

MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN
FEDERATION
FEDERAL STATE AUTONOMOUS EDUCATIONAL
INSTITUTION OF HIGHER EDUCATION
"NORTH CAUCASUS FEDERAL UNIVERSITY"

Suggestions for Teaching
to practical classes in the discipline
Network intrusion detection systems based on AI
for bachelors of training
11.03.02 Infocommunication Technologies and Communication Systems
Profile "Intelligent Data Processing in Infocommunication Systems and Networks"

Stavropol
2026

CONTENTS

Practical lesson No1. Architecture and principles of building intelligent IDS	3
Practical Lesson No2. Preparing and analyzing Datasets for AI training.....	7
Practical Lesson No3. Feature Engineering from Network Traffic	10
Practical Lesson No4. Comparative Analysis of Classical Machine Learning Algorithms for IDS.....	14
Practical Lesson No5. Metrics for Assessing the Quality of Intelligent IDS	18
Practical Lesson No6. Deep Learning for Securing Network Applications and Databases.....	21
Practical Lesson No7. Intelligent Traffic Analysis at Different Layers of the OSI Model.....	25
Practical Lesson No8. Organization of additional training of models and adaptation to new threats.....	29
Practical Lesson No9. Stress Testing of Intelligent IDS by Adversarial AI Methods	33

Practical lesson No1. Architecture and principles of building intelligent IDS

Objective: To study the architectural features of intelligent intrusion detection systems and the principles of transition from signature analysis to anomaly analysis.

Theoretical substantiation:

Traditional intrusion detection systems (IDS) have relied for decades on a signature-based analysis method that seeks exact matches between incoming traffic and a database of known attacks. However, the explosion of new vulnerabilities and the use of obfuscation techniques have made this approach ineffective against zero-day attacks. The introduction of artificial intelligence has shifted the security paradigm towards behavioral analysis, where the system is trained to recognize a "normal" traffic profile and instantly respond to any statistically significant deviations.

Intelligent IDS is a complex system consisting of data capture modules (sensors), preprocessing units, a neural network classifier, and a decision-making system. The key difference is the presence of a training phase, during which the AI model forms a multidimensional feature space that describes legitimate activity. This allows the system not only to look for "evil" code, but to understand the context of network interaction and identify hidden patterns that are inaccessible to the human eye or simple rules.

The architecture of such a system should provide high processing speed (inference) so as not to become a bottleneck in the network infrastructure. Modern approaches involve the use of graphics processing units (GPUs) or specialized accelerators for neural networks. An important aspect is also the integration of the AI module with existing security tools (firewalls, SIEM systems), which turns IDS from a passive observer into an active element of the intrusion prevention system (IPS).

Equipment and materials

Hardware. To perform practical work, you need a modern workstation or personal computer with the following minimum characteristics:

Processor: Multi-core processor (at least 4 cores) with a clock speed of 2.5 GHz or higher (Intel Core i5/i7 or AMD Ryzen 5/7 level) to provide parallel processing of network packets.

Random Access Memory (RAM): At least 16 GB (processing large network traffic datasets in RAM requires a significant amount of resources).

Graphics Card (GPU): A discrete graphics accelerator that supports CUDA technology (e.g., NVIDIA GeForce RTX 20 series or higher) is recommended to accelerate neural network training.

Storage: Free disk space of at least 50 GB (for storing traffic dumps in .pcap format and training datasets).

Network adapter: At least 1 Gbps data transfer rate.

Software. The laboratory bench should be based on modern software for data analysis and cybersecurity:

Operating system: Windows 10/11 or Linux (Ubuntu 20.04 LTS and above is recommended for better compatibility with data tools).

Development environment: Python programming language version 3.9 and higher. Interactive development environment Jupyter Notebook (as part of the Anaconda distribution) or VS Code / PyCharm.

Specialized Python libraries:

- Pandas, NumPy – for preprocessing network data;
- Scikit-learn – for the implementation of classical machine learning algorithms;
- TensorFlow or PyTorch – for building and training deep neural networks;
- Matplotlib, Seaborn – for visualizing results and analyzing anomalies.

Network analysis tools:

- Wireshark and Tcpdump – for capturing and primary analysis of network packets;
- CICFlowMeter – for Feature Extraction from .pcap traffic files.

Teaching materials:

Training Datasets: NSL-KDD, CIC-IDS2017, UNSW-NB15 (must be preloaded on workstations).

Code Snippets for implementing data loading and normalization modules.

Safety Instructions

When performing work in a computer class, it is prohibited:

1. Run neural network training algorithms with an infinite loop, which can lead to overheating of equipment.
2. Use the developed intellectual analysis tools to carry out unauthorized attacks on the nodes of the local computer network of the university.
3. Independently change the configuration of system software and network protocols without the consent of the teacher.
4. In case of signs of abnormal operation of the equipment (burning smell, strong noise of the cooling system, critical slowdown of the interface), immediately stop calculations and inform the system administrator.

Practical tasks:

1. Conduct a comparative analysis of signature and anomalous approaches to network security.
2. Describe the functions of a network sensor in the context of data preparation for AI.
3. Develop a diagram of interaction between the components of an intelligent IDS for a small enterprise.
4. Choose the optimal location for the IDS in the typical network topology (in front of or behind the router).
5. Justify the choice between cloud and on-premise hosting of the traffic analyzer.
6. Describe the feedback loop mechanism for retraining the model.
7. To form a list of hardware requirements for the operation of a neural network in a 10-gigabit channel.

8. Analyze the impact of traffic encryption (TLS/SSL) on the capabilities of an AI analyzer.
9. Develop an algorithm for the system's actions when a critical anomaly is detected.
10. Prepare a technical specification for the design of an intelligent IDS module.

Questions for self-examination:

1. What is the main weakness of the signature detection method?
2. What are the components of an intelligent IDS?
3. What is a normal network behavior profile?
4. How does AI help detect zero-day attacks?
5. What hardware resources are the most critical for the operation of neural network IDS?
6. What is the correlation between the detection accuracy and the speed of the algorithm?
7. Why are intelligent systems more prone to false positives?

References:

1. Olifer, V. G., Olifer, N. A. Computer Networks. Principles, Technologies, Protocols: Textbook for Higher Educational Institutions. 6th ed. St. Petersburg, Piter Publ., 2020. 992 p.
2. Shangin, V. F. Informatsionnaya bezopasnost' komp'yuternykh sistem i seti: Uchebnoe posobie [Information security of computer systems and networks: Textbook]. Moscow: INFRA-M Publ., 2017. 416 p. (in Russian).
3. Russell, S., Norvig, P. Artificial intelligence: a modern approach. 2nd ed. Moscow, Williams Publ., 2019. 1408 p. (in Russian).
4. Lukatsky, A. V. Detection of attacks. St. Petersburg, BHV-Petersburg Publ., 2003. 608 p. (in Russian).
5. Biryukov, A. A. Information Security: Protection and Attack. 2nd ed. Moscow: DMK Press, 2017. 434 p.

Practical Lesson No2. Preparing and analyzing Datasets for AI training

Objective: To master the methods of preprocessing network data and conducting exploratory traffic analysis to prepare training samples.

Theoretical substantiation:

The success of any intelligent system depends on 80% of the quality and representativeness of the data on which it learns. In the field of cybersecurity, specialized datasets are used, such as NSL-KDD, CIC-IDS2017 or UNSW-NB15. These datasets contain aggregated information about millions of network sessions, each of which is described by many parameters: from protocol and connection duration to complex statistical features. Working with such data requires a deep understanding of the structure of network protocols.

The data preparation process includes cleaning up noise, handling missing values, and transforming data types. Since machine learning algorithms only work with numbers, a critical step is to encode categorical features (for example, the names of the TCP, UDP, ICMP protocols) into numeric vectors. Improper data processing at this stage can lead to the model "remembering" random noise instead of identifying real attack patterns.

In addition, network traffic is characterized by a strong class imbalance: there are thousands of times more normal connections in a real network than intrusion attempts. Exploratory data analysis (EDA) makes it possible to visualize these differences and apply sampling balancing techniques. Only after normalization and standardization of numerical features can the data be fed into the neural network without the risk of overfitting or shifting gradients.

Equipment and materials

Hardware. To perform practical work, you need a modern workstation or personal computer with the following minimum characteristics:

Processor: Multi-core processor (at least 4 cores) with a clock speed of 2.5 GHz or higher (Intel Core i5/i7 or AMD Ryzen 5/7 level) to provide parallel processing of network packets.

Random Access Memory (RAM): At least 16 GB (processing large network traffic datasets in RAM requires a significant amount of resources).

Graphics Card (GPU): A discrete graphics accelerator that supports CUDA technology (e.g., NVIDIA GeForce RTX 20 series or higher) is recommended to accelerate neural network training.

Storage: Free disk space of at least 50 GB (for storing traffic dumps in .pcap format and training datasets).

Network adapter: At least 1 Gbps data transfer rate.

Software. The laboratory bench should be based on modern software for data analysis and cybersecurity:

Operating system: Windows 10/11 or Linux (Ubuntu 20.04 LTS and above is recommended for better compatibility with data tools).

Development environment: Python programming language version 3.9 and higher. Interactive development environment Jupyter Notebook (as part of the Anaconda distribution) or VS Code / PyCharm.

Specialized Python libraries:

- Pandas, NumPy – for preprocessing network data;
- Scikit-learn – for the implementation of classical machine learning algorithms;
- TensorFlow or PyTorch – for building and training deep neural networks;
- Matplotlib, Seaborn – for visualizing results and analyzing anomalies.

Network analysis tools:

- Wireshark and Tcpdump – for capturing and primary analysis of network packets;
- CICFlowMeter – for Feature Extraction from .pcap traffic files.

Teaching materials:

Training Datasets: NSL-KDD, CIC-IDS2017, UNSW-NB15 (must be preloaded on workstations).

