

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания
по выполнению практических работ
по дисциплине «**Модели машинного обучения, сохраняющие
конфиденциальность**»
для студентов направления подготовки
02.04.01 Математика и компьютерные науки
профиль «Искусственный интеллект в кибербезопасности»

Ставрополь, 2026

Содержание

<u>Практическая работа 1. "Методы машинного обучения"</u>	3
<u>Практическая работа 2. "Применение методов машинного обучения"</u>	6
<u>Практическая работа 3. «Введение в использование TenSEAL»</u>	8
<u>Практическая работа 4. «Исследование линейной регрессии, сохраняющей конфиденциальность.»</u>	11
<u>Практическая работа 5. «Работа с приближенными числами в гомоморфном шифровании»</u>	14
<u>Практическая работа 6. «Разработка сверточной нейронной сети, сохраняющей конфиденциальность, на базе набора данных MNIST»</u>	17

Практическая работа 1. "Методы машинного обучения"

Цель работы: Изучить основные методы машинного обучения, включая линейную и логистическую регрессию, с использованием библиотеки PyTorch на языке Python.

Общий пример: Пример реализации линейной и логистической регрессии в PyTorch.

```
import torch
import torch.nn as nn
import torch.optim as optim
import numpy as np

# Создаем синтетические данные для линейной регрессии
x_train = torch.tensor([[1.0], [2.0], [3.0], [4.0]], dtype=torch.float32)
y_train = torch.tensor([2.0, 4.0, 6.0, 8.0], dtype=torch.float32)

# Линейная модель
model = nn.Linear(1, 1)
criterion = nn.MSELoss()
optimizer = optim.SGD(model.parameters(), lr=0.01)

# Обучение модели
for epoch in range(1000):
    optimizer.zero_grad()
    output = model(x_train)
    loss = criterion(output, y_train)
    loss.backward()
    optimizer.step()

print("Коэффициенты линейной модели:", model.weight.item(), model.bias.item())

# Логистическая регрессия
x_train = torch.tensor([[1.0], [2.0], [3.0], [4.0]], dtype=torch.float32)
y_train = torch.tensor([0, 0, 1, 1], dtype=torch.float32)

class LogisticRegression(nn.Module):
    def __init__(self):
        super(LogisticRegression, self).__init__()
        self.linear = nn.Linear(1, 1)

    def forward(self, x):
        return torch.sigmoid(self.linear(x))

model = LogisticRegression()
criterion = nn.BCELoss()
optimizer = optim.SGD(model.parameters(), lr=0.01)

for epoch in range(1000):
    optimizer.zero_grad()
    output = model(x_train)
    loss = criterion(output, y_train)
    loss.backward()
    optimizer.step()

print("Коэффициенты логистической модели:", model.linear.weight.item(),
      model.linear.bias.item())
```

Задачи на работу:

1. Установить и настроить библиотеку PyTorch.
2. Реализовать линейную регрессию на синтетических данных.
3. Обучить модель и визуализировать результаты.
4. Реализовать логистическую регрессию и провести предсказание.
5. Сравнить точность обеих моделей на тестовых данных.

Задания для студентов по вариантам:

Вариант 1:

- Реализовать линейную регрессию с другим набором данных (например, квадратичная зависимость).

Вариант 2:

- Реализовать логистическую регрессию с 2D-данными (например, две группы точек).

Вариант 3:

- Добавить регуляризацию L2 в линейную регрессию и сравнить результаты.

Вариант 4:

- Использовать другой оптимизатор (Adam) и проанализировать влияние на сходимость модели.

Вариант 5:

- Реализовать многомерную линейную регрессию (несколько входных переменных).

Вариант 6:

- Реализовать многомерную логистическую регрессию и проверить точность.

Вариант 7:

- Визуализировать процесс обучения модели с помощью графиков потерь.

Вариант 8:

- Реализовать стохастический градиентный спуск (SGD) с разными параметрами batch size.

Вариант 9:

- Использовать датасет sklearn для линейной регрессии и сравнить с PyTorch-версией.

Вариант 10:

- Разработать бинарный классификатор на основе логистической регрессии с более сложными данными.

Практическая работа 2. "Применение методов машинного обучения"

Цель работы: Изучить применение сверточных нейронных сетей (CNN) с использованием библиотеки PyTorch на наборе данных MNIST.

