МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное

учреждение высшего образования

**"Южно-Уральский государственный университет"**

**(национальный исследовательский университет)**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

**Разработка системы упреждения перепадов напряжения в электросети с применением нейросетевых технологий**

КУРСОВАЯ РАБОТА

по дисциплине «Программная инженерия»

ЮУрГУ – 02.03.02.2017.1150651.КР

|  |  |
| --- | --- |
| Нормоконтролер:  кандидат физ.-мат наук,  Доцент кафедры СП  \_\_\_\_\_\_\_\_\_\_\_\_\_\_Иванов С.А.  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2017 г. | Научный руководитель:  кандидат физ.-мат наук,  доцент кафедры СП  \_\_\_\_\_\_\_\_\_\_ Иванов С.А.  Автор работы:  студент группы КЭ - 301  \_\_\_\_\_\_\_\_\_\_ Харрасов И.Т.  Работа защищена  с оценкой: \_\_\_\_\_\_\_\_\_\_\_  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2017 г. |

Челябинск-2017

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет»**

**(национальный исследовательский университет)**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

УТВЕРЖДАЮ

Зав. кафедрой СП

\_\_\_\_\_\_\_\_\_\_ Л.Б.Соколинский

10.02.2017

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине «Программная инженерия»

студенту группы КЭ-301

Харрасову Ильдару Тагировичу

обучающемуся по направлению   
02.03.02 «Фундаментальная информатика и информационные технологии»

1. **Тема работы**Разработка системы упреждения перепадов напряжения в электросети с применением нейросетевых технологий
2. **Срок сдачи студентом законченной работы:** 31.05.2017 г.
3. **Исходные данные к работе**
4. Хайкин Саймон. Нейронные сети. Полный курс- 2-е издание.: Пер. с англ. Издательский дом «Вильямс», 2006.
5. MachineLearning.ru Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных. [Электронный ресурс] URL:

http://www.machinelearning.ru/wiki/index.php?title=Заглавная\_страница

1. Боровиков В.П. Нейронные сети. STATISTICA Neural Networks: Методология и технологии современного анализа данных – М.: Горячая линия —Телеком, 2008.
2. Рутковская Д. Нейронные сети, генетические алгоритмы и нечёткие системы - Д. Рутковская, М. Пилиньский, Л. Рутковский. — М.: Горячая линия — Телеком, 2006. — 452 с.
3. **Перечень подлежащих разработке вопросов**
   1. Обзор аналогов и научной литературы по теме, поиск готовых решений для сравнения и реализации задачи, составление списка литературы.
   2. Определение исходных данных для подачи на вход нейронной сети. Методика сегментирования, разбиения на части и так далее.
   3. Построение обучающей выборки, обучение на данных из контрольного набора, оценка на «боевой» выборке, подготовка набора тестов для анализа: результатов, эффективности работы и так далее.
4. **Дата выдачи задания:** «14» февраля 2017 г.

Научный руководитель

к.ф - м.н., доцент кафедры СП, С.А. Иванов

Задание принял к исполнению И.Т. Харрасов

Оглавление

[ВВЕДЕНИЕ 5](#_Toc483957945)

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 7](#_Toc483957946)

[1.1 Анализ аналогичных проектов 7](#_Toc483957947)

[1.1.1 StatСоветник 7](#_Toc483957948)

[1.1.2 NeuroPro 8](#_Toc483957949)

[1.2 Анализ решений 10](#_Toc483957950)

[1.3 Определение требований 11](#_Toc483957951)

[2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ 12](#_Toc483957952)

[2.1 Диаграммы вариантов использования 12](#_Toc483957953)

[2.2 Диаграммы деятельности 17](#_Toc483957954)

[3 Реализация и тестирование 18](#_Toc483957955)

[3.1 Обзор используемых программных средств проектирования системы 18](#_Toc483957956)

[3.2 Реализация программной системы 19](#_Toc483957957)

[3.3 Реализация интерфейса системы 21](#_Toc483957958)

[3.3 Экспериментальная часть 22](#_Toc483957959)

[Заключение 26](#_Toc483957960)

[Литература 27](#_Toc483957961)

[ПРИЛОЖЕНИЕ 1 «Интерфейс системы» 28](#_Toc483957962)

[ПРИЛОЖЕНИЕ 2 «Результаты исследования» 30](#_Toc483957963)

# **ВВЕДЕНИЕ**

ОСНОВНЫЕ ПОНЯТИЯ

**Временной ряд** — последовательность регистрируемого сигнала (наблюдаемая). Такой подход используется, когда нет возможности построить уравнения движения.

**Искусственная нейронная сеть** (artificial neural network, ANN*)*, или просто **нейронная сеть** — это математическая модель, а также ее программные или аппаратные реализации, построенная в некотором смысле по образу и подобию сетей нервных клеток живого организма.

**Рекуррентные нейронные сети** (англ. Recurrent neural network; RNN) — вид нейронных сетей, в которых имеется обратная связь. При этом под обратной связью подразумевается связь от логически более удалённого элемента к менее удалённому.

**Обучающая выборка** (training sample) — выборка, по которой производится настройка (оптимизация параметров) модели зависимости.

**Тестовая (или контрольная) выборка** (test sample) — выборка, по которой оценивается качество построенной модели. Если обучающая и тестовая выборки независимы, то оценка, сделанная по тестовой выборке, является **несмещённой***.*

АКТУАЛЬНОСТЬ ТЕМЫ

Нейронные сети активно используется в тех областях, где невозможно использовать формальные методы или формальные методы выдают зачастую неадекватные решения: задачи классификации, распознавания образов, прогнозирование временных рядов и т.д.

Временные ряды активно используются в различных областях науки (прогнозирование погоды, выявление закономерностей в различных системах), медицины (электрокардиограммы и электроэнцефалограммы) и экономики (фондовые рынки). Применения нейронных сетей в задачах анализа и прогнозирования временных рядов позволяют:

1. Автоматизировать вышеупомянутые процессы.
2. Позволяют устранять шумовые выбросы и помехи, связанные с человеческим факторами и различными сбоями в технике.
3. Повышают точность в задачах прогнозирования.

ЦЕЛИ И ЗАДАЧИ ИССЛЕДОВАНИЯ

В данной работе рассматривается разработка системы, позволяющей прогнозировать временной ряд любой размерности, и обеспечивать наибольшую точность вычислений.

Основные задачи:

1. Изучить методы работы с временными рядами.

2. Определить модели нейронных сетей, используемые для данного временного ряда.

3. Составить обучающую выборку для временного ряда.

4. Обучить модели и протестировать их эффективность.