Code Snippets for implementing data loading and normalization modules.

Safety Instructions

When performing work in a computer class, it is prohibited:

5. Run neural network training algorithms with an infinite loop, which can lead to overheating of equipment.
6. Use the developed intellectual analysis tools to carry out unauthorized attacks on the nodes of the local computer network of the university.
7. Independently change the configuration of system software and network protocols without the consent of the teacher.
8. In case of signs of abnormal operation of the equipment (burning smell, strong noise of the cooling system, critical slowdown of the interface), immediately stop calculations and inform the system administrator.

Practical tasks:

1. Download the NSL-KDD dataset and explore the description of its features (41 parameters).
2. Find and remove duplicates in network records using the Pandas library.
3. Implement encoding of text names of protocols using the One-Hot Encoding method.
4. Normalize numerical features (duration, number of bytes) in the range $[0, 1]$.
5. Build a histogram of the distribution of attack types in the dataset.
6. Calculate a feature correlation matrix to identify redundant parameters.
7. Implement data partitioning into training, validation, and test samples.
8. Visualize data in 2D space using the t-SNE method.
9. Apply the Random Undersampling method to balance classes.
10. Export the cleaned and prepared dataset to .csv format for further use.

Questions for self-examination:

1. Which datasets are standards for IDS testing?
2. Why can't I train a model on data from only one type of traffic?
3. What is data normalization and why is it needed?
4. How does class imbalance affect the quality of AI training?

5. What is the difference between Label Encoding and One-Hot Encoding?
6. What does the correlation matrix of traits show?
7. Why allocate a separate validation sample?

References:

1. Geron, O. Applied Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools and Techniques for Creating Intelligent Systems. Moscow, Williams Publ., 2018. 688 p. (in Russian).
2. Müller, A., Guido, S. Introduction to Machine Learning with Python. A Guide for Data Scientists. Moscow, Williams Publ., 2017. 392 p.
3. Gras, D. Data Science. Data Science from Scratch. – St. Petersburg: BHV-Petersburg, 2017. – 336 p.
4. Plus, J. Vander. Python for Complex Problems: Data Science and Machine Learning. St. Petersburg, Piter Publ., 2018. 576 p. (in Russian).
5. Brink, H., Richards, J., Feverolf, M. Machine learning. St. Petersburg, Piter Publ., 2017. 336 p. (in Russian).

Practical Lesson No3. Feature Engineering from Network Traffic

Objective: To learn how to extract the most informative features from raw network traffic to improve the accuracy of intelligent detection.

Theoretical substantiation:

Feature Engineering is the art of transforming raw network packets into structured information that reveals the intrusion as efficiently as possible. In the context of IDS, simple packet parameters (IP addresses, ports) are often not informative enough. To detect complex distributed attacks (DDoS) or slow network scans, it is necessary to generate second-order statistical features, such as the average number of packets per second or the standard deviation of payload size.

An important area is Flow analysis. The network flow combines packets within a single session, which allows AI to analyze the dynamics of interaction between nodes. Inter-arrival time features are critical for detecting botnets, which

often use periodic beacons to communicate with the command center. Highlighting such features requires processing traffic in rolling time windows.

The Feature Selection process also plays a key role. Data redundancy leads to the "curse of dimensionality", when the model spends computational resources on analyzing unimportant parameters, which reduces the performance of IDS. The use of mathematical statistics methods and importance assessment algorithms allows you to leave only those 15-20 features that are crucial for the classification of a particular type of threat, ensuring a balance between accuracy and reaction speed.

Equipment and materials

Hardware. To perform practical work, you need a modern workstation or personal computer with the following minimum characteristics:

Processor: Multi-core processor (at least 4 cores) with a clock speed of 2.5 GHz or higher (Intel Core i5/i7 or AMD Ryzen 5/7 level) to provide parallel processing of network packets.

Random Access Memory (RAM): At least 16 GB (processing large network traffic datasets in RAM requires a significant amount of resources).

Graphics Card (GPU): A discrete graphics accelerator that supports CUDA technology (e.g., NVIDIA GeForce RTX 20 series or higher) is recommended to accelerate neural network training.

Storage: Free disk space of at least 50 GB (for storing traffic dumps in .pcap format and training datasets).

Network adapter: At least 1 Gbps data transfer rate.

Software. The laboratory bench should be based on modern software for data analysis and cybersecurity:

Operating system: Windows 10/11 or Linux (Ubuntu 20.04 LTS and above is recommended for better compatibility with data tools).

Development environment: Python programming language version 3.9 and higher. Interactive development environment Jupyter Notebook (as part of the Anaconda distribution) or VS Code / PyCharm.

Specialized Python libraries:

- Pandas, NumPy – for preprocessing network data;
- Scikit-learn – for the implementation of classical machine learning algorithms;

- TensorFlow or PyTorch – for building and training deep neural networks;
- Matplotlib, Seaborn – for visualizing results and analyzing anomalies.

Network analysis tools:

- Wireshark and Tcpdump – for capturing and primary analysis of network packets;

- CICFlowMeter – for Feature Extraction from .pcap traffic files.

Teaching materials:

Training Datasets: NSL-KDD, CIC-IDS2017, UNSW-NB15 (must be preloaded on workstations).

Code Snippets for implementing data loading and normalization modules.

Safety Instructions

When performing work in a computer class, it is prohibited:

9. Run neural network training algorithms with an infinite loop, which can lead to overheating of equipment.

10. Use the developed intellectual analysis tools to carry out unauthorized attacks on the nodes of the local computer network of the university.

11. Independently change the configuration of system software and network protocols without the consent of the teacher.

12. In case of signs of abnormal operation of the equipment (burning smell, strong noise of the cooling system, critical slowdown of the interface), immediately stop calculations and inform the system administrator.

Practical tasks:

1. Describe the statistical traffic profile typical of a SYN-flood attack.
2. Calculate the average packet size for a 5-minute traffic dump.
3. Implement an algorithm for counting unique destination ports for a single IP address (scan detection).

4. Identify features that describe TCP header flags and assess their importance to AI.
5. Calculate the entropy of packet payloads to identify encrypted control channels (C&C).
6. Implement time interval feature extraction (IAT) between batches in a session.
7. Rank the features of the CIC-IDS2017 dataset using the ExtraTreesClassifier algorithm.
8. Generate an aggregated attribute "number of bytes per packet" and analyze its distribution.
9. Evaluate the informative value of the TTL (Time To Live) attribute to determine source substitution (spoofing).
10. Create a Python script to automatically extract features from a .pcap file.

Questions for self-examination:

1. What is a Network Flow and how is it different from a single packet?
2. What are the most common features for DDoS detection?
3. Why do I need to measure the time intervals between packets?
4. What is the "curse of dimensionality" in machine learning?
5. How does entropy help identify harmful activity?
6. What methods are used for automatic feature selection?
7. How do traffic characteristics depend on the application protocol used (HTTP vs DNS)?

References:

1. Khaikin, S. Neural Networks: A Complete Course. 2nd ed. Moscow, Williams Publ., 2018. 1104 p. (in Russian).
2. Bishop, K. Recognition of Images and Machine Learning. Moscow: Intellect Publishing House, 2020. 960 p.
3. Stallings, V. Sovremennye komp'yuternye seti [Modern computer networks]. 2nd ed. St. Petersburg, Piter Publ., 2003. 783 p. (in Russian).

4. Kurilo, A. P. Audit of Information Security. Moscow, IT-Service Publ., 2011. 234 p. (in Russian).
5. Vapnik V. N. Teoriya statisticheskogo obucheniya [Theory of statistical learning]. Moscow, Nauka Publ., 1999. 356 p. (in Russian).

Practical Lesson No4. Comparative Analysis of Classical Machine Learning Algorithms for IDS

Objective: To compare the effectiveness of different machine learning methods (decision trees, SVMs, k-NN) in the task of classifying network intrusions.

Theoretical substantiation:

Complex neural networks are not always required to build an effective IDS; often, classical machine learning (ML) algorithms show excellent results with significantly lower computational costs. Supervised Learning algorithms are the main ones in this field. For example, Decision Trees build a logical sequence of rules based on traffic characteristics, which makes their work transparent and easy to interpret for the system administrator.

A more advanced method is Random Forest, which combines hundreds of decision trees into an ensemble. This approach minimizes the risk of overfitting and allows you to achieve high accuracy even on noisy network data. The Support Vector Method (SVM), in turn, is effective at finding a clear boundary between the "norm" and "attack" classes in an arrogant feature space, but it requires more training time and is sensitive to data scaling.

When choosing an algorithm for an intelligent IDS, it is necessary to consider not only Accuracy, but also training and prediction times. In high-speed networks, it is critical that packet classification occurs in microseconds. Comparative analysis allows you to find a trade-off between the algorithm's ability to detect advanced threats and its ability to work in real time on standard server hardware.

Equipment and materials

Hardware. To perform practical work, you need a modern workstation or personal computer with the following minimum characteristics:

Processor: Multi-core processor (at least 4 cores) with a clock speed of 2.5 GHz or higher (Intel Core i5/i7 or AMD Ryzen 5/7 level) to provide parallel processing of network packets.

Random Access Memory (RAM): At least 16 GB (processing large network traffic datasets in RAM requires a significant amount of resources).

Graphics Card (GPU): A discrete graphics accelerator that supports CUDA technology (e.g., NVIDIA GeForce RTX 20 series or higher) is recommended to accelerate neural network training.

Storage: Free disk space of at least 50 GB (for storing traffic dumps in .pcap format and training datasets).

Network adapter: At least 1 Gbps data transfer rate.

Software. The laboratory bench should be based on modern software for data analysis and cybersecurity:

Operating system: Windows 10/11 or Linux (Ubuntu 20.04 LTS and above is recommended for better compatibility with data tools).

