Mendelova univerzita v Brně
Provozně ekonomická fakulta

Klasifikace paketů

Neuronové sítě v aplikacích

Zpracovali:

Bc. Patrik Michalák
Bc. Tereza Plíšková

Brno 2019

Obsah

1 Úvod a cíl práce

Práce se zabývá návrhem a implementací neuronové sítě pro klasifikaci sady paketů. Cílem práce je dosáhnout co nejlepších klasifikačních výsledku při klasifikaci třídy sady paketů na základě určených atributů. Modifikacemi počtu atributů (features), počtu záznamů (records) a nastavení učícího algoritmu lze pozorovat změny učící křivky a výsledné úspěšnosti.

2 Data

Byl vybrán dataset, který obsahuje 2 160 668 záznamů. Takové množství záznamů je zcela postačující pro rozdělení na trénovací, validační a testovací množinu. Každý záznam z původního datasetu se skládá z 28 atributů, kde poslední atribut určuje, do jaké třídy byla sada paketů zařazena. Budeme tedy implementovat tzv. učení s učitelem.

Dataset byl převzat z volně dostupného zdroje na internetu. Je vhodný především pro jeho velké množství poskytovaných informací, a s velkou pravděpodobností bude dosažena úspěšnost nad 90%, i pro jednoduchou neuronovou síť.

Jak již bylo zmíněno výše, jedná se o dataset obsahující informace o sadách paketů. Každý záznam představuje jeden z vybraných typu DoS[1] útoků, nebo normální pakety, nejedná se však pouze o jeden paket, ale o celou sadu, která byla útočníkem zaslána.

Pakety jsou roztříděny do pěti tříd a to následovně:

	Název třídy	**Počet záznamů**
1	Normal	1935959
2	UDP-Flood	201344
3	Smurf	12590
4	SIDDOS	6665
5	HTTP-FLOOD	4110

Tab. 1 Rozložení záznamů do tříd

Jak je vidět v Tab. 1 rozložení vzorků do tříd není rovnoměrné. Počet paketů označených jako Normal tvoří přibližně 4/5 z celkového počtu. I přesto je počet dat nejméně zastoupené třídy plně dostačující pro dosáhnutí uspokojivých výsledků.

Všechny záznamy jsou úplné, tedy pro každý záznam existuje platná informace pro každý atribut.

2.1 Rozdělení dat

V rámci projektu byl vytvořen skript datasetFilter.pl, který slouží pro generování rozličných sad dat z původního datasetu. Hlavní funkcí skriptu je rozdělení dat do sad:

[1] Denial of service – zamítnutí poskytnutí služby na určitou dobu

- Sada tří csv souborů – trénovací data (70%), data pro validaci (15%) a testovací data (15%).
- Sada dvou csv souborů dataset_N_per_class_train a dataset_N_per_class_test, kde N určuje zvolený počet záznamů na jednu kalcifikační třídu (celkem N×5 záznamů). Aby nedocházelo k opakování stejných záznamů, první z výše zmíněných souborů obsahuje data ze začátku původního datasetu a druhý z konce původního datasetu.

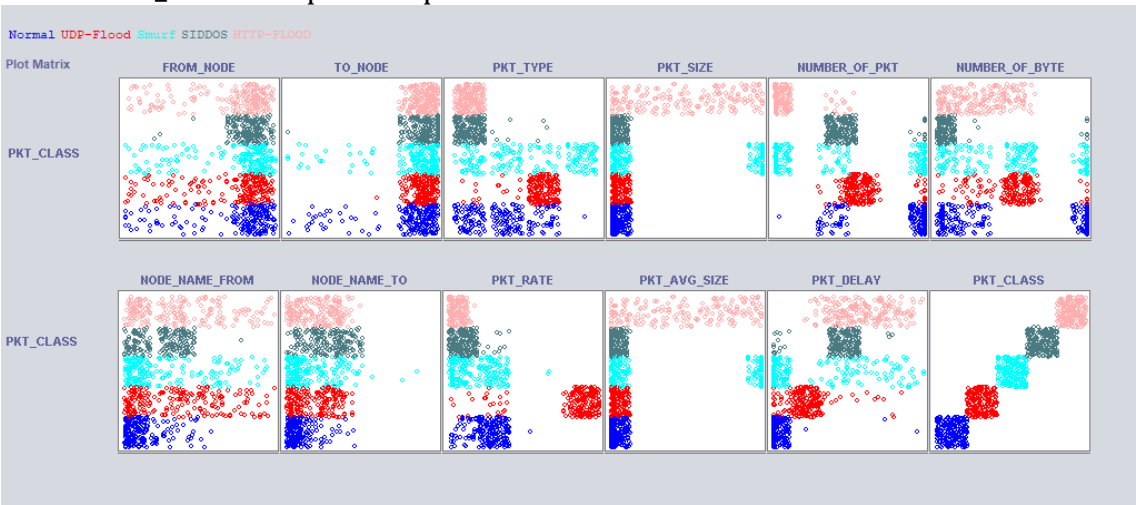
Skript dále provádí minoritní úpravy jednotlivých záznamů:

- Převedení dat, kde je hodnotou řetězec, na číselník numerické hodnoty
- Odstranění nepodstatných atributů
- Formátování numerických hodnot s desetinnou složkou
- Formátování oddělovače atributů na programem přijatelný tvar

2.2 Závislost vybraných dat na výsledné hodnotě

Z celkového počtu 27 atributů klasifikujících jednotlivé záznamy byly vybrány následující:

1. FROM_NODE – číslo portu, ze kterého byla sada paketů poslána
2. TO_NODE – číslo portu, na který byla sada paketů poslána
3. PKT_TYPE – typ protokolu {tcp, ack, cbr, ping} (ve výše zmíněném skriptu převedeno na číselník)
4. PKT_SIZE – velikost sady paketů
5. NUMBER_OF_PKT – počet paketů
6. NUMBER_OF_BYTE – počet bytů
7. NODE_NAME_FROM – název uzlu posílající pakety
8. NODE_NAME_TO – název uzlu přijímající pakety
9. PKT_RATE – počet přenesených paketů za sekundu
10. PKT_AVG_SIZE – průměrná velikost jednoho paketu
11. PKT_DELAY – zpoždění paketů



Obr. 1 Vykreslení závislosti vybraných atributů na klasifikační třídě

2.3 Normalizace

Normalizace je vhodná především pro atributy, které mají příliš velký rozptyl hodnot. Tento problém může navodit nekonzistenci v rozhodování neuronové sítě, a proto byly všechny atributy záznamů normalizovány pomocí techniky Mean normalization

$$x_i := \frac{x_i - \mu_i}{\sigma_i}$$

3 Evaluace hypotézy

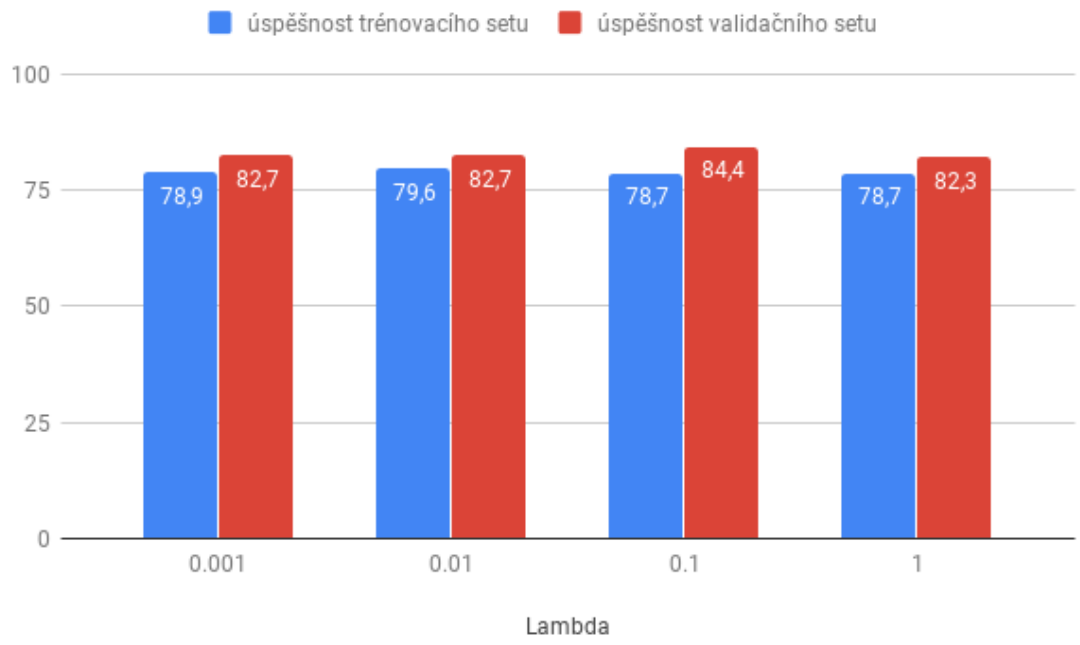
Pro výběr správného modelu a nastavení parametrů učícího algoritmu jsme zkoumali závislosti změn parametru k celkové úspěšnosti. U každé z možností jsme porovnávali úspěšnost trénovacího a validačního datasetu. Výchozí hodnoty při evaluaci byly stanoveny následovně:

- Regularizační parametr Lambda = 0,1
- Maximální počet epoch (iterací) optimalizační funkce = 50
- Počet neuronů na skryté vrstvě = 10
- Počet atributů (features) = 11
- Počet záznamů na třídu pro trénovací set = 500 (celková velikost 5×500 = 2 500 záznamů)
- Počet záznamů na třídu pro validační set = 20 (celková velikost 5×20 = 100 záznamů)

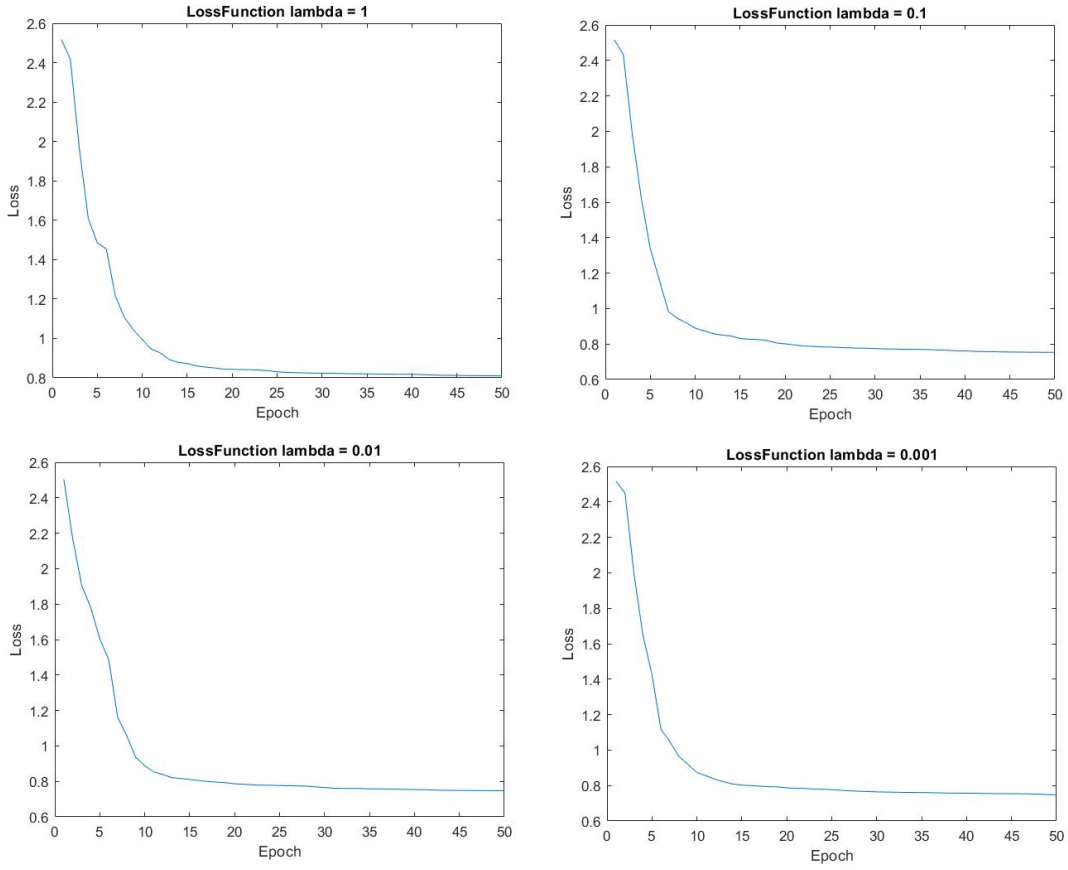
Po každé modifikaci parametrů bylo vyhodnoceno, zda má změna parametru vliv (zlepšení / zhoršení) na učící algoritmus.