Общий пример: Пример загрузки данных MNIST, построения и обучения простой сверточной нейронной сети.

```
import torch
import torch.nn as nn
import torch.optim as optim
import torchvision
import torchvision.transforms as transforms

# Загрузка и предобработка данных MNIST
transform = transforms.Compose([transforms.ToTensor(),
                                transforms.Normalize((0.5,), (0.5,))])
trainset = torchvision.datasets.MNIST(root='./data', train=True, download=True,
                                       transform=transform)
trainloader = torch.utils.data.DataLoader(trainset, batch_size=64, shuffle=True)

# Определение сверточной нейронной сети
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 16, kernel_size=3, padding=1)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(16, 32, kernel_size=3, padding=1)
        self.fc1 = nn.Linear(32 * 7 * 7, 128)
        self.fc2 = nn.Linear(128, 10)

    def forward(self, x):
        x = self.pool(torch.relu(self.conv1(x)))
        x = self.pool(torch.relu(self.conv2(x)))
        x = x.view(-1, 32 * 7 * 7)
        x = torch.relu(self.fc1(x))
        x = self.fc2(x)
        return x

# Инициализация модели, функции потерь и оптимизатора
model = CNN()
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)

# Обучение модели
for epoch in range(5):
    for images, labels in trainloader:
        optimizer.zero_grad()
        outputs = model(images)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()

print("Обучение завершено")
```

Задачи на работу:

1. Установить и настроить библиотеку PyTorch.

2. Загрузить набор данных MNIST и подготовить изображения.
3. Реализовать сверточную нейронную сеть с двумя сверточными слоями.
4. Обучить модель и протестировать на тестовых данных.
5. Проанализировать точность модели.

Задания для студентов по вариантам:

Вариант 1:

- Добавить еще один сверточный слой и повторно обучить модель.

Вариант 2:

- Изменить функцию активации (например, заменить ReLU на LeakyReLU).

Вариант 3:

- Добавить слой batch normalization после каждого сверточного слоя.

Вариант 4:

- Изменить оптимизатор на SGD и проанализировать влияние на обучение.

Вариант 5:

- Реализовать dropout-слой перед полносвязными слоями.

Вариант 6:

- Использовать другой размер ядра (например, 5×5 вместо 3×3).

Вариант 7:

- Добавить аугментацию данных (например, повороты и смещения изображений).

Вариант 8:

- Провести тестирование модели на измененном (размытом) наборе данных.

Вариант 9:

- Реализовать функцию предсказания для одиночного изображения.

Вариант 10:

- Добавить графики изменения ошибки и точности в процессе обучения.

Практическая работа 3. «Введение в использование TenSEAL»

Цель работы: Ознакомиться с основами гомоморфного шифрования и научиться использовать библиотеку TenSEAL для выполнения операций над зашифрованными данными.

Общий пример: Пример создания контекста CKKS, шифрования векторов, выполнения операций сложения и умножения над зашифрованными данными и дешифрования результатов.

```
import tenseal as ts

# Создание контекста CKKS
context = ts.context(ts.SCHEME_TYPE.CKKS, poly_modulus_degree=8192,
coeff_mod_bit_sizes=[40, 20, 40])
context.global_scale = 2 ** 40
context.generate_galois_keys()

# Исходные данные
vector1 = [1.0, 2.0, 3.0, 4.0]
vector2 = [2.0, 3.0, 4.0, 5.0]

# Шифрование данных
enc_vector1 = ts.ckks_vector(context, vector1)
enc_vector2 = ts.ckks_vector(context, vector2)

# Выполнение операций
enc_sum = enc_vector1 + enc_vector2
enc_prod = enc_vector1 * enc_vector2

# Дешифрование результатов
print("Расшифрованная сумма:", enc_sum.decrypt())
print("Расшифрованное произведение:", enc_prod.decrypt())
```

Задачи на работу:

1. Установить и настроить библиотеку TenSEAL.
2. Создать контекст CKKS с определенными параметрами.
3. Зашифровать векторы с помощью созданного контекста.
4. Выполнить операции сложения и умножения над зашифрованными векторами.
5. Расшифровать результаты операций и проанализировать их.

Задания для студентов по вариантам:

Вариант 1:

- Создать контекст CKKS с параметрами: полиномиальная степень 8192, коэффициент масштабирования 2^{40} .