Development environment: Python programming language version 3.9 and higher. Interactive development environment Jupyter Notebook (as part of the Anaconda distribution) or VS Code / PyCharm.

Specialized Python libraries:

- Pandas, NumPy – for preprocessing network data;
- Scikit-learn – for the implementation of classical machine learning algorithms;
- TensorFlow or PyTorch – for building and training deep neural networks;
- Matplotlib, Seaborn – for visualizing results and analyzing anomalies.

Network analysis tools:

- Wireshark and Tcpdump – for capturing and primary analysis of network packets;
- CICFlowMeter – for Feature Extraction from .pcap traffic files.

Teaching materials:

Training Datasets: NSL-KDD, CIC-IDS2017, UNSW-NB15 (must be preloaded on workstations).

Code Snippets for implementing data loading and normalization modules.

Safety Instructions

When performing work in a computer class, it is prohibited:

13. Run neural network training algorithms with an infinite loop, which can lead to overheating of equipment.

14. Use the developed intellectual analysis tools to carry out unauthorized attacks on the nodes of the local computer network of the university.

15. Independently change the configuration of system software and network protocols without the consent of the teacher.

16. In case of signs of abnormal operation of the equipment (burning smell, strong noise of the cooling system, critical slowdown of the interface), immediately stop calculations and inform the system administrator.

Practical tasks:

1. Implement a network attack classifier based on the Decision Tree algorithm.

2. Train a Random Forest model and evaluate the accuracy gains compared to a single tree.

3. Apply the Support Vector Method (SVM) to separate the two most similar types of attacks.

4. Implement the k-nearest neighbors (k-NN) method and analyze the impact of the k parameter on the result.

5. To study the operation of the Naive Bayes algorithm on the characteristics of network protocols.

6. Estimate the timing of the prediction (inference) for each algorithm.

7. Cross-validate the models (K-fold) to check their stability.

8. Visualize the "importance of features" derived from the Random Forest model.
9. Implement logistic regression as a baseline for comparison.
10. Compile a comparative table of algorithms based on accuracy and performance metrics.

Questions for self-examination:

1. Why is a Random Forest more stable than a single Decision Tree?
2. In what cases are SVMs preferable to other classification methods?
3. What are the limitations of the k-NN algorithm when working with big data?
4. What is "refitting" a model and how to notice it?
5. Which algorithm from classic ML is the fastest for IDS?
6. Why do we need cross-validation?
7. How to interpret the importance of features in a random forest model?

References:

1. Potapov, A. S. Recognition of Images and Machine Learning: A Textbook. St. Petersburg, ITMO University Publ., 2015. 164 p. (in Russian).
2. Bostrem, G. Osnovy premenirovaniya na Python: Uchebnoe posobie [Fundamentals of Python Programming: A Textbook]. Moscow, DMK Press Publ., 2018. 250 p.
3. Lutz, M. Studying Python. 5th ed. St. Petersburg: Dialectics, 2019. 832 p.
4. Sebastian, R. Python and Machine Learning. Moscow: DMK Press, 2017. 418 p.
5. Forsythe, D., Pons, J. Computer Vision. Modern Approach. Moscow, Williams Publ., 2004. 928 p.

Practical Lesson No5. Metrics for Assessing the Quality of Intelligent IDS

Objective: To study specialized metrics for assessing AI-based protection systems and learn how to interpret the results of the models.

Theoretical substantiation:

In cybersecurity tasks, the standard Accuracy metric is often misleading due to a strong class imbalance. If 99% of traffic is normal, then a model that always says "normal" will have 99% accuracy, but will miss all attacks. Therefore, deeper metrics are used to evaluate intelligent IDS: Precision, Recall, and F1-score. Precision shows what percentage of blocked traffic was actually an attack, and Recall shows what percentage of real attacks the system was able to detect.

IDS pays special attention to errors of the first and second types. False Positive leads to the blocking of legitimate users and financial losses to the business. False Negative is a missed attack that can result in data theft. An information security engineer must be able to configure the Threshold in such a way as to minimize the most critical type of error for a particular organization.

To visualize the effectiveness of an AI model, ROC and PR (Precision-Recall) curves are used. The area under the curve (AUC) is a universal indicator of the quality of the algorithm, regardless of the selected threshold of operation. Understanding these metrics allows you to objectively compare different neural network architectures and choose the one that provides the most security with minimal interference with the user experience.

Equipment and materials

Hardware. To perform practical work, you need a modern workstation or personal computer with the following minimum characteristics:

Processor: Multi-core processor (at least 4 cores) with a clock speed of 2.5 GHz or higher (Intel Core i5/i7 or AMD Ryzen 5/7 level) to provide parallel processing of network packets.

Random Access Memory (RAM): At least 16 GB (processing large network traffic datasets in RAM requires a significant amount of resources).

Graphics Card (GPU): A discrete graphics accelerator that supports CUDA technology (e.g., NVIDIA GeForce RTX 20 series or higher) is recommended to accelerate neural network training.

Storage: Free disk space of at least 50 GB (for storing traffic dumps in .pcap format and training datasets).

Network adapter: At least 1 Gbps data transfer rate.

Software. The laboratory bench should be based on modern software for data analysis and cybersecurity:

Operating system: Windows 10/11 or Linux (Ubuntu 20.04 LTS and above is recommended for better compatibility with data tools).

Development environment: Python programming language version 3.9 and higher. Interactive development environment Jupyter Notebook (as part of the Anaconda distribution) or VS Code / PyCharm.

Specialized Python libraries:

- Pandas, NumPy – for preprocessing network data;
- Scikit-learn – for the implementation of classical machine learning algorithms;
- TensorFlow or PyTorch – for building and training deep neural networks;
- Matplotlib, Seaborn – for visualizing results and analyzing anomalies.

Network analysis tools:

- Wireshark and Tcpdump – for capturing and primary analysis of network packets;
- CICFlowMeter – for Feature Extraction from .pcap traffic files.

Teaching materials:

Training Datasets: NSL-KDD, CIC-IDS2017, UNSW-NB15 (must be preloaded on workstations).

Code Snippets for implementing data loading and normalization modules.

Safety Instructions

When performing work in a computer class, it is prohibited:

1. Run neural network training algorithms with an infinite loop, which can lead to overheating of equipment.
2. Use the developed intellectual analysis tools to carry out unauthorized attacks on the nodes of the local computer network of the university.
3. Independently change the configuration of system software and network protocols without the consent of the teacher.
4. In case of signs of abnormal operation of the equipment (burning smell, strong noise of the cooling system, critical slowdown of the interface), immediately stop calculations and inform the system administrator.

Practical tasks:

1. Build a Confusion Matrix for IDS results.
2. Manually calculate Precision, Recall, and F1-score metrics based on an error matrix.
3. Evaluate the false positive rate (FPR) of the model.
4. Plot the ROC curve and calculate the AUC value.
5. Analyze the impact of changing the Threshold on Recall and Precision.
6. Construct the Precision-Recall curve and explain its shape.
7. Calculate the MCC (Matthews correlation coefficient) metric for an unbalanced sample.
8. To conduct a comparative analysis of two IDS models by F1-score.
9. Estimate the economic damage from False Positive in a banking network scenario.
10. Prepare an AI model validation report using the Scikit-learn library.

Questions for self-examination:

1. Why does high Accuracy not guarantee network security?
2. What is a Type 1 error in the context of intrusion detection?
3. In what situations is Recall more important than Precision?
4. How does the probability threshold affect the number of blocked users?
5. What does $AUC = 0.5$ mean?

6. Why is the F1-score metric better than a simple averaging of accuracy and completeness?
7. How are class imbalances and rating metric choices related?

References:

1. Flah, P. Machine Learning. The Science and Art of Building Algorithms That Model Data. Moscow, DMK Press Publ., 2015. 400 p.
2. Kuzin, A. V. Computer Networks: A Textbook. Moscow, Forum Publ., 2014. 192 p. (in Russian).
3. Rashid, T. Sozdaem neural set. St. Petersburg, Alfa-kniga Publ., 2017. 272 p. (in Russian).
4. Goodfellow, Y., Bengio, I., Courville, A. Glubokoe obuchenie [Deep learning]. Moscow, DMK Press Publ., 2017. 652 p.
5. Ian, G. Deep learning. Adaptive computing and machine learning. Moscow, DMK Press Publ., 2018. 740 p. (in Russian).

Practical Lesson No6. Deep Learning for Securing Network Applications and Databases

Objective: To study deep neural network architectures and learn how to use them to detect complex attacks on web applications and database management systems.

Theoretical substantiation:

Unlike classical machine learning, deep learning is able to independently extract hierarchical features from data, which is especially important when analyzing unstructured information such as HTTP requests or database logs. Multilayer perceptrons (MLPs) and convolutional neural networks (CNNs) allow you to identify complex patterns of malicious activity that are difficult to formalize using expert rules. For example, when analyzing SQL injections, a neural network can take into account not only the presence of keywords, but also their relative location and statistical weight in the context of a normal query.

To protect databases, the use of recurrent neural networks (RNNs) or long short-term memory networks (LSTMs) becomes critical. These architectures are designed to work with data sequences, which allows the system to "remember" previous user actions and identify anomalies in the transaction chain. If the typical DBA profile changes abruptly to a series of uncharacteristic requests to tables with personal data, the intelligent IDS will record this as a potential information theft, even if each individual request looks legitimate.

Designing such systems requires careful selection of activation functions, regularization methods (e.g., Dropout), and optimizers (Adam, SGD). Training deep models requires significant computing power (GPUs), but the result in the form of high classification accuracy and the ability to summarize data justifies the cost. At this stage, it is important to understand how the structure of the neural network correlates with the nature of the protected resource: web traffic, API calls, or database system events.

