energies

MDPI

Article

Methodology for Management of the Protection System of Smart Power Supply Networks in the Context of Cyberattacks

Igor Kotenko [1,*], Igor Saenko [1], Oleg Lauta [2] and Mikhail Karpov [3]

[1] Laboratory of Computer Security Problems, St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), 39, 14th Liniya, 199178 St. Petersburg, Russia; ibsaen@comsec.spb.ru
[2] Department of Integrated Information Security, Admiral Makarov State University of Maritime and Inland Shipping, 5/7 Dvinskaya St., 198035 St. Petersburg, Russia; laos-82@yandex.ru
[3] Department of Information and Telecommunication Security, Saint-Petersburg Signal Academy, 3 Tikhoretsky Av., 194064 St. Petersburg, Russia; karpuh.djan@mail.ru
* Correspondence: ivkote@comsec.spb.ru

Abstract: This paper examines an approach that allows one to build an efficient system for protecting the information resources of smart power supply networks from cyberattacks based on the use of graph models and artificial neural networks. The possibility of a joint application of graphs, describing the features for the functioning of the protection system of smart power supply networks, and artificial neural in order to predict and detect cyberattacks is considered. The novelty of the obtained results lies in the fact that, on the basis of experimental studies, a methodology for managing the protection system of smart power supply networks in conditions of cyberattacks is substantiated. It is based on the specification of the protection system by using flat graphs and implementing a neural network with long short-term memory, which makes it possible to predict with a high degree of accuracy and fairly quickly the impact of cyberattacks. The issues of software implementation of the proposed approach are considered. The experimental results obtained using the generated dataset confirm the efficiency of the developed methodology. It is shown that the proposed methodology demonstrates up to a 30% gain in time for detecting cyberattacks in comparison with known solutions. As a result, the survivability of the Self-monitoring, Analysis and Reporting technology (SMART) grid (SG) fragment under consideration increased from 0.62 to 0.95.

Keywords: power supply; protection system; graph theory; SMART grid system; data transmission network; cyberattack; control methodology; LSTM neural network

Citation: Kotenko, I.; Saenko, I.; Lauta, O.; Karpov, M. Methodology for Management of the Protection System of Smart Power Supply Networks in the Context of Cyberattacks. *Energies* **2021**, *14*, 5963. https://doi.org/10.3390/en14185963

Academic Editor: Srđan Skok

Received: 15 August 2021
Accepted: 15 September 2021
Published: 20 September 2021

1. Introduction

The Self-monitoring, Analysis and Reporting technology (SMART) grid is a power grid technology that uses information and communication networks and technologies to collect information about energy production and energy consumption to automatically improve the efficiency, reliability, economic benefits, and the sustainability of electricity generation and distribution [1–9].

The SMART grid has the following features:

- Application of open information and communication networks, protocols and technologies for collecting information on energy production and energy consumption;
- Active bidirectional scheme of interaction in real time of the information exchange between all elements and participants of the network (from power generators to terminal devices of power consumers);
- Coverage of the entire technological chain of the electric power system from energy producers and power distribution networks to the end consumers;
- Constant exchange between the network elements of information about the parameters of electrical energy, modes of consumption and generation, the amount of energy consumed and planned consumption, and commercial information [1].

On the one hand, the use of the SMART grid makes it possible to reduce the cost of the electrical network, solve the problem of technological limitation of electricity when consumed near peak capacities, use a large number of renewable energy sources, and also switch from a centralized topology of the electrical network to a highly distributed topology.

On the other hand, the pace at which the modern field of open information and communication networks, protocols, and technologies is developing exposes the world community to a number of unprecedented threats and vulnerabilities. At the same time, the greatest danger is caused by cyberattacks.

In the past few years, a characteristic trend of our time has been an increase in the number of cyberattacks on critical information infrastructure (CII) and strategic industrial facilities, which can lead to the disabling by attackers of systems that support human life and the emergence of global man-made disasters. The main element of CII is an integrated telecommunication network, including SMART grid (SG) power supply systems, in which controlled objects should allow remote control, and systems for assessing the situation and emergency automation should reduce excessive requirements for the reserves of power and information capacities [2].

The effects of computer attacks, first of all, are aimed at disrupting the performance of the SG protection tools that are combined with a unified management and monitoring system. However, in this system, it is possible to distinguish the following negative features. First, it requires one to have socket specialists who are usually not enough. Secondly, specialists are not able to process all the incoming threat messages during the working day. Finally, energy companies use a large number of various means of protecting information and communications, which are not always conjugated with each other. Considering that information security specialists need to analyze thousands of events daily, the task of viewing and filtering such a number of data, as well as managing the protection tools can be solved only by applying the automation tools. For the continuous monitoring of the state of geographically distributed defense means and the implementation of proactive measures to neutralize cyberattacks, it is necessary to take into account the peculiarities of the SG functioning, the interaction of defense means, the indicators characterizing the effectiveness of its work, and the constantly changing ways of implementing cyberattacks. All this gave rise to the search for new methods of managing the SG protection system in the context of cyberattacks [3,10–13].

Considering that the behavior of the SG protection system in the context of cyberattacks can be represented as a sequence of random events with a finite or countable number of outcomes, it can be assumed that well-tested Markov chains can be used to describe it. However, the Markov process mechanisms lose their meaning when analyzing complex, multifaceted systems, such as the SG protection system, in which processes can proceed not only sequentially, but also contain events independent of each other.

In this work, in order to study the SG protection system as the system in an antagonistic confrontation (the cyberattacks versus the protection system), we will apply the method of constructing models using flat, or plane, graphs. A flat (planar) graph is a graph that can be included in a plane; that is, it can be drawn on a plane in such a way that its edges intersect only at their endpoints. Flat graphs are widely used as probabilistic automata in modeling structures such as simple cycles, trees, forests, etc.

The results of the modeling of information processes in the SG technological data transmission networks (TDTN) using flat graphs are then used in our methodology for the substantiation and implementation of the architecture of the SG protection system (PS) based on its capabilities to identify and further predict cyberattack impacts.

The analysis showed that one of the most efficient prediction methods is the usage of artificial neural networks with long short-term memory (LSTM). The property of recurrence allows an artificial neural network (ANN) to "refer" to the results of its work in the past and fulfill a predictive analysis. However, the efficiency of LSTM operation largely depends on the quality of the formation of the features and the context of datasets used for the LSNM network learning. For complex systems such as the SG PS, the formation of training

datasets very often becomes a very hard problem, making it difficult to use LSNM networks or significantly reducing the effect of their use. It should be noted that the use of open access test datasets for training LSNM networks does not solve this problem, since the computer networks on which the test datasets are formed, as a rule, do not take into account the specifics of TDTN in the SG.

Thus, we can say that the methodology proposed in the paper is aimed at solving the scientific problem of increasing the efficiency of detecting and predicting computer attacks in SG TDTN using LSTM networks by improving the quality of the formation of training datasets based on SG modeling using flat graphs.

Solving this problem requires overcoming a number of current challenges, among which are the following:

- Flat graph-based development and implementation of the formal models, which provide the specifications necessary for constructing a LSTM network and training and testing datasets, i.e., the information resource model, the security threat model, and the PS functioning model;
- Selection and justification of hyperparameters necessary to construct the LSTM network;
- Development of an approach to generate datasets with TDTN information content necessary for training and testing LSTM networks;
- Integration of the developed approaches according to the required criteria into a unified methodology of the SG PS management in the context of cyberattacks.

To overcome the above challenges, the methods of graph theory, probability theory, and machine learning, including deep learning and others, were used. The fairness of the models and approaches developed on their basis was verified as a result of their practical implementation and experiments, for which modern proven software technologies (frameworks and libraries) were used. The results of the experiments have shown that more efficient approaches are needed to provide proactive SG protection.

Therefore, the main contribution of the paper, demonstrating the range of possible applications of flat graphs for describing the features of functioning of the SG protection system under the influence of cyberattacks, as well as LSTM neural networks for making effective management decisions on the implementation of proactive SG protection, is undoubtedly relevant.

In addition, the contribution of the paper is as follows:

- Structures of long-term dependencies in the SG traffic, which allow for revealing its characteristic features in the interests of the early detection of cyberattacks;
- A new approach to cyberattack detection based on the use of flat graphs and LSTM neural networks;
- A software tool that implements the proposed approach;
- A dataset with SG traffic containing anomalies from the impact of cyberattacks;
- An experimental evaluation of the proposed approach.

The novelty of the obtained results lies in the fact that, on the basis of experimental studies, a methodology for managing the SG protection system in conditions of cyberattacks is substantiated. It is based on the specification of the SG protection system by using flat graphs and implementing a neural network with long short-term memory. Such results make it possible to predict with a high degree of accuracy and fairly quickly the impact of cyberattacks, on which the basis of the proactive protection measures can be developed. This is a significant advantage of the proposed method.

The further structure of the paper is as follows. Section 2 reviews related works on the research topic. Section 3 describes the theoretical foundations of the proposed methodology for the management of the SG protection system, based on modeling the protection system and predicting the impact of cyberattacks. Section 4 presents the implementation issues of the proposed methodology. Section 5 outlines the experimental results and its comparative evaluation. Section 6 contains conclusions and further research directions.

2. Related Work

In international practice, the abbreviation SMART stands for "Self-Monitoring, Analysis and Reporting Technology", i.e., the technology that implies the independent monitoring, analysis and transmission of monitoring results, and network resource management. Typically, the SMART grid refers to the hardware and software architecture that contributes to the efficiency of energy management. Along with the SG, the concepts of the Modern grid, Wise grid, Future grid, Empowered grid, and Intelligrid are used [4–7]. Sometimes SG systems are called "smart", "intelligent" or "adaptive-active" power supply systems [8,9].

Security threats of the SG power supply are included in the five most probable risks (together with the risks of epidemics, critical weather conditions, financial collapses, and extreme natural disasters) and in the list of the six most critical factors in terms of possible damage (together with the risks of using weapons of mass destruction, natural disasters, weather anomalies, and the lack of drinking water). That is why the security of SG management [10–13] is one of the priority directions for development of the energy complex all over the world, as it is critical for their effective functioning.

All of the SG security threats can be characterized by two parameters: firstly, the likelihood of the threat being realized, and, secondly, the potential damage to the energy company (organization, enterprise). Usage of these parameters to select a model of the threats to SG resources allows one to find the "golden mean" when building a protection system, choose network management techniques, and make decisions to minimize risks. Today, there are a huge number of diverse and very common methods for managing SG security systems, which in turn are divided into three main groups.

The first group [14–18] summarizes the methods based on quantitative indicators and criteria. The measure of ranking of the threat models (criterion) is the permissible level of possible damage from information and technical impact on SG resources and the assessment of the profit factor from investments in protective measures. Quantitative methodologies follow the requirements of ISO 27001 and 27002, NIST, and COBIT IV. Although these methodologies take into account a predetermined risk appetite, they do not consider the variability of the SG defense system design. In addition, the disadvantages of these techniques include the complexity of their implementation and the high level of labor costs. The complexity of quantitative methods also lies in the fact that the decision taken for each potential threat must be taken into account in the strategy for eliminating the consequences of a cyberattack [19]. For example, in [20], the quantitative ranking of risks for the SG is taken into account. However, the method of managing the security system through a cloud computing service considered in this work is of interest. Nonetheless, this technique contains a number of negative factors associated with the problems of cloud resources.

The second group of techniques [21–24] consists of qualitative techniques. The methodologies of this group take into account the security threats to SG resources by quality criterion. Qualitative methods boil down to finding an optimal solution, a balance between the costs of building a protection system and the resulting effect (cost/benefit analysis), i.e., the quality of the protection system. As a rule, the methods use the mathematical apparatus of game theory (matrix games). The disadvantages of qualitative methods include the high complexity of calculating the results of a risk analysis for the financial justification of the feasibility of investing in the implementation of the SG protection system according to one or another threat model, as well as the insufficient visibility of the results of qualitative methods. Techniques using qualitative criteria are similar in nature to the facilitated risk analysis process (FRAP) technique [25,26].

The third approach [27–31] is a combined (mixed) one. It combines the approaches used in both the first and the second groups of techniques. Most often, combined techniques are used in small energy companies. The weaknesses of this group of methods are insufficient analytical data on the predicted damage of the cyberattack impact, as well as the use of a minimum set of factors in risk assessment.

Thus, in [32–35], a structured approach to assessing the model of threats to the SG information and telecommunications resources, namely CRAMM (Risk Analysis and Management Method from Central Computer and Telecommunication Agency) and MEHARI (MEthod for Harmonized Analysis of Risk) methods, are presented, an integrated representation of the information security threat parameters is employed, but the peculiarities of building the SG protection system are practically not considered.

The information security management methodology of Microsoft Security Assessment Tool (MSAT) [34,36] is interesting not only for its threat model ranking system, but also for the implementation of the information security threat decision-making system and for assessing the effectiveness of the measures taken. However, it is usually implemented on local SG power grids. The MSAT system is based on the Risk Management Manual [23]. It performs the following functions: (1) risk assessment; (2) decision support; (3) implementation of control; and (4) evaluation of the effectiveness of the program. This application (app) is targeted at companies with less than 1000 employees and is designed to help one better understand the potential information security issues.

All of the above approaches to the management of SG protection systems are either based on a deep analysis of the potential risks (probable damage), or a selectively ranked construction of the SG PS. Therefore, we propose an architecture-oriented approach to managing the SG security system that goes beyond the abstract representation and dispenses with the technical details.

Our approach covers the identification and assessment of threats to the impact of cyberattacks, modeling the PS architecture, situational PS management based on a neural network algorithm with long short-term memory, as well as reducing the risks and assessing the effectiveness of the predictions and countermeasures taken. At the end of the paper, we will take a closer look at the proposed active security solutions for SMART grids and their implementation.

3. Theoretical Foundations of the Methodology for Management of the SG Protection System

Many works, for example [37,38], are devoted to the theoretical foundations of the theory of planar (flat) graphs. Therefore, let us consider in more detail their application for building the model of the SG protection system functioning. This model, in turn, includes three models: the model of protected information resources, the cyberattack threat model, and the model of functioning of the SG TDTN protection system.

3.1. Model of Protected Information Resources of SG

The components of the SG TDTN, which contain protected information resources (PIR), are services and software and hardware systems that implement logically complete functionality of the SG TDTN. Information security threats are unique to each element of the system. However, each component of the TDTN must comply with the security policy requirements.

To create a model, we construct an N-root graph GT_i, which reflects the PIR of the TDTN (Figure 1). The top of the GT_i graph reflects the main goal of the information and technical impact of an adversary's cyberattack. By the term "adversary" we mean an attacker (or an organized group of attackers) whose purpose is to disrupt the effective functioning of the SG as a critical infrastructure object that affects the life of society.

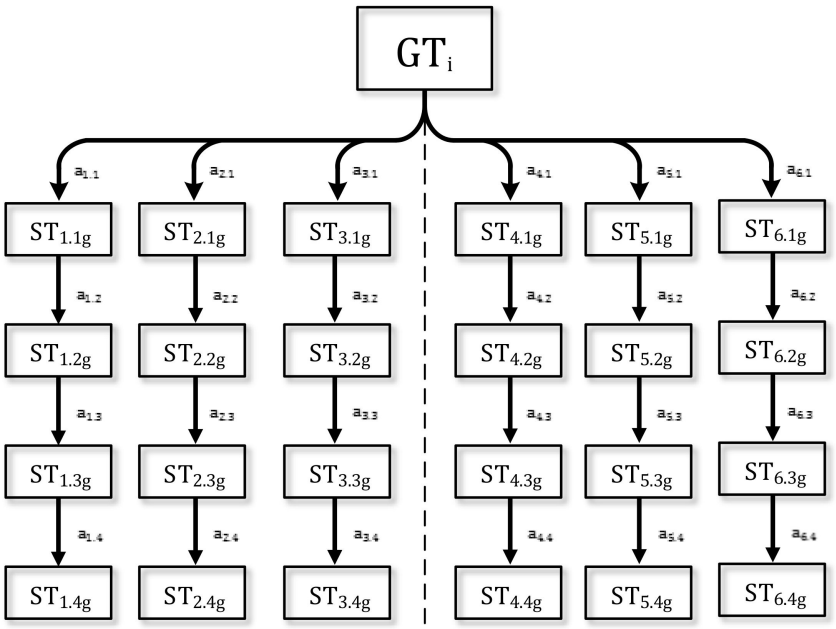

Figure 1. Oriented graph "Protected resources of TDTN".

The nodes $\{ST_{i.jg}\}$ of the graph GT_i represent the secondary goals (subgoals) of the impact of cyberattacks, and the arcs $\{a_{i.j}\}$ reflect the significance of these subgoals. The subgoals are grouped into groups $\{GT_{ij}\}$, which are subgraphs of the graph G_i.

The groups disclose one of the aspects of information security (confidentiality GT_{i1}, integrity GT_{i2}, and availability GT_{i3} of the information), as well as the required parameters of the telecommunications component of the TDTN (intelligence security GT_{i4}, sustainability GT_{i5}, and the TDTN throughput GT_{i6}) that is to be protected. GT_{i5} can be decomposed into four more oriented subgraph chains characterizing "Noise immunity", "Reliability", "Vitality", and "Cyber resilience" of the TDTN (Figure 2).

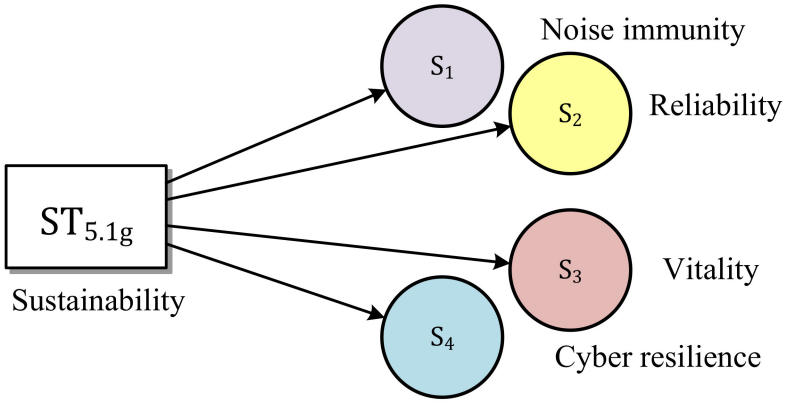

Figure 2. Components of the sustainability graph for the SG TDTN.