3.1 Regularizační parametr Lambda

Parametr Lambda slouží k definování váhy regularizačního termu. Když má parametr příliš nízkou hodnotu, může dojít k přetrénování neuronové sítě. Jako výchozí hodnota regularizačního parametru Lambda byla zvolena konstanta 0,1. Ta se ukázala být vhodně zvolenou. Úspěšnost validačního datasetu není menší než úspěšnost trénovacího datasetu, takže nedochází k přetrénovaní.



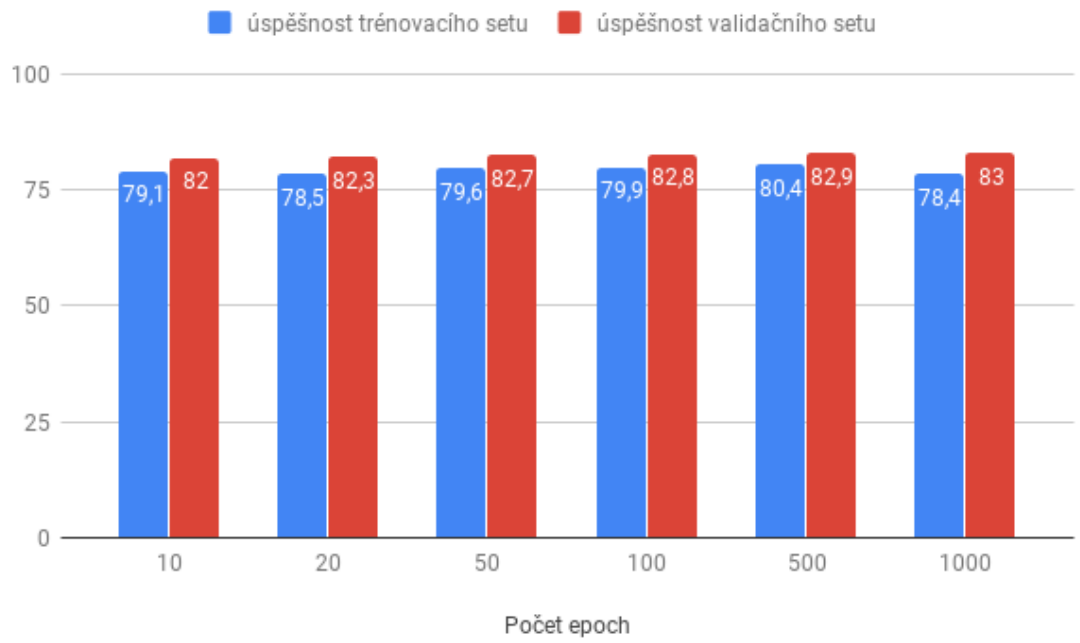
Obr. 2 Porovnání úspěšnosti trénovacího a validačního datasetu při změně parametru Lambda