Equipment and materials

Hardware. To perform practical work, you need a modern workstation or personal computer with the following minimum characteristics:

Processor: Multi-core processor (at least 4 cores) with a clock speed of 2.5 GHz or higher (Intel Core i5/i7 or AMD Ryzen 5/7 level) to provide parallel processing of network packets.

Random Access Memory (RAM): At least 16 GB (processing large network traffic datasets in RAM requires a significant amount of resources).

Graphics Card (GPU): A discrete graphics accelerator that supports CUDA technology (e.g., NVIDIA GeForce RTX 20 series or higher) is recommended to accelerate neural network training.

Storage: Free disk space of at least 50 GB (for storing traffic dumps in .pcap format and training datasets).

Network adapter: At least 1 Gbps data transfer rate.

Software. The laboratory bench should be based on modern software for data analysis and cybersecurity:

Operating system: Windows 10/11 or Linux (Ubuntu 20.04 LTS and above is recommended for better compatibility with data tools).

Development environment: Python programming language version 3.9 and higher. Interactive development environment Jupyter Notebook (as part of the Anaconda distribution) or VS Code / PyCharm.

Specialized Python libraries:

- Pandas, NumPy – for preprocessing network data;
- Scikit-learn – for the implementation of classical machine learning algorithms;
- TensorFlow or PyTorch – for building and training deep neural networks;
- Matplotlib, Seaborn – for visualizing results and analyzing anomalies.

Network analysis tools:

- Wireshark and Tcpdump – for capturing and primary analysis of network packets;
- CICFlowMeter – for Feature Extraction from .pcap traffic files.

Teaching materials:

Training Datasets: NSL-KDD, CIC-IDS2017, UNSW-NB15 (must be preloaded on workstations).

Code Snippets for implementing data loading and normalization modules.

Safety Instructions

When performing work in a computer class, it is prohibited:

5. Run neural network training algorithms with an infinite loop, which can lead to overheating of equipment.
6. Use the developed intellectual analysis tools to carry out unauthorized attacks on the nodes of the local computer network of the university.
7. Independently change the configuration of system software and network protocols without the consent of the teacher.

8. In case of signs of abnormal operation of the equipment (burning smell, strong noise of the cooling system, critical slowdown of the interface), immediately stop calculations and inform the system administrator.

Practical tasks:

1. Develop a multilayer perceptron architecture for traffic classification from the CIC-IDS2017 suite.
2. Tokenize HTTP requests to feed into a neural network (convert text to a numeric vector).
3. Implement a Dropout layer to prevent the model from being overtrained on specific attacks.
4. To compare the effectiveness of ReLU and Sigmoid activation functions in the hidden layers of the neural network.
5. Train the model to detect Cross-Site Scripting (XSS) attacks in URL parameters.
6. Configure the Adam optimizer and estimate the rate of convergence of the Loss function.
7. Describe the structure of an LSTM network for analyzing a sequence of SQL commands.
8. Conduct an experiment to change the number of hidden layers and assess the impact on the F1-score metric.
9. Visualize neural network weights to understand which parts of the query are most "suspicious" to the model.
10. Export the trained model to ONNX format for embedding in a web firewall (WAF).

Questions for self-examination:

1. What is the main advantage of deep learning over classical algorithms (e.g., SVM)?
2. What tasks in information security do recurrent neural networks (RNNs) solve?
3. Why is the Dropout method used in neural network training?

4. How does tokenization of text queries work in the context of intelligent IDS?
5. What is overtraining and how can it be detected by training schedules?
6. Why is it preferable to use a GPU to train deep models?
7. Which Loss functions are most applicable to attack classification tasks?

References:

1. Goodfellow, Y., Bengio, I., Courville, A. Glubokoe obuchenie [Deep learning]. Moscow, DMK Press Publ., 2017. 652 p.
2. Rashid, T. Sozdaem neural set. St. Petersburg, Alfa-kniga Publ., 2017. 272 p. (in Russian).
3. Shangin, V. F. Informatsionnaya bezopasnost' komp'yuternykh sistem i seti: Uchebnoe posobie [Information security of computer systems and networks: Textbook]. Moscow: INFRA-M Publ., 2017. 416 p. (in Russian).
4. Geron, O. Applied Machine Learning with Scikit-Learn and TensorFlow. Moscow, Williams Publ., 2018. 688 p. (in Russian).
5. Schollet, F. Glubokoe obuchenie na Python [Deep learning in Python]. St. Petersburg, Piter Publ., 2018. 400 p. (in Russian).

Practical Lesson No7. Intelligent Traffic Analysis at Different Layers of the OSI Model

Objective: To study the specifics of the use of AI algorithms for the analysis of network anomalies at the channel, network, transport and application levels of the open systems interaction model.

Theoretical substantiation:

Information security threats can occur at any of the seven layers of the OSI model, and there is no one-size-fits-all AI model to mitigate them. At the network (L3) and transport (L4) layers, intelligence focuses on packet headers and flow statistics (IP addresses, ports, TCP flags). Here, AI effectively detects port scanning

attempts, spoofing, and volumetric DDoS attacks by analyzing anomalous bursts in the number of packets or disruption of connection logic.

At the application layer (L7), the task of intelligent IDS becomes more complicated: it is necessary to analyze the payload of packets. Here, AI encounters a variety of protocols (HTTP, DNS, SSH, FTP), each of which has its own specifics. For example, DNS tunneling detection requires the analysis of domain name entropy and request rate, and the detection of attacks in mail traffic requires natural language processing (NLP). The intelligent agent must be adapted to a specific protocol to minimize false positives.

The integrated approach implies a cascading structure of neural networks, where each model is responsible for its own layer or group of protocols. This allows not only to detect an attack but also to accurately determine its vector. For example, if the transport layer model sees suspicious activity on port 80, the data is transmitted to a specialized application layer model for a detailed analysis of the HTTP session. Such in-depth intelligent protection significantly increases the overall stability of the network infrastructure.

Equipment and materials

Hardware. To perform practical work, you need a modern workstation or personal computer with the following minimum characteristics:

Processor: Multi-core processor (at least 4 cores) with a clock speed of 2.5 GHz or higher (Intel Core i5/i7 or AMD Ryzen 5/7 level) to provide parallel processing of network packets.

Random Access Memory (RAM): At least 16 GB (processing large network traffic datasets in RAM requires a significant amount of resources).

Graphics Card (GPU): A discrete graphics accelerator that supports CUDA technology (e.g., NVIDIA GeForce RTX 20 series or higher) is recommended to accelerate neural network training.

Storage: Free disk space of at least 50 GB (for storing traffic dumps in .pcap format and training datasets).

Network adapter: At least 1 Gbps data transfer rate.

Software. The laboratory bench should be based on modern software for data analysis and cybersecurity:

Operating system: Windows 10/11 or Linux (Ubuntu 20.04 LTS and above is recommended for better compatibility with data tools).

Development environment: Python programming language version 3.9 and higher. Interactive development environment Jupyter Notebook (as part of the Anaconda distribution) or VS Code / PyCharm.

Specialized Python libraries:

- Pandas, NumPy – for preprocessing network data;
- Scikit-learn – for the implementation of classical machine learning algorithms;
- TensorFlow or PyTorch – for building and training deep neural networks;
- Matplotlib, Seaborn – for visualizing results and analyzing anomalies.

Network analysis tools:

- Wireshark and Tcpdump – for capturing and primary analysis of network packets;
- CICFlowMeter – for Feature Extraction from .pcap traffic files.

Teaching materials:

Training Datasets: NSL-KDD, CIC-IDS2017, UNSW-NB15 (must be preloaded on workstations).

Code Snippets for implementing data loading and normalization modules.

Safety Instructions

When performing work in a computer class, it is prohibited:

1. Run neural network training algorithms with an infinite loop, which can lead to overheating of equipment.
2. Use the developed intellectual analysis tools to carry out unauthorized attacks on the nodes of the local computer network of the university.
3. Independently change the configuration of system software and network protocols without the consent of the teacher.

4. In case of signs of abnormal operation of the equipment (burning smell, strong noise of the cooling system, critical slowdown of the interface), immediately stop calculations and inform the system administrator.

Practical tasks:

1. Generate an AI feature set to detect ARP spoofing at the channel layer.
2. Analyze anomalies in IP packet headers (L3) to identify fragmented attacks.
3. To develop an algorithm for classifying sessions based on TCP flags (SYN, ACK, FIN, PSH).
4. Search for signs of DNS tunneling in traffic dumps using machine learning methods.
5. Compare HTTP and HTTPS Protocol Traffic Profiles: What Can AI See in an Encrypted Stream?
6. Conduct Packet Length Analysis to detect hidden botnet control channels.
7. Implement an anomaly detector in SSH traffic (detection of Brute Force attacks).
8. Describe an AI model for data integrity monitoring in the FTP protocol.
9. Develop a logic for data transfer between the L4 detector and the L7 detector in the IDS.
10. Visualize traffic anomalies at different OSI layers using the Seaborn library.

Questions for self-examination:

1. At what levels of the OSI model do today's intelligent IDS operate?
2. What are the most important traffic indicators for analysis at the network layer (L3)?
3. What is the complexity of application layer (L7) analysis compared to transport analysis?
4. How can AI detect an attack in encrypted traffic without decrypting it?
5. What is traffic entropy and how does it help identify tunneling?

6. Why use a cascade of neural networks to protect different layers of OSI?
7. Which application layer protocols are most vulnerable to smart attacks?

References:

1. Olifer, V. G., Olifer, N. A. Computer Networks. Principles, Technologies, Protocols: Textbook for Higher Educational Institutions. 6th ed. St. Petersburg, Piter Publ., 2020. 992 p.
2. Tanenbaum, E., Weseroll, D. Computer Networks. 5th ed. St. Petersburg, Piter Publ., 2012. 960 p.
3. Stallings, V. Sovremennye komp'yuternye seti [Modern computer networks]. 2nd ed. St. Petersburg, Piter Publ., 2003. 783 p. (in Russian).
4. Lukatsky, A. V. Detection of attacks. St. Petersburg, BHV-Petersburg Publ., 2003. 608 p. (in Russian).
5. Biryukov, A. A. Information Security: Protection and Attack. 2nd ed. Moscow: DMK Press, 2017. 434 p.