In the works [39,40], the mutual influence of noise immunity, survivability, reliability, and cyber-stability of the information resources of technological data transmission networks is described in some detail. Showing that the individual properties of technical systems can be considered together, within the framework of the concept of technical stability, using unified fuzzy logic descriptions, the authors place special emphasis on the complex influence of the balance of all four components S_1, S_2, S_3 и S_4. Let us describe the subgraphs $\{GT_{ij}\}$ with the following expressions:

(1) Confidentiality of information

$$GT_{i1} = \left(GT_i, ST_{1.1g}, ST_{1.2g}, ST_{1.3g}, ST_{1.4g}, a_{1.1}, a_{1.2}, a_{1.3}, a_{1.4}\right); \quad (1)$$

(2) Integrity of information

$$GT_{i2} = \left(GT_i, ST_{2\ .1g}, ST_{2.2g}, ST_{2.3g}, ST_{2.4g}, a_{2.1}, a_{2.2}, a_{2.3}, a_{2.4}\right); \quad (2)$$

(3) Availability of information

$$GT_{i3} = \left(GT_i, ST_{3.1g}, ST_{3.2g}, ST_{3.3g}, ST_{3.4g}, a_{3.1}, a_{3.2}, a_{3.3}, a_{3.4}\right); \quad (3)$$

(4) TDTN intelligence security

$$GT_{i4} = \left(GT_i, ST_{4.1g}, ST_{4.2g}, ST_{4.3g}, ST_{4.4g}, a_{4.1}, a_{4.2}, a_{4.3}, a_{4.4}\right); \quad (4)$$

(5) TDTN sustainability

$$GT_{i5} = \left(GT_i, ST_{5.1g}, ST_{5.2g}, ST_{5.3g}, ST_{5.4g}, a_{5.1}, a_{5.2}, a_{5.3}, a_{5.4}\right); \quad (5)$$

(6) TDTN throughput

$$GT_{i6} = \left(GT_i, ST_{6.1g}, ST_{6.2g}, ST_{6.3g}, ST_{6.4g}, a_{6.1}, a_{6.2}, a_{6.3}, a_{6.4}\right). \quad (6)$$

In Figure 1, the dotted line delineates the protected information and telecommunication resources. However, this distinction is conditional. In real TDTN, it is rather difficult to separate the rigidly interconnected components of information and telecommunications security.

Each of the directed subgraphs $\{GT_{ij}\}$ included in the graph GT_i comprises levels that characterize the assets directly related to the protected resource. We will take into account the following levels:

- Disruption to the functioning of the TDTN ($ST_{1.1g}$, $ST_{2.1g}$, $ST_{3.1g}$, $ST_{4.1g}$, $ST_{5.1g}$, $ST_{6.1g}$);
- Hardware resource of the TDTN ($ST_{1.2g}$, $ST_{2.2g}$, $ST_{3.2g}$, $ST_{4.2g}$, $ST_{5.2g}$, $ST_{6.1g}$);
- Software resource of the TDTN ($ST_{1.3g}$, $ST_{2.3g}$, $ST_{3.3g}$, $ST_{4.3g}$, $ST_{5.3g}$, $ST_{6.1g}$);
- Protected information and telecommunications resource of the TDTN ($ST_{1.4g}$, $ST_{2.4g}$, $ST_{3.4g}$, $ST_{4.4g}$, $ST_{5.4g}$, $ST_{6.4g}$).

Having determined the specific weight of each of the vertices of the subgraphs GT_{i1}, ..., GT_{i6}, as well as the price of each of their arcs, we obtain a weighted N-root planar graph for which we can determine the reachability matrix of each vertex of the subgraphs ($ST_{i.1g}$, $ST_{i.2g}$, ..., $ST_{i.ng}$).

By the terms "specific weight" we mean the importance of a particular protected information and telecommunications resource of the TDTN when choosing a defensive strategy or information security policy. Thus, the specific weight of the vertex to which the path μ_1 leads is calculated as follows:

$$L_{[\mu 1]} = \sum_{a_{i,j} \in \mu_1} A_{ij}, \quad (7)$$

where A_{ij} is the weight of the arc $a_{i.j}$, belonging to the path μ_1.

3.2. Cyberattack Threat Model

A cyberattack is a planned deliberate impact on the protected information resource, information infrastructure, technical means, or programs that solve the problem of receiving, transmitting, processing, storing, and reproducing protected information in order to cause characteristic functional or structural changes [39].

Targeted or structural changes in the critical infrastructure objects targeted by the cyberattack are to reduce the level of information and telecommunications security of the SG TDTN.

Let us represent the set of possible options for the implementation of the effects of a cyberattack on the TDTN in the form of a concatenation of the graph M (set of cyberattacks) and the graph R (set of PIRs), as shown in Figure 3.

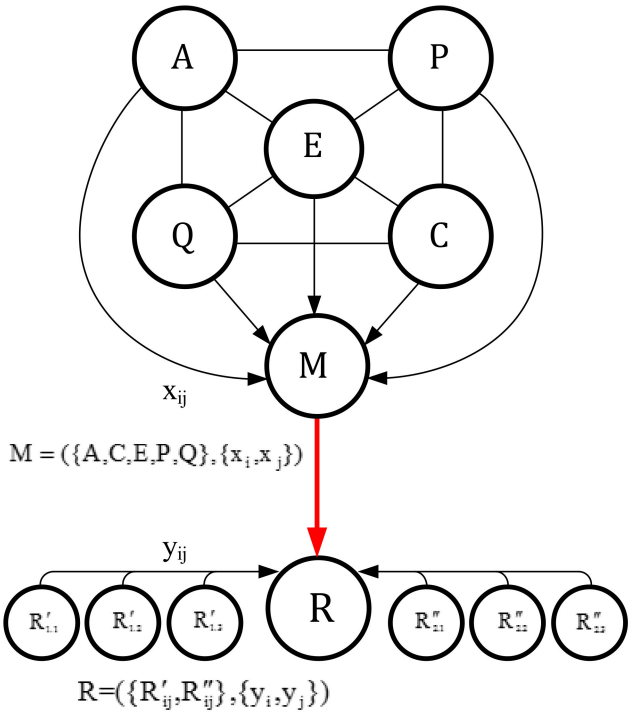

Figure 3. Context diagram of the cyberattack effects on the TDTN.

Let us represent $M = \{m_k, k = 1, 2 \ldots, K\}$ in the form of a set of possible options for the implementation of information influences on the TDTN. Let us introduce into consideration $\beta = \{b_n, n = 1, 2 \ldots, N\}$ is the set of private scalar indicators of the effectiveness of cyberattacks on the TDTN, for which the normalization condition is valid.

$$\sum_{n=1}^{N} b_n = 1, \tag{8}$$

where n is the private indicator identifier.

Let us denote $M^* = \{m_{k1} \Rightarrow m_{k2} \Rightarrow \ldots \Rightarrow m_{k\forall}\}$ as a subset of the preferred options for choosing an attacker's impact strategy, ordered by efficiency, where $\forall$ is the number of preferred options for implementing a cyberattack. A subset M^* is defined using the following expression:

$$(M, A, C, E, P, Q, \beta\} \underset{\varphi}{\rightarrow} M^*, \tag{9}$$

where φ is the rule of choosing the subset M^* from the set M, using the indicator β (optimality criterion).

In the context diagram presented in Figure 3, the vertex M is composed of vectors of the information and technical impact (E, Q, C), as well as the information and psychological (A, P) impact on the TDTN. Here E is the vector of the set of indicators of software and technical espionage (intelligence) in relation to the TDTN and protected resources, Q is the vector of the set of indicators of passive (providing) cyberattacks, C is the vector of the set of indicators of active cyberattacks on the TDTN resources (destruction, manipulation, blocking, substitution, and so on), A is the vector of the set of indicators of information tools and methods of influencing the personnel of the TDTN, and P is the vector of the set of indicators of the psychotronic and cognitive impact on the personnel of the TDTN. In the future, we will not consider the information-psychological impacts (vectors A and P), taking into account that for the SG this area of impacts refers to the limitations.

Let us represent the threat model for the implementation of cyberattacks in the form of a directed graph M, as shown in Figure 4.

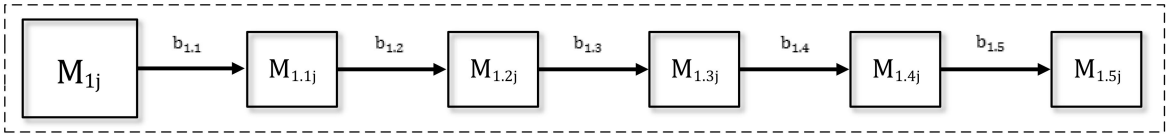

Figure 4. Oriented graph "Threats of the impact of the cyberattacks on the resources of the TDTN".

The state of the graph M is described by the expression

$$M = \left(M_{1j}, M_{1.1j}, M_{1.2j}, M_{1.3j}, M_{1.4j}, M_{1.5j}, b_{1.1}, b_{1.2}, b_{1.3}, b_{1.4}, b_{1.5}\right). \quad (10)$$

We will also use the following expression for shorthand:

$$M = \left(M_{ij}, B_{i.j}\right). \quad (11)$$

In the graph M, the vertices $M_{1.1j}$ is the object of the threat impact (element of the TDTN); $M_{1.2j}$ is a protected TDCT resource; $M_{1.3j}$ is a cyberattack implementation; $M_{1.4j}$ is a violation of the information security indicators; $M_{1.5j}$ is a violation of the information and technical characteristics of the TDTN.

Let us consider in more detail the components of the "Threats" graph. The top of the graph M_{1j} represents a set of threats influenced by the impact of cyberattacks. A variant of the classifier of the vertex M_{1j} is shown in Table 1.

Table 1. Classifier of the model of threats.

Intelligence Service	Penetration	Attack	Anchoring
$M_{1.1}$—vulnerability detection	$M_{1.4}$—activation of hidden malicious code (backdoor)	$M_{1.6}$—implementation of undeclared software capabilities	$M_{1.9}$—duplication and distribution of malware in the system
$M_{1.2}$—system analysis	$M_{1.5}$—exploitation of a vulnerability	$M_{1.7}$—copying PIRs	$M_{1.10}$—masking (hiding) malware in the operating system
$M_{1.3}$—exfiltration		$M_{1.8}$—destruction (substitution) of the PIRs	

By detailing this classifier, we will break down each of the stages of a computer attack into implementation options. So, when scouting the ports of the router's network services ($M_{1.1}$), the following actions are used: scanning network services ($M_{1.1.1}$), discovering peripheral devices ($M_{1.1.2}$), discovering network configuration parameters ($M_{1.1.3}$), and so on.

The classifiers of the components of the TDTN threat graph are represented by an undirected flat subgraph M_2. Thus, the formal model of threats (Figure 5) consists of a direct compositional sum ($\oplus$) of the graphs M_1 (directly the threat) and M_2 (classifiers of the components of the graph M_1), defined by the following expression:

$$\Gamma_M = M_1 \oplus M_2. \quad (12)$$

The vertices (classifiers) of the subgraph M_2 are united by the edge $b_{2.6}$, which means the characteristic dependence of the classifiers from each other. For example, the component of the classifier $M_{2.5tx}$ "TDTN throughput" has both forward and reverse connectivity with the component $M_{2.1tx}$ "border router of the open segment of the TDTN".

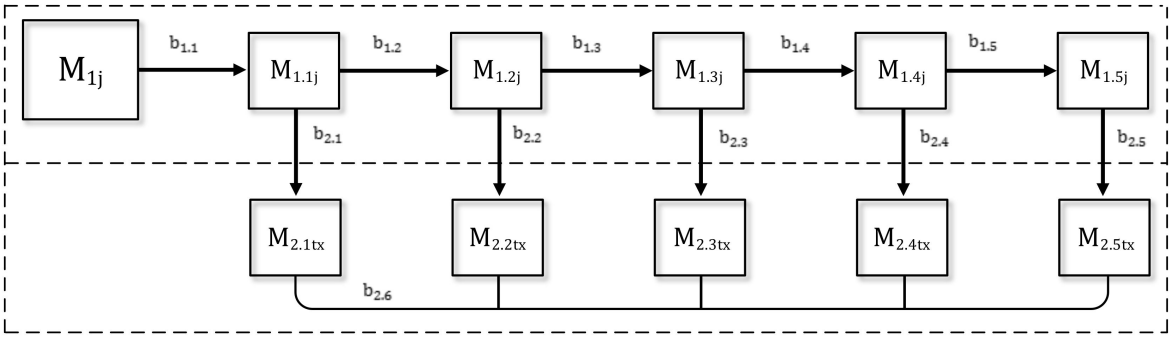

Figure 5. Formal model of cyberattack impact threats.

3.3. Protection System Model of the SG TDTN

The protection system model shown in Figure 6 is a kind of barrier between the graphs M and GT.

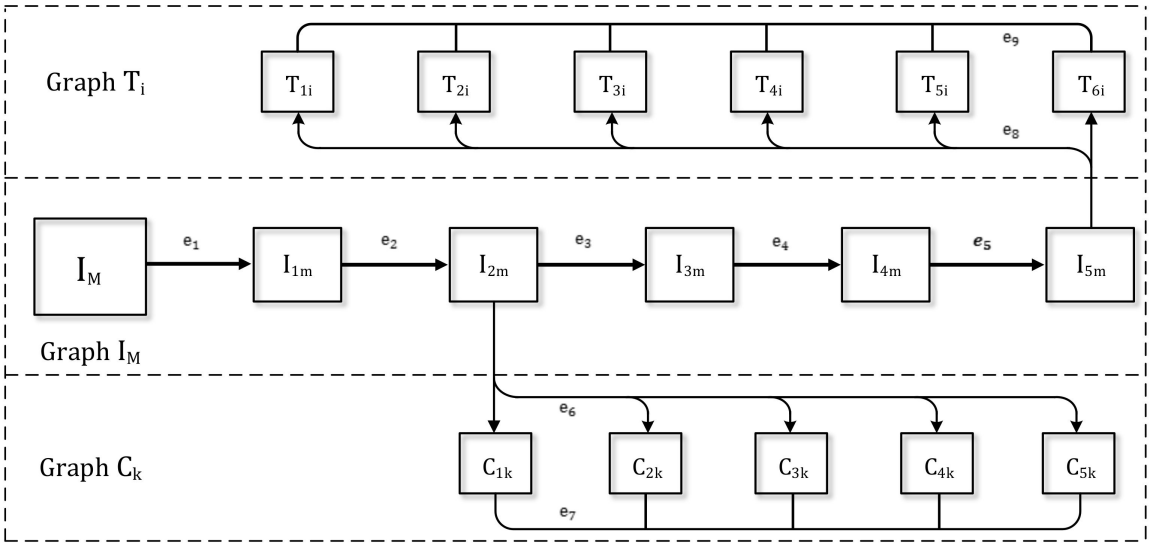

Figure 6. Formal model of the TDTN protection system.

In this model, three directed graphs can be distinguished: I_M, T_i, and C_k. Digraph I_M consists of the following nodes: I_M—threat identifier; I_{1m}—target of the cyberattack; I_{2m} —channels of influence of the cyberattack; I_{3m}—requirements for the elements of the TDTN protection system; I_{4m}—protection system architecture; I_{5m}—coefficient of efficiency of the TDTN protection means.

The digraph T_i contains, as the vertices, the efficiency coefficients of the TDTN protection means for each protected resource. The digraph C_k contains the channels of the threat's impact.

The possible channels of threats of the impact of the cyberattack on the resources of the SG TDTN are represented in Figure 7. The protection system model takes into account the following threat channels: physical, program, organizational, technical, and social.

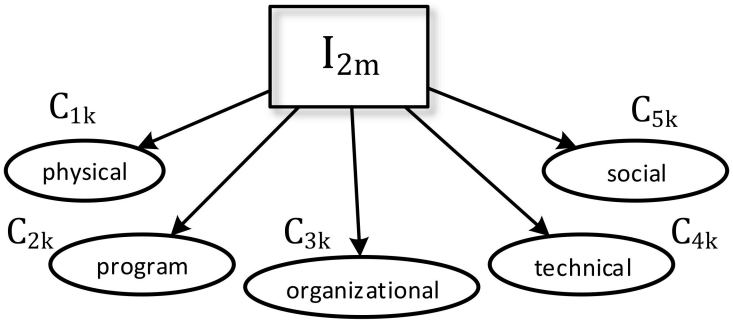

Figure 7. Possible channels of threats of the cyberattack impact on the resources of the SG TDTN.

Besides the subgraphs T_i and C_k, in the formal threat model, there is a bipartite subgraph D_g, which, with its vertices $\{D_{x5g}\}$, characterizes the requirements for the protection means of each PIR, and, with the vertices $\{D_{x5gi}\}$, denotes the chosen defense strategy that excludes the threat (Figure 8).

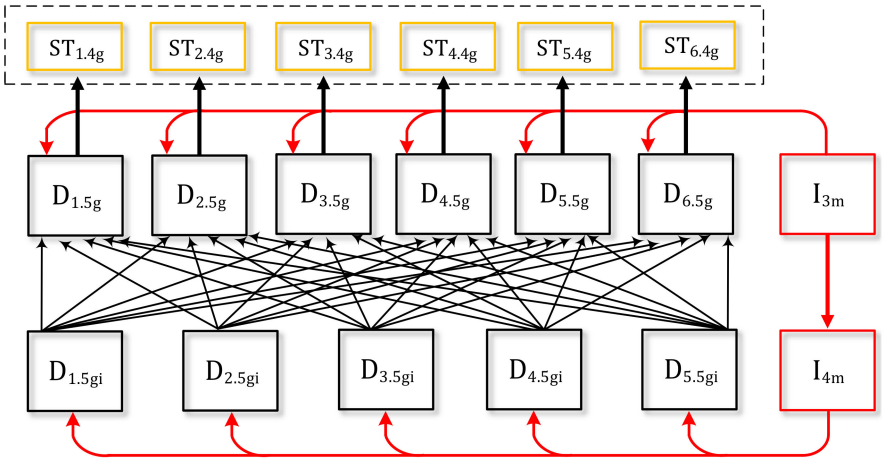

Figure 8. Bipartite subgraph of the choice of means of protection of the TDTN and the requirements for the means of protection of each information resource.

Thus, the formal model of the protection system not only constitutes a lexicological scheme with the digraph protected resources, but also contains a matrix of requirements for each individual asset protection tool [2,40].

3.4. Model of Functioning of the SG Protection System

Thus, in the model of functioning of the SG protection system, there are three digraph models: a threat model, a protection system model, and a protected information and telecommunication resource model.

The relationship of these digraphs is expressed by the concatenations

$$M \vdash I_M \vdash GT. \quad (13)$$

The concatenation operation means that there is only one arc that connects one vertex of each graph M, I_M, and GT to each other (Figure 9).