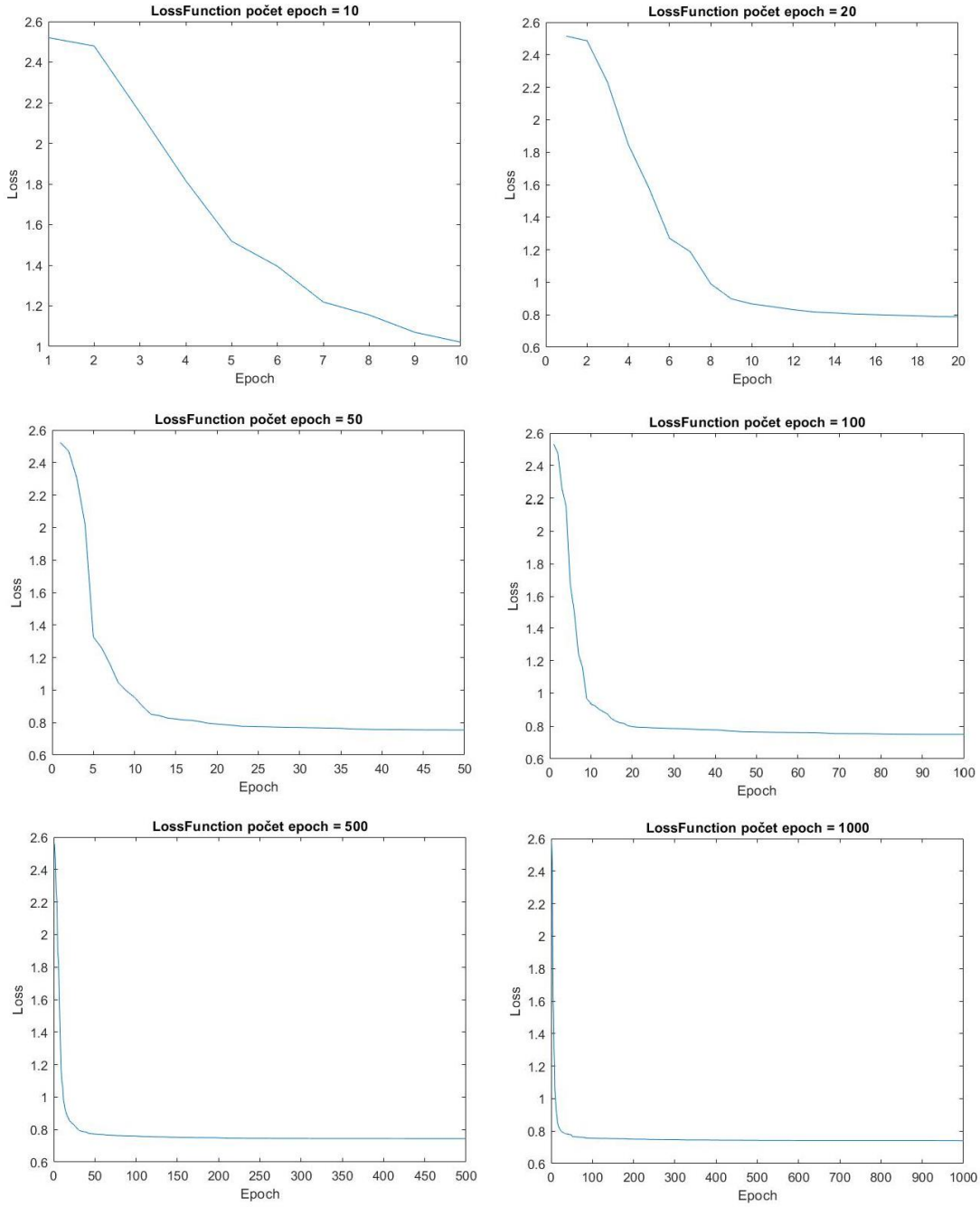
Obr. 3 Vykreslení učících křivek při změně parametru Lambda

3.2 Počet epoch

Výběr počtu epoch, neboli maximálního počtu iterací optimalizační funkce, nezávisí tolik na celkové úspěšnosti algoritmu. Důležité je najít optimální počet iterací, aby rozdíl hodnot mezi posledními iteracemi loss funkce byl co nejvíce ustálen. Když se podíváme na učící křivky viz Obr. 5, v případě, kdy je počet epoch roven 10 není funkce dostatečně ustálena. Naopak při přibližně 50 iteracích je loss funkce skoro dokonale ustálena. Proto jako defaultní počet epoch bylo zvoleno právě 50 iterací.



