ДЕТЕКЦИЈА НА АНОМАЛИИ ВО SCADA СИСТЕМ

СКЛАДИШТА И ОБРАБОТКА НА ПОДАТОЦИ

БЛАГОЈ ХРИСТОВ 157/2016

ВОВЕД

- Модернизација на сите области од човековото живеење – позитивни и негативни страни
  - Smart технологијата овозможува полесен живот
  - Изложена на сајбер напади врз автоматизираните smart системи
- Потреба од детекција и спречување на овие напади

ОПИС НА ПРОБЛЕМОТ

- Виртуелно симулиран град C-Town базиран на водоводна мрежа со средна големина

- Со воведување на smart технологија доаѓа до појава на аномалии во системот
  - ниски нивоа на вода во резервоар T5
  - прелевање на резервоар T1

ГРАФИЧКА ПРЕТСТАВА НА C-TOWN

ПОДАТОЦИ

- Три дадени податочни множества со податоци отчитувани на секој час:

  - **dataset03:** податоци со времетраење од 365 денови пред инсталација на smart технологијата (без напади) – *множество за тренинг*
  - **dataset04:** податоци со времетраење од 174 денови по инсталацијата на smart технологијата, во сооднос 88% нормални спрема 12% напади (не сите напади се точно обележани) – *множество за валидација*
  - **test_dataset:** податоци со времетраење од 87 денови во сооднос 80% нормални спрема 20% напади – *множество за тест*

FEATURES

- SCADA систем за снабдување на податоци и управување
  - аквизицијата на податоците се врши преку девет PLC
    - сензор за ниво на вода во резервоарите
    - сензор за статус на пумпата (ON/OFF)
    - сензор за проток низ пумпата
    - сензор за влезен и излезен притисок на станицата за пумпање
- Вкупно 43 feature-и

TSFRESH

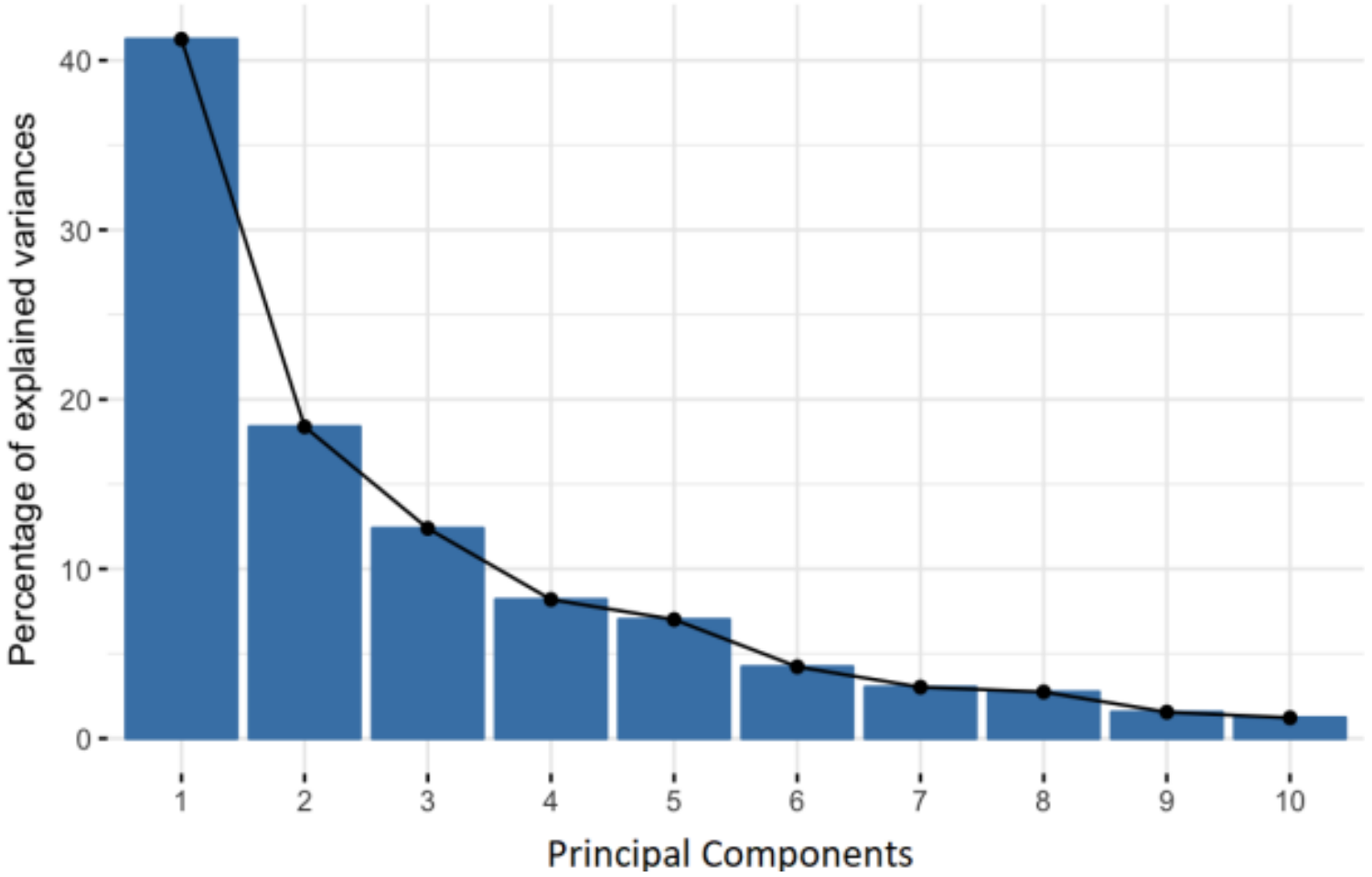
- Со користење на Python библиотеката tsfresh се извлекуваат нови feature-и
- Од 20,000+ нови feature-и се избираат само најзначајните 150
- Како да знаеме кои се најзначајни?