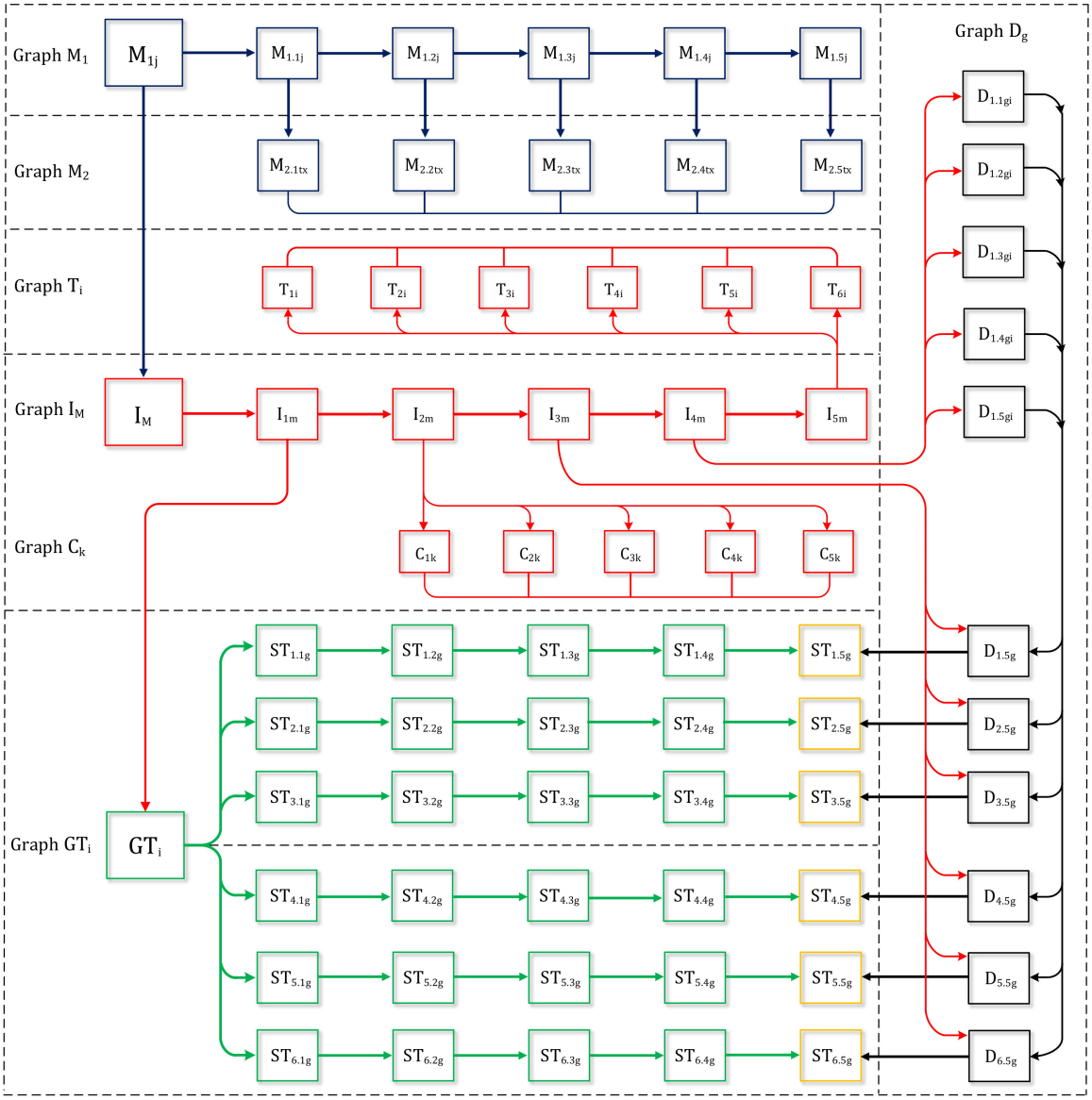

Figure 9. Formal model of "Cyberattack threat/TDTN protection system/Cyberattack recipient".

As seen from Figure 9, subgraphs D_g, C_{xk}, T_{xi}, and M_2 are connected by compositions with their generating digraphs.

Thus, the model of functioning of the SG protection system can be represented by the following expression:

$$(M_1 \oplus M_2) \vdash \{(I_M \oplus C_{xk})(I_M \oplus T_{xi})\} \vdash GT\perp (I_M \oplus GT \oplus D_g). \quad (14)$$

From Expression (14) it can be seen that I_M is a transit graph with respect to the graph M.

Thus, the model of the functioning of the SG protection system is developed as a model of functioning of complex opposing systems. Modeling is focused on solving the problem of distributing a heterogeneous resource among interdependent elements.

The model makes it possible to obtain qualitative assessments of the threat parameters and automate the assessment process, as well as dynamically recalculate the results obtained when the external environment or individual components and systems of the SG TDTN change.

3.5. A Technique for Using a LSTM Neural Network for Early Detection of Cyberattacks

LSTM neural networks are a subtype of the more general recurrent neural networks (RNNs). The main area of RNN usage as a deep learning model is applications with time series and sequential data [41]. LSTM neural networks extend the capabilities of traditional RNNs. They are highly effective in solving problems of classification and the forecasting of time series in conditions of a priori uncertainty of the boundaries of time intervals between events [42]. As a result, LSTM networks are successfully used in many areas related to anomaly detection, in particular, in speech recognition and generation [43–45], text document processing [46], intrusion detection [47], etc. Therefore, the idea of using LSTM networks is key in our work.

The key feature of the LSTM networks, such as RNNs as a whole, is their ability to store information or state of a cell for further use in the network (Figure 10).

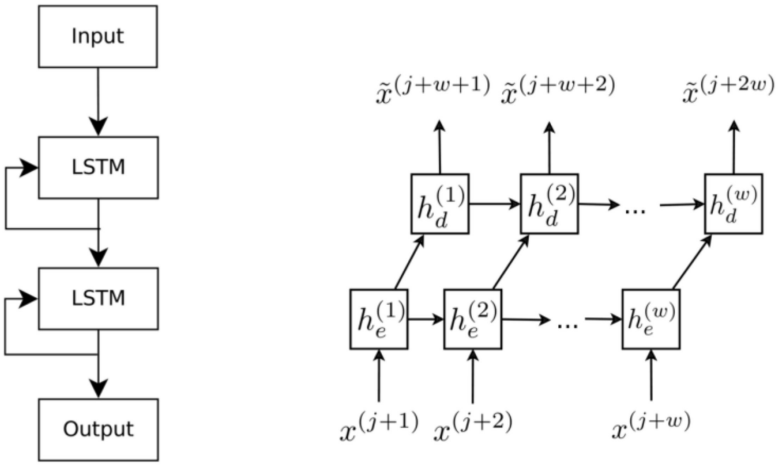

Figure 10. Neural network architecture and prediction scheme.

This makes them especially suitable for analyzing temporal data that changes over time. LSTM networks are used for tasks such as speech recognition, text translation, and in this case, for network anomaly detection.

LSTM can remove information from the cell state. This process is governed by structures called gates. Filters allow information to pass through based on certain conditions. They consist of a sigmoidal neural network layer and a pointwise multiplication operation. The sigmoid layer returns numbers from zero to one that indicate how much of each block of information should be passed down the network. Zero in this case means "do not miss anything", one means "pass everything".

Let us consider the LSTM cell operation algorithm step by step:

1. The information that can be removed from the cell state is determined. This decision is made by a sigmoidal layer called "the forget gate layer". It counts h_{t-1} and x_t and returns a number between 0 and 1 for each value from the cell state. As such, "1" means "keep completely" and "0" means "discard completely". The calculation is made according to the following formula: $f_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_f)$, where f_t—output for forget gate layer, σ—sigmoidal transfer function, W_i—weight for input layer gate, $[h_{t-1}, x_t]$—set-theoretic union operation h_{t-1} and x_t, b_f—offset for forget gate layer.
2. A decision is made about what new information will be stored in the cell state. This stage has two parts.
    a. First, a sigmoidal layer called "the input layer gate" determines which values to update: $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$, where i_t—output for input gate layer, b_i—offset for input gate layer.

b. Then the tanh layer builds a vector of new candidate values $\widetilde{C}_t$ that can be added to the cell state: $\widetilde{C}_t = \sigma\left(W_C \cdot [h_{t-1}, x_t] + \widetilde{b}_C\right)$, where $\widetilde{C}_t$—output for tanh gate layer, W_C—weight for tanh layer, $\widetilde{b}_C$—offset for tanh layer.

3. Updating the old cell state value C_{t-1} based on C_t: $C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t$, where C_t —state of cell t, $\widetilde{C}_t$—possible future state (candidate state) of cell t.
4. Generation of output data:
   a. Using the sigmoidal layer, it is determined what information will be output from the cell state: $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$, where o_t—output for output layer gate, W_o—weight for output layer gate, b_o—offset for the output layer gate.
   b. Cell state values are passed through a tanh layer to output values in the -1 to 1 range, and are multiplied with the sigmoidal layer outputs to output only the information required: $h_t = o_t \cdot \tan h(C_t)$, where o_t—output for output layer gate, $\tan h$—hyperbolic tangent.

3.6. Synthesis Technique for the SG Protection System

Despite the fact that the mathematical model of the SG protection system is very abstract, it has a certain advantage. To describe it, only one integral numerical indicator is required—the probability of making a timely error-free decision.

In addition, the integral probability of solving the tasks set by the system is considered in the model as an efficiency criterion, which corresponds to the target settings of the structure. In this case, the vector criterion in the factor space "efficiency—survivability" of the system is considered as an optimization criterion.

We mean that the system has two abstract states: a state when a timely error-free decision $p(t)$ is made, and a state when an erroneous and untimely decision $(1 - p(t))$ is made (Figure 11).

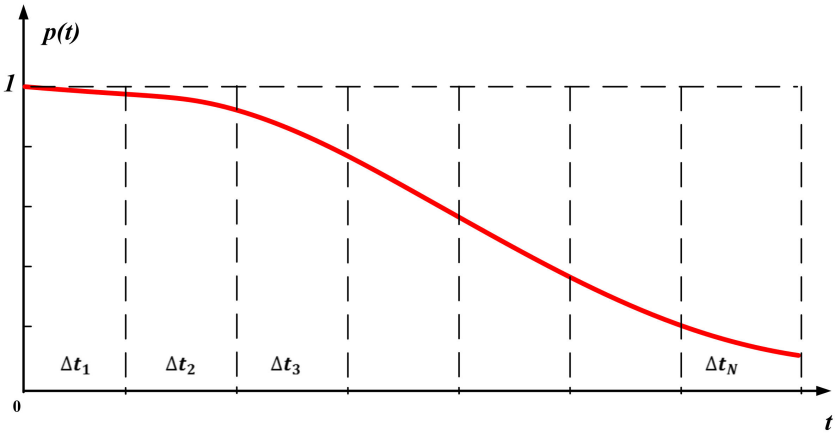

Figure 11. The process of evolutionary dependence of the survivability of the system on time.

Thus, the functional $\phi_1(p(t), T') = \frac{1}{2}\int_0^{T'} p(t)dt$ determines the average probability of a timely error-free solution on the time interval $(0, T')$. The mathematical expectation of the probability of a timely error-free decision determines the indicator of the survivability of the SG management system $\beta' = E(\phi_1(x(t), T'))$. The functional $\phi_2(p(t), T'') = \frac{1}{2}\int_0^{T''} (p(t) - a)_+ dt$, where the function $x_+ = \max(x, 0)$, $0 < a < 1$, determines the fraction of time spent by the process above level a, where $\beta'' = E(\phi_2(x(t), T''))$ determines the mathematical expectation of this fraction.

Based on this, the task of synthesizing the SG protection system can be formulated as follows: it is necessary to determine the structure at which

$$\min_{S(m_k)} C = f_1(S(m_k)), \ \max_{S(m_k)} P = f_2(S(m_k), \overline{\lambda}, \overline{\mu}, \overline{\varphi}(t), \overline{Q}, \overline{\varepsilon}, \overline{\delta}) \quad (15)$$

and restrictions on the main criterion functions are fulfilled

$$P \geq P_{min}; C \geq C_{min}, \tag{16}$$

besides supporting criteria $K_{3min} \leq K_{3k} \leq K_{3max}$, $K_{pk} \leq K_{pmax}$, $\forall k \in \{1, \ldots, S\}$, respectively, mean the vectors of the intensity of the tasks coming for processing to each element of the protection system, the vectors of the intensity of their solution and the vectors of the lifetimes of situations containing the indicated tasks; vectors Q and ε denote the corresponding probabilities of an erroneous solution of tasks by the elements of the protection system, determined by their functional Q and group tasks, in accordance with the nature of the coordination links in the system, determined by the matrix δ.

The problem posed is a two-parameter problem of vector optimization with mutually opposite criteria in the factor space "efficiency—survivability". Its general solution can be found through the use of various methods of scalarization of vector criteria. The essence of the most effective of them comes down to the standardization of criteria and their subsequent additive convolution.

To determine the values of the normalizing criteria for cost and efficiency, direct and inverse optimization problems are considered. The statement of the direct problem is reduced to the following: it is necessary to determine

$$\max_{S(m_k)} P = f_2(S(m_k), \overline{\lambda}, \overline{\mu}, \overline{\varphi}(t), \overline{Q}, \overline{\varepsilon}, \overline{\delta}) \tag{17}$$

under restrictions on the main criterion functions and under the restrictions on the auxiliary functions considered above. The inverse problem can be formally represented as follows:

$$\min_{S(m_k)} C(S(m_k)), C = C_1 + \sum_{k=1}^{S-1} m_k C_k \tag{18}$$

with the appropriate restrictions on the main criterion and auxiliary functions. In view of the significant nonlinearity of the criterial function $P(\cdot)$ in synthesis problems in the first approximation, it is more convenient to use the parameter t of the total time losses for solving operational problems.

Then the direct synthesis problem will be as follows:

$$\min_{S(m_k)} t\big(S(m_k), \overline{\lambda}, \overline{\mu}, \overline{\varphi}(t)\big). \tag{19}$$

At the same time, restrictions on the main criterion auxiliary functions should be carried out. In this case, the criteria $t(\cdot)$ and $P(\cdot)$ and, consequently, the solution of the problem, are determined either analytically under the conditions of sufficiently strict constraints, or by the method of statistical modeling.

Direct and inverse problems can be solved within the framework of one optimization procedure, during the implementation of which both the first and the second functionals are fixed. Analytical methods of calculations have a number of indisputable advantages, which include their simplicity, clarity, and the ability to effectively interpret the results. On the basis of statistical calculation methods, data can be obtained that describe the main trends in the changes in the structure of the management system of the SG protection system.

4. Implementation Issues of the SG Protection System Management Methodology

4.1. General Description of the SG Protection System Management Methodology

The proposed methodology for the SG protection system management in conditions of cyberattacks contains three stages.

At the first, auxiliary, stage, the SG information resources and its protection system are modeled, both in conditions of cyberattacks and without them. The modeling is performed by building and using formal models. As a result of this simulation, the values of the

transition probabilities are determined. This stage can be called the learning stage. To determine the transition probabilities, the elements of the planar graphs considered above are used.

At the second, main stage, the prediction and detection of cyberattacks are carried out based on the use of a neural network with LSTM. Based on the machine learning methods, the analysis of transition probabilities, as well as anomalies in the functioning of thw SG elements and its protection system, caused by the impact of cyberattacks, is carried out. The recurrent LSTM neural network detects anomalies using a threshold value.

At the third stage, based on predicting and detecting cyberattacks, decisions are made to change the logical structure of the SG protection system and assess the impact of these changes on the overall SG survivability, which can be determined using one of known methods [48].

Let us assume that the TDTN protection system uses m types of protection means and n copies of each ones. Then the probability P_{det} of reaching the k-th PIR by the attacker is determined by the formula of total probability as a result of solving the following statistical problem:

$$P_{det} = \sum_{i=1}^{n} P(H_i) Z_i\left(N_i^Z\right) + \sum_{j=i+1}^{2^n} P\left(H_j\right) Z_j\left(N_j^Z \ldots N_n^Z\right), \quad (20)$$

where N_i^Z is a number of crucial nodes of the TDTN connected with PIR of the i-th type, exposed of the cyberattack; N^Z is a total number of crucial nodes of the TDTN connected with all of the protected PIRs; $P(H_i)$ is the probability of the hypotheses about the achievement of a cyberattack of the protected PIR of the i-th type; $Z_i\left(N_i^Z\right) = \frac{(N^Z - N_i^Z)}{N^Z}$ is the weight of the i-th protection mean for PIR; $Z_j\left(N_j^Z \ldots N_n^Z\right) = Z_{maxj} + \frac{\sum_{j=1}^{m-1} Z_j}{n-m+1}$ is the total weight of the protective mean used in the SG TDTN protection system.

Basing on the fact that the TDTN protection system includes n protection means, the complete group of events of the cyberattack on a particular protection means will be determined by a set of hypotheses H_i, the total number of which is 2^n.

4.2. Software Implementation

To calculate the strong and weak components of the composition graph, we define the correspondence:

$$G = (X,\ A), \quad (21)$$

where X—nodes of the graph G, denoting states, A—arcs of the graph G, specifying connections between states.

The mixed asymmetric graph G in accordance with Expression (21) is shown in Figure 12. We use G as a knowledge model, which is information about the complete alphabet of events (signs of threats, security criteria, incidence of events, and strong and weak components). We also take into account the weight coefficients of the arcs of the graph G in the knowledge model:

$$L_{(\alpha)} = \sum_{(x_i, x_j) \in \alpha} z_{ij}, \quad (22)$$

where L is the total weight of paths of the mixed antisymmetric graph G, z_{ij} is the arc (edge) weight, α is the cardinality of the path.

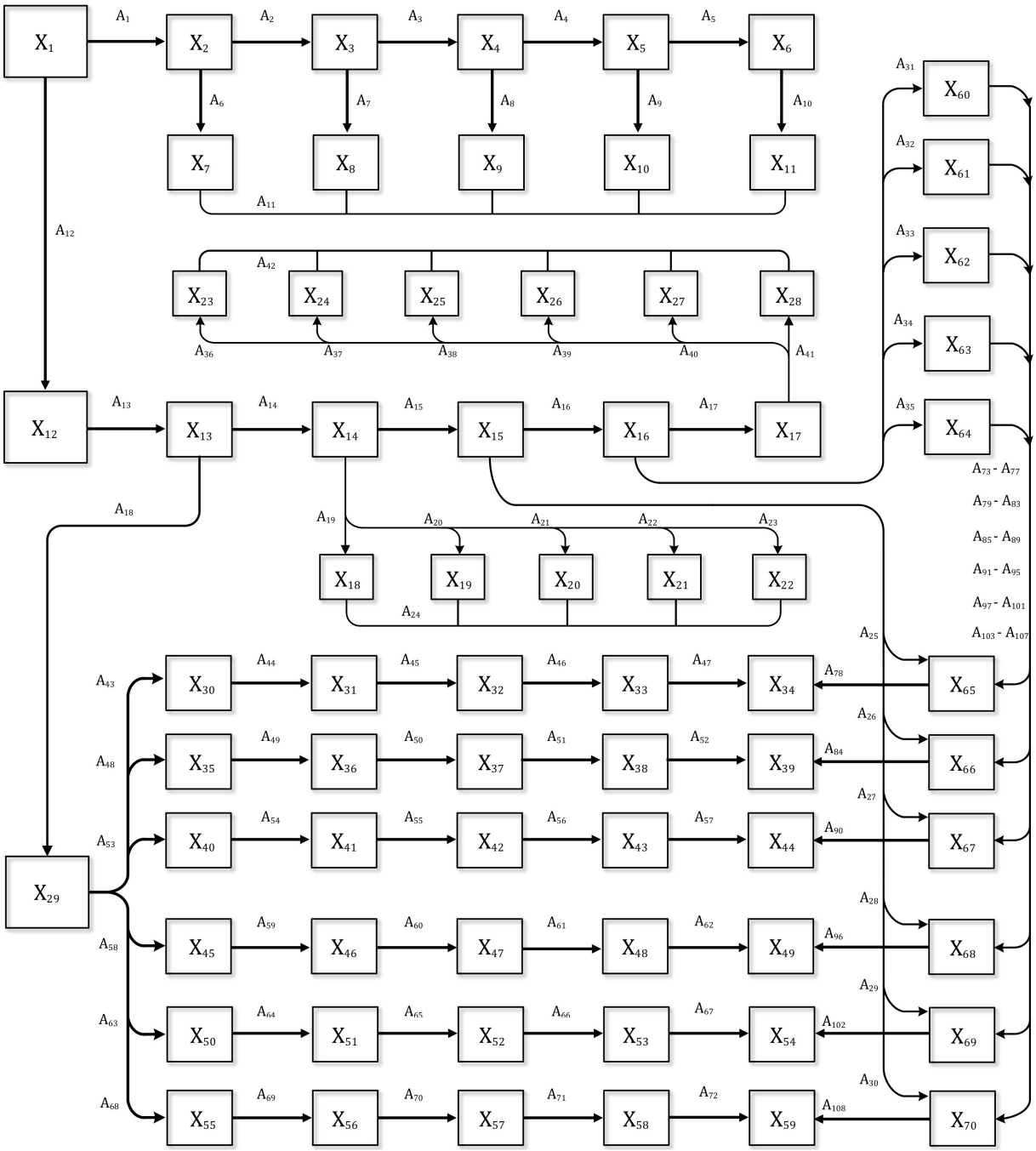

Figure 12. Mixed asymmetric graph $\mathbf{G} = (\mathbf{X}, \mathbf{A})$.