Practical Lesson No8. Organization of additional training of models and adaptation to new threats

Objective: To study the life cycle of an AI model in cybersecurity and to master methods to combat the degradation of system accuracy when the network landscape changes.

Theoretical substantiation:

One of the main problems of AI-based systems is "model drift". The network environment is dynamic: new applications are introduced, user patterns are changing, and new types of attacks (Zero-day) are emerging. If a model trained a year ago is not updated, its accuracy will inevitably decrease, and the number of missed attacks will increase. To maintain the effectiveness of intelligent IDS, it is necessary to build processes for continuous monitoring of the quality of AI work.

The retraining methodology can include the concept of Active Learning. In this scenario, the system identifies those traffic patterns that it is "unsure" about and passes them on to a security expert for markup. Once confirmed, whether it is a norm or an attack, the model updates its weights to adapt to new conditions without a complete reinstallation. This allows IDS to evolve with threats while maintaining a high level of relevance to the knowledge base.

Intelligent system administration also includes working with MLOps (Machine Learning Operations). This is a set of practices for automating the deployment, testing, and monitoring of AI models in an industrial network. An information security specialist must monitor metrics such as Data Drift and Concept Drift. Responding to these changes in a timely manner ensures that the intrusion detection system does not become a worthless asset, but is a reliable intelligent shield for the organization.

Equipment and materials

Hardware. To perform practical work, you need a modern workstation or personal computer with the following minimum characteristics:

Processor: Multi-core processor (at least 4 cores) with a clock speed of 2.5 GHz or higher (Intel Core i5/i7 or AMD Ryzen 5/7 level) to provide parallel processing of network packets.

Random Access Memory (RAM): At least 16 GB (processing large network traffic datasets in RAM requires a significant amount of resources).

Graphics Card (GPU): A discrete graphics accelerator that supports CUDA technology (e.g., NVIDIA GeForce RTX 20 series or higher) is recommended to accelerate neural network training.

Storage: Free disk space of at least 50 GB (for storing traffic dumps in .pcap format and training datasets).

Network adapter: At least 1 Gbps data transfer rate.

Software. The laboratory bench should be based on modern software for data analysis and cybersecurity:

Operating system: Windows 10/11 or Linux (Ubuntu 20.04 LTS and above is recommended for better compatibility with data tools).

Development environment: Python programming language version 3.9 and higher. Interactive development environment Jupyter Notebook (as part of the Anaconda distribution) or VS Code / PyCharm.

Specialized Python libraries:

- Pandas, NumPy – for preprocessing network data;
- Scikit-learn – for the implementation of classical machine learning algorithms;
- TensorFlow or PyTorch – for building and training deep neural networks;
- Matplotlib, Seaborn – for visualizing results and analyzing anomalies.

Network analysis tools:

- Wireshark and Tcpdump – for capturing and primary analysis of network packets;
- CICFlowMeter – for Feature Extraction from .pcap traffic files.

Teaching materials:

Training Datasets: NSL-KDD, CIC-IDS2017, UNSW-NB15 (must be preloaded on workstations).

Code Snippets for implementing data loading and normalization modules.

Safety Instructions

When performing work in a computer class, it is prohibited:

1. Run neural network training algorithms with an infinite loop, which can lead to overheating of equipment.
2. Use the developed intellectual analysis tools to carry out unauthorized attacks on the nodes of the local computer network of the university.
3. Independently change the configuration of system software and network protocols without the consent of the teacher.
4. In case of signs of abnormal operation of the equipment (burning smell, strong noise of the cooling system, critical slowdown of the interface), immediately stop calculations and inform the system administrator.

Practical tasks:

1. Audit the trained model on "new" traffic and record a drop in the Call metric.
2. Implement a mechanism for collecting "uncertain" model predictions (where the probability of the class is close to 0.5).
3. Simulate a Concept Drift situation: a drastic change in legitimate traffic (e.g., the introduction of new enterprise software).
4. Develop a script for automatic model retraining on new data (Incremental Learning).
5. To study methods for saving neural network states (Checkpoints) for rollback to previous versions.
6. Form a regulation for checking the accuracy of IDS (weekly/monthly reports).
7. Describe the role of a human expert in an Active Learning system for cybersecurity.
8. To conduct a comparative analysis of the accuracy of the model "before" and "after" the process of additional training.
9. Implement an Administrator Alerts system in case of a critical decrease in AI accuracy.
10. Prepare an AI model lifecycle diagram for a large data center.

Questions for self-examination:

1. What is "model drift" and why is it dangerous?
2. What is the difference between Data Drift and Concept Drift?
3. How does Active Learning work?
4. Why do I need to save Checkpoints during training?
5. What is Incremental Learning?
6. What MLOps automation tools are applicable in cybersecurity?
7. How often is it necessary to conduct additional training of neural networks in a real network?

References:

1. Brink, H., Richards, J., Feverolf, M. Machine learning. St. Petersburg, Piter Publ., 2017. 336 p. (in Russian).
2. Plus, J. Vander. Python for Complex Problems: Data Science and Machine Learning. St. Petersburg, Piter Publ., 2018. 576 p. (in Russian).
3. Müller, A., Guido, S. Introduction to Machine Learning with Python. Moscow, Williams Publ., 2017. 392 p. (in Russian).
4. Flah, P. Machine Learning. Science and Art of Algorithm Construction. Moscow: DMK Press, 2015. 400 p.
5. Khaikin, S. Neural Networks: A Complete Course. 2nd ed. Moscow, Williams Publ., 2018. 1104 p. (in Russian).

Practical Lesson No9. Stress Testing of Intelligent IDS by Adversarial AI Methods

Objective: To study the vulnerabilities of machine learning algorithms themselves and learn how to conduct IDS tests for resistance to Adversarial Attacks.

Theoretical substantiation:

As the protection becomes intelligent, attackers begin to use adversarial artificial intelligence (Adversarial AI) to bypass it. Adversarial attacks are about looking for minimal changes in the vector of traffic signs that do not change the essence of the attack, but force the neural network to classify it as "normal." For example, adding small pauses between packets or introducing random bytes into a payload can completely blind the neural network anomaly detector.

There are two main types of such attacks: evasion attacks, which are carried out at the stage of system operation, and poisoning attacks, which are aimed at distorting the learning process. In the case of poisoning, the attacker injects specific traffic into the training sample, which makes the model consider certain types of malicious activity legitimate. This creates a hidden "backdoor vulnerability" in the

protection system itself, which is almost impossible to detect with standard monitoring methods.

To combat adversarial attacks, Adversarial Training methods and anomaly detectors are used in the input data itself. At the testing stage, the information security engineer must stress test the system using tools for generating adversarial examples. Only an understanding of the vulnerabilities of AI algorithms allows you to create a truly reliable intrusion detection system that can withstand not only conventional hackers, but also automated AI-based attack systems.

Equipment and materials

Hardware. To perform practical work, you need a modern workstation or personal computer with the following minimum characteristics:

Processor: Multi-core processor (at least 4 cores) with a clock speed of 2.5 GHz or higher (Intel Core i5/i7 or AMD Ryzen 5/7 level) to provide parallel processing of network packets.

Random Access Memory (RAM): At least 16 GB (processing large network traffic datasets in RAM requires a significant amount of resources).

Graphics Card (GPU): A discrete graphics accelerator that supports CUDA technology (e.g., NVIDIA GeForce RTX 20 series or higher) is recommended to accelerate neural network training.

Storage: Free disk space of at least 50 GB (for storing traffic dumps in .pcap format and training datasets).

Network adapter: At least 1 Gbps data transfer rate.

Software. The laboratory bench should be based on modern software for data analysis and cybersecurity:

Operating system: Windows 10/11 or Linux (Ubuntu 20.04 LTS and above is recommended for better compatibility with data tools).

Development environment: Python programming language version 3.9 and higher. Interactive development environment Jupyter Notebook (as part of the Anaconda distribution) or VS Code / PyCharm.

Specialized Python libraries:

- Pandas, NumPy – for preprocessing network data;
- Scikit-learn – for the implementation of classical machine learning algorithms;

- TensorFlow or PyTorch – for building and training deep neural networks;
- Matplotlib, Seaborn – for visualizing results and analyzing anomalies.

Network analysis tools:

- Wireshark and Tcpdump – for capturing and primary analysis of network packets;

- CICFlowMeter – for Feature Extraction from .pcap traffic files.

Teaching materials:

Training Datasets: NSL-KDD, CIC-IDS2017, UNSW-NB15 (must be preloaded on workstations).

Code Snippets for implementing data loading and normalization modules.

Safety Instructions

When performing work in a computer class, it is prohibited:

1. Run neural network training algorithms with an infinite loop, which can lead to overheating of equipment.
2. Use the developed intellectual analysis tools to carry out unauthorized attacks on the nodes of the local computer network of the university.
3. Independently change the configuration of system software and network protocols without the consent of the teacher.
4. In case of signs of abnormal operation of the equipment (burning smell, strong noise of the cooling system, critical slowdown of the interface), immediately stop calculations and inform the system administrator.

Practical tasks:

1. Explore the concept of the Adversarial Example in the context of network packets.
2. Implement a simple evasion attack by modifying packet time intervals (Jittering).

3. Conduct an experiment to bypass the classifier using the FGSM (Fast Gradient Sign Method) algorithm on traffic data.
4. Simulate a "data poisoning" attack in a training sample and assess the consequences.
5. To study the effect of the "protective distillation" method on the stability of the neural network.
6. Conduct stress testing of IDS using adversarial generation tools (e.g., ART or CleverHans libraries).
7. To assess the robustness (stability) of the model to random noise in the network channel.
8. Implement adversarial learning (adding malicious examples to the training set to increase protection).
9. To compare the resistance of classical algorithms (Random Forest) and deep neural networks to adversarial influences.
10. Develop a report to assess the security of the IDS intelligent module from external manipulation.