PRINCIPAL COMPONENT ANALYSIS

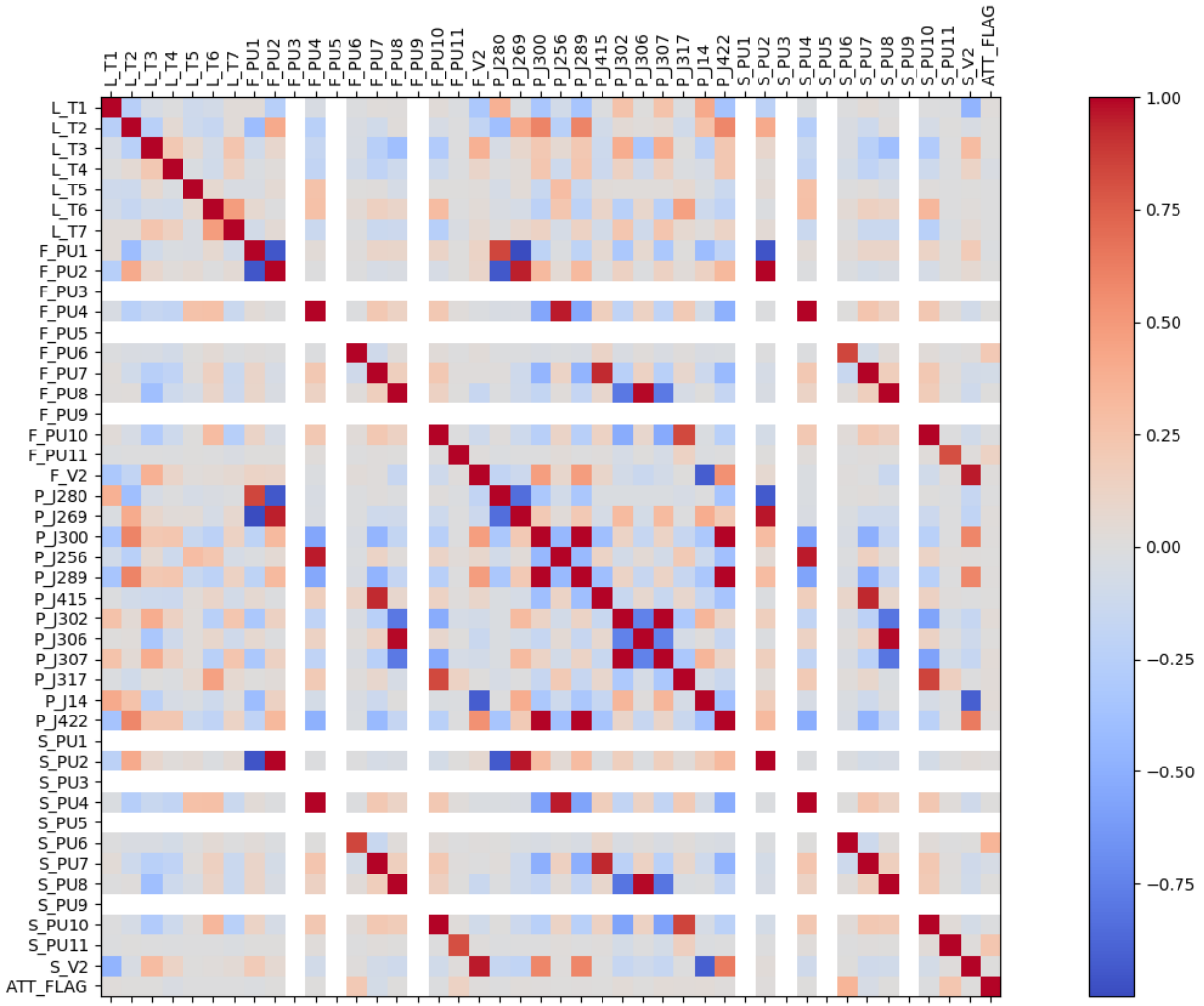
- Постапка за редукција на димензионалноста на податочното множество