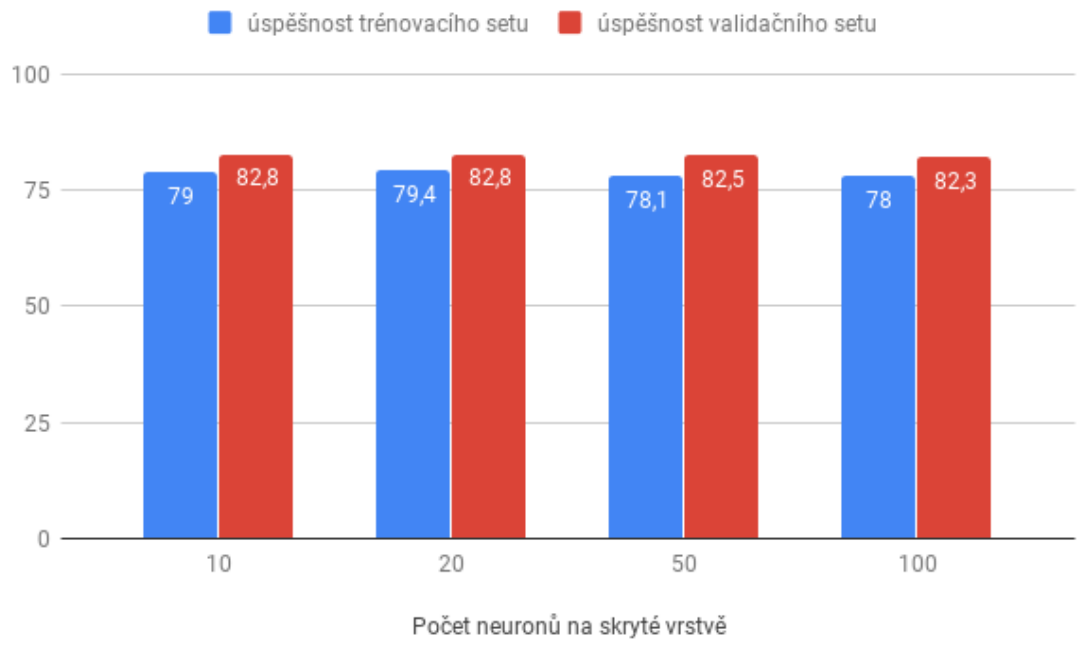
Obr. 4 Porovnání úspěšnosti trénovacího a validačního datasetu při změně počtu epoch



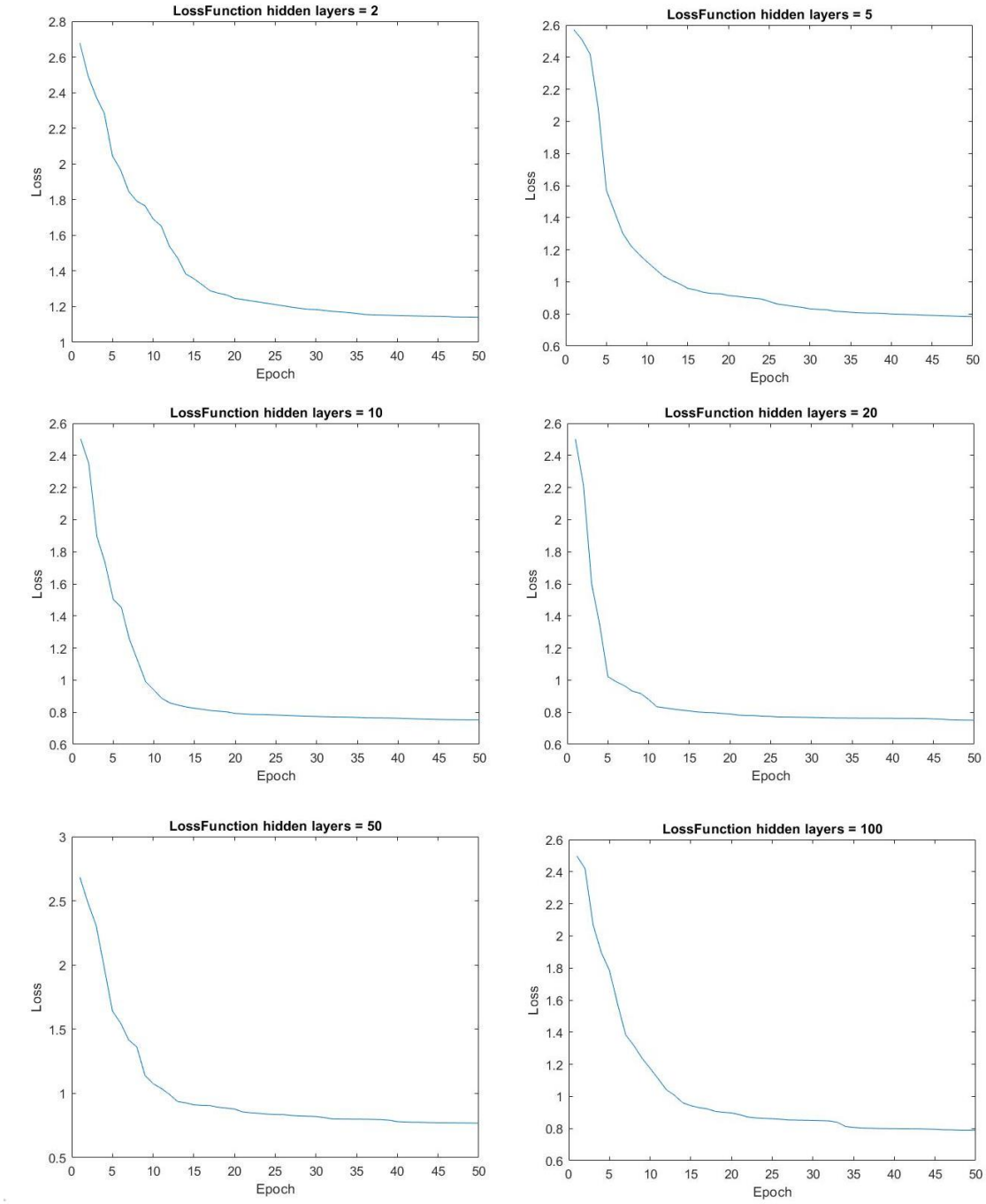
Obr. 5 Vykreslení učících křivek při změně počtu epoch (iterací)