Questions for self-examination:

1. What is an Adversarial Attack on AI?
2. What is the difference between Evasion and Poisoning attacks?
3. How can a small change in traffic signs fool a neural network?
4. What is Adversarial Training and how does it help defense?
5. What Python libraries are used to test the resilience of AI models?
6. Is it possible to fully protect an AI system from adversarial examples?
7. Why are deep neural networks often more vulnerable to adversarial attacks than simple algorithms?

References:

1. Potapov, A. S. Recognition of Images and Machine Learning: A Textbook. St. Petersburg, ITMO University Publ., 2015. 164 p. (in Russian).
2. Russell, S., Norvig, P. Artificial intelligence: a modern approach. 2nd ed. Moscow, Williams Publ., 2019. 1408 p. (in Russian).

3. Bishop, K. Recognition of Images and Machine Learning. Moscow: Intellect Publishing House, 2020. 960 p.
4. Vapnik V. N. Teoriya statisticheskogo obucheniya [Theory of statistical learning]. Moscow, Nauka Publ., 1999. 356 p. (in Russian).
5. Kuzin, A. V. Computer Networks: A Textbook. Moscow, Forum Publ., 2014. 192 p. (in Russian).

MINISTRY OF SCIENCE AND HIGHER EDUCATION
OF THE RUSSIAN FEDERATION
FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION OF
HIGHER EDUCATION
"NORTH CAUCASUS FEDERAL UNIVERSITY"

METHODOLOGICAL RECOMMENDATIONS
on the organization of independent work of students
in the discipline "Network intrusion detection systems based on AI"
for students of the field of study
11.03.02 Infocommunication Technologies and Communication Systems,
Profile "Intelligent Data Processing in Infocommunication Systems and
Networks"

Stavropol, 2026

Methodical instructions for the organization of students' independent work are compiled in accordance with the requirements of the discipline program "AI-Based Network Intrusion Detection Systems» for bachelors of the direction of training 11.03.02. The textbook is devoted to the planned educational work of students, performed in extracurricular (classroom) time on the assignment and under the methodological guidance of the teacher, but without his direct participation, with partial, direct participation of the teacher.

CONTENTS

1. Independent work as the most important form of the educational process	41
2. Goals and main objectives of the SRS	42
3. Types of independent work	43
4. Organisation of SSDs.....	44
5. General recommendations for the organization of independent work	46
6. Independent work of a student is a necessary link in the formation of a researcher.....	49
7. Methodical recommendations for students on certain forms of independent work.....	51
8. Educational, methodological and information support of the discipline.....	57

1. Independent work as the most important form of the educational process

Independent work is the planned educational, research, research work of students, performed in extracurricular (classroom) time on the assignment and under the methodological guidance of the teacher, but without his direct participation (with the partial direct participation of the teacher, leaving the leading role behind the work of students).

Independent work of students at the university is an important type of educational and scientific activity of the student. Independent work of students plays a significant role in the rating technology of education. Education at the university includes two parts that are almost the same in volume and mutual influence - the process of learning and the process of self-study. Therefore, the SRS should become an effective and purposeful work of the student.

The concept of modernization of Russian education defines the main tasks of vocational education - "training a qualified worker of the appropriate level and profile, competitive in the labor market, competent, responsible, fluent in his profession and oriented in related fields of activity, capable of effective work in the specialty at the level of world standards, ready for constant professional growth, social and professional mobility."

The solution of these problems is impossible without increasing the role of students' independent work on educational material, increasing the responsibility of teachers for the development of independent work skills, for stimulating the professional growth of students, fostering creative activity and initiative.

Society imposes a fairly wide list of requirements on a modern specialist, among which it is important that graduates have certain abilities and ability to independently obtain knowledge from various sources, systematize the information received, and assess a specific financial situation. The formation of such a skill occurs during the entire period of study through the participation of students in laboratory classes, the performance of control tasks and tests, the writing of term papers and graduation qualification papers. At the same time, the independent work of students plays a decisive role in the entire educational process.

The forms of independent work of students are diverse. In this discipline "Network intrusion detection systems based on artificial intelligence" the following are provided:

Use of the material of the educational and methodological complex of the discipline

At the first stage, it is necessary to familiarize yourself with the mandatory minimum content of the basic educational program in the discipline studied. Next, it is necessary to study the work program of the discipline, which considers the content of the topics of the lecture course, the relationship between the topics of lectures and laboratory classes, topics and types of independent work. For each type of independent work, certain forms of reporting are provided presented in the curriculum for the discipline "Network intrusion detection systems based on artificial intelligence".

Work with literature

For the successful development of the discipline, it is necessary to independently study the presented topics in detail according to the recommended sources of information that are presented in the curriculum.

Independent work introduces students to scientific creativity, the search for and solution of urgent modern problems.

2. Goals and main objectives of the SRS

The leading goal of the organization and implementation of SRS should coincide with the goal of teaching a student - the preparation of a bachelor with higher education. In the organization of SRS, an important and necessary condition is the formation of the ability to work independently to acquire knowledge, skills and the ability to organize educational and scientific activities.

The purpose of students' independent work is to master fundamental knowledge, professional skills and abilities in the profile, experience in creative and research activities. Independent work of students contributes to the development of

independence, responsibility and organization, a creative approach to solving problems of the educational and professional level.

The objectives of the SRS are:

- systematization and consolidation of the theoretical knowledge and practical skills of students;
- deepening and expanding theoretical knowledge;
- formation of skills to use regulatory, legal, reference documentation and special literature;
- development of cognitive abilities and activity of students: creative initiative, independence, responsibility and organization;
- formation of independent thinking, abilities for self-development, self-improvement and self-realization;
- development of research skills;
- the use of material collected and obtained in the course of independent studies at seminars, laboratory classes, when writing term papers and final qualification papers, for effective preparation for final tests and exams.

3. Types of independent work

In the educational process of a higher professional educational institution, two types of independent work are distinguished: classroom, under the guidance of a teacher, and extracurricular. A close relationship between these types of work provides for differentiation and effectiveness of the results of its implementation and depends on the organization, content, logic of the educational process (interdisciplinary connections, promising knowledge, etc.):

Classroom independent work on the discipline is carried out in classes under the direct guidance of the teacher and on his assignment.

Extracurricular independent work is performed by the student on the instructions of the teacher, but without his direct participation.

The main types of independent work of students without the participation of a teacher in this discipline are:

- formation and assimilation of the content of lecture notes on the basis of educational literature recommended by the lecturer, including information educational resources (electronic textbooks, electronic libraries, etc.);
- preparation for laboratory work, their design.

The main types of independent work of students with the participation of teachers are:

- ongoing consultations;
- acceptance and defense of laboratory work (during the L/R).

4. Organisation of SSDs

The methodology for organizing students' independent work depends on the structure, nature and features of the discipline "Network Intrusion Detection Systems Based on Artificial Intelligence", the number of hours for its study, the type of tasks for independent work of students, the individual qualities of students and the conditions of educational activity.

The process of organizing students' independent work includes the following stages:

- preparatory (definition of goals, drawing up a program, preparation of methodological support, preparation of equipment);
- the main one (implementation of the program, the use of methods of information search, assimilation, processing, application, transfer of knowledge, recording of results, self-organization of the work process);
- final (assessment of the significance and analysis of the results, their systematization, assessment of the effectiveness of the program and methods of work, conclusions about the directions of labor optimization).

Students' Activities for the Formation and Development of Skills of Educational Independent Work

In the process of independent work, the student acquires the skills of self-organization, self-control, self-management, self-reflection and becomes an active independent subject of educational activity.

Performing independent work under the supervision of a teacher, the student must:

- master the minimum content submitted for independent work of students and proposed by the teacher in accordance with the educational standards of higher education;
- plan independent work in accordance with the schedule of independent work proposed by the teacher;
- the student must carry out independent work in the organizational forms provided for by the curriculum and the teacher's work program;
- perform independent work and report on its results in accordance with the schedule for submitting results, types and terms of reporting on students' independent work.

A student can:

in addition to the one proposed by the teacher (with justification and agreement with him) and the minimum mandatory content determined by the program for this discipline:

- independently determine the level (depth) of the study of the content of the material;
- to suggest topics and questions for independent study;
- within the framework of the general schedule for the performance of independent work, offer a reasonable individual schedule for the implementation and reporting on the results of independent work;
- offer their own options for organizational forms of independent work;
- use for independent work methodological manuals, manuals, development of the list proposed by the teacher;
- to use not only control, but also self-control of the results of independent work in accordance with the methods of self-control proposed by the teacher or chosen independently.

Independent work of students should have an important impact on the formation of the personality of the future bachelor, it is planned by the student

independently. Each student independently determines the mode of his work and the degree of work spent on mastering the educational content in each discipline. He performs extracurricular work according to a personal individual plan, depending on his preparation, time and other conditions.

5. General recommendations for the organization of independent work

The main form of independent work of the student is the study of lecture notes, their supplementation, recommended literature, active participation in laboratory classes. But for successful educational activity, its intensification, it is necessary to take into account the following subjective factors:

1. Knowledge of school program material, the presence of a solid system of gaps necessary for mastering the main university courses. This is especially important for mathematical disciplines. It is necessary to distinguish gaps in knowledge that make it difficult to assimilate new material from small abilities. By spending effort to overcome these gaps, the student will ensure normal academic performance and believe in his abilities.

2. Availability of skills, skills of mental work:

- a) the ability to take notes at lectures and when working with a book;
- b) mastery of logical operations: comparison, analysis, synthesis, generalization, definition of concepts, rules of systematization and classification.