- Зашифровать векторы: [1.0, 2.0, 3.0, 4.0] и [2.0, 3.0, 4.0, 5.0].
- Выполнить сложение и умножение зашифрованных векторов.
- Расшифровать и вывести результаты.

Вариант 2:

- Создать контекст СККС с параметрами: полиномиальная степень 16384, коэффициент масштабирования 2^{30} .
- Зашифровать векторы: [0.5, 1.5, 2.5, 3.5] и [1.0, 2.0, 3.0, 4.0].
- Выполнить сложение и умножение зашифрованных векторов.
- Расшифровать и вывести результаты.

Вариант 3:

- Создать контекст СККС с параметрами: полиномиальная степень 8192, коэффициент масштабирования 2^{50} .
- Зашифровать векторы: [1.1, 2.2, 3.3, 4.4] и [0.9, 1.8, 2.7, 3.6].
- Выполнить сложение и умножение зашифрованных векторов.
- Расшифровать и вывести результаты.

Вариант 4:

- Создать контекст СККС с параметрами: полиномиальная степень 16384, коэффициент масштабирования 2^{35} .
- Зашифровать векторы: [2.0, 4.0, 6.0, 8.0] и [1.0, 3.0, 5.0, 7.0].
- Выполнить сложение и умножение зашифрованных векторов.
- Расшифровать и вывести результаты.

Вариант 5:

- Создать контекст СККС с параметрами: полиномиальная степень 8192, коэффициент масштабирования 2^{45} .
- Зашифровать векторы: [3.0, 6.0, 9.0, 12.0] и [1.5, 3.0, 4.5, 6.0].
- Выполнить сложение и умножение зашифрованных векторов.
- Расшифровать и вывести результаты.

Вариант 6:

- Создать контекст СККС с параметрами: полиномиальная степень 16384, коэффициент масштабирования 2^{40} .
- Зашифровать векторы: [0.1, 0.2, 0.3, 0.4] и [0.9, 0.8, 0.7, 0.6].

- Выполнить сложение и умножение зашифрованных векторов.
- Расшифровать и вывести результаты.

Вариант 7:

- Создать контекст СККС с параметрами: полиномиальная степень 8192, коэффициент масштабирования 2^{35} .
- Зашифровать векторы: [5.0, 10.0, 15.0, 20.0] и [2.0, 4.0, 6.0, 8.0].
- Выполнить сложение и умножение зашифрованных векторов.
- Расшифровать и вывести результаты.

Вариант 8:

- Создать контекст СККС с параметрами: полиномиальная степень 16384, коэффициент масштабирования 2^{50} .
- Зашифровать векторы: [7.0, 14.0, 21.0, 28.0] и [3.5, 7.0, 10.5, 14.0].
- Выполнить сложение и умножение зашифрованных векторов.
- Расшифровать и вывести результаты.

Вариант 9:

- Создать контекст СККС с параметрами: полиномиальная степень 8192, коэффициент масштабирования 2^{30} .
- Зашифровать векторы: [4.4, 8.8, 13.2, 17.6] и [2.2, 4.4, 6.6, 8.8].
- Выполнить сложение и умножение зашифрованных векторов.
- Расшифровать и вывести результаты.

Вариант 10:

- Создать контекст СККС с параметрами: полиномиальная степень 16384, коэффициент масштабирования 2^{45} .
- Зашифровать векторы: [9.0, 18.0, 27.0, 36.0] и [4.5, 9.0, 13.5, 18.0].
- Выполнить сложение и умножение зашифрованных векторов.
- Расшифровать и вывести результаты.

Практическая работа 4. «Исследование линейной регрессии, сохраняющей конфиденциальность.»

Цель работы: Изучить основы работы с линейной регрессией на зашифрованных данных с использованием библиотеки TenSEAL.

