networks generally have trouble generating entirely new information, instead it creates slight variations of existing data. As such it may not be able to be used to train a network to predict entirely unknown threats.

Overall, SGAN networks may not be an effective solution to training network attack classifiers, however, additional research may be conducted. In particular, the question of network hyperparameter tuning remains open, as it may allow us to prevent overfitting and improve model accuracy. Additionally, the research was conducted only on a single dataset, it is worth exploring additional datasets to further evaluate the proposed solution.

References

1. Chasaki D., Wu Q., Wolf T. Attacks on Network Infrastructure // 2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN) / HI, USA. Lahaina: IEEE, 2011. C.1-1. URL: https://ieeexplore.ieee.org/document/6005919

2. Anderson R. Security Engineering: A Guide to Building Dependable Distributed Systems. Hoboken: Wiley Data and Cybersecurity, 2020. 1232 c. URL: https://ieeexplore.ieee.org/book/9820859

3. Lan K., Hussain A., Dutta D. The Effect of Malicious Traffic on the Network // Proc. Passive and Active Measurement Wksp. (PAM). 2009. № 10. C.1-1 URL: https://www.researchgate.net/publication/228611227_The_Effect_of_Malicious_Traffic_on_the_Network

4. John W., Olovsson T. Detection of malicious traffic on back-bone links via packet header analysis // Campus-Wide Information Systems. 2008. № 25 (5). C.342-358 URL: https://www.emerald.com/insight/content/doi/10.1108/10650740810921484/full/html

5. Network Traffic Analysis and Intrusion Detection Using Packet Sniffer / M. Qadeer та ін. // 2010 Second International Conference on Communication Software and Networks. Singapore, 2010. C.313-317. URL: https://ieeexplore.ieee.org/document/5437681

6. Wang W., Gombault S., Guyet T. Towards Fast Detecting Intrusions: Using Key Attributes of Network Traffic // 2008 The Third International Conference on Internet Monitoring and Protection. Bucharest, Romania, 2008. C.86-91. URL: https://ieeexplore.ieee.org/document/4561331

7. Network Intrusion Detection System: A Machine Learning Approach' / M. Panda та ін. // Intelligent Decision Technologies (IDT) Journal, IOS Press. 2011. № 4 (5). C.347-356 URL: https://content.iospress.com/articles/intelligent-decision-technologies/idt00117

8. Augustus O. Semi-supervised learning with generative adversarial networks // arXiv 1606.01583. 2016. № 0. C.1-3 URL: https://arxiv.org/abs/1606.01583

9. Pasupa, K., Tungjitnob, S., & Vatathanavaro, S. (2020). Semi-supervised learning with deep convolutional generative adversarial networks for canine red blood cells morphology classification. *Multimedia Tools and Applications*, 1, 1-1. Retrieved from https://doi.org/10.1007/s11042-020-08767-z

AUTHORS

Volokyta Artem – Ph. D., Associate Professor, Department of Computer Engineering NTUU “KPI”.

Scientific interests: real time systems, information security, distributed computations

Dremov Artem – student of the Department of Computer Engineering NTUU “KPI”.

Scientific interests: real time systems, cybersecurity, machine learning.