3. The specifics of cognitive mental processes: attention, memory, speech, observation, intelligence and thinking. Poor development of each of them becomes a serious obstacle in learning.

4. Good working capacity, which is ensured by normal physical condition. After all, serious learning is a large, versatile and diverse work. The result of learning is evaluated not by the amount of information communicated, but by the quality of its assimilation, the ability to use it and the development of one's ability to further independent education.

5. Correspondence of the chosen activity, profession to individual abilities. It is necessary to develop the ability to self-regulate one's emotional state and eliminate circumstances that disturb the business mood, interfere with the planned work.

6. Mastering the optimal style of work that ensures success in activities. Alternating work and pauses in work, rest periods, individually justified sleep duration, preference for evening or morning classes, stress resistance at exams and features of preparation for them,

7. The level of demands on oneself, determined by the established self-esteem.

An adequate assessment of knowledge, advantages, disadvantages is an important component of a person's self-organization, without it it is impossible to successfully manage one's behavior and activity.

One of the main features of education in higher school is that constant external control is replaced by self-control, the active role in learning belongs not so much to the teacher as to the student.

Knowing the basic methods of scientific organization of mental work, it is possible to achieve the best results with the least expenditure of time, money and labor efforts.

The effectiveness of assimilation of incoming information depends on the working capacity of a person at a particular moment of his activity.

Working capacity is the ability of a person to work with a high degree of intensity for a certain time. There are internal and external factors of working capacity.

Internal factors of working capacity include intellectual characteristics, will, and state of health.

External:

- organization of the workplace, work and rest schedule;
- the level of labor organization - the ability to obtain information and use information;
- the amount of mental load.

The outstanding Russian physiologist N. E. Vvedensky identified the following conditions for the productivity of mental activity:

- Any work should be entered gradually;
- dimensionality and rhythm of work. Different people have a more or less different pace of work;
- the usual sequence and systematicity of activity;
- correct alternation of work and rest.

Rest does not imply obligatory complete inaction on the part of a person, it can be achieved by a simple change of business. During the day, the working capacity changes. The most fruitful is the *morning time (from 8 to 14 hours)*, and the maximum working capacity falls on the period from 10 to 13 hours, then *the afternoon (from 16 to 19 hours)* and *the evening (from 20 to 24 hours)*. It is better to study the material that is very difficult to understand at the beginning of each period of time (preferably in the morning) after a good rest. After 1-1.5 hours, breaks of 10 - 15 minutes are needed, after 3 - 4 hours of work, the rest should be long - about an hour.

An integral part of the scientific organization of mental labor is the mastery of the technique of mental labor.

The time that the student has to complete the curriculum consists of two components: one of them is classroom work at the university according to the schedule of classes, the other is extracurricular independent work. Tasks and materials for independent work are given during classes according to the schedule, in the same classes the teacher controls independent work, and also assists students in the correct organization of work.

Self-study will require intensive mental work, which must not only be properly organized, but also stimulated. At the same time, it is very important to be able to maintain steady attention to the material being studied. The development of attention requires significant volitional efforts. That is why, if a student notices that he is often distracted during independent studies, he needs to force himself to concentrate. Such

a procedure must be done constantly, as this is attention training. Sustained attention appears when a person treats the matter with interest.

It is necessary to properly organize your classes in time: 50 minutes - work, 5-10 minutes - break; after 3 hours of work, a break - 20-25 minutes. Otherwise, increasing fatigue will lead to instability of attention. A very significant factor affecting the increase in mental performance is systematic physical education. The organization of active recreation provides for the alternation of mental and physical activity, which completely restores a person's working capacity.

6. Independent work of a student is a necessary link in the formation of a researcher

The progress of science and technology, information technology leads to a significant increase in scientific information, which imposes higher requirements not only on the moral and ethical qualities of a person, but also, in particular, constantly increasing requirements in the field of education - renewal, modernization of general and professional knowledge, skills of a specialist.

Any education should act as a dynamic process inherent in a person and continuing throughout his life. Mastering scientific thought and the language of science is a necessary component in the self-organization of a future research specialist. This is understood not so much as the accumulation of knowledge, but the mastery of scientifically based ways of acquiring it. This, generally speaking, is the main task of the university.

The specificity of the university educational process, in the organization of which more and more space is given to the independent work of the student, is that it seems to be the last and most adequate link for the implementation of this task. For during the study at the university, the development of the style and skills of educational (cognitive) activity takes place, the rational nature of which will contribute to the constant renewal of the knowledge of a highly qualified university graduate.

However, there are certain difficulties before this path, in particular, the transition of the student from the synthetic process of learning in secondary school to the analytical one in higher school. This is due both to the new content of education (expansion of general education and deepening of professional training) and to new, hitherto unknown forms of training (lectures, seminars, laboratory classes, etc.). The student receives not only the knowledge provided by the program and teaching aids, but he must also become familiar with the ways of acquiring knowledge in such a way as to be able to assess what we know, how we know it and how we have achieved this knowledge. All this is achieved through one's own independent work.

This is also because self-acquired knowledge is more operational, it becomes personal property, as well as a motive for behavior, develops intellectual traits, attention, observation, criticality, and the ability to evaluate. The role of the teacher is mainly to guide the accumulation of knowledge (in relation to first-year students), and in the subsequent years of study, in the senior years, in the joint identification of problems and care for the student's independent search, as well as control over their activities. It should be noted that it is not possible to limit oneself only to the acquisition of knowledge provided for by the program of the discipline studied, it is necessary to constantly deepen the knowledge gained, focusing it on some narrow certain area corresponding to the interests of the student. In-depth study of all subjects provided for by the program is possible in practice, and good organization of work allows you to save time, which creates conditions for a deep, systematic, interested study of the topic independently chosen by the student.

Of course, all the advice, examples, recommendations in this area, given by the teacher, or certain publications, or other sources, do not guarantee any success without the manifestation of one's own activity in this matter, i.e. they do not give ready-made recipes, but should contribute to the analysis of one's own work, its goals, organization in accordance with individual characteristics. Taking into account personal capabilities, existing living and working conditions, skills, on the basis of these recommendations, it is possible to develop an individually justified set

of methods, ways, find one's own style or improve it, so that, having studied certain material, have time to assess its significance, suitability and possibilities of its application, in order to ultimately ensure the success of one's studies from future professional activity

7. Methodical recommendations for students on certain forms of independent work

The system of university education implies much greater independence of students in planning and organizing their activities.

Work with literature

When working with literature, it is necessary to select literature, learn how to read it correctly, and keep notes. For the selection of literature in the library, alphabetical and systematic catalogs are used.

It is important to remember that rational skills of working with a book are always a great saving of time and effort.

The correct selection of textbooks is recommended by the teacher who reads the lecture course. The necessary literature can also be indicated in the methodological developments for this course.

Studying the material from the textbook, you should move on to the next question only after the correct understanding of the previous one, describing on paper all the calculations and calculations (including those that are omitted in the textbook or given in the lecture for independent conclusion).

In the study of any discipline, independent individual work plays a large and important role.

Particular attention should be paid to the definition of the basic concepts of the course. The student should analyze in detail the examples that explain such definitions, and be able to build similar examples independently. It is necessary to achieve an accurate idea of what you are studying. It is useful to make reference notes. When studying the material from the textbook, it is useful to supplement the

lecture notes in the notebook (in specially designated margins). There you should also note the questions selected by the student for consultation with the teacher.

It is recommended to highlight the conclusions obtained as a result of the study in the notes so that they are better remembered when rereading the notes.

Experience shows that many students are helped by compiling a list of reference signals containing the most important and most frequently used formulas and concepts. Such a sheet helps to memorize formulas, the main points of the lecture, and can also serve as a permanent reference for the student.

There are two types of reading; primary and secondary. *Primary reading* is attentive, unhurried reading, in which you can dwell on difficult places. After it, not a single incomprehensible tin should remain. The content may not always be understood after the primary reading.

The task of *secondary* reading is the complete assimilation of the meaning of the whole (this reading may not be the second, but the third or fourth).

Reading a scientific text is a part of cognitive activity. Its purpose is to extract the necessary information from the text. The effectiveness of the action largely depends on how conscious the reader is of his own inner attitude when referring to the printed word (to find the necessary information, to assimilate the information fully or partially, to critically analyze the material, etc.).

There are *four main attitudes in reading a scientific text*:

1. information retrieval (the task is to find, highlight the information sought)
2. assimilative (the reader's efforts are aimed at understanding and remembering as fully as possible both the information presented by the author and the entire logic of his reasoning)
3. analytical-critical (the reader seeks to critically comprehend the material, analyzing it, determining his attitude to it)
4. creative (creates in the reader a readiness in one form or another – as a starting point for his reasoning, as an image for action by analogy, etc. – to use the author's judgments, the course of his thoughts, the result of observation, the developed methodology, to supplement them, to subject them to a new check).

The existence of several types of reading *is associated with the presence of different attitudes to addressing a scientific text:*

1. Bibliographic – viewing catalog cards, recommendation lists, summary lists of journals and articles for the year, etc.;

2. Viewing – used to search for materials containing the necessary information, usually it is used immediately after working with lists of references and catalogs, as a result of such viewing the reader establishes which of the sources will be used in further work;

3. introductory – implies a continuous, sufficiently detailed reading of the selected articles, chapters, individual pages, the purpose is to get acquainted with the nature of the information, to find out what issues are submitted by the author for consideration, to sort the material;

4. Studying – assumes a thorough mastery of the material; in the course of such reading, the reader's trust in the author, readiness to accept the information presented, the attitude to the most complete understanding of the material is realized;

5. Analytical-critical and creative reading are two types of reading that are close to each other in that they participate in solving research problems. The first of them involves a directed critical analysis of both the information itself and the ways in which it is obtained and presented by the author; the second is the search for those judgments, facts, on which or in connection with which the reader considers it necessary to express his own thoughts.