Общий пример: Пример создания контекста CKKS, шифрования входных данных, вычисления предсказаний линейной модели на зашифрованных данных и расшифрования результатов.

```
import tenseal as ts
import numpy as np

# Создание контекста CKKS
context = ts.context(ts.SCHEME_TYPE.CKKS, poly_modulus_degree=8192,
coeff_mod_bit_sizes=[40, 20, 40])
context.global_scale = 2 ** 40
context.generate_galois_keys()

# Коэффициенты модели линейной регрессии
weights = [0.5, -1.2, 3.4]
bias = 0.7

# Входные данные
inputs = [2.0, -1.0, 3.0]

# Шифрование входных данных
enc_inputs = ts.ckks_vector(context, inputs)
enc_weights = ts.ckks_vector(context, weights)

# Вычисление линейной комбинации (скалярное произведение)
enc_output = enc_inputs.dot(enc_weights) + bias

# Дешифрование результата
print("Расшифрованный результат:", enc_output.decrypt())
```

Задачи на работу:

1. Установить и настроить библиотеку TenSEAL.
2. Создать контекст CKKS с заданными параметрами.
3. Реализовать линейную регрессию на зашифрованных данных.
4. Провести операции шифрования, вычисления и дешифрования.
5. Проанализировать результаты.

Задания для студентов по вариантам:

Вариант 1:

- Использовать коэффициенты модели [0.4, -0.8, 2.5] и bias 1.2.
- Зашифровать входные данные [1.0, 2.0, -3.0].
- Вычислить линейную регрессию на зашифрованных данных.
- Расшифровать и вывести результат.

Вариант 2:

- Использовать коэффициенты модели [1.5, -0.5, 3.0] и bias -0.2.
- Зашифровать входные данные [-2.0, 3.0, 0.5].
- Вычислить линейную регрессию на зашифрованных данных.
- Расшифровать и вывести результат.

Вариант 3:

- Использовать коэффициенты модели [-0.7, 2.3, 1.1] и bias 0.5.
- Зашифровать входные данные [0.2, -1.5, 2.7].
- Вычислить линейную регрессию на зашифрованных данных.
- Расшифровать и вывести результат.

Вариант 4:

- Использовать коэффициенты модели [0.8, -2.1, 3.6] и bias 1.0.
- Зашифровать входные данные [-1.0, 2.5, 0.3].
- Вычислить линейную регрессию на зашифрованных данных.
- Расшифровать и вывести результат.

Вариант 5:

- Использовать коэффициенты модели [1.2, 0.3, -1.5] и bias -0.7.
- Зашифровать входные данные [3.0, -2.0, 1.5].
- Вычислить линейную регрессию на зашифрованных данных.
- Расшифровать и вывести результат.

Вариант 6:

- Использовать коэффициенты модели [2.0, -0.6, 1.8] и bias 0.4.
- Зашифровать входные данные [1.0, -3.0, 2.2].
- Вычислить линейную регрессию на зашифрованных данных.
- Расшифровать и вывести результат.

Вариант 7:

- Использовать коэффициенты модели [-0.9, 1.7, 0.6] и bias 2.1.
- Зашифровать входные данные [-2.3, 1.0, 3.5].
- Вычислить линейную регрессию на зашифрованных данных.
- Расшифровать и вывести результат.

Вариант 8:

- Использовать коэффициенты модели $[0.5, 1.2, -2.3]$ и bias 0.9.
- Зашифровать входные данные $[2.1, -1.7, 0.8]$.
- Вычислить линейную регрессию на зашифрованных данных.
- Расшифровать и вывести результат.

Вариант 9:

- Использовать коэффициенты модели $[1.9, -0.3, 2.8]$ и bias -1.5.
- Зашифровать входные данные $[0.6, -2.4, 1.3]$.
- Вычислить линейную регрессию на зашифрованных данных.
- Расшифровать и вывести результат.

Вариант 10:

- Использовать коэффициенты модели $[-1.4, 2.5, 0.7]$ и bias 1.8.
- Зашифровать входные данные $[3.2, -0.9, 1.1]$.
- Вычислить линейную регрессию на зашифрованных данных.
- Расшифровать и вывести результат.

Практическая работа 5. «Работа с приближенными числами в гомоморфном шифровании»

Цель работы: Изучить работу с приближенными числами в CKKS-гомоморфном шифровании с использованием библиотеки TenSEAL.

Общий пример: Пример шифрования, выполнения математических операций и расшифрования приближенных чисел в CKKS.

```
import tenseal as ts

# Создание контекста CKKS
context = ts.context(ts.SCHEME_TYPE.CKKS, poly_modulus_degree=8192,
coeff_mod_bit_sizes=[40, 20, 40])
context.global_scale = 2 ** 40
context.generate_galois_keys()

# Исходные данные (приближенные числа)
vector = [3.14159, 2.71828, 1.41421]

# Шифрование данных
enc_vector = ts.ckks_vector(context, vector)

# Выполнение операций
enc_result = enc_vector * 2.5 + 1.1

# Дешифрование результата
print("Расшифрованные значения:", enc_result.decrypt())
```

Задачи на работу:

1. Установить и настроить библиотеку TenSEAL.
2. Создать контекст CKKS с заданными параметрами.
3. Реализовать шифрование приближенных чисел.
4. Выполнить арифметические операции над зашифрованными числами.
5. Проанализировать потери точности при расшифровании.