The mixed asymmetric graph G, shown in Figure 12, reflects the concatenation of the graphs of the "Threat Models" (Figures 4 and 5), "Protection Systems" (Figures 6–8), and "Protected Resources" (Figure 1). This transformation allows us to highlight the strong and weak partial subgraphs of the graph G; adjacent arcs; and also set the cost of each oriented route that is necessary to select the optimal security policy.

The software for the proposed methodology is implemented in Python using the Pandas library, which was used to process and analyze the data. The Pandas library is written in the C, Cython, and Python programming languages. The presented library makes Python a powerful tool for data analysis and makes it possible to build pivot tables, perform groupings, and provide convenient access to tabular data at a high level. In addition to the Pandas library, the NumPy library was used, which is a lower-level tool that provides work with high-level mathematical functions, as well as multidimensional arrays

(tensors). In general, the software implementation is based on the iterative optimization method. This approach is that each node x of the graph $G = (X, A)$ is associated with a sequence of states: $s_{x,t} \in S^n$, $t \in \{0, \ldots m\}$. The states are updated according to the following expression:

$$s_{x,t+1} = F(s_{x,t}, \sum(x, a) \in G, M(s_{x,t}, s_{a,t}, w_{x,a})). \quad (23)$$

In other words, at the first iterations, for each edge a of the graph $G = (X, A)$, the alphabet of events is calculated using the function M (the event depends on the states of the nodes $s_{x,t}$ and the weights of the edges $s_{a,t}$), and then all of the events are summed and the state of the node is updated using the function F (both functions are parameterized as usual with variable learning parameters). In our case, function F is implemented by LSTM. Given that, the algorithms for traversing the knowledge model G take into account the differences between the types of links and estimate the weights of different links $L_{(\alpha)}$ from the point of view of the problem of ensuring the security of the protected SMART resources, and do not use a common set of weights that was initially identified during training on the graph.

The LSTM artificial neural network used in the experiment is organized in accordance with the scheme shown in Figure 13.

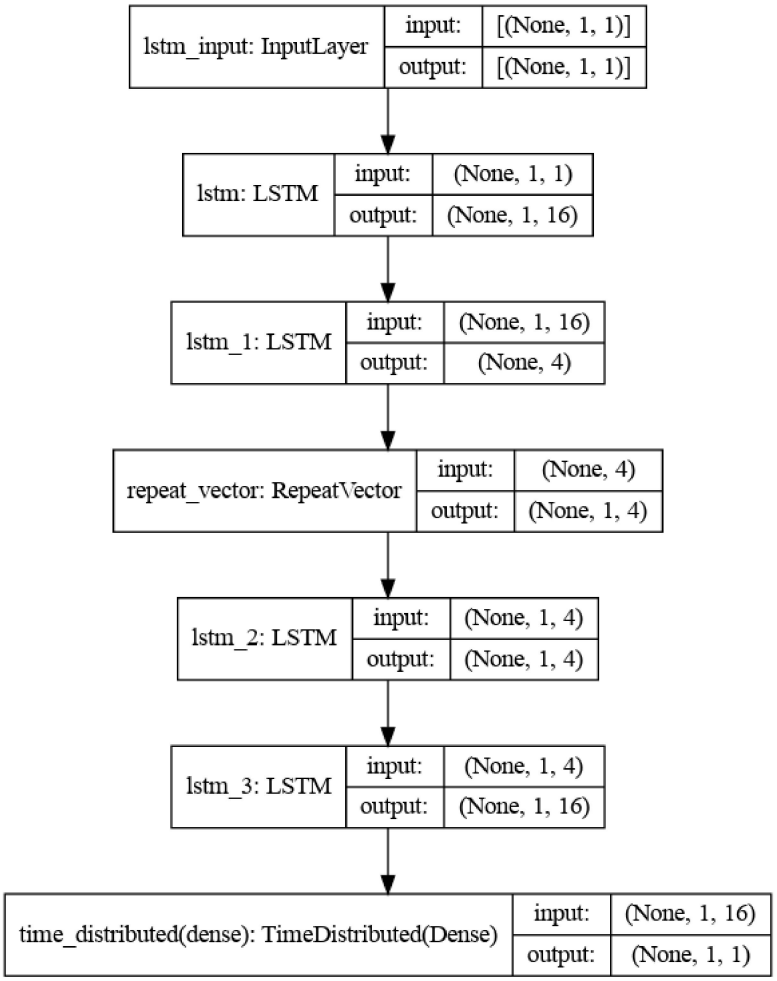

Figure 13. Diagram of the LSTM artificial neural network.

Hyperparameters of an artificial neural network are configurable parameters that allow one to control the learning process of the model. For example, in neural networks, you determine the number of hidden layers and the number of nodes in each layer. The performance of the model largely depends on the hyperparameters. Hyperparameter

tuning, also called hyperparameter optimization, is the process of finding a hyperparameter configuration that leads to better performance. This process usually requires significant computing resources and is performed manually. The presented hyperparameters of the neural network are optimized for our experiment and allow us to configure the network on a limited dataset to search for impacts on the network, to ensure the adequacy of the model.

Hyperparameters of the experimental neural network are: module optimizer—"Adam"; loss function (average absolute error)—"MAE"; the size of the data array for LSTM training is 10; the number of training epochs is 10; activation function (input layer)—"Than"; activation function (output layer)—"Relu"; the number of layers is 7; the dimension of the input/output tensor is 3/3.

The values of the hyperparameters for LSTM are determined a priori. With well-selected hyperparameter values, the LSTM network detects well-known computer attacks with a probability close to 1, and unknown attacks—with a probability exceeding 0.8.

Machine learning methods are implemented using the scikit-learn library, and neural networks are implemented using the Keras framework. The graphs were built using the Matplotlib module based on the obtained dataset. All calculations were performed in the Jupiter notebook integrated development environment. Simulation modeling based on GNS3 software was used to generate traffic.

Cyberattacks, such as distributed denial of service (DDoS), reconnaissance, backdoor, exploits, fuzzers, and "scanning the network and its vulnerabilities", were taken into account as implemented attacks. Considering the above, the traffic structure, packet header length, flags, checksum, and some others were considered as the main characteristics under study in the dataset.

For the experiment, using the IXIA Perfect Storm tool (Figure 14), a special dataset was generated. It includes the reference traffic and was used to train the system and analyze the traffic without anomalies and the abnormal traffic.

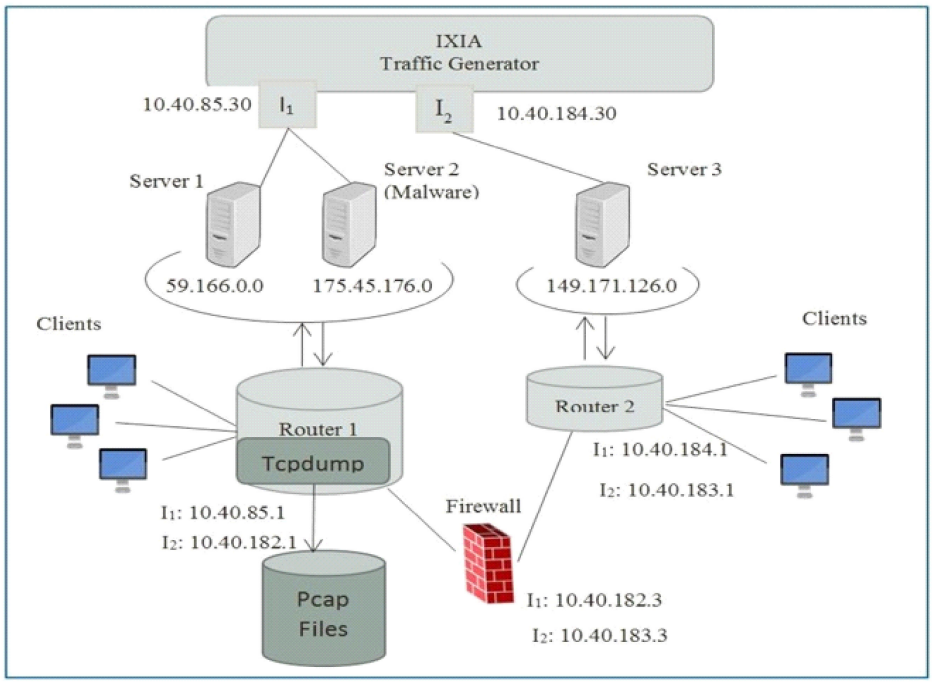

Figure 14. Tool for software implementation of the "IXIA Perfect Storm" cyberattack.

Abnormal traffic included 55.0% of cyberattacks, which are divided into ten types, namely: fuzzers (ε_1), analysis (ε_2), backdoors (ε_3), DDoS (ε_4), exploits (ε_5), generic (ε_6), reconnaissance (ε_7), shellcode (ε_8), worms (ε_9), and "scanning the network and its vulnerabilities" (ε_{10}).

Cyberattacks were carried out for the following protocols: dns, http, smtp, ftp, ftp-data, pop3, ssh, ssl, snmp, dhcp, radius, and irc. They were used to test the effectiveness of the method under consideration and to identify its merits over other methods.

Determination of the efficiency criterion for the recognition of the cyberattacks (μ_P) by the neural network is performed by the formula

$$\mu_P = \sum_{k=1}^{l} \sum_{m=1}^{n} P_{PA_k}\left(\varepsilon_{Y_k}^{A_k} y_k\right) K\left(\varepsilon_{Y_m}^{A_m}\right), \quad (24)$$

where $k = 1, \ldots, l$ is a number of supposed types of cyberattacks $M = \left(M_{ij}, B_{i,j}\right)$ (see Table 1); $m = 1, \ldots, n$ is a number of the supposed attacked SG resources—subgraphs $GT_{i1} - GT_{i6}$ (see Figures 3–6); P_{PA_k} is a probability of the error-free cyberattack recognition, $k = 1, \ldots, l$; $\varepsilon_{Y_k}^{A_k}$ is a set of the types of the cyberattacks on SG resources under a priori alphabet of attack events (A_k) and a priori alphabet of signs of cyberattacks (Y_k); $\{y_k\}$ are the vectors of signs of cyberattacks, $k = 1, \ldots, l$; $K\left(\varepsilon_{Y_m}^{A_m}\right)$ is a gain function from recognizing attack targets ϑ_{Y_n}, which are classified in the alphabet of cyberattacks.

Next, using the Argus and Bro-IDS tools, the data was analyzed and tagged into 44 features with a class label. The total number of records is equal to 82,332. Table 2 shows a sample from the resulting dataset with the indication of the main features.

Table 2. Dataset of network traffic (a sample).

Record Number	Dataset Features												Attack Category
	service	*state*	*dpkts*	*sbytes*	*dbytes*	*ct_dst_src_ltm*	*is_ftp_login*	*ct_ftp_cmd*	*ct_flw_http_mthd*	*ct_src_ltm*	*ct_srv_dst*	*is_sm_ips_ports*	**label**	
1	ospf	INT	20	0	1280	1	2	0	0	0	1	1	0	Reconnaissance
2	ospf	INT	20	0	1280	1	2	0	0	0	1	1	0	Reconnaissance
3	ospf	INT	20	0	1280	1	2	0	0	0	1	1	0	Backdoor
4	ospf	INT	20	0	1280	1	2	0	0	0	1	1	0	DoS
5	sctp	INT	2	0	104	1	2	0	0	0	1	1	0	Exploits
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
82328	udp	INT	2	0	1510	1	1	0	0	0	1	5	0	Fuzzers
82329	udp	INT	4	0	1216	1	1	0	0	0	1	6	0	Fuzzers
82330	udp	INT	4	0	1216	1	1	0	0	0	1	6	0	Fuzzers
82331	tcp	FIN	10	6	590	1	1	0	0	0	1	5	0	Fuzzers
82332	tcp	FIN	10	6	590	1	2	0	0	0	2	4	0	Fuzzers

The following features are used in Table 2:

service—a service used (may be as *ospf*, *http*, *ftp*, *smtp*, *ssh*, *dns*, *ftp*, *udp*, *tcp* etc. or '-');
state—indicates the state and its dependent protocol (ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, and URN or '-' if the state is not used);
dpkts—the number of packages from the destination to the source;
sbytes—the number of bytes in the transaction from the source to the destination;
dbytes—the number of bytes in the transaction from the destination to the source;
ct_dst_src_ltm—the number of connections between the same destination address and the initial port in the last 100 connections;
is_ftp_login—is equal to 1 if the user has access to the file transfer protocol (FTP) session by password, otherwise—0;
ct_ftp_cmd—the number of connections that have a command on the FTP session;
ct_src_ltm—the number of connections with the same source address in the 100 last connections;

ct_srv_dst—the number of compounds containing the same service and destination address in the last 100 connections;
is_sm_ips_ports—if the source and destination IP addresses are equal and the port numbers are equal, then this variable is 1, otherwise—0;
label—is equal 0 for normal and 1 for attacked records.

In the dataset, there are nine attack categories: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms.

The Tcpdump tool was used to capture raw, unprocessed traffic. Cyberattacks were carried out using the KaliLinux distribution kit.

As the scenario under study, the traffic corresponding to the SG TDTN of St. Petersburg (Russia) was selected. This network contains 50 high-voltage substations (substations 110–220 kV), 2200 distribution points and transformer substations, as well as more than 80,000 metering devices (Figure 15).

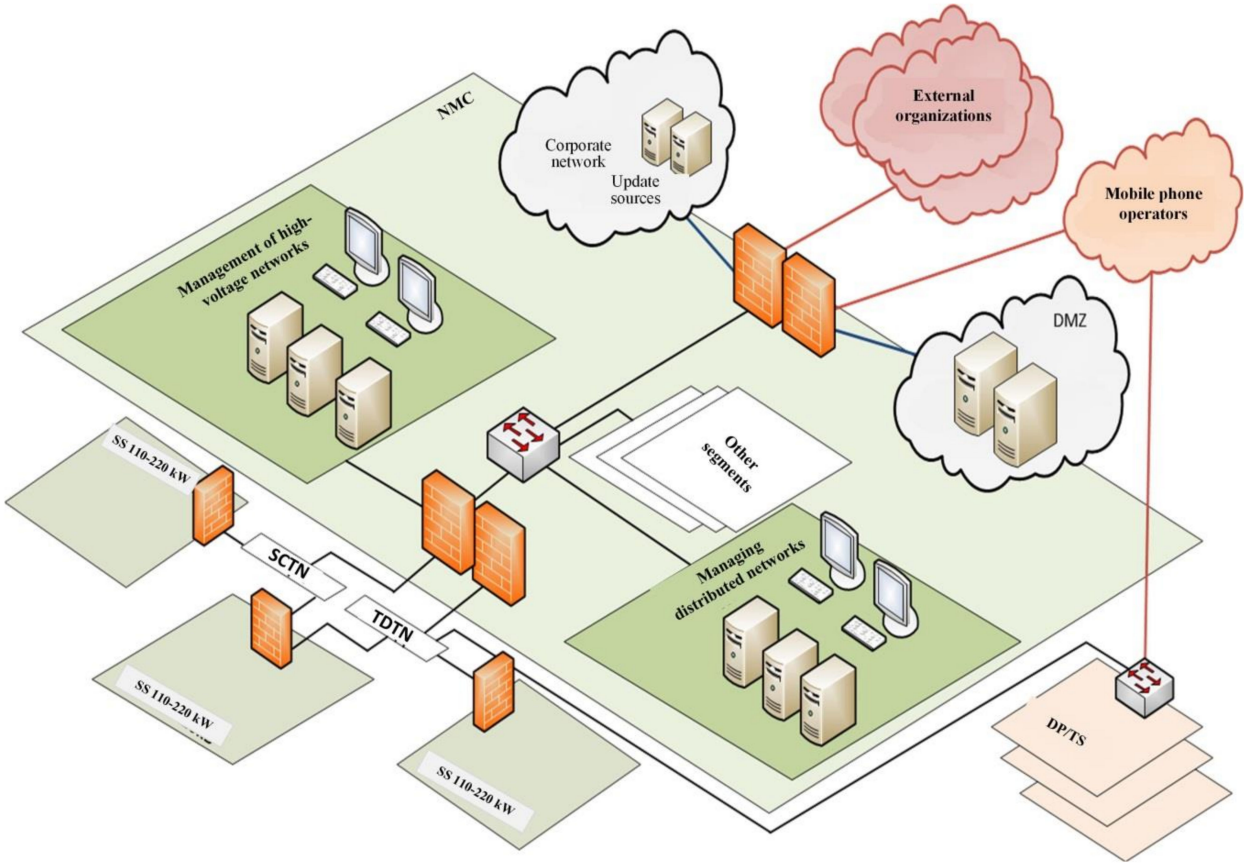

Figure 15. Functional diagram of the metropolitan SMART grid.

The management of the TDTN elements and management of the network security is carried out from the central communication node (CCN).

The simulated traffic was a set of data of interest to operators and dispatchers of the SG power system. This data contained the following parameters:

- Power factor values;
- Power quality parameters in the entire system;
- Distributed measuring system parameters;
- Equipment condition parameters;
- Parameters of the state of protection means;
- Information about the places of damage and failures;
- Load of transformers and lines;
- Profiles and forecasts of electricity consumption and some other parameters.

5. Experimental Results

5.1. Evaluation of the Management Methodology for the SG Protection System

To solve the problem associated with predicting and detecting a cyberattack, it is very important not only to determine machine learning algorithms or neural networks, but also to highlight the parameters that are most susceptible to anomalous deviation during the course of the attacker's influence.

Consider a particular case of using the graph $G = (X, A)$ by a LSTM network. Let us construct a transitive adjacency graph G' to prevent a DDoS attack. Based on the threat model for SG resources, the classifier of the threat model $X_7(X_{7.1}, X_{7.2}, X_{7.3}, X_{7.4}, X_{7.5})$ contains from 1 to n threats. Let $X_{7.1}$ be a DDoS cyberattack type that affects the protected SG resources of the "information availability" (X_{44}) and "network bandwidth" (X_{59}) types. Let us formulate a context diagram for a particular case of a cyberattack on the SG resources $G' = (X', A')$, where $X_1, X_2, X_3, \ldots, X_{70} \in X'$ and $A_1, A_2, A_3, \ldots, A_{51} \in A'$ (Figure 16).