- Се состои од три главни чекори:
  1) Стандардизација на податоците со цел сите подеднакво да влијаат врз анализата
  2) Пресметување на матрица на коваријанса **Σ**
  3) Пресметување на сопствени вектори и сопствени вредности на **Σ** за да се одредат *principal components*
  4) Избирање на првите n компоненти – n се одредува со grid search

PRINCIPAL COMPONENT ANALYSIS



КОРЕЛАЦИЈА НА ПОДАТОЦИТЕ

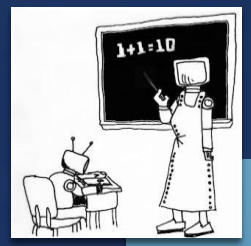


АЛГОРИТМИ

- Мал процент на позитивни класи во големо податочно множество => supervised или unsupervised ?

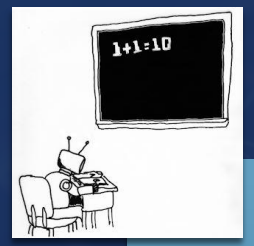
- Зошто да не и двете!

АЛГОРИТМИ



Supervised

- K-Nearest Neighbor
- Support Vector Classifier



Unsupervised

- K-Means Clustering
- One-Class SVM
- Multivariate Gaussian Distribution
- Isolation Forest
- Autoencoder Neural Network