СОДЕРЖАНИЕ РАБОТЫ

В первой главе «Анализ предметной области» описан анализ предметной области, заключающийся в обзоре аналогов и сервисов, работающих с временными рядами на основе нейросетевых технологий, а также решений, полученных по данной задаче.

Во второй главе «Проектирование программной системы» приводится список требований к проектируемой системе, диаграмма вариантов использования и их описание, а также описана архитектура системы.

В третьей главе «Реализация и тестирование» приводится описание реализации приложения на основе поставленного списка требований и

протоколы тестирования.

В заключении приводятся основные результаты работы и рассматриваются дальнейшие пути развития приложения.

# **1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

## **1.1 Анализ аналогичных проектов**

Использование нейросетевых технологий в различных областях хозяйства приводит к повышению спроса на сервисы, занимающихся вычислением различных задач, применяющих нейронные сети. Одними из них оказались сервисы StatСоветник и «NeuroPro».

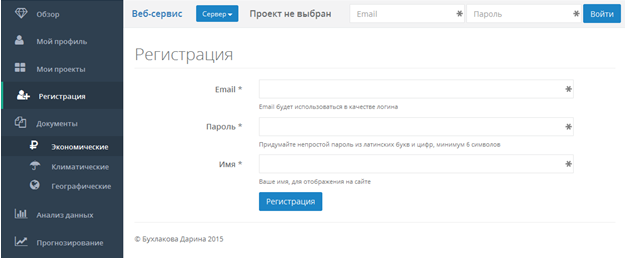
## **1.1.1 StatСоветник**

Программа предназначена для визуализации научных расчетов на примере решения задач анализа и прогнозирования временных рядов. В разработанном веб-приложении (см. рис. 1) для анализа временных рядов, использованы следующие статистические модели:

* аддитивная;
* мультипликативная;
* модель экспоненциального сглаживания;
* модель Хольта или двойного экспоненциального сглаживания;
* модель Хольта-Винтерса или тройного экспоненциального сглаживания.

Программа обладает следующими возможностями:

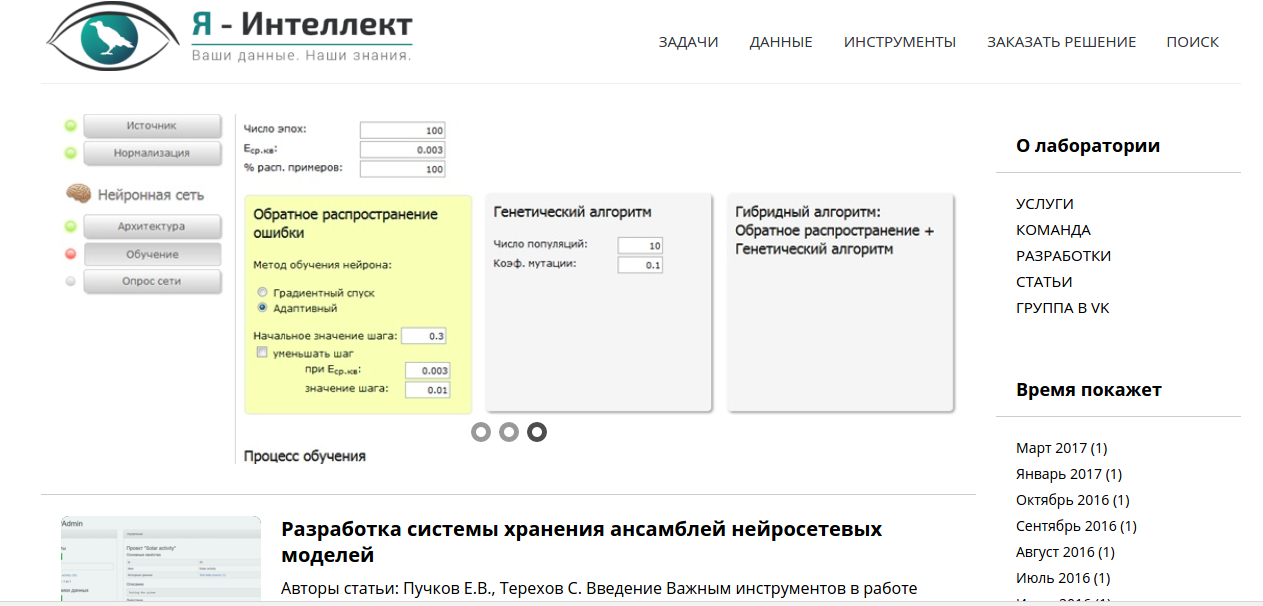
1. клиент-серверная архитектура приложения с тонким клиентом;
2. кроссплатформенность клиентской и серверной части;
3. реализация удаленного и авторизованного доступа к веб-сервису;
4. использование современных средств визуализации данных;
5. наличие интерфейса, позволяющего использовать методы анализа данных для рядового пользователя.



**Рис. 1.** Страница веб-сервиса StatСоветник

**Рис.**2**:** Интерфейс сервиса StatСоветник

Данное приложение является частью интернет-сервиса «Я-Интеллект», данный сервис имеет более широкий спектр услуг, связанный с анализом данных. На сайте показан данный сервис.