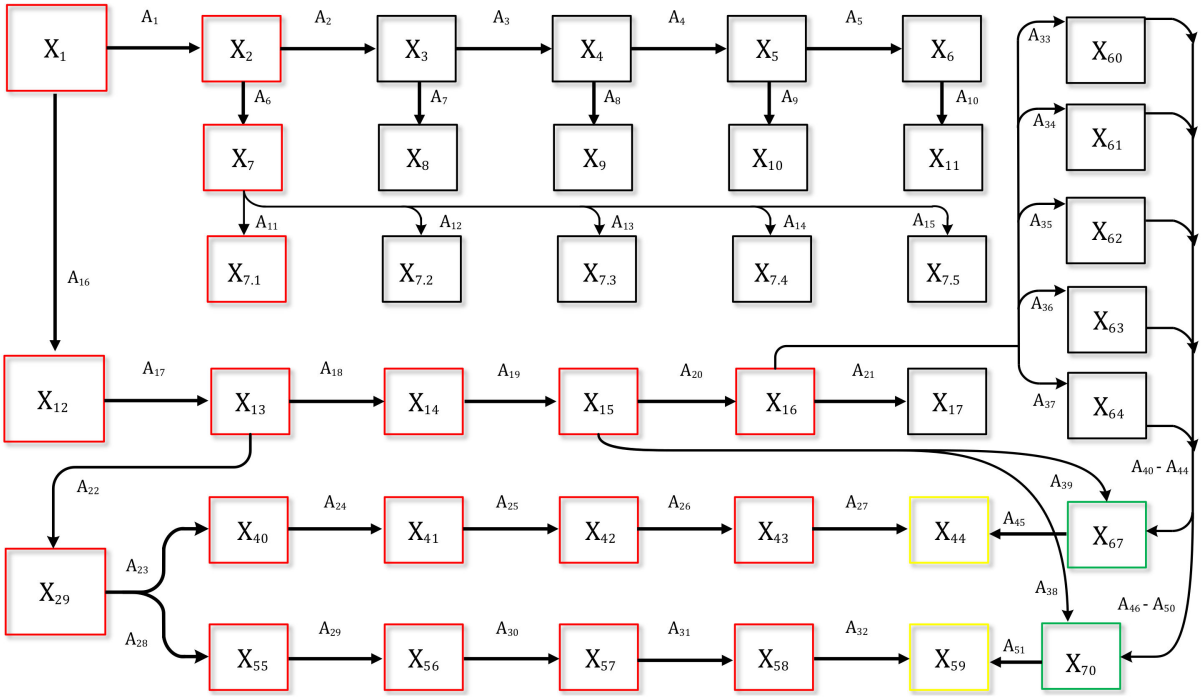

Figure 16. Mixed asymmetric adjacency graph $G' = (X', A')$.

After detecting and classifying the attack on the SG resources, the recurrent neural network compares the attack countermeasures. This involves the elements of the D_g subgraph (see Figure 8). The elements of the bipartite subgraph D_g define the protection strategy $X_{16}(X_{60}, X_{61}, X_{62}, X_{63}, X_{64})$ of the SG resources X_{44} of "information availability" and X_{59} of "network bandwidth" against attacks such as $X_{7.1}$ (attacks such as DDoS) based on the requirements of $X_{15}(X_{67}, X_{70})$ to the protection system to prevent attacks such as $X_{7.1}$ (Figure 17):

$$X_{44} \oplus X_{67} \oplus X_{16}(X_{60} \ldots X_{64}). \quad (25)$$

When detecting the DDoS cyberattack, each protocol was initially considered separately based on the dataset of the network traffic (Figures 18–24).

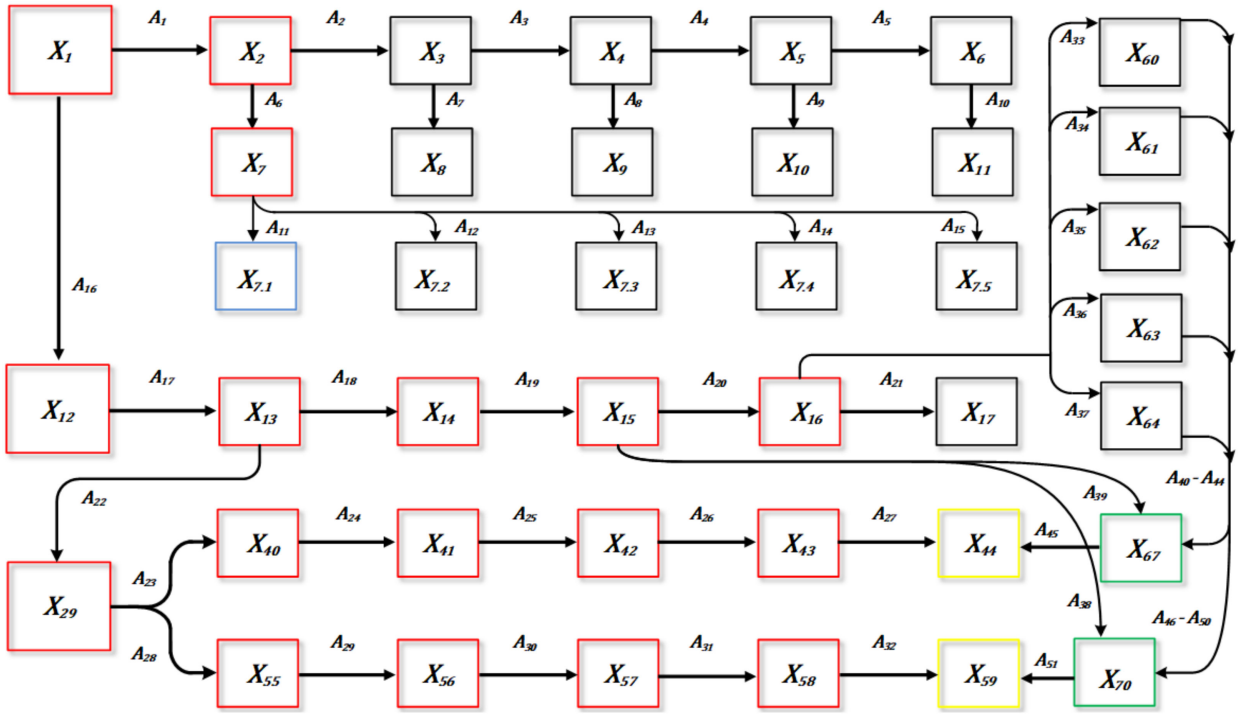

Figure 17. Implementation of the defense response against exposure to SG resources for the graph $G' = (X', A')$ to prevent DDOS attack.

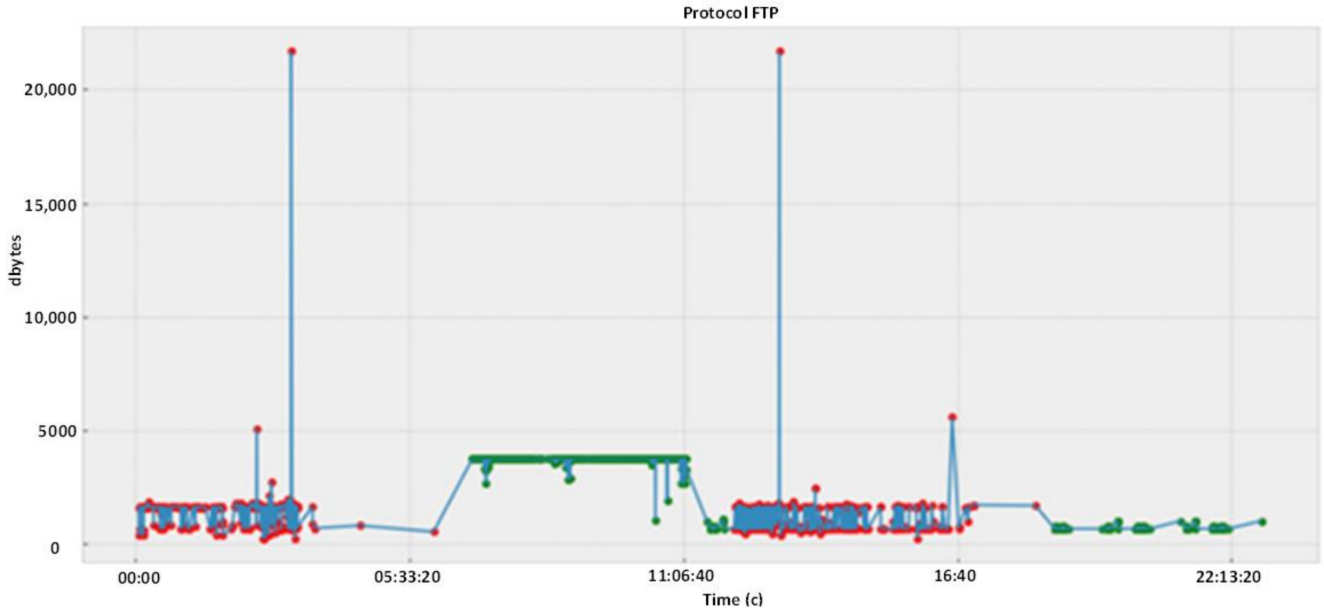

Figure 18. FTP protocol.

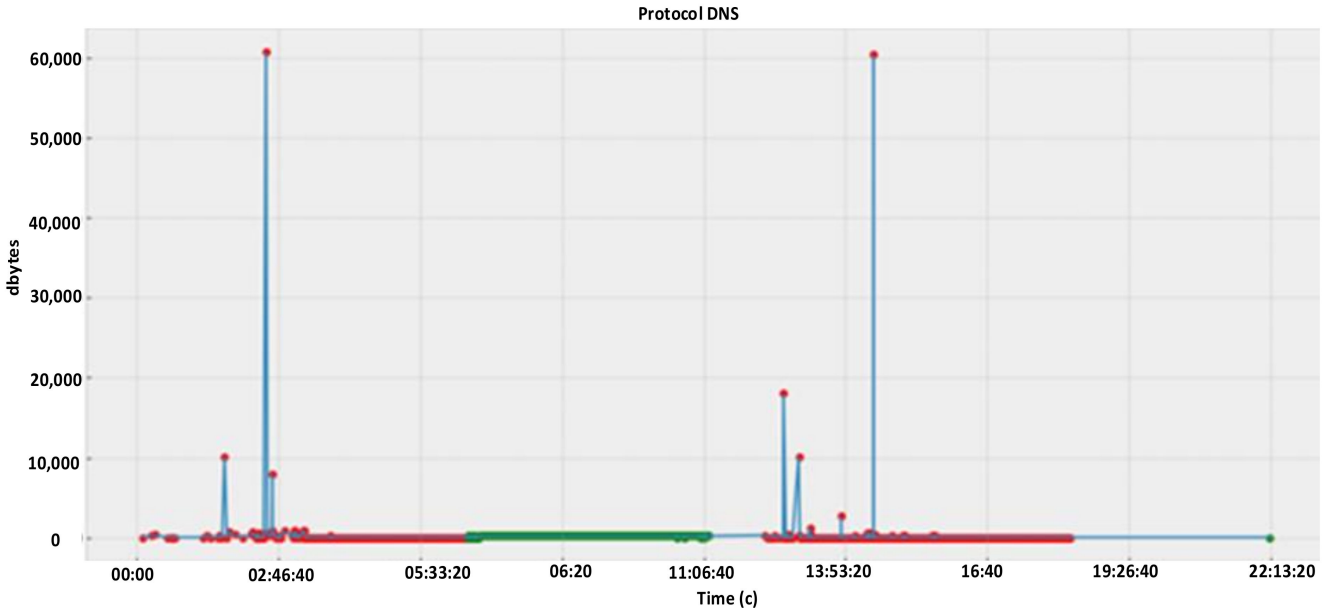

Figure 19. DNS protocol.

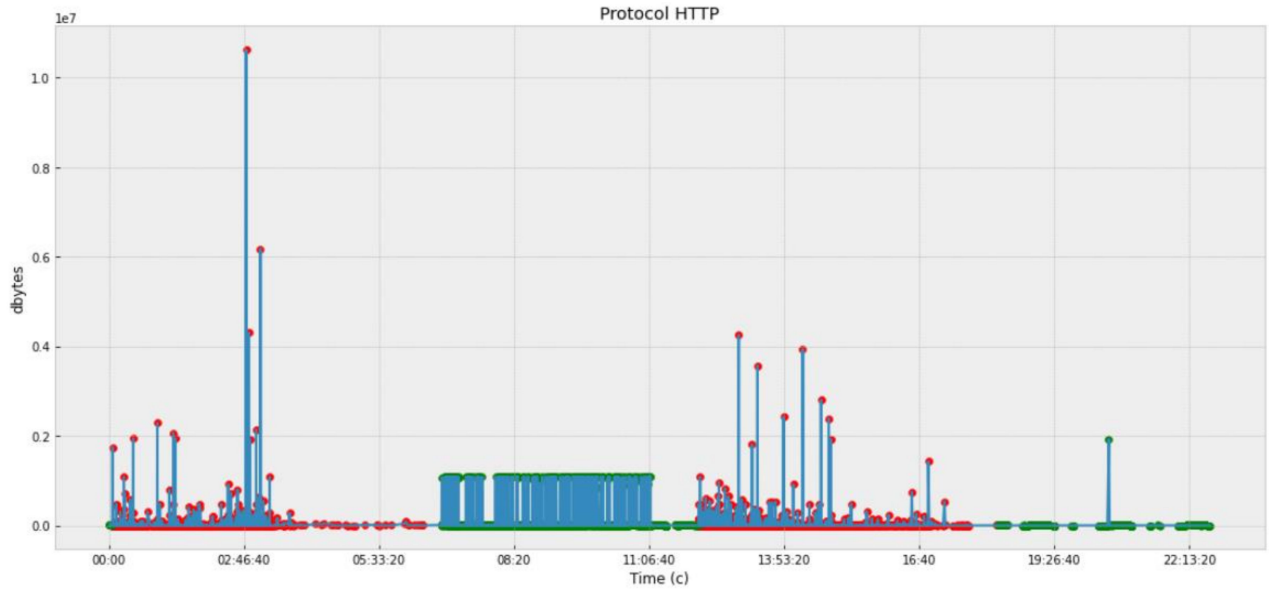

Figure 20. HTTP protocol.

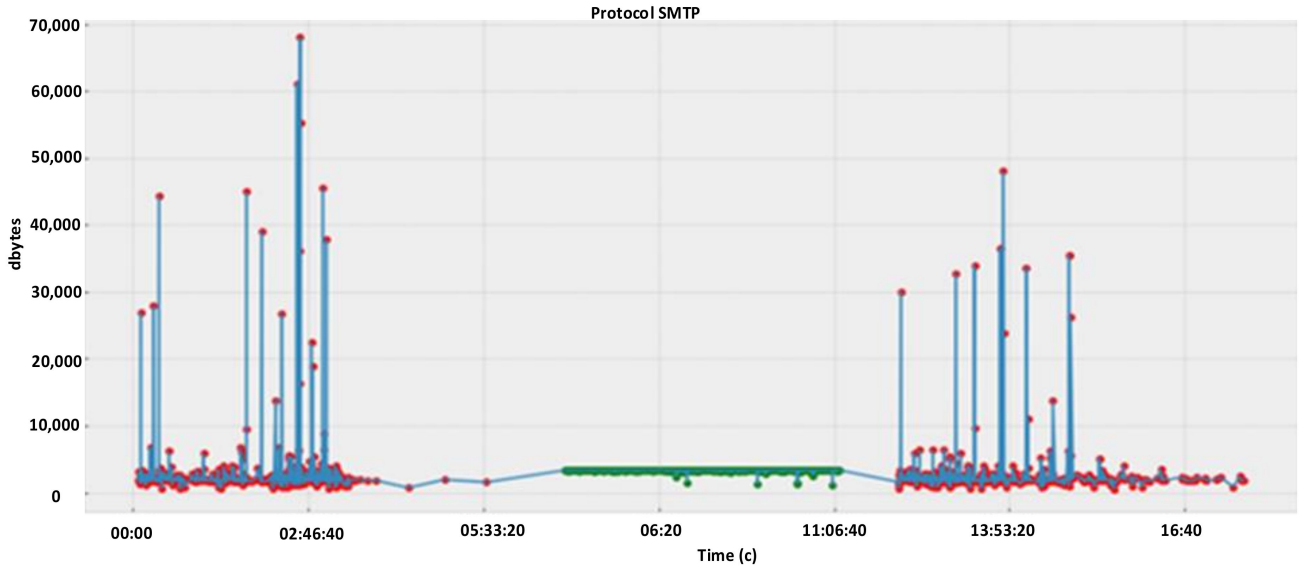

Figure 21. SMTP protocol.

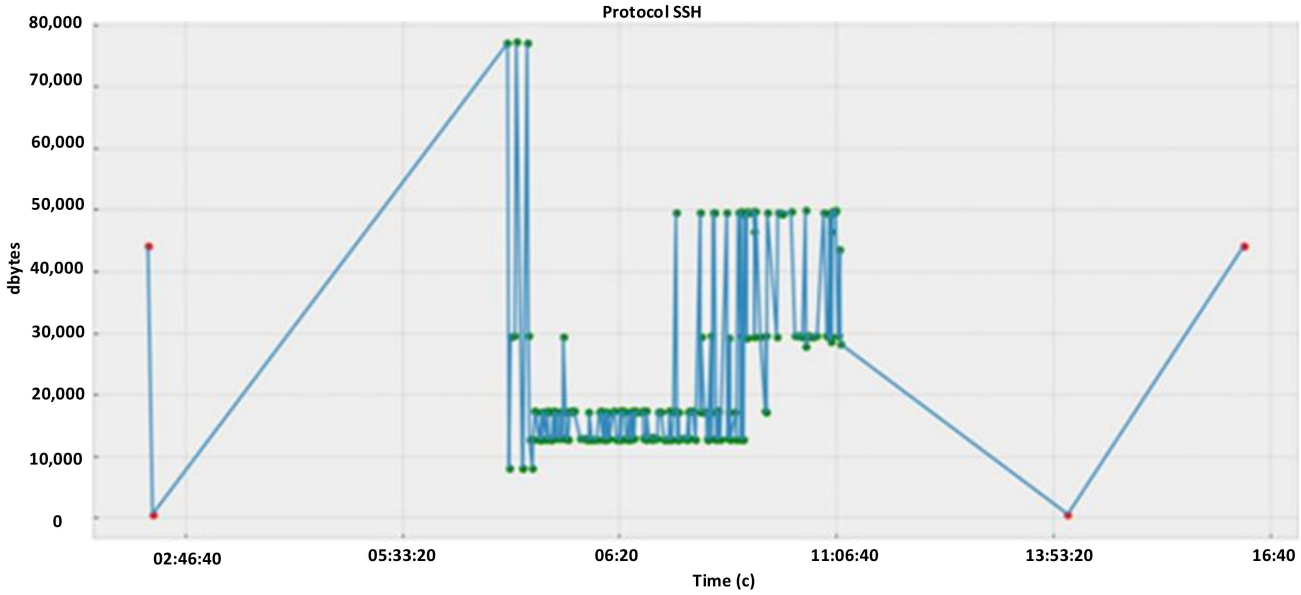

Figure 22. SSH protocol.

