



PREGLED METODA I RAZVOJ MODELA U PROGRAMSKOM JEZIKU *PYTHON* ZA PREPOZNAVANJE I PREDVIĐANJE NASTAJANJA NASLAGA U IZMJENJIVAČIMA TOPLINE

Nikola Rimac, Željka Ujević Andrijić
nrimac@fkit.hr
zujevic@fkit.hr
lam.fkit.hr

Jedan od uvijek aktualnih problema u rafinerijskim i drugim postrojenjima procesne industrije je nastajanje naslaga kod izmjenjivača topline. Nastale naslage mogu značajno smanjiti djelotvornost izmjenjivača topline. Najvažniji troškovi proizašli iz nastajanja naslaga su: gubici u proizvodnji, troškovi dodatne potrošnje energije te, posljedično, veće ekološke kazne zbog emisija štetnih plinova. Prema tome, očita je potreba za kontinuiranim praćenjem stvaranja naslaga na izmjenjivačima topline kako bi se omogućilo njihovo preventivno i prediktivno održavanje.

ε -NTU

$$\varepsilon = \frac{Q_s}{Q_{\max}} = \frac{\dot{m}_h c_{p,h} (T_{h,i} - T_{h,u})}{\dot{m}_h c_{p,h} (T_{t,u} - T_{h,u})} \quad NTU = \frac{KA}{C_{\min}}$$