**Рис. 2.** Страница сайта "Я-Интеллект"

## **1.1.2 NeuroPro**

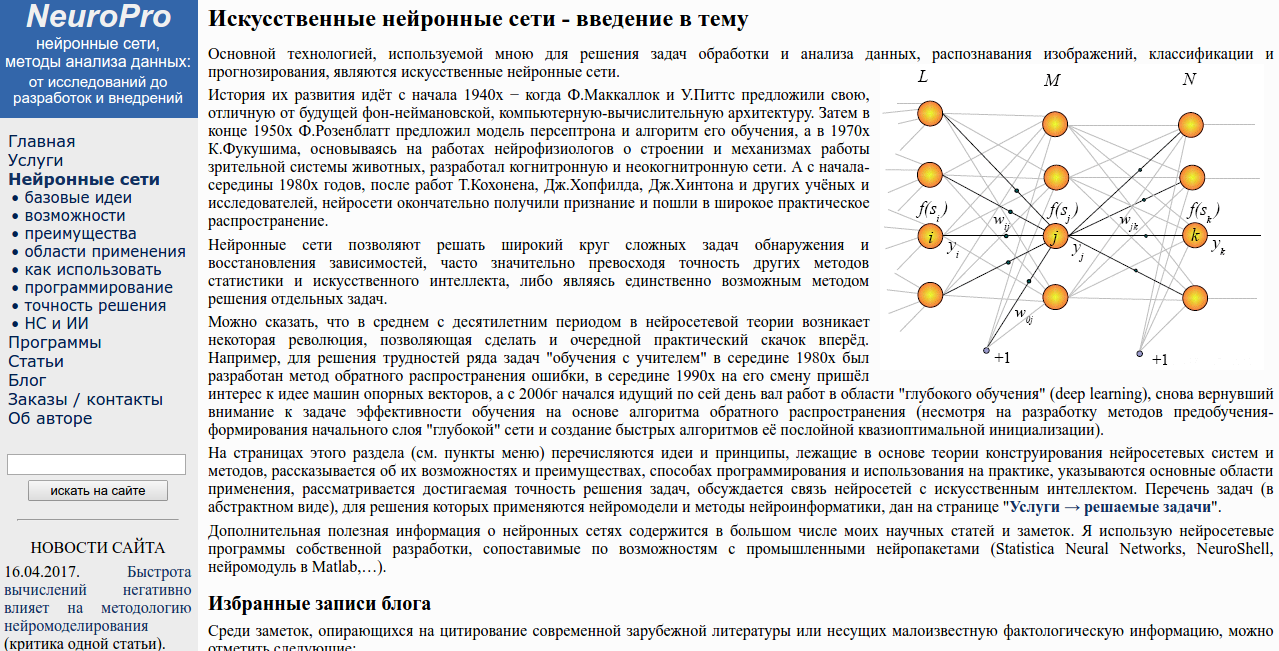
Сайт В. Царегородцева (см. рис. 3), в котором владелец сайта предлагает своим клиентам услуги, связанные с анализом данных, созданию интеллектуальных компьютерных программ для решения задач прогнозирования и диагностики, а также описания результатов научных исследований по теории и практике изучения и применения нейросетей.

Круг задач обработки информации очень широк и многообразен, например:

* обнаружение взаимосвязей и зависимостей между разнообразными измерениями и показателями;
* классификация − отнесение объектов и ситуаций к тем или иным (заранее заданным или автоматически определяемым) категориям;
* прогнозирование временных рядов, обнаружение нетипичных событий во временных рядах;
* распознавание образов (изображений) и видео, создание систем технического зрения;
* сбор и обработка сигналов в реальном времени, очистка (фильтрация) показателей от шума;
* выбор и оптимизация (идентификация) параметров объекта по экспериментальным данным, диагностика состояния (контроль) объекта, управление, автоматическая адаптация к изменившимся условиям;
* моделирование развития процесса;
* обнаружение новых или аномальных ситуаций.

Автор для решения вышеупомянутых задач использует следующие математические модели:

* *искусственные нейронные сети* (artificial neural networks) − широкий набор архитектур и алгоритмов, включая современные технологии глубокого обучения (deep learning);
* генетические и эволюционные алгоритмы оптимизации;
* *математическую статистику*: регрессионный, [кластерный анализ](http://www.neuropro.ru/sol1_9.shtml), метод главных компонент, проверка статистических гипотез,…
* другие методы идентификации зависимостей и "*машинного обучения*" (machine learning);
* методы искусственного интеллекта и технологии data mining;
* разнообразные (в т.ч. неклассические и эмпирические) методы моделирования, визуализации данных.



**Рис. 3.** Сайт NeuroPro

## **1.2 Анализ решений**

Для решения задач прогнозирования используется нейронные сети различных топологий. Например, для решения задач прогнозирования временных рядов часто используется нейронные сети прямого распространения и обратного распространения ошибок. Эти сети хорошо подходят для задач с малой зашумленностью. Однако на практике временные ряды бывают сильно зашумлены, и эффективность таких НС падает. Для решения данного типа задач используются реккурентные нейронные сети(RNN) с задержкой времени. Одним из самых популярных RNN для решения данной задачи становится долгая короткосрочная память (LSTM, long short-time memory).

В данной главе выполнен анализ аналогичных проектов и существующих решений.

Выбраны следующие методы реализации решения задачи: выделение обучающей выборки с помощью реккурентных нейронных сетей. На вход подается файл определенного формата, выход нейронной сети составляют два числа: ошибка при обучающейся выборке и ошибка при тестовой выборке, а также нейронная сеть выдает временной ряд, составляющей и трех графиков: исходные данные, график поученной обучающей выборки и график полученной тестовой выборки.

## **1.3 Определение требований**

В ходе проектирования приложения были определены функциональные и нефункциональные требования к разрабатываемому приложению:

Функциональные требования:

1. Система должна принимать на вход, заданный пользователем файл определенного формата.
2. Система должна выделять из таблицы обучающую выборку, определенная пользователем.
3. Система должна вывести графики временных рядов, исходную, и полученную в результате обучения нейронной сети.
4. Система может поддерживать самостоятельный выбор пользователем набор параметров нейронной сети для прогнозирования временного ряда.

Нефункциональные требования:

1. Система должна быть реализована на языке Python.

# **2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ**

## **2.1 Диаграммы вариантов использования**

Для проектирования приложения был использован язык графического описания для объектного моделирования UML. Была построена модель взаимодействия актера «Пользователь» с приложением в виде диаграммы вариантов использования (см. рис. 4).

АКТЕРЫ, ВЗАИМОДЕЙСТВУЮЩИЕ С СИСТЕМОЙ



**Рис. 4.** Диаграмма вариантов использования

Пользователь — лицо, которое использует действующую систему для выполнения конкретной функции.

КРАТКОЕ ОПИСАНИЕ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

Пользователь настраивает конфигурацию нейронной сети, т. е. выбирает топологию нейронной сети, количество эпох, функцию потерь и функцию активации, а также выделяет обучающую выборку из данных. Пользователь также выбирает файлы, хранящие данные.

Нейронные сети запускают процесс обучения, исходя из конфигураций, поданных пользователем, при окончании эпох, нейронная сеть выводит на консоль средние квадратические функции потерь данных при обучении и тестировании, а также в отдельном окне отображаются графики временных рядов, исходные данные и данные, полученные при обучении/тестировании. В графике будет показано расхождение между исходными и полученными данными.

СПЕЦИФИКАЦИЯ ОСНОВНЫХ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

В табл. 1 приведено описание прецедента «Ввести данные».

**Табл. 1.** Спецификация вариантов использования

|  |
| --- |
| ***UseCase:* Ввести данные** |
| *ID:* 1 |
| *Аннотация:* Пользователь вводит имя файла и путь к ней |
| *Главные актеры:* Пользователь |
| *Второстепенные актеры*: нет |
| *Предусловия:* нет |
| *Основной поток:*  1. Прецедент начинается, когда Пользователь вводит данные  2. Пользователь вводит путь до файла  3. Система запоминает выбор Пользователя |
| *Постусловия:*  Система запоминает название файла и путь к нему |
| *Альтернативные потоки: -* |

В табл. 2 приведено описание прецедента «Создать обучающую выборку».