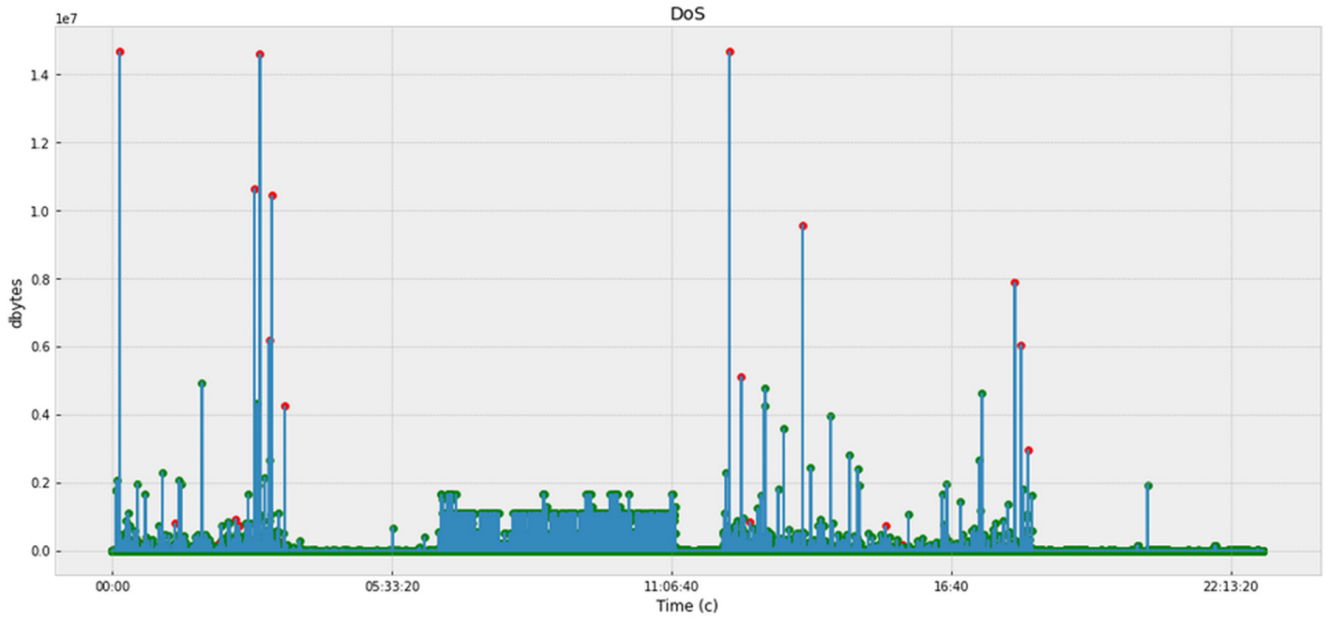

Figure 23. DDoS attack.

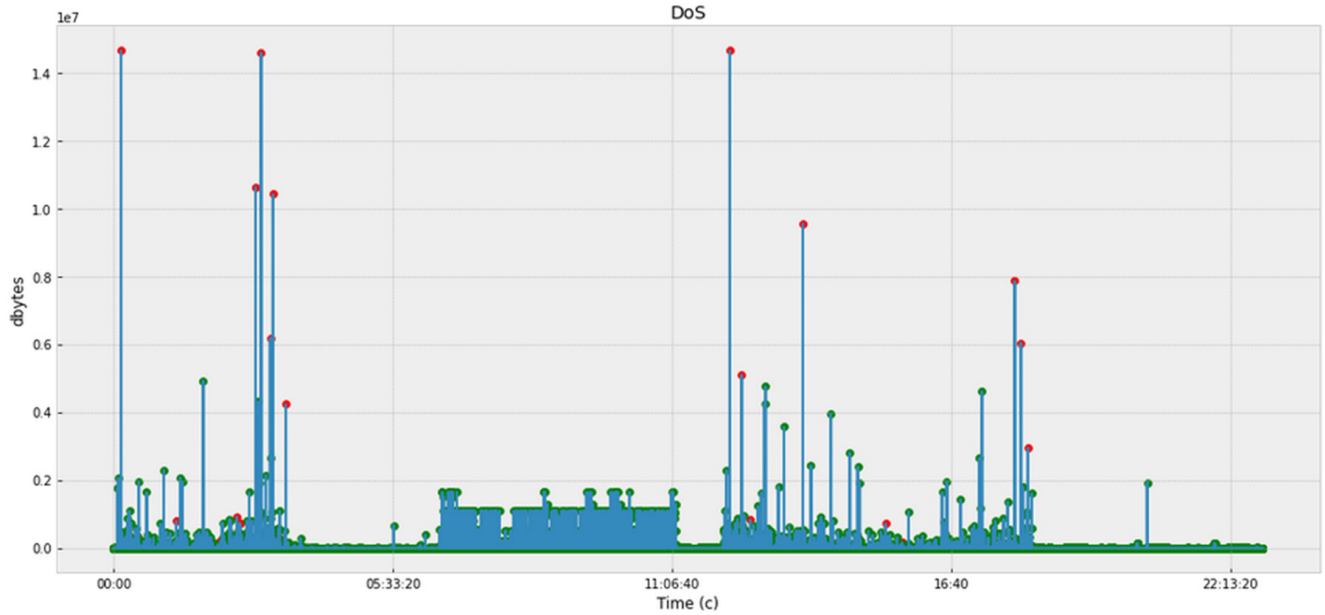

Figure 24. All protocols.

In Figures 18–24, abnormal zones are designated in red, and normal zones are designated in green. Blue lines display network packets. The Y axis displays the length of the network packet, and the X one is a time axis. As can be seen from Figures 18–24, it is almost impossible to visually distinguish between normal and abnormal behavior.

We use the LSTM autoencoder to predict anomalies in the time series (Figure 25).

Due to the strong correlation of multivariate time series and the multiscale nature of a process with fast (long-term) and slow (short-term) subprocesses, feedforward neural networks usually perform poorly. A more accurate predictive model can be developed using a neural network with LSTM cells.

Let us look at the training process of the LSTM model (Figure 26).

```
In [76]: model = autoencoder_model(X_train)
         model.compile(optimizer='adam', loss='mae')
         model.summary()

         Model: "autoencoder"
         _________________________________________________________________
         Layer (type)                 Output Shape              Param #
         =================================================================
         input_1 (InputLayer)         [(None, 1, 1)]            0
         _________________________________________________________________
         lstm (LSTM)                  (None, 1, 16)             1152
         _________________________________________________________________
         lstm_1 (LSTM)                (None, 4)                 336
         _________________________________________________________________
         repeat_vector (RepeatVector) (None, 1, 4)              0
         _________________________________________________________________
         lstm_2 (LSTM)                (None, 1, 4)              144
         _________________________________________________________________
         lstm_3 (LSTM)                (None, 1, 16)             1344
         _________________________________________________________________
         time_distributed (TimeDistri (None, 1, 1)              17
         =================================================================
         Total params: 2,993
         Trainable params: 2,993
         Non-trainable params: 0
```

Figure 25. LSTM neural layers.

```
In [64]: history = model.fit(
             X_train, X_train,
             epochs=10,
             batch_size=10,
             validation_split=0.05,
             shuffle=False,
         )

         Epoch 1/10
         5475/5475 [==============================] - 19s 3ms/step - loss: 0.0160 - val_loss: 0.0013
         Epoch 2/10
         5475/5475 [==============================] - 16s 3ms/step - loss: 0.0032 - val_loss: 9.0731e-04
         Epoch 3/10
         5475/5475 [==============================] - 22s 4ms/step - loss: 0.0028 - val_loss: 0.0012
         Epoch 4/10
         5475/5475 [==============================] - 22s 4ms/step - loss: 0.0030 - val_loss: 6.0797e-04
         Epoch 5/10
         5475/5475 [==============================] - 22s 4ms/step - loss: 0.0025 - val_loss: 3.1750e-04
         Epoch 6/10
         5475/5475 [==============================] - 22s 4ms/step - loss: 0.0025 - val_loss: 0.0014
         Epoch 7/10
         5475/5475 [==============================] - 23s 4ms/step - loss: 0.0025 - val_loss: 0.0028
         Epoch 8/10
         5475/5475 [==============================] - 22s 4ms/step - loss: 0.0024 - val_loss: 5.4495e-04
         Epoch 9/10
         5475/5475 [==============================] - 23s 4ms/step - loss: 0.0025 - val_loss: 0.0023
         Epoch 10/10
         5475/5475 [==============================] - 23s 4ms/step - loss: 0.0026 - val_loss: 4.3728e-04
```

Figure 26. Training process of the model.

We disabled data movement during model training by setting the parameter *shuffle = False*, because order is important in the time series data (you cannot allow random sampling). Figure 27 is a plot of the loss function showing how the model was trained. It can be seen from this graph that the loss function decreases during training.

```
In [64]: history = model.fit(
             X_train, X_train,
             epochs=10,
             batch_size=10,
             validation_split=0.05,
             shuffle=False,
         )

         Epoch 1/10
         5475/5475 [==============================] - 19s 3ms/step - loss: 0.0160 - val_loss: 0.0013
         Epoch 2/10
         5475/5475 [==============================] - 16s 3ms/step - loss: 0.0032 - val_loss: 9.0731e-04
         Epoch 3/10
         5475/5475 [==============================] - 22s 4ms/step - loss: 0.0028 - val_loss: 0.0012
         Epoch 4/10
         5475/5475 [==============================] - 22s 4ms/step - loss: 0.0030 - val_loss: 6.0797e-04
         Epoch 5/10
         5475/5475 [==============================] - 22s 4ms/step - loss: 0.0025 - val_loss: 3.1750e-04
         Epoch 6/10
         5475/5475 [==============================] - 22s 4ms/step - loss: 0.0025 - val_loss: 0.0014
         Epoch 7/10
         5475/5475 [==============================] - 23s 4ms/step - loss: 0.0025 - val_loss: 0.0028
         Epoch 8/10
         5475/5475 [==============================] - 22s 4ms/step - loss: 0.0024 - val_loss: 5.4495e-04
         Epoch 9/10
         5475/5475 [==============================] - 23s 4ms/step - loss: 0.0025 - val_loss: 0.0023
         Epoch 10/10
         5475/5475 [==============================] - 23s 4ms/step - loss: 0.0026 - val_loss: 4.3728e-04
```

Figure 27. Loss function.

We used the mean absolute error (MAE) to calculate the "reconstruction error" because it gave us better results compared to the mean square error (MSE) and the root mean square error (RMSE):

$$\mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N}\left|Z(t) - \hat{Z}(t)\right|, \tag{26}$$

$$\mathrm{RMSE} = \left(\frac{1}{N}\sum_{t=1}^{N}\left(Z(t) - \hat{Z}(t)\right)^2\right)^{-2}, \tag{27}$$

where Z(t) is the actual value of the time series, $\hat{Z}$(t) is the projected value predicted by the algorithm.

In both the MSE and RMSE, the errors are squared before they are averaged, resulting in higher weights assigned to larger errors. This makes the model more sensitive to noise that can cause false alarms. Since our data is inherently noisy, we have determined that an "anomaly" is a spike or trend that lasts at least 5 s. Therefore, in this model, we need a loss function that is more "forgiving" for small spikes in one or two functions.

The simplest way to determine what is an anomaly is as follows: "Anything above a fixed threshold is considered an anomaly, otherwise a normal value."

By plotting the distribution of the loss function (Figure 28) in the training set, an appropriate threshold value can be determined to identify the anomaly. In doing so, we need to check that this threshold is set above the "noise level". Any anomalies noted should be statistically significant above the background noise.

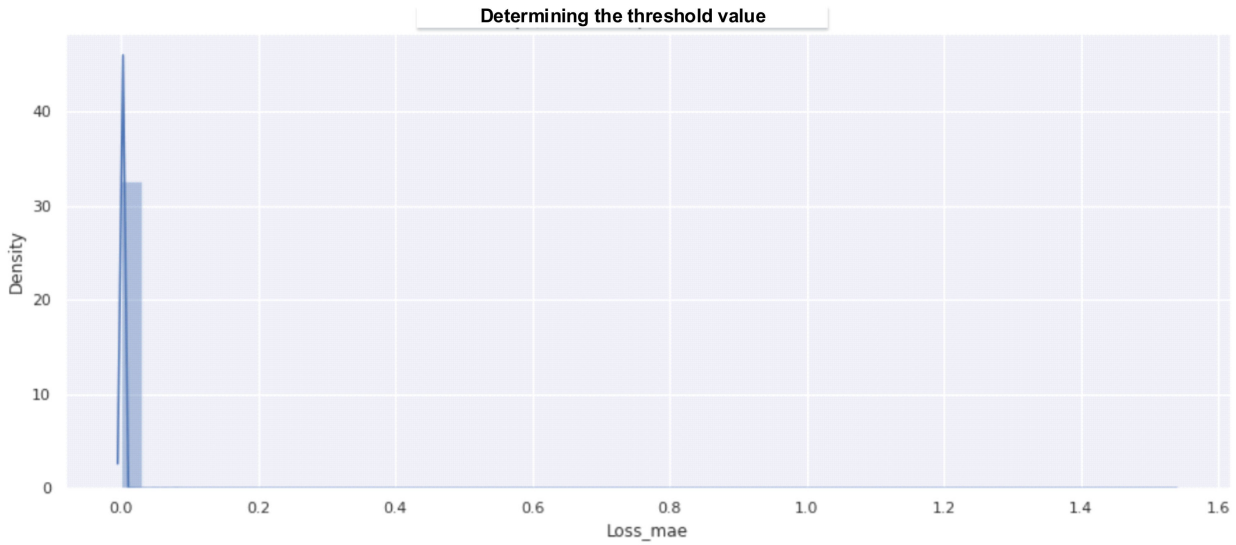

Figure 28. Distribution of learning loss.

Figure 28 shows the recovery error measured by the MAE. From the distribution of losses, a threshold value of 0.001 can be set to detect an anomaly. Figure 29 outlines the threshold value.

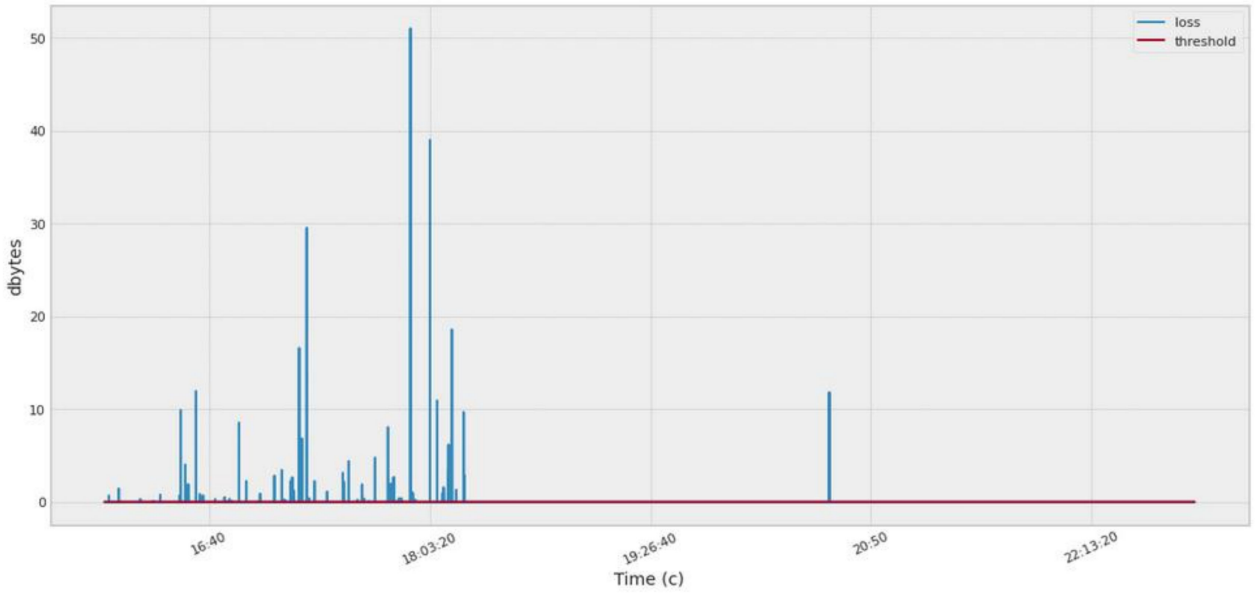

Figure 29. Static threshold value on test data.

Mathematically, the static threshold is calculated as the overall mean plus 2 standard MAE steps for all of the trained data (Figure 30).

```
In [63]: X_test_pred = model.predict(X_test)

In [64]: test_mae_loss = np.mean(np.abs(X_test_pred - X_test), axis=1)

In [65]: test_mae_loss_avg_vector = np.mean(test_mae_loss, axis=1)

In [66]: THRESHOLD = np.mean(test_mae_loss_avg_vector) + 3 * np.std(test_mae_loss_avg_vector)
         THRESHOLD

Out[66]: 0.00060480021676109 12
```

Figure 30. Calculation of static threshold value.

Figure 31 depicts the result of the anomaly detection using the neural network.

As can be seen from this figure, the neural network is very sensitive to sudden bursts. The drawing is noisy and full of anomalies, although it is not. "Noise" is seasonality, which tells us that we should use a dynamic threshold that is sensitive to the behavior of the data.

We decided to use the exponential moving average (EMA) threshold to detect anomalies:

$$\mathrm{EMA_n} = \left(\mathrm{Value} \cdot \frac{\alpha}{1+\mathrm{N}}\right) + \left(\mathrm{EMA_{n-1}} \cdot \left(1 - \frac{\alpha}{1+\mathrm{N}}\right)\right), \tag{28}$$

where N is the number of values of the original function to calculate the moving average, a is a coefficient that can be selected randomly, ranging from 0 to 1.

Figure 32 shows the average loss of the indicators: the static threshold is highlighted in purple, the moving average threshold is in red, and the exponential average threshold is in green.