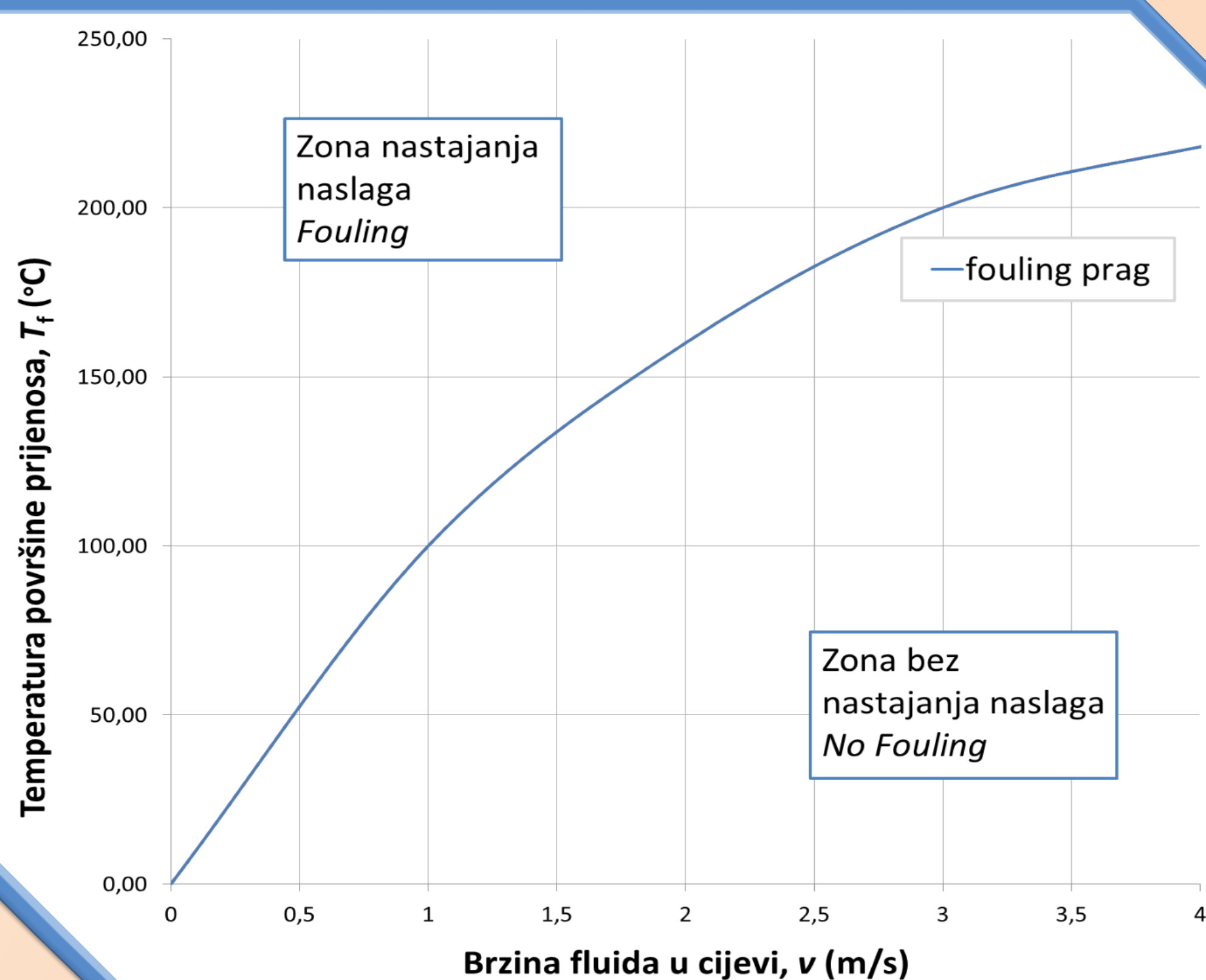
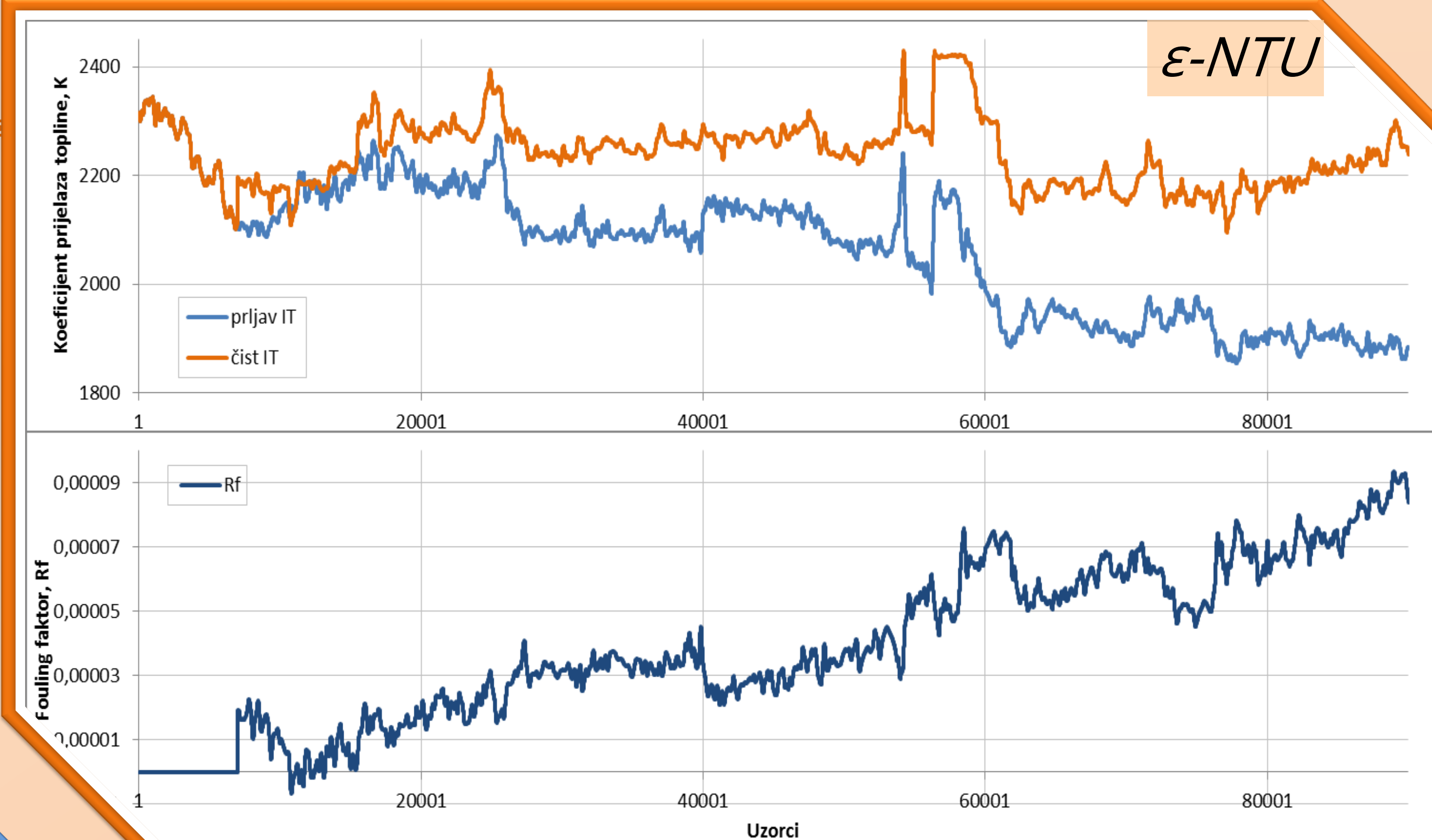
$$Q_{\max} = C_{\min} (\Delta T_{\max}) \quad T_{LM} = \frac{\Delta T_1 - \Delta T_2}{\ln \frac{\Delta T_1}{\Delta T_2}} = \frac{(T_{t,u} - T_{h,i}) - (T_{t,i} - T_{h,u})}{\ln \frac{T_{t,u} - T_{h,i}}{T_{t,i} - T_{h,u}}}$$
$$Q_s = KA \Delta T_{LM}$$