**Табл. 2.** Спецификация вариантов использования

|  |
| --- |
| ***UseCase:* Создать обучающую выборку** |
| *ID:* 2 |
| *Аннотация:* Пользователь выделяет обучающую выборку из исходных данных |
| *Главные актеры:* Пользователь |
| *Второстепенные актеры*: нет |
| *Предусловия:* нет |
| *Основной поток*   1. Прецедент начинается, когда Пользователь вводит десятичное число от 0 до 1 2. Пользователь вводит количество эпох 3. Система запоминает выбор Пользователя |
| *Постусловия:*  Система использует количество эпох для прогонки данных нейронных сетей |
| *Альтернативные потоки:* нет |

В табл. 3 приведено описание прецедента «Обучить нейронную сеть».

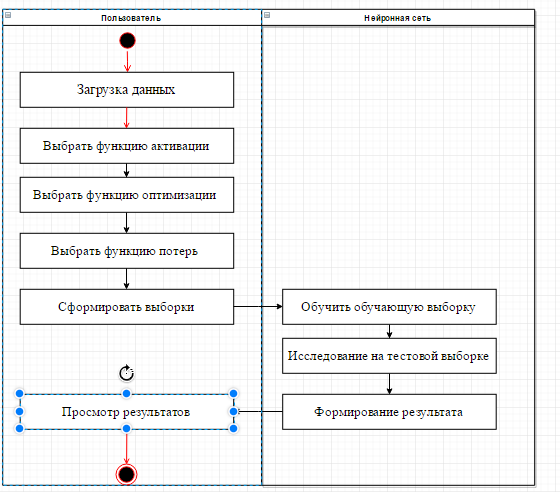
**Табл. 3.** Спецификация вариантов использования

|  |
| --- |
| ***UseCase:* Обучить нейронную сеть** |
| *ID:* 3 |
| *Аннотация:* Пользователь настраивает нейронную сеть |
| *Главные актеры:* Пользователь |
| *Второстепенные актеры*: нет |
| *Предусловия:* нет |
| *Основной поток:*   1. Прецедент начинается, когда Пользователь выбирает топологию нейронной сети и функцию оптимизации 2. Пользователь выбирает функцию ошибок и функцию активации 3. Система запоминает выбор Пользователя |
| *Постусловия:*  Система обучает нейронную сеть по данным параметрам |
| *Альтернативные потоки:* нет |

В табл. 4 приведено описание прецедента «Протестировать на реальных данных».

|  |
| --- |
| ***UseCase:*  Протестировать на реальных данных** |
| *ID:* 4 |
| *Аннотация:* Система тестирует на данных, оставшихся от исходных после разд |
| *Главные актеры:* Пользователь |
| *Второстепенные актеры*: нет |
| *Предусловия:*   1. Ввести данные 2. Создать обучающую выборку 3. Обучить нейронную сеть |
| *Основной поток:*   1. Прецедент начинается, когда все предыдущие прецеденты были выполнены |
| *Постусловия:*  Система выводит результаты обучения на консоль |
| *Альтернативные потоки:* нет |

## **2.2 Диаграммы деятельности**



**Рис. 5.** Диаграмма деятельности системы

На основе требований к системе была разработана диаграмма деятельности, представленная на рис. 5. В данной диаграмме было показано разделение деятельности между пользователем и нейронной сетью. Пользователю отведена настройка параметров сети и сформирование выборки, обучение по ней и тестирование по реальным данным - нейронной сети. Запуск и остановка работы нейронной сети лежит на пользователе.

# 

# **3 Реализация и тестирование**

## 3.1 Обзор используемых программных средств проектирования системы

Система была реализована на языке Python, который имеет простой синтаксис и для которого существует большое количество библиотек, решающих задачи по прогнозированию временных рядов.

Реализация данной системы выполнена с помощью следующих библиотек (или пакетов библиотек):

1. *Scikit-learn* — пакет библиотек, поддерживающие работу с различными методами анализа данных (машинное обучение, нейронные сети и т. д.) из этого пакета были использованы следующие библиотеки:
   1. *Pandas* — позволяет работать с различными структурами данных и операций над ними (например, считывание данных из файла)
   2. *NumPy* — расширение языка Python, позволяющий работать с большими многомерными массивами и матрицами, а также имеет большую библиотеку высокоуровневых математических функций для выполнения различных операций над этими массивами
   3. *Matplotlib —* библиотека языка программирования Python для визуализации данных в виде двумерной(2D) или трехмерной(3D) графики
   4. *Keras —* библиотека для работы с нейронными сетями. Она содержит различные функции для работы с нейронными сетями и позволяет упрощать многие операции с ними. Достоинствами данной библиотеки являются:
      1. Удобство для пользователя — сводит количество пользовательских действий общего характера к минимуму
      2. Модульность — каждая функция, используемая в Keras, является модулем.
      3. Расширяемость — к существующей нейронной сети можно добавить новые модули.

К недостаткам Keras можно отнести плохую гибкость.

Вышеупомянутые библиотеки являются свободно распространяемым ПО.

## **3.2 Реализация программной системы**

Перед созданием нейронной сети был сформирован документ, содержащий примерно 1100 значений, который будет формировать временной ряд. Как перед обучением, так и перед предсказанием данные из документа были преобразованы в тип float32. Затем была сформирована обучающая выборка из документа, сформированная из документа.

Для создания и обучения модели и проведения экспериментов был создан скрипт на языке Python. На рис. 6 изображен листинг формирования обучающей и тестовой выборки.

**Рис. 6.** Листинг формирования обучающейся и тестовой выборки

dataframe = pandas.read\_csv('/home/darius/Курсач/graphics.csv',

usecols = [1], engine = 'python', skipfooter = 3)

dataset = dataframe.values

dataset = dataset.astype('float32')

scaler = MinMaxScaler(feature\_range = (0, 1))

dataset = scaler.fit\_transform(dataset)

train\_size = int(len(dataset) \* 0.7)

test\_size = len(dataset) - train\_size

train ,test = dataset[0:train\_size, :], dataset[train\_size:len(dataset), :]

print(len(train), len(test))

def create\_dataset(dataset, look\_back = 1):

dataX, dataY = [], []

for i in range(len(dataset) - look\_back - 1):

a = dataset[i:(i+look\_back), 0]

dataX.append(a)

dataY.append(dataset[i + look\_back, 0])

return numpy.array(dataX), numpy.array(dataY)

look\_back = 1

trainX, trainY = create\_dataset(train, look\_back)

testX, testY = create\_dataset(test, look\_back)

trainX = numpy.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))

testX = numpy.reshape(testX, (testX.shape[0], 1, testX.shape[1]))