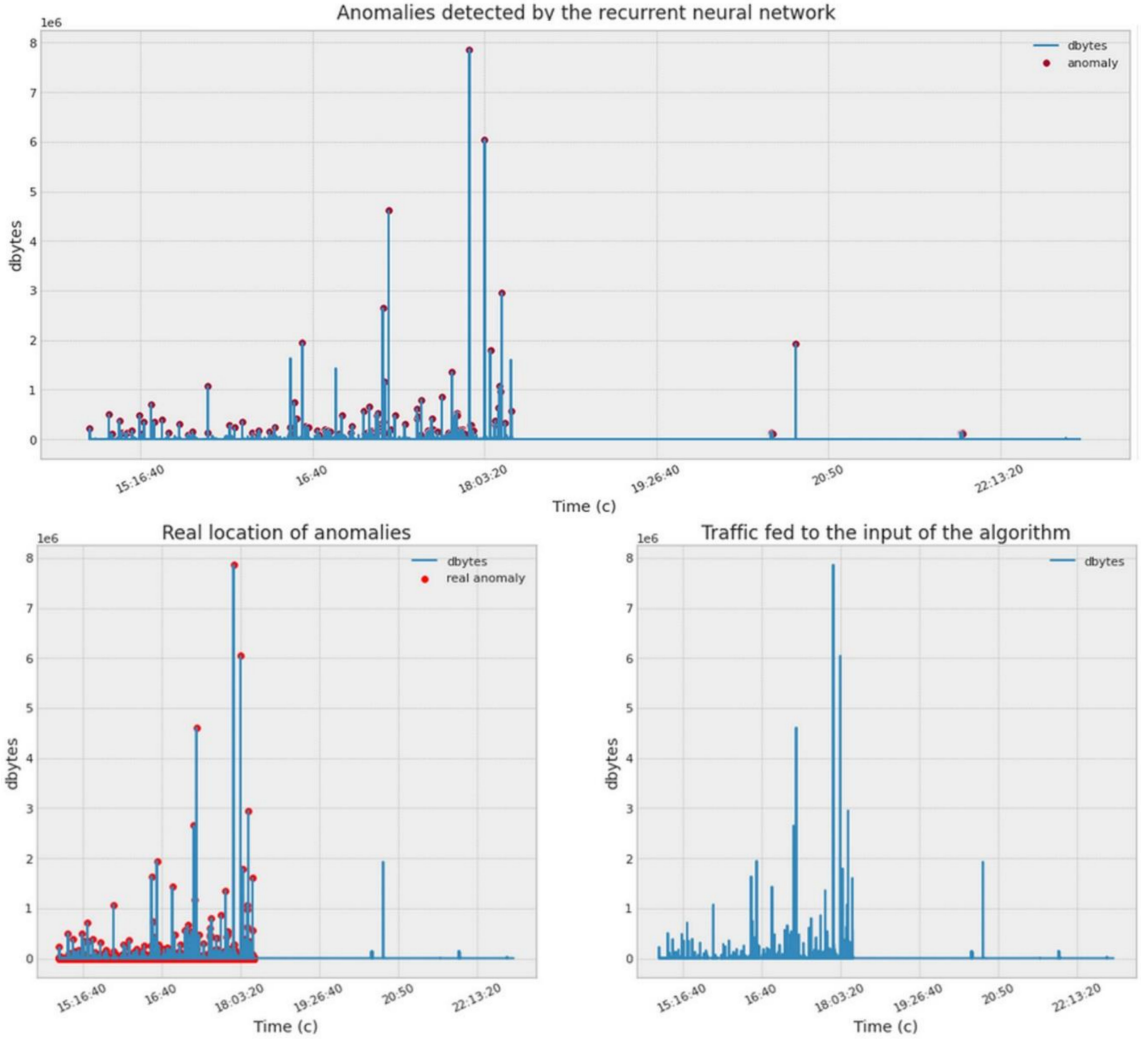

Figure 31. The result of the algorithm.

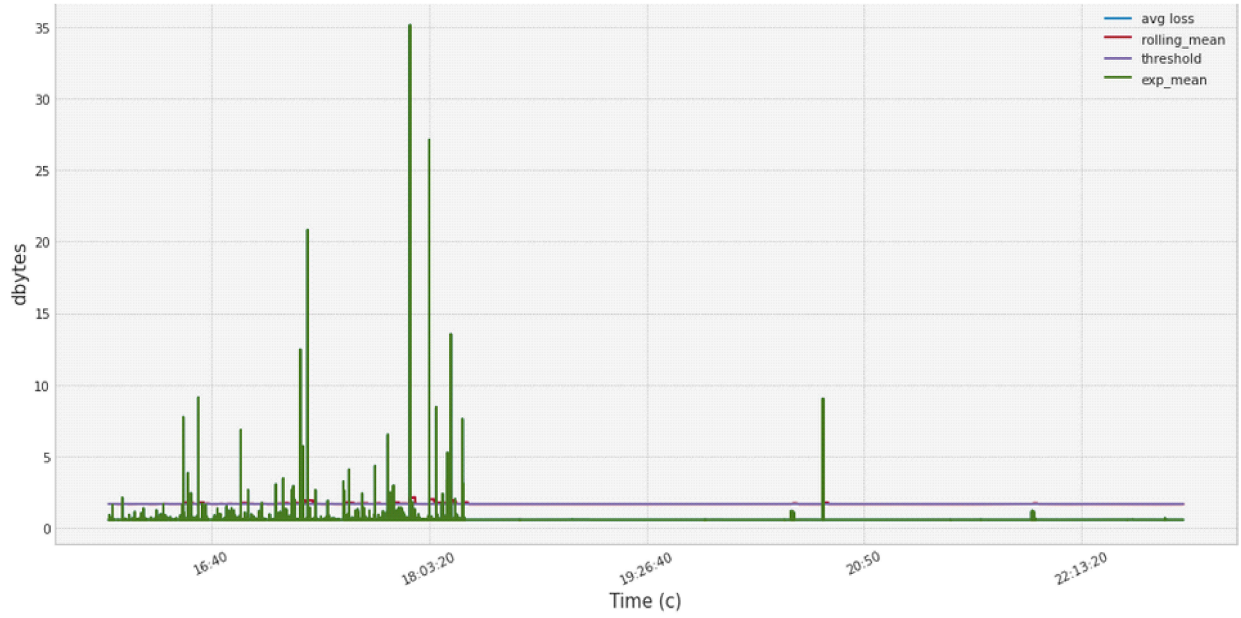

Figure 32. Dynamic threshold on test data.

After calculating the loss distribution and anomaly threshold, we can visualize the model output (Figure 33). As it can be seen from this figure, the number of false positives has decreased because the trained model was not so sensitive to emissions.

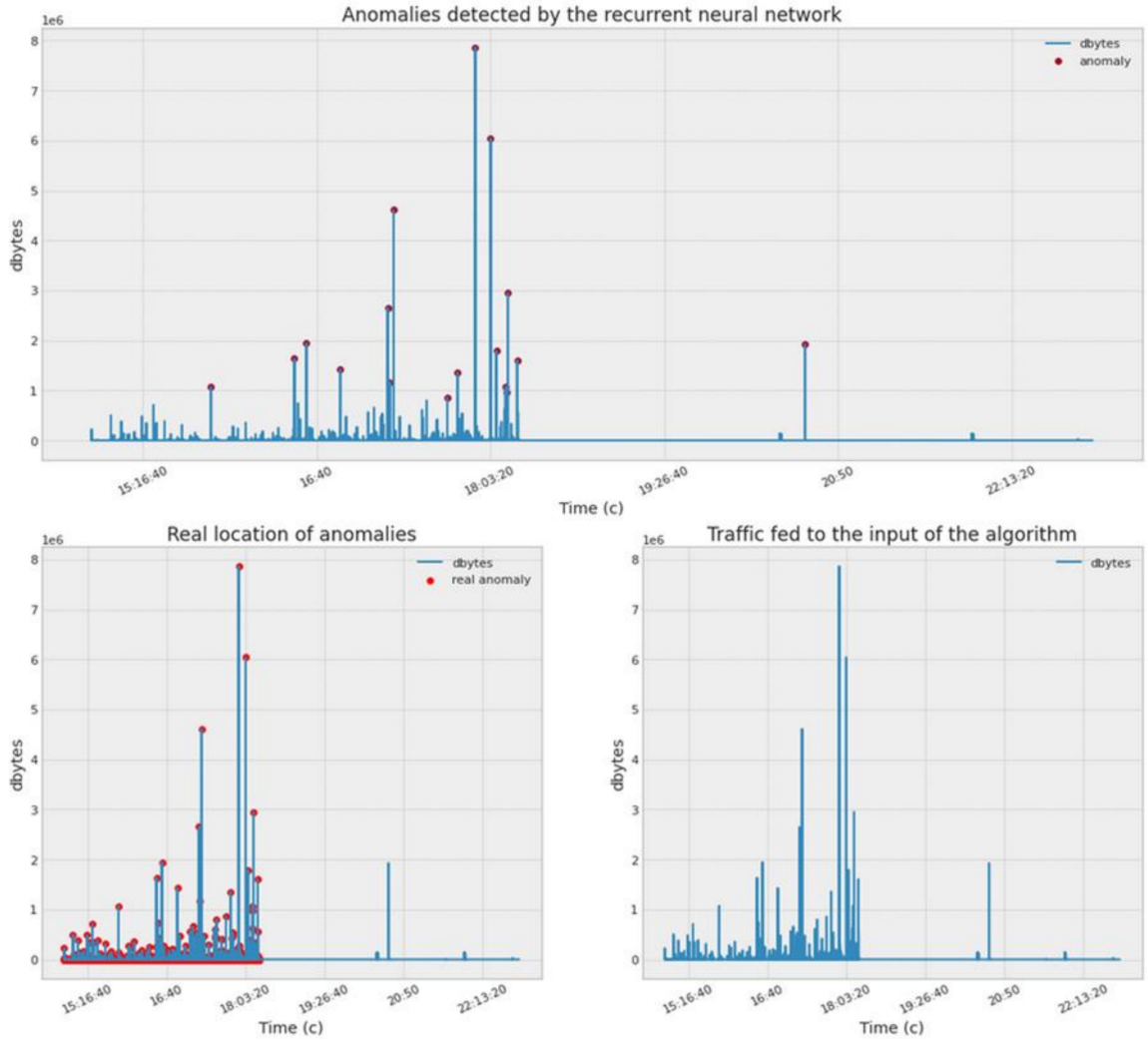

Figure 33. The result of the algorithm.

Let us approximate the graph (Figure 34).

An analysis of the experimental results showed that the ability of the recurrent neural network not only to learn, but also to develop rules for resolving collisions associated with the anomalous behavior of traffic controlled by LSTM allows for early warning of intrusions into the SG network from the outside.

Based on these warnings, the synthesis of the management system of the SG protection system is carried out. A timely, error-free decision to change the protection system affects the overall survivability of the management system.

The calculation results showed that before the management decisions were made to change the protection system, the survivability of the SG fragment under consideration was 0.75. After the synthesis of the protection system, the survivability increased to 0.93.

Thus, the proposed method for controlling the active protection of information and telecommunication SG resources allows to not only detect cyberattacks in a timely manner, but also to take measures to control the protection in a mode close to real time.

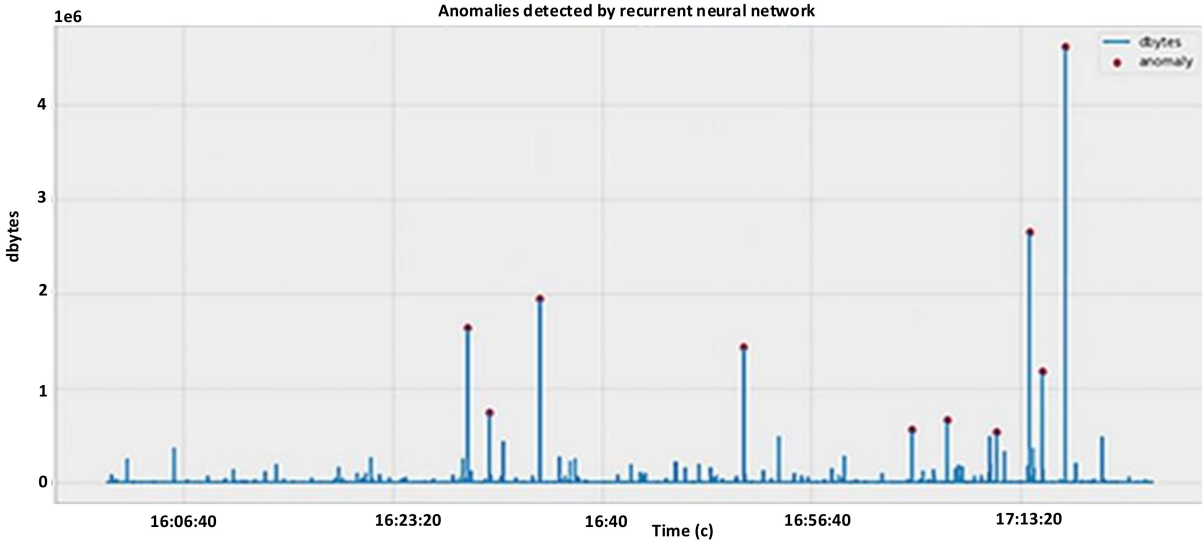

Figure 34. Approximation up to 5000 packets.

5.2. Comparative Evaluation of the Management Methodology for the SG Protection System

Formally, the management process can be represented as a tuple ⟨Y, U, O⟩, where Y is a vector of common coordinates that characterizes the system problem and the management goal; U is a vector of influence; O is a control object (Figure 35).

Figure 35. Management process.

The goal of the management process is determined by the relation $Z_Y \subset K_Y$, where K_Y is the aggregate set of Y values at which the states of the control object meet the requirements, and Z_Y is the aggregate set of Y values that arise during the operation of the control object.

In real management systems, when there is not enough a priori information about the control object, the action U is carried out by the controller P (Figure 36), which is functionally interconnected with the output states of the control object and the independent influences imposed on it. In such a situation, it is necessary, depending on the state of the vectors C_o (side effects on the control object) and C_p (side effects on the controller P), to constantly change the properties and the order of the controller functioning. A generalized diagram of the management process is shown in Figure 36, where Q_o, Q_{C_o}, and Q_{C_p} are signs-identifiers of the properties of the control object and vectors C_o and C_p.

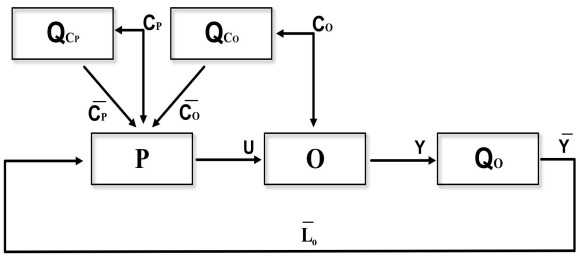

Figure 36. The process of adaptive control with a priori insufficient information about changing the parameters of the control object and the states of the controller.

For those management systems that operate under conditions of continuously changing influences, i.e., when the steady-state regime is absent at all (and this is fully true for

the SG), we will give a definition of their resilience. A management system is resilient if its output parameters remain limited under conditions of exposure to limited magnitude disturbances. The argument $\overline{L}_o$ reflects in Figure 36 the fact that the properties of P change when the properties of the control object change as a result of external and internal independent disturbances.

The classical model of a management system is a feedback model with real-time adjustable coefficients. The coefficients of such a controller are adjusted during each control cycle in accordance with the estimated system parameters. A closed-loop control is a process in a system where a controlled (controlled) variable is constantly monitored and compared with a setpoint, i.e., with a reference variable. Depending on the result of such a comparison, the input variable of the system is changed so that the output variable is adjusted to the specified value, regardless of all of the deviations. As a result of such a reaction of the system, a closed flow of actions arises. The use of feedback in the management of the SG protection system makes it possible to take into account information not only about the desired process, but also about the actual process of functioning of all of the components of the protection system.

Another well-known model of a management system is the Lyapunov reference model [49]. Adaptive control systems using the Lyapunov model are designed so that the output of the controlled model matches the output of a predefined model that has the desired characteristics. Such a system should be asymptotically stable, i.e., the controlled system keeps track of the parameters of the reference model with zero error. Moreover, transient processes at the stage of adaptive (learning) control have guaranteed limits.

Let us carry out a comparative analysis of the proposed management system of the SG protection system with those discussed above. Stability, convergence rate, operation under noise conditions (targeted and natural influences), required memory size, etc. were chosen as the comparison criteria. The results of the comparative analysis are presented in Table 3. The following symbols are used in the table: "−" is the worst indicator; "+" is the average indicator; "+ +" is the best indicator.

Table 3. The results of the comparative analysis of the management system.

Criterion	Closed-Loop Control with Adjustable Coefficients	Adaptive Control with Reference Model	LSTM Based Management
Convergence rate	+ +	+	+
Resilience of the feedback	−	+ +	+
Tracking error	+	+ +	+
Software interference minimization	+ +	−	+ +
Complexity of the control program	−	+	+
Real-time operation	+	+	+ +
Robustness of the model mismatch	−	+	+ +
System response time	−	+	+ +

The system response time (t_R) when detecting abnormal traffic deviations was measured within the specified limits ($0 < t_R < 500$MC). The system with a neural network core showed the best result of—on average, for 10 epochs of experiments—358.3 ms. For systems with feedback control and adjustable coefficients, as well as adaptive control using a reference model, the response time was 521.4 ms and 476.9 ms, respectively. Thus, it can be seen that the proposed approach demonstrates a gain in time for detecting cyberattacks in comparison with the known solutions up to 30%.

In order to determine the dependence of the cyberattack detection time on the volume of analyzed data, several experiments were carried out using the proposed method for two additional datasets. They differ from the base one, containing, as mentioned above,

82,332 records. The first additional dataset contained about 164,000 (i.e., it was about two times larger than the baseline). The second additional dataset contained approximately 41,000 records (i.e., it was about two times smaller than the base one). The experiments have shown that for the first additional dataset, the average attack detection time was 1150 ms, and for the second—150 ms. Thus, the dependence of the time to detect attacks on the volume of analyzed data has a power-law character with a low exponent. The dependence of the system response time on the amount of processed data is presented in Figure 37.

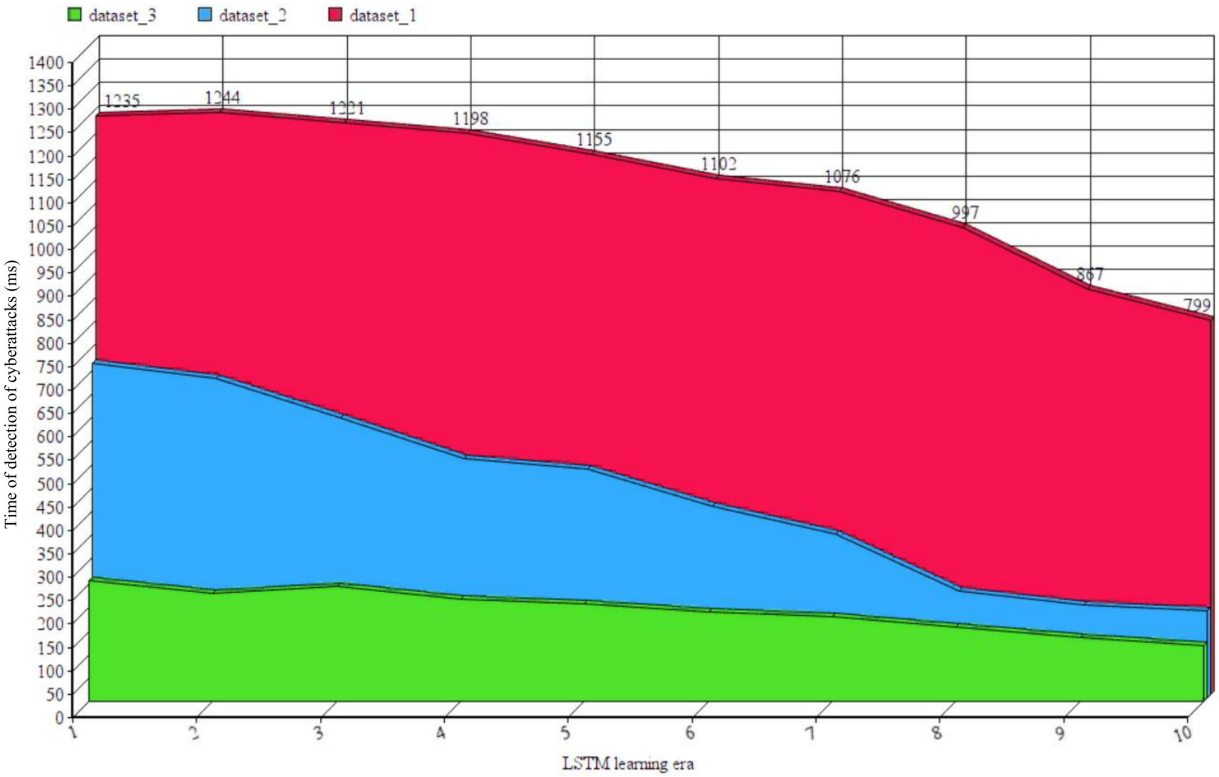

Figure 37. Dependences of the system response time from the volume of the processed data and the learning era.