$$\varepsilon = \frac{(T_{h,i} - T_{h,u})}{(T_{t,u} - T_{h,u})} = \frac{KA \Delta T_{LM}}{C_{\min} (\Delta T_{\max})} \xrightarrow{\Delta T_{\max} = (T_{t,u} - T_{h,u})} (T_{h,i} - T_{h,u}) = \frac{KA \Delta T_{LM}}{C_{\min}} \rightarrow NTU = \frac{(T_{h,i} - T_{h,u})}{\Delta T_{LM}}$$

$$K_{\varepsilon} = \frac{\dot{m}_h c_{p,h} NTU_{\varepsilon}}{A}$$

$$K_f = \frac{\dot{m}_h c_{p,h} NTU}{A}$$

$$R_f = \frac{1}{K_f} - \frac{1}{K_{\varepsilon}}$$



Polu-empirijski (engl. *threshold*) modeli

Ovisno o fizikalno-kemijskim karakteristikama fluida te termodinamičkim uvjetima u kojima se nalazi izmjenjivač topline, određuje se potencijal nastajanja naslaga.

$$\frac{dR_f}{dt} = \alpha Re^{\beta} \exp\left(-\frac{E_a}{RT_f}\right) - \gamma \tau$$

Ebert-Panchal model

$$\frac{dR_f}{dt} = \alpha Re^{\beta} Pr^{-0.33} \exp\left(-\frac{E_a}{RT_f}\right) - \gamma \tau$$

Panchal *et al.* model

$$\frac{dR_f}{dt} = \frac{K}{h_i} \exp\left(-\frac{E_a}{RT_f}\right) S$$

Polley *et al.* model

Zbog složenog procesa nastajanja naslaga, većeg broja tipova izmjenjivača topline i odstupanja od projektiranih radnih uvjeta, ne postoji široko primjenjiva standardna metoda, poluempirijski ni teorijski model za predviđanje nastajanja naslaga. S druge strane, znanstvena i stručna zajednica sve više koristi blagodat digitalizacije, čijom se primjenom prikupljaju i arhiviraju velike količine mjernih podataka. Na temelju toga mogu se razvijati modeli temeljeni na podacima različitim metodama identifikacije. Primjeri takvih naprednih metoda su sustavi neizrazite logike, umjetne neuronske mreže, parcijalna regresija metodom najmanjih kvadrata, nelinearne autoregresijske metode i druge. Prikazani rezultati dobiveni su temeljem realnih podataka izmjenjivača topline na procesu hidrokreiranja u naftnoj industriji.