Обучение нейронной сети показан на рис.7 , описание листинга нейронной сети описано в табл. 4

**Рис. 7.** Листинг модели нейронной сети

model = Sequential()

model.add(LSTM(10, input\_shape = (1, look\_back)))

model.add(Dropout(0.5))

model.add(Dense(1))

model.add(Activation ("tanh"))

model.compile(loss = 'binary\_crossentropy', optimizer = 'rmsprop', metrics=['accuracy'])

model.fit(trainX, trainY, epochs = 100, batch\_size = 1, verbose = 2)

**Табл. 4.** Описание структуры используемой сети

|  |  |
| --- | --- |
| **Код** | **Описание** |
| model = Sequential() | Последовательное описание модели |
| model.add(LSTM(5, input\_shape = (1, look\_back))) | Добавление НС топологии LSTM с 5 нейронами и 2 входами |
| model.add(Dropout(0.5)) | Модуль с функцией выброса случайных значений |
| model.add(Dense(1)) | Полносвязный слой с одним выходом |
| model.add(Activation ("relu")) | Добваление модуля с функцией активации: ReLU |
| model.compile(loss = 'binary\_crossentropy', optimizer = 'adadelta', metrics=['accuracy']) | Добавление функции потерь — бинарная кросс-энтропия, функция оптимизации — adadelta, метрика, по которым считывается качество — accuracy (точность) |
| model.fit(trainX, trainY, epochs = 100, batch\_size = 1, verbose = 2) | Обучение нейронной сети |

## 3.3 Реализация интерфейса системы

На рис. 8 в ПРИЛОЖЕНИЕ 1 «Интерфейс системы» изображен интерфейс системы в момент обучения. Система представляет собой консольное приложение. В нем показывается следующие параметры — текущая эпоха, вес и точность — разницу между исходными и предсказанными данными. По окончании всех эпох, появляется окно с графиками временных рядов (*рис. 9*): временной ряд синего цвета — исходные данные, черного — данные, полученные во время обучения, розового — данные, полученные на тестовой выборке.

## 3.3 Экспериментальная часть

Для выявления лучшего наилучшего результата (графика прогнозирования временных рядов) используется:

1. Все функции потерь (mean\_squared\_error, mean\_absolute\_error, mean\_absolute\_percentage\_error, squared\_hinge, hinge, binary\_crossentropy, logcosh).
2. Все функции активации (softmax, elu, softplus, softsign, relu, tanh, sigmoid, hard\_sigmoid, linear).
3. Все функции оптимизации (sgd, rmsprop, adamax, adaline, adadelta, adam, nadam).

В качестве основной топологии была использована LSTM-сеть. Количество элементов в обучающей выборке было взято для 60 и 70 процентов, количество эпох = 100.

Для эксперимента были взяты две функции потерь — среднюю квадратическую потерь (MSE), и бинарная кросс-энтропия (binary-crossentropy).

**Табл. 4.** MSE, обучающая выборка - 60%

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | adagrad | sgd | adamax | nadam | adam | adadelta | rmsprop |
| 5 | 89.12 | 86.51 | 86.51 | 83.75 | 84.49 | 92.01 | 89.85 |
| 91.47 | 90.10 | 89.97 | 88.70 | 89.47 | 96.09 | 94.26 |
| 10 | 89.28 | 87.15 | 87.63 | 81.29 | 84.96 | 90.93 | 89.72 |
| 91.80 | 90.90 | 91.59 | 85.52 | 90.13 | 95.28 | 94.07 |
| 25 | 89.27 | 91.77 | 89.01 | 82.89 | 87.52 | 92.12 | 93.51 |
| 91.65 | 92.95 | 91.41 | 84.88 | 91.43 | 95.98 | 97.05 |
| 50 | 89.04 | 88.99 | 87.40 | 76.37 | 84.85 | 91.35 | 91.47 |
| 91.59 | 91.36 | 90.82 | 80.43 | 89.70 | 95.58 | 95.65 |
| 100 | 89.11 | 89.11 | 87.45 | 75.79 | 83.88 | 90.55 | 89.69 |
| 91.73 | 91.96 | 90.72 | 80.13 | 88.51 | 94.76 | 94.14 |
| 500 | 89.21 | 93.00 | 93.63 | 77.43 | 85.25 | 91.65 | 146.57 |
| 92.43 | 97.31 | 98.68 | 82.24 | 90.73 | 96.41 | 147.12 |

Одним из наилучших полученных результатов данной таблицы (см. табл. 5) является нейронная сеть, имеющая следующую конфигурацию:

- топология — LSTM;

- функция потерь — MSE;

- функция активации — linear;

- функция активации — rmsprop;

- количество нейронов — 500;

- обучающая выборка — 60% от всех данных.

График функции данного временного ряда представлен на рис. 10.

***Табл. 6.*** *MSE, обучающая выборка - 70%*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | adagrad | sgd | adamax | nadam | adam | adadelta | rmsprop |
| 5 | 89.41 | 88.26 | 86.26 | 85.39 | 85.15 | 91.14 | 89.70 |
| 91.27 | 90.83 | 90.34 | 90.71 | 90.51 | 93.77 | 93.16 |
| 10 | 89.59 | 87.23 | 86.87 | 84.23 | 85.03 | 92.21 | 90.77 |
| 91.45 | 91.21 | 90.65 | 89.93 | 90.72 | 94.96 | 94.72 |
| 25 | 89.50 | 89.89 | 87.91 | 83.38 | 86.26 | 92.09 | 92.67 |
| 91.41 | 91.30 | 90.72 | 86.68 | 90.72 | 94.98 | 96.43 |
| 50 | 89.28 | 91.19 | 88.55 | 86.33 | 88.24 | 92.03 | 91.51 |
| 91.30 | 91.94 | 91.16 | 87.55 | 91.89 | 95.07 | 94.87 |
| 100 | 89.43 | 89.55 | 87.47 | 79.94 | 84.44 | 91.42 | 91.90 |
| 91.55 | 91.24 | 90.59 | 85.09 | 90.33 | 94.86 | 96.14 |
| 500 | 89.09 | 89.68 | 86.99 | 81.56 | 82.47 | 90.45 | 88.64 |
| 91.34 | 91.44 | 90.61 | 85.00 | 85.85 | 94.03 | 92.82 |

Одним из наилучших полученных результатов данной таблицы (см. табл. 6) является нейронная сеть, имеющая следующую конфигурацию:

- топология — LSTM;

- функция потерь — MSE;

- функция активации — linear;

- функция отпимизации — rmsprop;

- количество нейронов — 500;

- обучающая выборка — 70% от всех данных.

График функции данного временного ряда представлен на рис. 11.