Figure 37 discusses 3 datasets and 10 neural network learning eras. The Y axis corresponds to the response time of the system (i.e., the cyberattack detection time). The graph has three different dependences that show that the cyberattack detection time is reduced with each learning era. However, additional computing power is required to ensure the model scalability.

To obtain a comparative assessment of the proposed methodology for the accuracy of detecting cyberattacks, work [47] was selected. In this work, an LSTM-based classifier of attacks was studied, and its assessment was carried out at the KDD Cup 1999 dataset. The experiments carried out in [47] showed that the LSTM model has a higher efficiency in detecting and classifying cyberattacks than such well-known classifiers as k-mearest neighbor, support vector machine, multilayer perceptron, decision tree, and naïve Bayes. In [47], the following values were obtained: detection rate (DR) is 98.88% and false alarm rate (FAR) is 10.04%. In our work, the values of these indicators, averaged over all datasets, are: DR = 98.55% and FAR = 8.95%. It can be seen that our values for the accuracy of detecting cyberattacks based on LSTM are close to that of [47], but the experimental conditions are not comparable—we used a more complex data set, which included contemporary attacks, and we believe that our approach allows us to achieve greater confidence by combining the LSTM theory and the flat graph theory.

The experiments included testing the theory first without using the neural network core of the control and management system and then using it. In both cases, the type,

duration, and sequence of computer attacks (fuzzers, analysis, backdoors, DDoS, rxploits, generic, reconnaissance, shellcode, worms, and "scanning the network and its vulnerabilities") on the information and telecommunications resources of the experimental computer network were identical.

The analysis of the experimental results revealed not only characteristic changes in the "pattern" of useful traffic, but also the fact that the ability of a recurrent neural network not only to learn, but also to develop rules for resolving collisions associated with abnormal behavior of traffic controlled by LSTM allows us to warn in a timely manner of intrusions into the SG network from the outside. Thus, the proposed method of managing the active protection of SG information and telecommunications resources allows one not only to detect cyberattacks in a timely manner, but also to take measures to control the protection in a mode close to real time.

On the basis of these warnings and recommendations, the synthesis of the SG protection system management system can be carried out. A timely, error-free decision to change the protection system affects the overall survivability of the control system. The calculation results showed that before making management decisions to change the protection system, the survivability of the SG fragment under consideration was 0.75. After the synthesis of the protection system, the survivability increased to 0.93. The efficiency of the protection system increased by 24% per unit of time. According to preliminary calculations, this efficiency will be at least 17% effective under real operating conditions of the SG functioning in comparison with the protection system without an additional control module using LSTM algorithms.

An analysis of the results of a comparative analysis of various management systems shows that the proposed methodology for managing the SG protection system based on the use of LSTM neural networks has a higher efficiency. Despite the fact that each of the considered methods has both positive and negative characteristics, it should be noted that the neural network control method has a number of positive qualities that are poorly implemented in the first two management systems. The dynamics of the system response is of particular importance in the management of the SG TDTN protection system time characteristic of the output variable ($\Delta\tau$) [50]. As practical experiments show, $\Delta\tau$ is minimal for LSTM-controlled systems. In addition, the control system based on LSTM is resistant to interference when exposed to cyberattacks.

5.3. Discussion

The experiments showed, first of all, that when predicting the impact of cyberattacks in order to develop control decisions, it is very important not only to determine the machine learning algorithm or the neural network, but also to identify the parameters that are most susceptible to abnormal deviations during the attacker's exposure. In addition, the experiments demonstrated that the proposed methodology for managing the protection system, using the flat graph specification and a neural network with LSTM, has a fairly high efficiency. The main advantage of this approach is the ability to work in real time, as well as the ability to work with any kind of traffic. Revealing the fact of the impact of cyberattacks is carried out in a few microseconds, depending on the performance of the computer technology.

Other advantages of this approach include the low demands on system resources. In addition, there are practically no restrictions on the linearity of the system in the LSTM-based control system. Such a control system is effective in conditions of interference in the communication channel caused by the impact of cyberattacks. The recurrence properties provide constant additional training and management of the TDTN protection system in real time, which gives it an advantage over other systems.

However, despite the above listed advantages, this method is not a panacea because, at this stage, it is not possible to monitor the tracking errors that occur during the operation of the management systems under the influence of cyberattacks. Insufficient accuracy can lead to a discrepancy between the characteristics of the system and the conditions for the

functioning of the management system. In addition, at this stage of implementation, the management program is very complex.

It should be noted that the conducted studies are still only demonstrating the potential and effectiveness of the proposed approach to managing the security system in SG TDTN. The practical implementation and further improvement of this methodology determine further directions of the research.

6. Conclusions

This paper considered the methodology for using flat graphs for modeling a protected resource and a management system, as well as the neural network with LSTM for predicting the impact of cyberattacks on SG TDTN and developing management solutions for the protection system. The methodology proposed allows one not only to form the architecture of the SG TDTN protection system, but also to audit the SG security in real time.

In the simulation, the principle of solving the problems of distributing a heterogeneous resource among interdependent elements was used with the further implementation of the differentiated approach to creating an integrated protection of elements of smart power supply networks. The effectiveness of this method over others was shown, and the possibility of timely management of the protection system of smart power supply networks was substantiated. The issues of software implementation of the proposed approach were considered. The experimental results obtained using the generated datasets confirmed the high efficiency of the proposed approach.

In this paper, we did not consider such subjective parameters as "model complexity" and the model parameter uncertainty. They will primarily depend on the volume of tasks to be solved, the number of reference parameters to be set, the available computing power, data quality, etc. The values of the hyperparameters for LSTM in our research were determined a priori. As it was shown, the LSTM network detects well-known computer attacks with a probability close to 1, and unknown attacks with a probability exceeding 0.8.

Machine learning methods were implemented using the scikit-learn library, and neural networks using the Keras framework. The graphs were built using the Matplotlib module based on the obtained dataset. All of the calculations were performed in the Jupiter notebook integrated development environment. A simulation based on GNS3 software was used to generate traffic.

Cyberattacks, such as DDoS, reconnaissance, backdoor, exploits, fuzzers, and "scanning the network and its vulnerabilities", were taken into account as implemented attacks. The traffic structure, packet header length, flags, checksum, and some others were considered as the main characteristics under study in the dataset.

The proposed methodology, based on graph models and the practical implementation on LSTM networks, makes it possible not only to detect cyberattacks in a timely manner, but also to take measures for the active protection of SG resources in real or near real time. The use of flat graphs makes it possible to take into account (when modeling the action of an attacker when implementing the cyberattacks) the possibility of applying the means of protection, the state of the information and telecommunications network in SG, and the methods of organizing management and communication. Experiments have shown that the proposed methodology demonstrates up to a 30% gain in time for detecting cyberattacks in comparison with the known solutions. As a result, the survivability of the SG fragment under consideration increased from 0.62 to 0.95.

The main difficulty of the application of the methodology is seen in the complexity of specifying the processes, protocols, and technologies implemented in the SG in the absence of acceptable datasets for learning LSTM networks and in the absence of a common topology for constructing SG TDTN. This defines one of the further areas of research. In addition, further research will be aimed at integrating the proposed methodology into the existing monitoring and management systems for the SG protection system and optimizing the output parameters.

Author Contributions: I.K. was responsible for the conceptualization and methodology; I.S. and O.L. analyzed the data; M.K. conceived and designed the experiment. All of the authors wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research is being supported by the grant of RSF #21-71-20078 in SPC RAS.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kotenko, I.V.; Saenko, I.B.; Kotsynyak, M.A.; Lauta, O.S. Assessment of cyber-resilience of computer networks based on simulation of cyberattacks by the stochastic networks conversion method. *SPIIRAS Proc.* **2017**, *6*, 160–184. [CrossRef]
2. Bagretsov, S.A.; Lauta, O.S.; Klimenko, A.I.; Balenko, E.G. Method of Assessing Regions of Controlled Balance in Information and Telecommunications Network. In Proceedings of the 2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), Vladivostok, Russia, 1–4 October 2019; IEEE: New York, NY, USA, 2019; pp. 1–5.
3. Kotenko, I.; Saenko, I.; Lauta, O. Analytical Modeling and Assessment of Cyber Resilience on the Base of Stochastic Networks Conversion. In Proceedings of the 2018 10th International Workshop on Resilient Networks Design and Modeling (RNDM), Longyearbyen, Norway, 27–29 August 2018; IEEE: New York, NY, USA, 2019; pp. 1–8.
4. Modern Distribution Grid Project. Available online: https://gridarchitecture.pnnl.gov/modern-grid-distribution-project.aspx (accessed on 9 August 2021).
5. Henderson, M.I.; Novosel, D.; Crow, M.L. *Electric Power Grid, Modernization Trends, Challenges, and Opportunities*; IEEE: Picataway, NJ, USA, 2017.
6. Agüero, J.R.; Takayesu, E.; Novosel, D. Modernizing the grid: Challenges and opportunities for a sustainable future. *IEEE Power Energy Mag.* **2017**, *14*, 74–83. [CrossRef]
7. Tan, D.; Novosel, D. Energy challenge, power electronics & systems (PEAS) technology and grid modernization. *CPSS Trans. Power Electron. Appl.* **2017**, *2*, 3–11.
8. The Main Provisions of the Concept of an Intelligent Power System with an Active-Adaptive Grid. Available online: http://www.fsk-ees.ru/upload/docs/ies_aas.pdf (accessed on 9 August 2021).
9. Quadrennial Energy Review (QER). Available online: http://www.ieee-pes.org/qer (accessed on 9 August 2021).
10. Pallotti, E.; Mangiatordi, F. Smart Grid Cyber Security Requirements. In Proceedings of the 2011 10th International Conference on Environment and Electrical Engineering, Rome, Italy, 8–11 May 2011; IEEE: New York, NY, USA, 2019; pp. 1–4.
11. Alvarez, E.; Lopez, A.C.; Gómez-Aleixandre, J.; de Abajo, N. On-line minimization of running costs, greenhouse gas emissions and the impact of distributed generation using microgrids on the electrical system. In Proceedings of the 2009 IEEE PES/IAS Conference on Sustainable Alternative Energy (SAE), Valencia, Spain, 28–30 September 2009; IEEE: New York, NY, USA, 2009; pp. 1–10.
12. SMART GRID: An introduction. 27 June 2011. Available online: https://www.energy.gov/oe/downloads/smart-grid-introduction (accessed on 9 August 2021).
13. Concepts, Elements and Tools for the Smart Grid Methodology. Available online: https://ec.europa.eu/energy/sites/ener/files/documents/xpert_group3_methodology.pdf (accessed on 9 August 2021).
14. Smart Grid Program. Available online: https://www.nrcan.gc.ca/climate-change/green-infrastructure-programs/smart-grids/19793 (accessed on 9 August 2021).
15. Protection Profile for the Gateway of a Smart Metering System (Smart Meter Gateway PP). Available online: https://www.commoncriteriaportal.org/files/ppfiles/pp0073b_pdf.pdf (accessed on 9 August 2021).
16. ENISA Smart Grid Security Recommendations. Available online: https://www.enisa.europa.eu/publications/ENISA-smart-grid-security-recommendations (accessed on 9 August 2021).
17. ISO/IEC 27005:2008. Information Technology—Security techniquesInformation security risk management. Available online: https://www.iso.org/standard/75281.html (accessed on 9 August 2021).
18. ISO/IEC 27019:2017. Information technology—Security techniques—Information security controls for the energy utility industry. Available online: https://www.iso.org/standard/68091.html (accessed on 9 August 2021).
19. Kendrick, D.; Groom, L.; Stewart, J.; Watson, M.; Mulvaney, C.; Casterton, R. Risk watch: Cluster randomised controlled trial evaluating an injury prevention program. *Inj. Prev.* **2016**, *13*, 93–98. [CrossRef] [PubMed]
20. Fang, X.; Misra, S.; Xue, G.; Yang, D. Managing smart grid information in the cloud: Opportunities, model, and applications. *IEEE Netw.* **2012**, *26*, 32–38. [CrossRef]
21. Prasad, I. Smart Grid Technology: Application and control. *Int. J. Adv. Res. Elect. Electron. Instrum. Eng.* **2014**, *3*, 9533–9542.
22. Müller, K.J. Verordnete Sicherheit—das Schutzprofil für das Smart Metering Gateway. *Datenschutz Datensicherheit* **2014**, *35*, 547–551. [CrossRef]
23. Protection Profile for the Security Module of a Smart Metering System (Security Module PP). Available online: http://www.commoncriteriaportal.org/files/ppfiles/pp0077b_pdf.pdf (accessed on 9 August 2021).

24. Camachi, B.E.M.; Ichim, L.; Popescu, D. Cyber Security of Smart Grid Infrastructure. In Proceedings of the 2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 17–19 May 2018; IEEE: New York, NY, USA, 2018; pp. 000303–000308.
25. Peltier, T.R. Risk Management: The facilitated risk analysis and assessment process. In *Information Security Fundamentals*, 2nd ed.; Peltier, T.R., Ed.; Auerbach Publications: New York, NY, USA, 2013; pp. 60–110.
26. Nurul, A.H.; Zaheera, Z.A.; Puvanasvaran, A.P.; Zakaria, N.A.; Ahmad, R. Risk assessment method for insider threats in cyber security: A review. *Int. J. Adv. Comp. Sci. Appl.* **2018**, *9*, 16–19.
27. Tankard, C. Advanced persistent threats and how to monitor and deter them. *Netw. Secur.* **2011**, *2011*, 16–19. [CrossRef]
28. Appropriate Security Measures for Smart Grids. Available online: https://www.enisa.europa.eu/publications/appropriate-security-measures-for-smart-grids (accessed on 9 August 2021).
29. Parikh, T.P.; Patel, A.R. Cyber security: Study on attack, threat, vulnerability. *Int. J. Res. Mod. Eng. Emerg. Technol.* **2017**, *5*, 1–7.
30. Sterbenz, J.P.G.; Hutchison, D.; Çetinkaya, E.K.; Jabbar, A.; Rohrer, J.P.; Schöller, M.; Smith, P. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Comp. Netw.* **2010**, *54*, 1245–1265. [CrossRef]
31. Kotenko, I.; Polubelova, O.; Saenko, I.; Doynikova, E. The ontology of metrics for security evaluation and decision support in SIEM systems. In Proceedings of the 2013 International Conference on Availability, Reliability and Security (ARES 2013), Regensburg, Germany, 2–6 September 2013; pp. 638–645.
32. El Fray, I. A Comparative Study of Risk Assessment Methods, MEHARI & CRAMM with a New Formal Model of Risk Assessment (FoMRA) in Information Systems. In *Computer Information Systems and Industrial Management. CISIM 2012. Lecture Notes in Computer Science*; Cortesi, A., Chaki, N., Saeed, K., Wierzchoń, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7564, pp. 428–442.
33. Syalim, A.; Hori, Y.; Sakurai, K. Comparison of Risk Analysis Methods: Mehari, Magerit, NIST800-30 and Microsoft's Security Management Guide. In Proceedings of the 2009 International Conference on Availability, Reliability and Security, Fukuoka, Japan, 16–19 March 2009; IEEE: New York, NY, USA, 2009; pp. 726–731.
34. MEHARI. Overview. Available online: http://meharipedia.x10host.com/wp/wp-content/uploads/2019/05/MEHARI-Overview-2019.pdf (accessed on 9 August 2021).
35. Kuzminykh, I.; Ghita, B.; Sokolov, V.; Bakhshi, T. Information Security Risk Assessment. *Encyclopedia* **2021**, *1*, 50. [CrossRef]
36. Microsoft Security Assessment Tool 4.0. Available online: https://www.microsoft.com/en-us/download/details.aspx?id=12273 (accessed on 9 August 2021).
37. Rahman, M.S. Current Research on Planar Graphs. In Proceedings of the 2007 International Conference on Information and Communication Technology, Dhaka, Bangladesh, 7–9 March 2007; IEEE: New York, NY, USA, 2007; pp. 148–149.
38. Dujmović, V.; Esperet, L.; Gavoille, C.; Joret, G.; Micek, P.; Morin, P. Adjacency Labelling for Planar Graphs (and Beyond). In Proceedings of the 2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS), Durham, NC, USA, 16–19 November 2020; IEEE: New York, NY, USA, 2020; pp. 577–588.
39. Kotenko, I.; Saenko, I.; Lauta, O.; Kribel, A. An Approach to Detecting Cyberattacks against Smart Power Grids Based on the Analysis of Network Traffic Self-Similarity. *Energies* **2020**, *13*, 5031. [CrossRef]
40. Schreiner, Z. Asset Management Optimization a New Approach to Protection Maintenance. In Proceedings of the 2004 Eighth IEE International Conference on Developments in Power System Protection, Amsterdam, The Netherlands, 5–8 April 2004; IEEE: New York, NY, USA, 2004; Volume 1, pp. 289–294.
41. Kaur, M.; Mohta, A. A Review of Deep Learning with Recurrent Neural Network. In Proceedings of the 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 27–29 November 2019; IEEE: New York, NY, USA, 2019; pp. 460–465.
42. Malinović, N.S.; Predić, B.B.; Roganović, M. Multilayer Long Short-Term Memory (LSTM) Neural Networks in Time Series Analysis. In Proceedings of the 2020 55th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST), Nis, Serbia, 10–12 September 2020; IEEE: New York, NY, USA, 2020; pp. 11–14.
43. Mustaqeem; Kwon, S. Att-Net: Enhanced emotion recognition system using lightweight self-attention module. *Appl. Soft Comput.* **2021**, *102*, 107101. [CrossRef]
44. Mustaqeem; Kwon, S. CLSTM: Deep feature-based speech emotion recognition using the hierarchical ConvLSTM network. *Mathematics* **2020**, *8*, 2133. [CrossRef]
45. Fujita, T.; Bai, W.; Quan, C. Long short-term memory networks for automatic generation of conversations. In Proceedings of the 2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Kanazawa, Japan, 26–28 June 2017; IEEE: New York, NY, USA, 2017; pp. 483–487.
46. Skovajsová, L. Long short-term memory description and its application in text processing. In Proceedings of the 2017 Communication and Information Technologies (KIT), Vysoke Tatry, Slovakia, 4–6 October 2017; IEEE: New York, NY, USA, 2017; pp. 1–4.
47. Kim, J.; Kim, J.; Thu, H.L.T.; Kim, H. Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection. In Proceedings of the 2016 International Conference on Platform Technology and Service (PlatCon), Jeju, Korea, 15–17 February 2016; IEEE: New York, NY, USA, 2016; pp. 1–5.
48. Alobaidi, I.A.; Sarvestani, S.S.; Hurson, A.R. Survivability Analysis and Recovery Support for Smart Grids. In Proceedings of the 2016 Resilience Week (RWS), Chicago, IL, USA, 16–18 August 2016; IEEE: New York, NY, USA, 2016; pp. 33–39.

49. Nguyen, N.T. Model-reference adaptive control. In *Model-Reference Adaptive Control*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 83–123.
50. Omitaomu, O.A.; Niu, H. Artificial intelligence techniques in smart grid: A survey. *Smart Cities* **2021**, *4*, 29. [CrossRef]