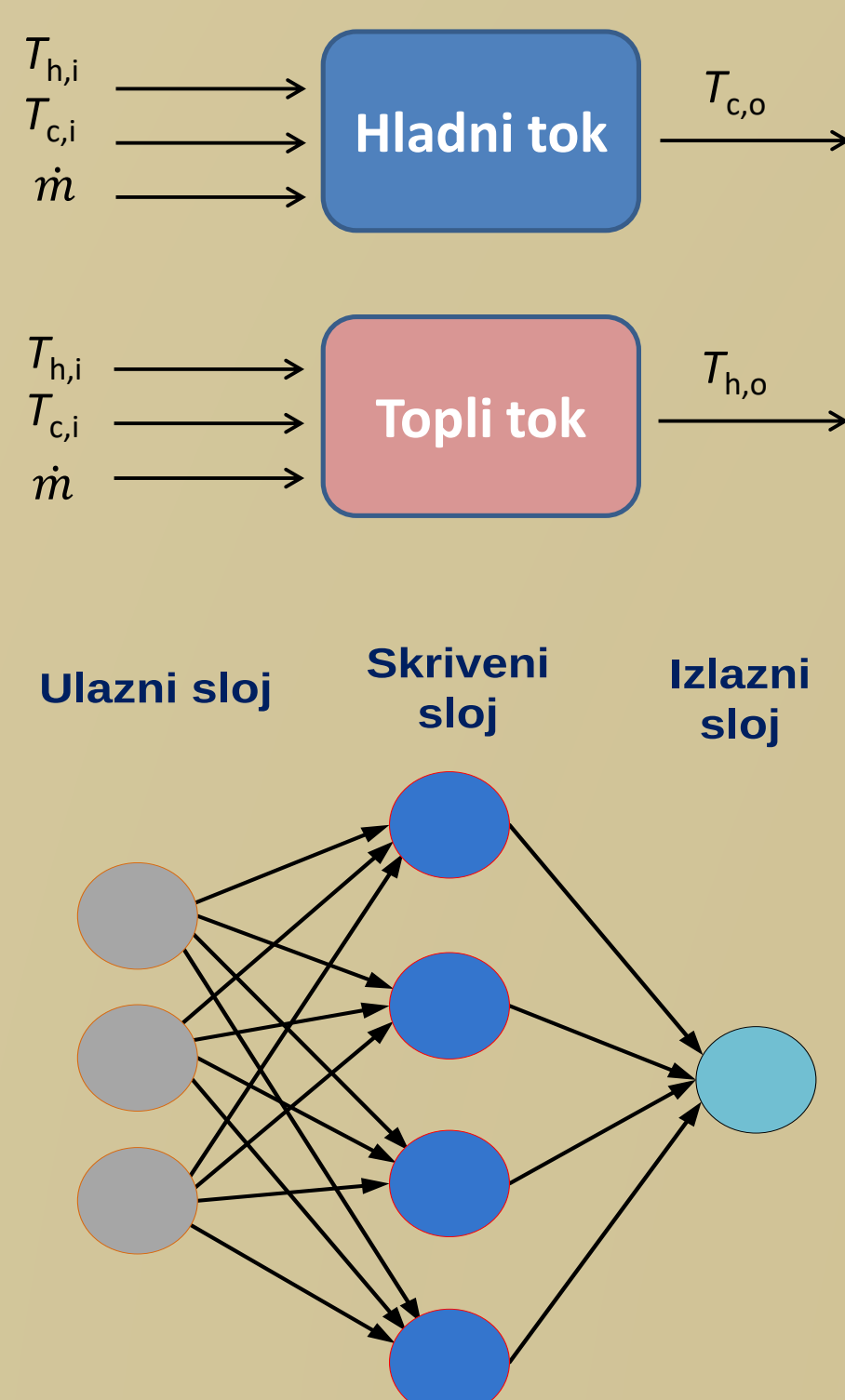
Ulazne varijable	Izlazne varijable
$T_{h,i}$ Ulazna temperatura tople struje	$T_{h,i}$ Ulazna temperatura tople struje
$T_{c,i}$ Ulazna temperatura tople struje	$T_{h,i}$ Ulazna temperatura tople struje
$\dot{m}$ Maseni protok hladne struje	$T_{h,i}$ Ulazna temperatura tople struje

$$K_f = \frac{\dot{m} c_p (T_{h,i} - T_{h,o})}{AF \ln \left(\frac{(T_{h,i} - T_{c,o})}{(T_{h,o} - T_{c,i})} \right)}$$