**Табл. 7.** Binary\_crossentropy, обучающая выборка — 60%

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | adagrad | sgd | adamax | nadam | adam | adadelta  (relu) | rmsprop  (softline) |
| 5 | - | - | - | - | - | 85.83 | 85.63 |
| - | - | - | - | - | 89.98 | 89.78 |
| 10 | - | - | - | - | - | 90.93 | 87.46 |
| - | - | - | - | - | 95.28 | 92.29 |
| 25 | - | - | - | - | - | 86.73 | 87.26 |
| - | - | - | - | - | 91.56 | 91.33 |
| 50 | - | - | - | - | - | 86.84 | 88.65 |
| - | - | - | - | - | 91.74 | 93.09 |
| 100 | - | - | - | - | - | 87.06 | 86.95 |
| - | - | - | - | - | 92.11 | 90.95 |
| 500 | - | - | - | - | - | 82.65 | 111.71 |
| - | - | - | - | - | 92.41 | 111.11 |

Одним из наилучших полученных результатов данной таблицы (см. табл. 7) является нейронная сеть, имеющая следующую конфигурацию:

- топология — LSTM;

- функция потерь — binary\_crossentropy;

- функция активации — softline;

- функция оптимизации — rmsprop;

- количество нейронов — 500;

- обучающая выборка — 60% от всех данных.

График функции данного временного ряда представлен на рис. 12.

**Табл. 8.** Binary\_crossentropy, обучающая выборка — 70%

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | adagrad | sgd | adamax | nadam | adam | adadelta  (relu) | rmsprop  (softline) |
| 5 | - | - | - | - | - | 91.02 | 85.63 |
| - | - | - | - | - | 92.88 | 89.78 |
| 10 | - | - | - | - | - | 92.40 | 87.46 |
| - | - | - | - | - | 94.78 | 92.29 |
| 25 | - | - | - | - | - | 94.04 | 87.26 |
| - | - | - | - | - | 96.85 | 91.33 |
| 50 | - | - | - | - | - | 94.11 | 88.65 |
| - | - | - | - | - | 97.23 | 93.09 |
| 100 | - | - | - | - | - | 93.74 | 86.95 |
| - | - | - | - | - | 97.16 | 90.95 |
| 500 | - | - | - | - | - | 92.33 | 111.71 |
| - | - | - | - | - | 96.06 | 111.11 |

Одним из наилучших полученных результатов данной таблицы (см. табл. 8) является нейронная сеть, имеющая следующую конфигурацию:

- топология — LSTM;

- функция потерь — binary\_crossentropy;

- функция активации — softline;

- функция оптимизации — rmsprop;

- количество нейронов — 500;

- обучающая выборка — 60% от всех данных.

График функции данного временного ряда представлен на рис. 13.

Из всех данных вышеуказанных таблиц наиболее близкие результаты к исходному показал следующий временной ряд (см. рис. 12), имеющий следующие показатели:

- топология — LSTM;

- функция потерь — binary\_crossentropy;

- функция активации — softline;

- функция оптимизации — rmsprop;

- количество нейронов — 500;

- обучающая выборка — 60% от всех данных.

## 

## **Заключение**

После изучения существующих решений были выбраны следующие методы решения задачи: выделение обучающей выборки и анализ временного ряда с помощью реккурентной нейронной сети.

Были разобраны все варианты, и многие из них не прошли начальную проверку. Например, в случае с функцией потерь «binary\_crossentropy» были исключены функции оптимизации adagrad, sgd, adamax, nadam, adam. Некоторые функции потерь, такие как mean\_absolute\_percentage\_error, hinge, logcosh и другие не не дают адекватного ответа.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ.

1. Создана обучающая выборка
2. С помощью созданной выборки обучена реккурентная нейронная сеть
3. Проведена экспериментальная проверка эффективности данной нейронной сети
4. Создано приложение с консольным интерфейсом

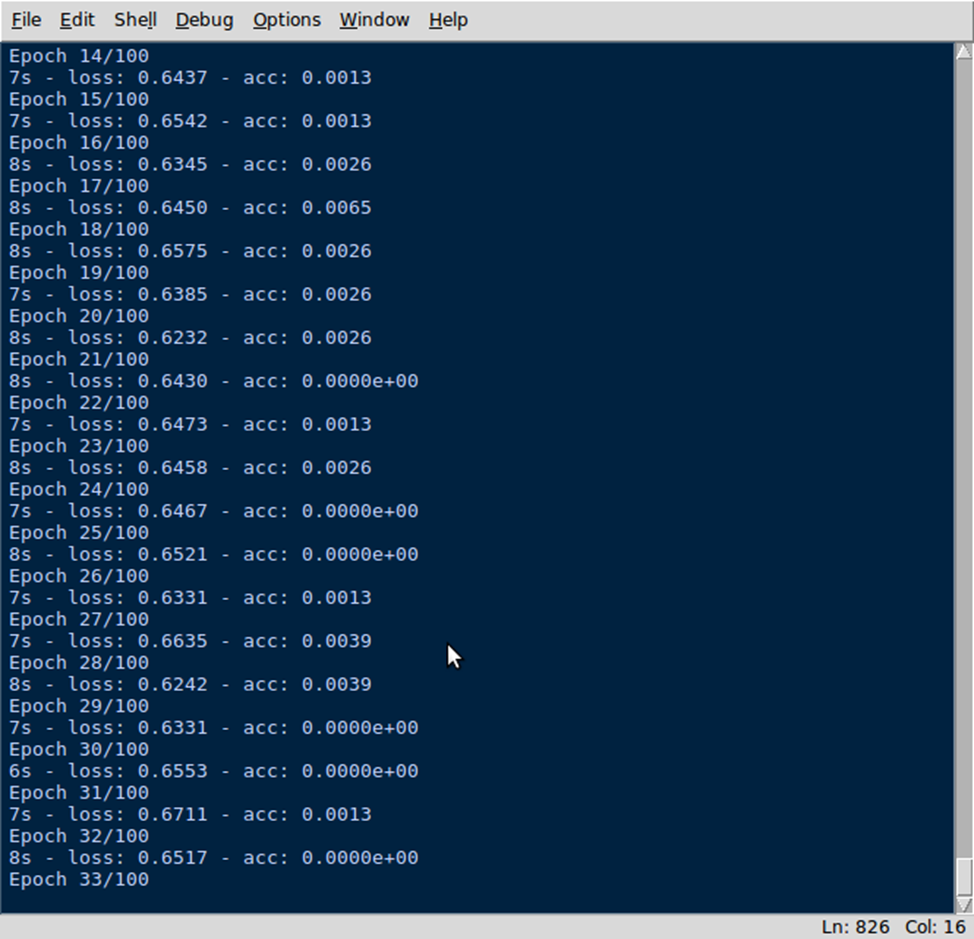
ДАЛЬНЕЙШИЕ НАПРАВЛЕНИЯ РАЗВИТИЯ

На данный момент временной ряд был рассмотрен на топологии LSTM, в дальнейшем планируется прогнозирование на других топологиях реккурентных нейронных сетей — GRU, сетях Эльмана и др. Также будет продолжено исследование на остальных функциях потерь. Планируется введение графического интерфейса.