Задания для студентов по вариантам:

Вариант 1:

- Зашифровать вектор [1.73205, 3.14159, 2.71828].
- Выполнить операцию: умножение на 3.5, затем сложение с 2.2.
- Расшифровать и вывести результат.

Вариант 2:

- Зашифровать вектор [0.7071, 1.4142, 2.2361].
- Выполнить операцию: умножение на 4.2, затем вычитание 1.5.
- Расшифровать и вывести результат.

Вариант 3:

- Зашифровать вектор [2.3026, 0.6931, 1.0986].
- Выполнить операцию: сложение с 3.3, затем деление на 2.
- Расшифровать и вывести результат.

Вариант 4:

- Зашифровать вектор [1.618, 2.718, 3.142].
- Выполнить операцию: умножение на 2.7, затем сложение с 0.8.
- Расшифровать и вывести результат.

Вариант 5:

- Зашифровать вектор [0.5772, 1.6180, 2.5029].
- Выполнить операцию: умножение на 5, затем сложение с 3.
- Расшифровать и вывести результат.

Вариант 6:

- Зашифровать вектор [1.732, 2.236, 2.645].
- Выполнить операцию: сложение с 1.5, затем умножение на 2.
- Расшифровать и вывести результат.

Вариант 7:

- Зашифровать вектор [2.3026, 1.0986, 0.6931].
- Выполнить операцию: вычитание 1.2, затем умножение на 3.4.
- Расшифровать и вывести результат.

Вариант 8:

- Зашифровать вектор [0.3010, 0.4771, 0.6990].
- Выполнить операцию: умножение на 6.5, затем вычитание 2.3.
- Расшифровать и вывести результат.

Вариант 9:

- Зашифровать вектор [1.204, 2.303, 3.505].
- Выполнить операцию: сложение с 0.9, затем умножение на 4.
- Расшифровать и вывести результат.

Вариант 10:

- Зашифровать вектор [2.6457, 3.7416, 4.1231].
- Выполнить операцию: вычитание 0.7, затем деление на 1.3.

- Расшифровать и вывести результат.

Практическая работа 6. «Разработка сверточной нейронной сети, сохраняющей конфиденциальность, на базе набора данных MNIST»

Цель работы: Изучить принципы работы сверточных нейронных сетей (CNN) на зашифрованных данных, используя библиотеку TenSEAL.

Общий пример: Пример загрузки данных MNIST, шифрования входных изображений, выполнения сверточных операций в зашифрованном виде и расшифрования результатов.

```
import tenseal as ts
import torch
import torchvision.transforms as transforms
from torchvision import datasets

# Загрузка и предобработка данных MNIST
transform = transforms.Compose([transforms.ToTensor(),
                                transforms.Normalize((0.5,), (0.5,))])
dataset = datasets.MNIST(root="./data", train=True, transform=transform,
                          download=True)

# Выбор одного изображения и преобразование в вектор
image, label = dataset[0]
image = image.view(-1).tolist()

# Создание контекста CKKS
context = ts.context(ts.SCHEME_TYPE.CKKS, poly_modulus_degree=8192,
                     coeff_mod_bit_sizes=[40, 20, 40])
context.global_scale = 2 ** 40
context.generate_galois_keys()

# Шифрование изображения
enc_image = ts.ckks_vector(context, image)

# Пример сверточной операции (упрощенный вариант)
kernel = [0.1] * len(image) # Упрощенное ядро
enc_result = enc_image.dot(kernel)

# Дешифрование результата
print("Расшифрованный результат свертки:", enc_result.decrypt())
```

Задачи на работу:

1. Установить и настроить библиотеку TenSEAL.
2. Загрузить набор данных MNIST и подготовить изображения.
3. Реализовать процесс шифрования изображений.
4. Выполнить сверточные операции на зашифрованных данных.
5. Проанализировать расшифрованные результаты и их качество.