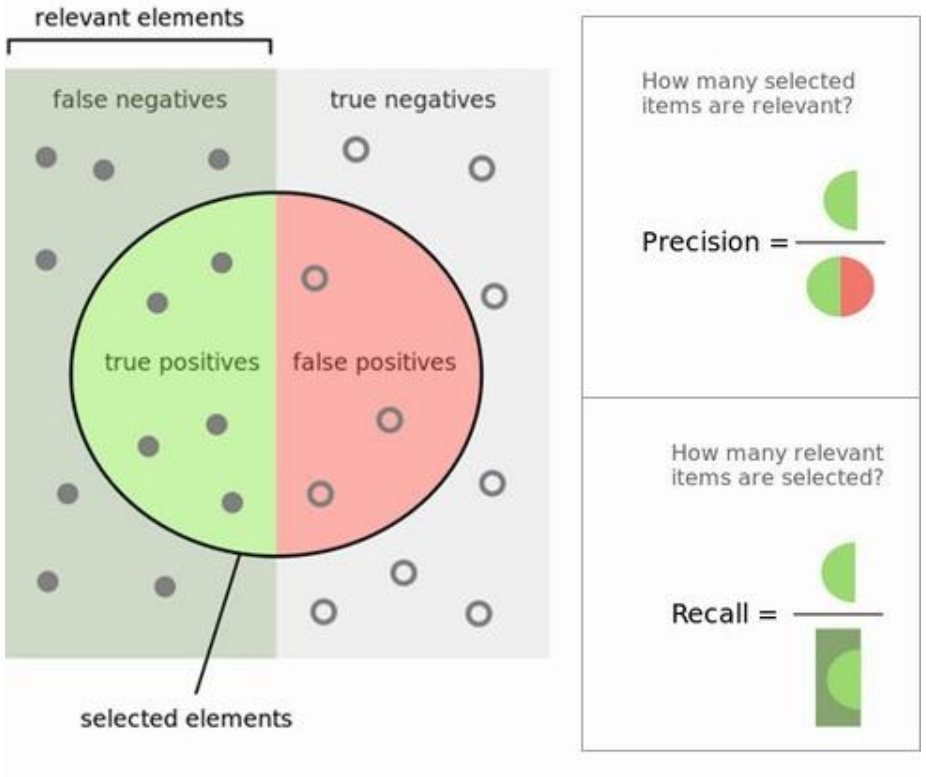
STRATIFIED K-FOLD CROSS-VALIDATION

- Се користи при одредување на оптимални параметри со grid search
- Поради нерамнотежа во податочното множество се користи *stratified* верзија од стандардниот K-Fold
- Определува K folds така што секој содржи подеднаков процент од позитивните класи
- За конкретниот проблем се користи 10-Fold cross validation

МЕТРИКА ЗА ОЦЕНУВАЊЕ

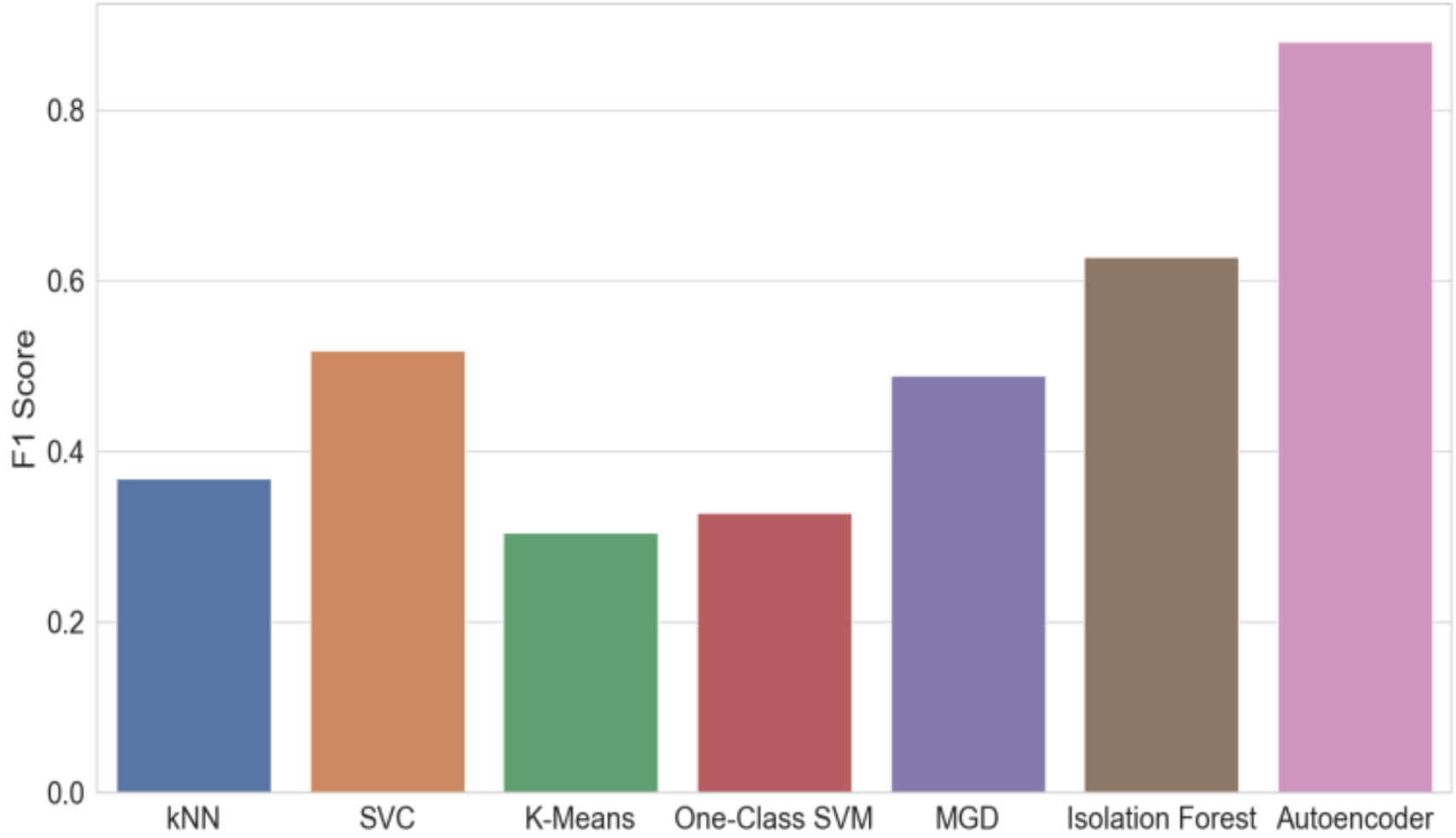
- Се користи F1 score метриката за добивање пореален приказ на точноста на алгоритмот