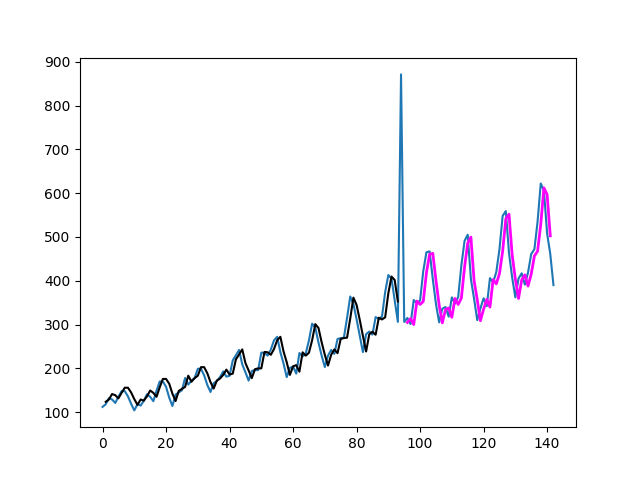
# **Литература**

1. Хайкин Саймон. Нейронные сети. Полный курс- 2-е издание.: Пер. с англ. Издательский дом «Вильямс», 2006.
2. MachineLearning.ru Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных. [Электронный ресурс] URL:
3. http://www.machinelearning.ru/wiki/index.php?title=Заглавная\_страница
4. Боровиков В.П. Нейронные сети. STATISTICA Neural Networks: Методология и технологии современного анализа данных – М.: Горячая линия —Телеком, 2008.
5. Рутковская Д. Нейронные сети, генетические алгоритмы и нечёткие системы - Д. Рутковская, М. Пилиньский, Л. Рутковский.**—** М.: Горячая линия — Телеком, 2006. — 452 с.
6. Ежов А. А. Нейрокомпьютинг и его применение в экономике и бизнесе: Учеб. пособие А. А. Ежов, С. А. Шумский. — М.: МИФИ, 1998. — 222 с.
7. Прогнозирование при помощи рекуррентных нейронных сетей. [Электронный ресурс] URL:http://datareview.info/article/prognozirovanie-pri-pomoshhi-rekurrentnyih-neyronnyih-setey/.
8. Знакомство с архитектурой LSTM-сетей. [Электронный ресурс] URL: http://datareview.info/article/znakomstvo-s-arhitekturoy-lstm-setey/
9. Лоскутов А.Ю. Анализ временных рядов. –– Курс лекций,Физический факультет МГУ.
10. {Я-Интеллект} Лаборатория анализа данных и оптимальных решений. [Электронный ресурс] URL: http://i-intellect.ru/.
11. Разработка системы хранения ансамблей нейросетевых моделей. [Электронный ресурс] URL: http://i-intellect.ru/articles/1938/.
12. NeuroPro. нейронные сети, методы анализа данных: от исследований до разработок и внедрений. [Электронный ресурс] URL: http://www.neuropro.ru/.
13. On the Suitability of Long Short-Term Memory Networks for Time Series Forecasting. [Электронный ресурс] URL: http://machinelearningmastery.com/suitability-long-short-term-memory-networks-time-series-forecasting/.
14. Keras. Documentation [Электронный ресурс] URL: https://keras.io.