Задания для студентов по вариантам:

Вариант 1:

- Выбрать 5 изображений из MNIST.

- Использовать ядро свертки размером 3×3 .
- Зашифровать изображения, выполнить свертку, расшифровать результат.

Вариант 2:

- Выбрать 10 изображений из MNIST.
- Применить два сверточных ядра 3×3 и 5×5 .
- Провести шифрование, вычисления и расшифрование.

Вариант 3:

- Выбрать одно изображение цифры «7».
- Применить ядро 3×3 и функцию активации ReLU.
- Проанализировать изменение значений после свертки.

Вариант 4:

- Выбрать 3 изображения цифры «5».
- Использовать ядро 5×5 и нормализацию значений после свертки.
- Дешифровать и интерпретировать результат.

Вариант 5:

- Выбрать случайные изображения из MNIST.
- Реализовать свертку с разными весами ядра и сравнить результаты.
- Провести анализ полученных данных.

Вариант 6:

- Выбрать 5 изображений.
- Применить свертку с ядром 3×3 , затем выполнить субдискретизацию (пулинг).
- Дешифровать и вывести итоговые значения.

Вариант 7:

- Использовать сверточное ядро с разреженными весами.
- Выбрать 4 изображения для обработки.
- Провести шифрование, сверточные операции и дешифрование.

Вариант 8:

- Выбрать изображения цифры «3».
- Реализовать сверточные операции и добавить второй сверточный слой.
- Дешифровать и проанализировать разницу между слоями.

Вариант 9:

- Выбрать 6 изображений из MNIST.
- Провести свертку и сравнить результаты с обычной (незашифрованной) CNN.
- Проанализировать различия в точности.

Вариант 10:

- Использовать 3 изображения и провести их поворот на 90 градусов перед обработкой.
- Применить сверточное ядро 5×5 и выполнить активацию.
- Дешифровать и рассмотреть влияние поворота на результат.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания

для обучающихся по организации и проведению самостоятельной работы
по дисциплине «МОДЕЛИ МАШИННОГО ОБУЧЕНИЯ, СОХРАНЯЮЩИЕ
КОНФИДЕНЦИАЛЬНОСТЬ»
для студентов направления подготовки

направленность (профиль):

Ставрополь
2026

СОДЕРЖАНИЕ

1. *Общие положения* 22
2. *Цель и задачи самостоятельной работы* 23
3. *Технологическая карта самостоятельной работы студента* 24
4. *Контрольные точки и виды отчетности по ним* 24
5. *Порядок выполнения самостоятельной работы студентом* 24
- 5.1. *Методические рекомендации по работе с учебной литературой* 24
6. *Список литературы для выполнения СРС* 28

1. Общие положения

Самостоятельная работа – планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов). Владиславу Вячеславовичу

Самостоятельная работа студентов (СРС) в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

К основным видам самостоятельной работы студентов относятся:

- формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);
- написание докладов;
- подготовка к семинарам, практическим и лабораторным работам, их оформление;
- составление аннотированного списка статей из соответствующих журналов по отраслям знаний (педагогических, психологических, методических и др.);
- выполнение учебно-исследовательских работ, проектная деятельность;
- подготовка практических разработок и рекомендаций по решению проблемной ситуации;
- выполнение домашних заданий в виде решения отдельных задач, проведения типовых расчетов, расчетно-компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин и т.д.;
- компьютерный текущий самоконтроль и контроль успеваемости на базе электронных обучающих и аттестующих тестов;
- выполнение курсовых работ (проектов) в рамках дисциплин;
- выполнение выпускной квалификационной работы и др.

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей изучаемой дисциплины, объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);

- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
 - заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).
- Самостоятельная работа по дисциплине «Информационные технологии командной работы и проектной деятельности» направлена на формирование следующих **компетенций**:

Код	Формулировка
ПК-2	Способен проводить научные исследования, на основе существующих методов в конкретной области профессиональной деятельности

2. Цель и задачи самостоятельной работы

Ведущая цель организации и осуществления СРС совпадает с целью обучения студента – формирование набора общенаучных, профессиональных и универсальных компетенций будущего бакалавра.

При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности. Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельной работы и лабораторных занятий.