Of all the types of reading considered, the main one for students is the learning one - it allows them to accumulate knowledge in various fields when working with educational literature. That is why this type of reading should be mastered in the first place. In addition, when mastering this type of reading, the main techniques are formed that increase the effectiveness of working with a scientific text.

The main types of systematized recording of what has been read:

1. Annotation is an extremely brief coherent description of the book (article) viewed or read, its contents, sources, nature and purpose;

2. Planning is a brief logical organization of the text, revealing the content and structure of the studied material;

3. Thesis is a concise reproduction of the author's main statements without involving factual material;

4. Quotation is a verbatim excerpt from the text of excerpts, extracts that most significantly reflect one or another thought of the author;

5. Note-taking is a brief and consistent presentation of the content of what has been read.

A synopsis is a complex way of presenting the content of a book or article in a logical sequence. A synopsis accumulates the previous types of notes, allows you to comprehensively cover the content of a book or article. Therefore, the ability to make a plan, theses, make extracts and other notes determines the technology of compiling a synopsis.

Methodical recommendations for compiling a synopsis:

1. Read the text carefully. Clarify the incomprehensible words in the reference literature. When writing down, do not forget to put the reference data in the margins of the note.

2. Highlight the main thing, make a plan.

3. Briefly formulate the main provisions of the text, note the author's argumentation.

4. Take notes of the material, clearly following the points of the plan. When taking notes, try to express the idea in your own words. Notes should be kept clearly.

5. Write down quotes correctly. When quoting, take into account the conciseness and significance of the thought.

In the text of the synopsis, it is desirable to give not only thesis statements, but also their proofs. When drawing up a synopsis, it is necessary to strive for the capacity of each sentence. The thoughts of the author of the book should be stated briefly, taking care of the style and expressiveness of what is written. The number of additional elements of the synopsis should be logically justified, the notes should

be distributed in a certain sequence that corresponds to the logical structure of the work. Margins should be left for clarification and addition.

Mastering the skills of taking notes requires purposefulness and daily independent work from the student.

Practical exercises

It should be emphasized that only after mastering the lecture material from a certain point of view (namely, from the point of view from which it is presented at the lectures) it will be consolidated in the laboratory classes both as a result of discussion and analysis of the lecture material, and with the help of completing the tasks of the laboratory work, the tasks set. Under these conditions, the student will not only learn the material well, but also learn how to apply it in practice, and will also receive an additional incentive (and this is very important) for active study of the lecture.

When performing tasks independently, it is necessary to justify each stage of the solution, based on the theoretical provisions of the course. If the student sees several ways to solve a problem (task), then it is necessary to compare them and choose the most rational one. It is useful to draw up a brief plan for solving the problem (task) before starting calculations. The solution of problems or examples should be presented in detail, calculations should be arranged in a strict order, separating auxiliary calculations from the main ones. Solutions, if necessary, should be accompanied by comments, diagrams, drawings and drawings.

It should be remembered that the performance of each educational task should be brought to the final logical answer required by the condition, and, if possible, with a conclusion. The performance of tasks of this type should be continued until the acquisition of solid skills in solving them.

Self-Test

After studying a certain topic from the notes in the notes and textbook, as well as solving a sufficient number of relevant tasks in laboratory classes and

independently, the student is recommended, using a sheet of reference materials, to reproduce from memory definitions, conclusions of formulas, formulations of the main provisions and evidence.

If necessary, you need to carefully understand the material again.

Sometimes the insufficiency of assimilation of a particular issue becomes clear only when studying further material. In this case, it is necessary to go back and repeat the poorly mastered material. An important criterion for mastering theoretical material is the ability to solve problems or pass a test on the material covered. However, it should be remembered that the correct solution to the problem can be obtained as a result of the use of mechanically memorized formulas without understanding the essence of theoretical provisions.

Consulting

If in the process of independent work on the study of theoretical material or when performing tasks, the student has questions that cannot be solved independently, it is necessary to contact the teacher to obtain explanations or instructions from him. In his questions, the student should clearly express what he is experiencing difficulties, the nature of this difficulty. Advice should also be sought if there are doubts about the correctness of the answers to the self-test questions.

Preparation for the test with a grade in the discipline "Information Security"

The study of many general professional and special disciplines ends with a test. Preparation for the test contributes to the consolidation, deepening and generalization of knowledge obtained in the process of study, as well as its application to solving practical problems. Preparing for the test, the student eliminates the existing gaps in knowledge, deepens, systematizes and organizes his knowledge.

To receive credit, the student must fully complete the tasks of the laboratory work of the classes, prepare a report on each laboratory work and a report on the results of independent study of topics, according to the curriculum. If there are debts on the current certification in this discipline, the student does not receive credit.

Systematic preparation for classes during the semester will allow you to use the time of the exam session to systematize knowledge.

Current certification of students is carried out by teachers who conduct laboratory classes in the discipline in the following forms: interview when defending laboratory work with the provision of a written report. Admission to laboratory work occurs if students have a printed version of the report. The defense of the report takes place in the form of an interview. The maximum number of points (according to the five-point assessment system) is received by the student if the design of the report meets the established requirements, and the report fully reveals the essence of the work. The grounds for reducing the grade are:

- poorly performed work on a personal computer,
- the written report is made in a sloppy form and with errors,
- the student does not give an exhaustive answer to the teacher's question,
- Gets confused or finds it difficult to demonstrate work on the computer.

The report can be sent for revision in the following cases:

- negligent execution and with errors,
- abbreviated content of the report.

The criteria for evaluating the interview are given in the Fund of Assessment Tools for the discipline "Information Security".

8. Educational, methodological and information support of the discipline

On this discipline, the teacher recommends the following literature:

The following are considered as the main literature:

1. Averchenkov V. I., Rytov M. Yu., Kuvyklin A. V., Gainulin T. R. Metody i sredstva inzhenerno-tekhnicheskogo zashchity informatsii [Methods and Means of Engineering and Technical Protection of Information]. 2nd ed., ster. Moscow, Flinta Publ., 2011. 187 p.
http://old.biblioclub.ru/93275_Metody_i_sredstva_inzhenerno_tekhnicheskoi_zashchity_informatsii_uchebnoe_posobie.html

2. Chipiga, A. F. (SevKavGTU). Multichannel measuring systems for monitoring the safety of objects: textbook. textbook / A. F. Chipiga, V. V. Fedorenko, I. V. Fedorenko; FBGOU Sev.-Kav. Gos. Techn. University. - Stavropol : SevKavSTU Publishing House, 2011. - 126 p. : ill. - References: p. 123-128 (58 titles)

3. Chipiga, A. F. Informatsionnaya bezopasnost' avtomatizirovannykh sistem [Information security of automated systems]. Textbook. Moscow, Gelios ARV Publ., 2010. 336 p. (in Russian).

4. Melnikov V. P., Kleymenov S. A., Petrakov A. M. Informatsionnaya bezopasnost' i zashchita informatsii [Information security and information protection]. Melnikov V. P., Kleymenov S. A., Petrakov A. M. ; ed. by S. A. Kleymenov. – 3rd ed., ster. – Moscow : Academy, 2008. – 331 p.

As additional literature, the following are considered:

1. Chipiga, A. F. (SevKavGTU). Information Security: Textbook. Manual (Practicum) / A. F. Chipiga, M. A. Lapina, V. V. Belous ; M-vo obrazovaniya i nauki Ros. Federatsii, GOU VPO Sev.-Kav. Gos. Techn. University. – Stavropol : SevKavGTU, 2009. – 282 p.

2. Sergeeva Y. S. Information Protection. Lecture Notes. Textbook. - Moscow: A-Prior, 2011. - 128 p.
http://old.biblioclub.ru/72670_Zaschita_informatsii_Konspekt_lektsii_Uchebnoe_posobie.html

3. Sychev Y. N., Fundamentals of Information Security. Educational and Practical Manual - Moscow: Eurasian Open Institute, 2010. - 328 p.
http://old.biblioclub.ru/90790_Osnovy_informatsionnoi_bezopasnosti_Uchebno_p_rakticheskoe_posobie.html

4. Shangin V. F., Protection of Computer Information. Effective Methods and Means - Moscow: DMK Press, 2010. - 544 p. APPROVED BY the UMO of universities on university polytechnic education as a textbook for students of higher educational institutions.

http://old.biblioclub.ru/86475_Zaschita_kompyuternoi_informatsii_Effektivnye_metody_i_sredstva.html

List of educational and methodological support for students' independent work in the discipline

1. Federal Law "On Information, Information Technologies and Information Protection"
2. The concept of national security
3. Federal Law of the Russian Federation "On Security"

List of resources of the information and telecommunication network "Internet" necessary for mastering the discipline

<http://biblioclub.ru/> – University Library Online "Biblioclub"

List of information technologies used in the implementation of the educational process in the discipline, including a list of software and information reference systems

When giving lectures, computer equipment is used, demonstration of presentation multimedia materials. In practical classes, students defend reports on the work that they perform independently or in a team using computer equipment and Internet technologies

Information reference systems:

Information and reference and information and legal systems used in the study of the discipline:

Information and reference and information and legal systems used in the study of the discipline:

- 1 <http://catalog.ncfu.ru/catalog/ncfu> – Official website of the Scientific Library of the FGAOU HE NCFU.
- 2 <http://www.consultant.ru> – Official website of Consultant Plus
- 3 <https://www.mnr.gov.ru> – Official website of the Ministry of Natural Resources and Environment of the Russian Federation

4 Information and Legal Portal "Garant" - <http://www.garant.ru/>

Software:

- 1 Viola Workstation 10
- 2 Alt Workstation K
- 3 Viola "Server"
- 4 Office software package - P7-Office
- 5 XSpider
- 6 McAfee
- 7 DLP systems