# **ПРИЛОЖЕНИЕ 1 «Интерфейс системы»**

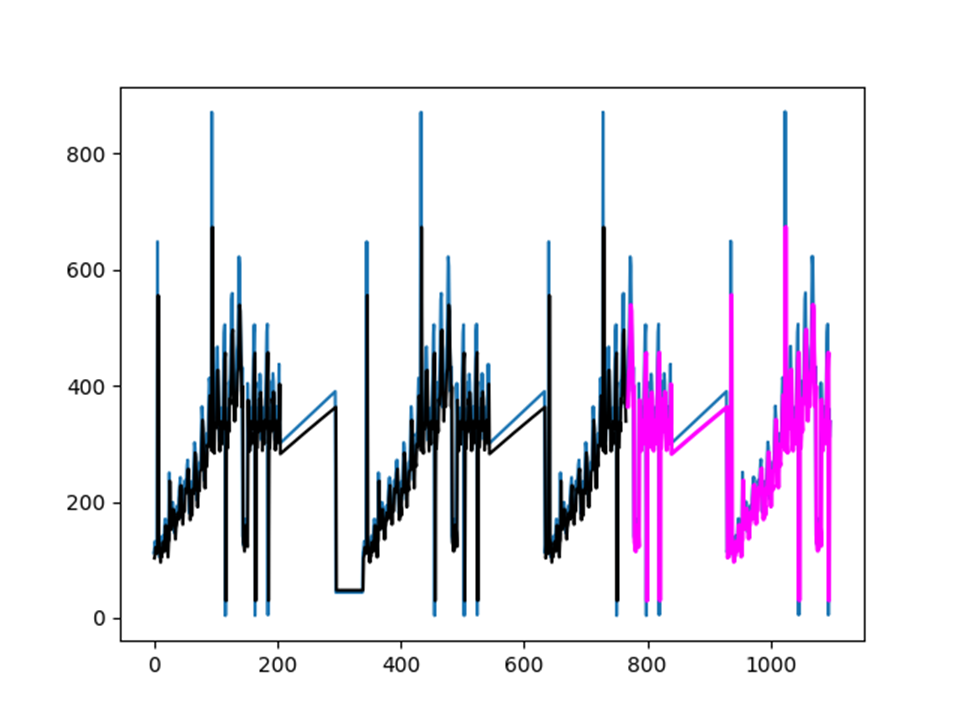


**Рис. 8.** Интерфейс системы в момент обучения

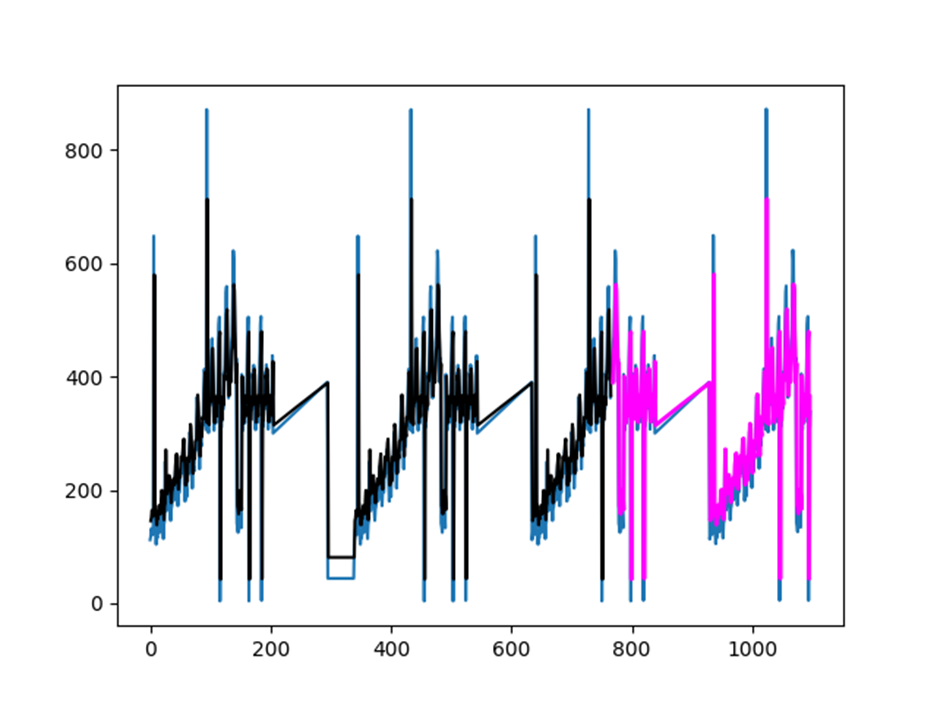


**Рис. 9** Окно с полученными временными рядами

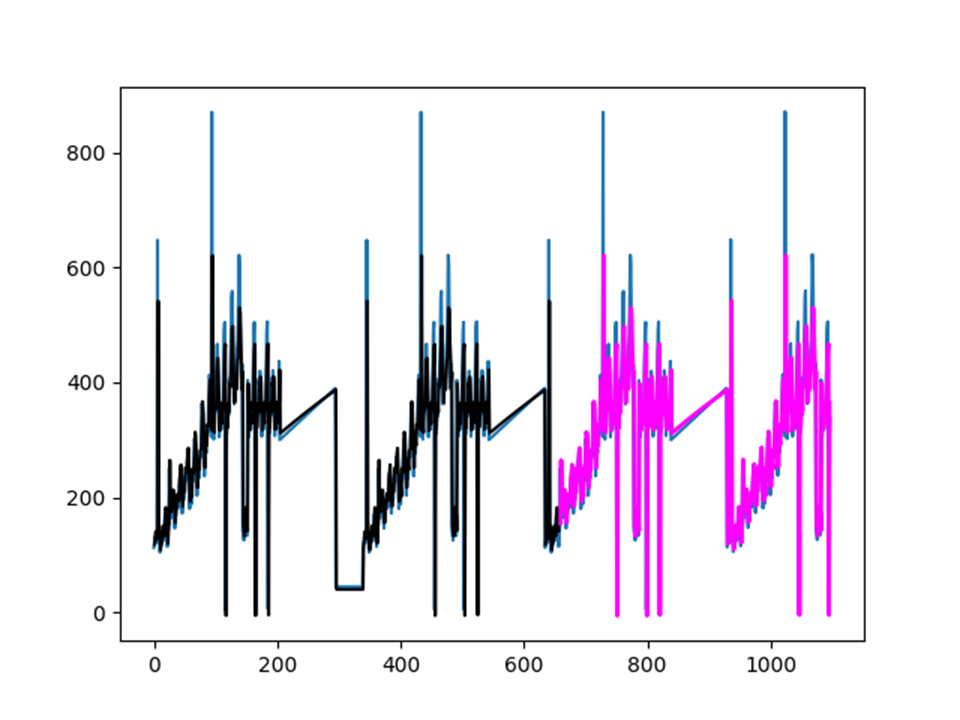
# **ПРИЛОЖЕНИЕ 2 «Результаты исследования»**



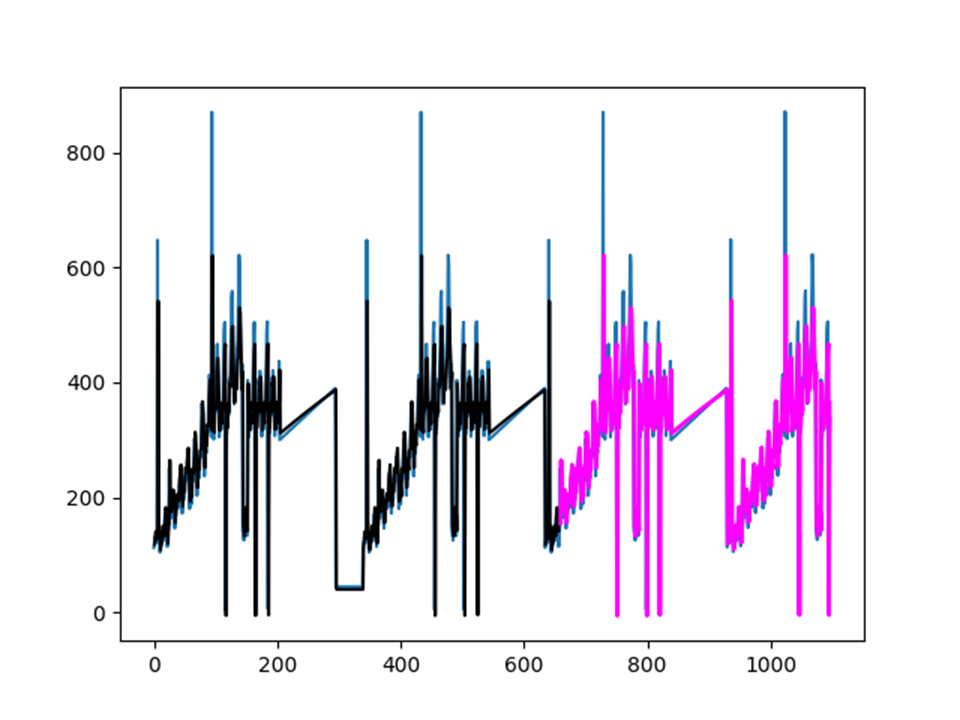
**Рис. 10.** MSE, Обучающая выборка - 60%, наилучший результат

****

**Рис. 11** MSE, обучающая выборка - 70%, наилучший результат

****

**Рис. 12** Binary\_crossentropy, обучающая выборка - 60%, наилучший результат

****

**Рис. 13** Binary\_crossentropy, обучающая выборка - 70%, наилучший результат