- Поголема точност на резултатот поради земање во предвид лажни позитивни и лажни негативни примероци.



$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

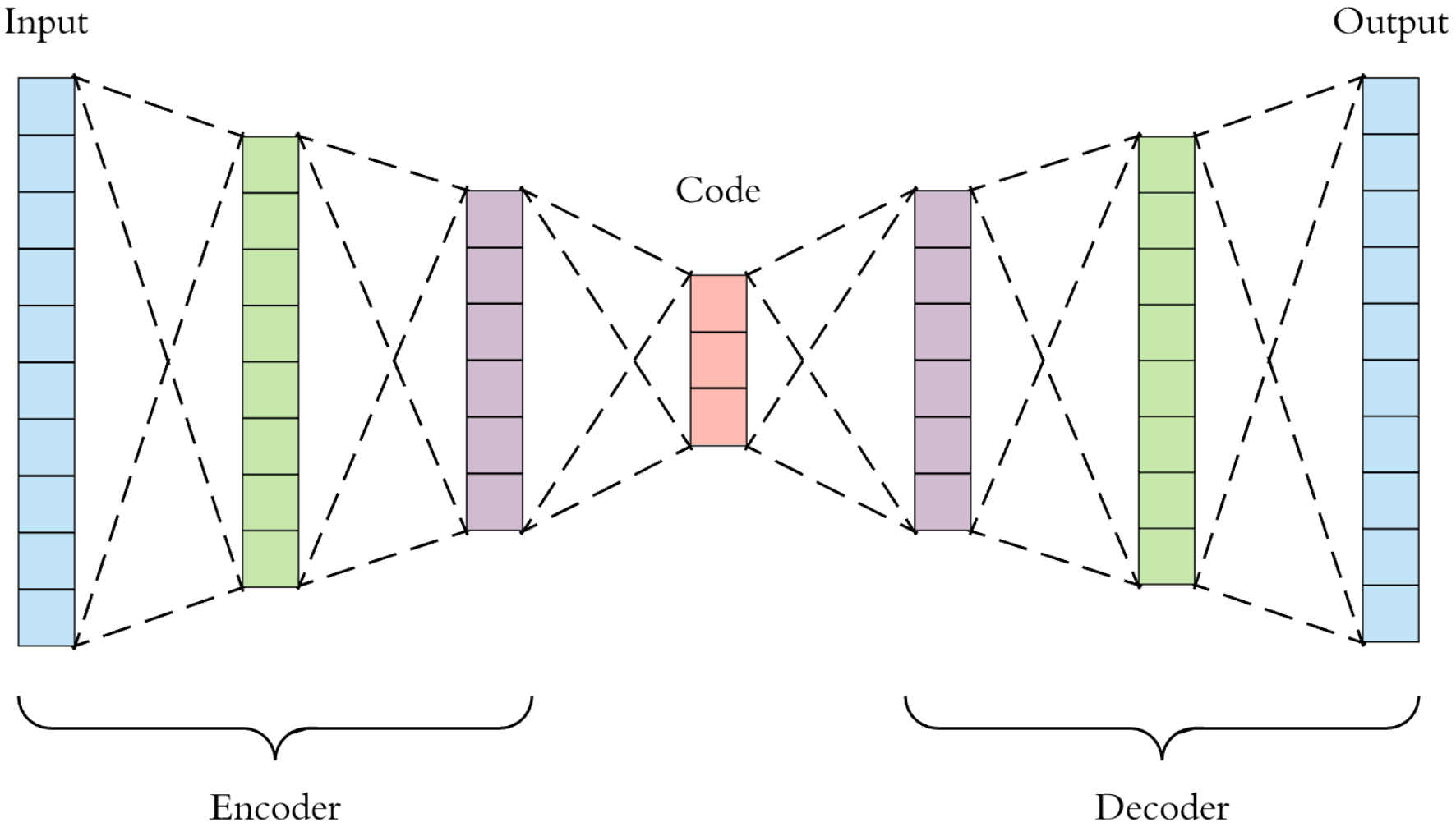
РЕЗУЛТАТИ


F1 Score
0.0
0.2
0.4
0.6
0.8
kNN
SVC
K-Means
One-Class SVM
MGD
Isolation Forest
Autoencoder

AUTOENCODER NEURAL NETWORK

- Најдобри резултати поради природата на мрежата
- Составена од три дела:
  - енкодер – слоеви за редуцирање на податоците (димензиона редукција и занемарување на шум)
  - код (латентна репрезентација) на податоците
  - декодер – слоеви за реконструкција на оригиналните податоци
- Симетрична скалеста форма во однос на кодот

AUTOENCODER NEURAL NETWORK



AUTOENCODER NEURAL NETWORK

- Тренирањето се врши само на множеството за тренинг кога системот е во нормален режим на работа

- Мрежата учи како да ги реконструира податоците по нивната деконструкција

- Множеството за валидација се користи за одредување на оптимални параметри (број на слоеви, број на неврони...)

AUTOENCODER NEURAL NETWORK

```
input_dim = new_train_data.shape[1]
encoding_dim = 179
input_layer = Input(shape=(input_dim, ))

encoder = Dense(int(encoding_dim * .8), activation='tanh', activity_regularizer=regularizers.l1(10e-5))(input_layer)
encoder = Dense(int(encoding_dim * .6), activation="relu")(encoder)
decoder = Dense(int(encoding_dim * .6), activation="relu")(encoder)
decoder = Dense(int(encoding_dim * .8), activation="tanh")(decoder)
decoder = Dense(input_dim, activation='tanh')(decoder)
autoencoder = Model(inputs=input_layer, outputs=decoder)

nb_epoch = 300
batch_size = 256
autoencoder.compile(optimizer='adam',
                    loss='mean_squared_error',