3. Технологическая карта самостоятельной работы студента

Коды реализованных компетенций	Вид деятельности студентов	Итоговый продукт самостоятельной работы	Средства и технологии оценки	Объем часов, в том числе (астр.)		
				СРС	Контактная работа с преподавателем	Всего
4 семестр						
ПК-2	Подготовка к лекциям	Конспект лекций	Собеседование	21,0	15,0	36,0
ПК-2	Подготовка к практическим занятиям	Отчет в электронном виде	Собеседование	21,0	15,0	36,0
ПК-2	Самостоятельное изучение литературы	Конспект	Собеседование	21,0	15,0	36,0
ПК-2	Самотестирование, подготовка к тестированию	Тестирование	Тестовые задания	21,0	15,0	36,0
Итого за 4 семестр				84,0	60,0	144,0
Итого				84,0	60,0	144,0

4. Контрольные точки и виды отчетности по ним

Контроль качества и сроков самостоятельной работы выполняется в соответствии с учебным графиком и оформляется в соответствии с заданием. Предусмотрена следующая рейтинговая оценка знаний

Описание шкалы оценивания

В рамках рейтинговой системы успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации.

Текущий контроль

5. Порядок выполнения самостоятельной работы студентом

5.1. Методические рекомендации по работе с учебной литературой

При работе с книгой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того на сколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют **четыре основные установки в чтении научного текста:**

информационно-поисковый (задача – найти, выделить искомую информацию)

усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения излагаемые автором, так и всю логику его рассуждений)

аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)

творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

Основные виды систематизированной записи прочитанного:

Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;

Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;

Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;

Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

5.2. Методические рекомендации по составлению конспекта лекций:

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта.

2. Выделите главное, составьте план.

3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора.

4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.

5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

5.3. Методические рекомендации по подготовке к практическим занятиям

Для того чтобы практические занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по вычитанному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на лабораторных

занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

Практические работы № 1-5,8 выполняются студентами заочной и очно-заочной формы обучения самостоятельно.

5.4. Методические рекомендации по самопроверке знаний

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется провести самопроверку усвоенных знаний, ответив на контрольные вопросы по изученной теме.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала – умение отвечать на вопросы для собеседования.

Критерии оценки:

Оценка «**зачтено**» выставляется студенту, если он показал минимальный, средний или высокий уровень сформированности компетенции ПК-2.

Оценка «**не зачтено**» выставляется студенту, если минимальный уровень сформированности компетенции ПК-2 не достигнут

Описание шкалы оценивания

Максимально возможный балл за весь текущий контроль устанавливается равным 55. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

Уровень выполнения контрольного задания	Рейтинговый балл (в % от максимального балла за контрольное задание)
Отличный	100
Хороший	80
Удовлетворительный	60
Неудовлетворительный	0

5.6. Методические рекомендации по подготовке к зачетам

Контроль самостоятельной работы студентов

Контроль самостоятельной работы проводится преподавателем в аудитории.

Предусмотрены следующие виды контроля: собеседование, оценка выполнения доклада и его презентации.

Подробные критерии оценивания компетенций приведены в Фонде оценочных средств для проведения текущей и промежуточной аттестации.

6. Список литературы для выполнения СРС

Перечень основной литературы

1. Бабенко Л. К. и др. Полностью гомоморфное шифрование (обзор) //Вопросы защиты информации. – 2015. – №. 3. – С. 3-26.
2. Бабенко Л. К. и др. Защищенные вычисления и гомоморфное шифрование //Программные системы: теория и приложения.–2014.–25 с. – 2014.

Перечень дополнительной литературы

1. Бабенко Л. К. и др. Методы полностью гомоморфного шифрования на основе матричных полиномов //Вопросы кибербезопасности. – 2015. – №. 1 (9). – С. 14-25.

Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине

1. Методические указания по организации самостоятельной работы студентов по дисциплине «Модели машинного обучения, сохраняющие конфиденциальность». Ставрополь: СКФУ, 2025.
2. Методические указания к практическим занятиям по дисциплине «Модели машинного обучения, сохраняющие конфиденциальность». Ставрополь: СКФУ, 2025.

**Перечень ресурсов информационно-телекоммуникационной сети «Интернет»,
необходимых для освоения дисциплины**

1. <http://biblioclub.ru/> – Университетская библиотека online "Библиоклуб"
2. <https://4brain.ru/liderstvo/> – Лидерство: уроки эффективного руководителя
3. <https://spravochnik.ru/psihologiya/> – Справочник по психологии