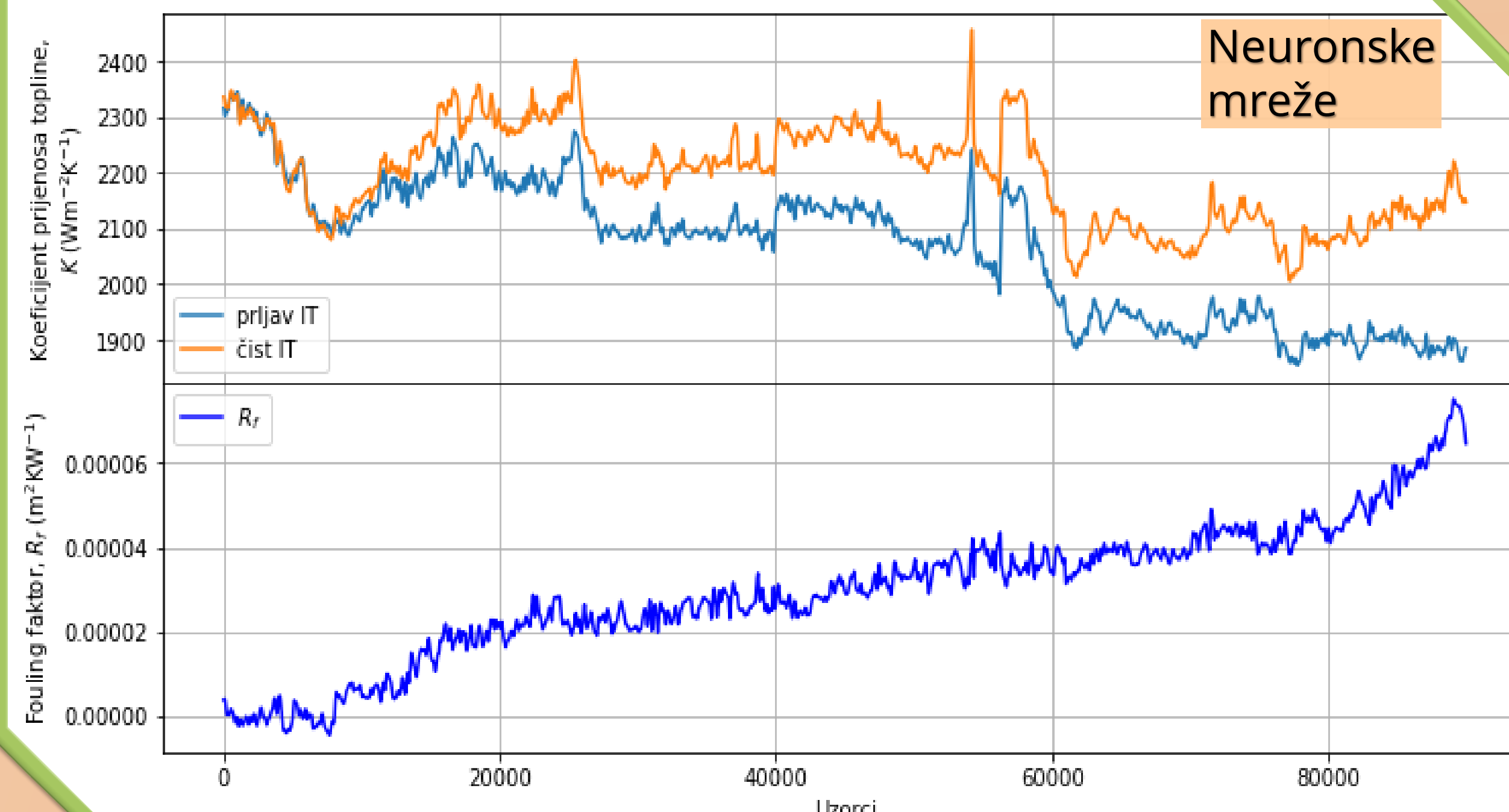
$$K_{\varepsilon} = \frac{\dot{m} c_p (T_{h,i} - T_{h,o,mod})}{AF \ln \left(\frac{(T_{h,i} - T_{c,o,mod})}{(T_{h,o,mod} - T_{c,i})} \right)}$$

$$R_f = \frac{1}{K_f} - \frac{1}{K_{\varepsilon}}$$

Model umjetne neuronske mreže



Keras



```
model = Sequential()
model.add(Dense(3, input_dim = 3, kernel_initializer = 'TruncatedNormal', activation = 'tanh'))
model.add(Dense(1, activation = 'Linear'))
model.compile(loss = 'mse', optimizer = 'Nadam', metrics = ['mae'])
Early_stop = keras.callbacks.callbacks.EarlyStopping(monitor = 'mae', patience = 5)
model.fit(trainX, trainY, epochs=100, batch_size = 32, validation_split = 0.2, callbacks=[Early_stop], verbose = 0)
model.evaluate(testX, testY, verbose = 0)
```

Python