                    metrics=['accuracy'])

checkpointer = ModelCheckpoint(filepath="model.h5",
                               verbose=0,
                               save_best_only=True)

tensorboard = TensorBoard(log_dir='./logs',
                          histogram_freq=0,
                          write_graph=True,
                          write_images=True)

history = autoencoder.fit(new_train_data, new_train_data,
                          epochs=nb_epoch,
                          batch_size=batch_size,
                          shuffle=True,
                          verbose=1,
                          callbacks=[checkpointer, tensorboard]).history
```

- **Градба:**
  - влезен слој со број на неврони еднаков на бројот на feature-и
  - два енкодирачки слоеви со број на неврони еднаков на 80% и 60% од бројот на feature-и соодветно
  - два декодирачки слоеви со број на неврони еднаков на 60% и 80% од бројот на feature-и соодветно
  - излезен слој со ист број на неврони како и влезниот

- **Активациски функции:**
  - Хиперболична *tanh* функција на надворешните скриени слоеви
  - *ReLU* (Rectified Linear Unit) функција за внатрешните скриени слоеви

- **Други параметри:**
  - метрика на евалуација: средна квадратна грешка (*MSE*)
  - алгоритам за оптимизација: *Adam*
  - број на епохи: 300
  - големина на *batch*: 256
  - *threshold:* 0.84

ЗАКЛУЧОК И ИДНА РАБОТА

- Проблемот за детекција на аномалии не е лесен, но има големо значење

- Да се подобрат и оптимизираат алгоритмите со дополнителен *grid search* на сите параметри

- Да се искористат *ensemble* методи за подобрување на резултатите – *stacking*, *boosting, bagging*