3.3 Počet neuronů na skryté vrstvě

Jako ideální počet neuronů na skryté vrstvě bylo zvoleno 10 neuronů. Při větším počtu neuronů teoreticky může docházet ke zlepšení úspěšnosti, ale zbytečně se komplikuje výpočetní náročnost algoritmu.



Obr. 6 Porovnání úspěšnosti trénovacího a validačního datasetu při změně počtu neuronů na skryté vrstvě

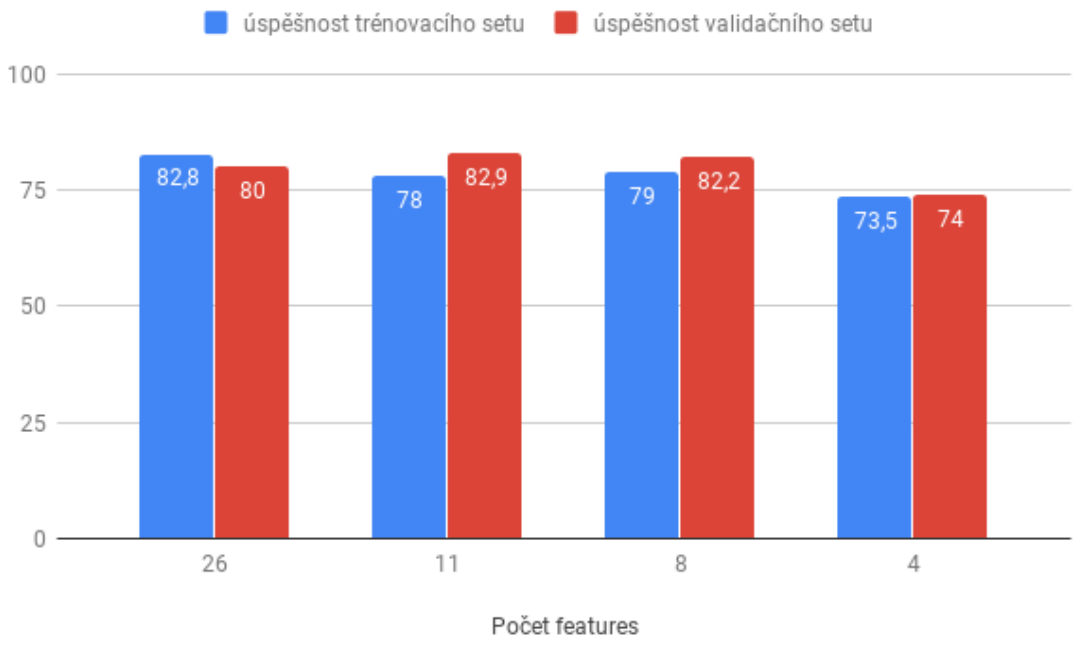


Obr. 7 Vykreslení učících křivek při změně počtu neuronů na skryté vrstvě

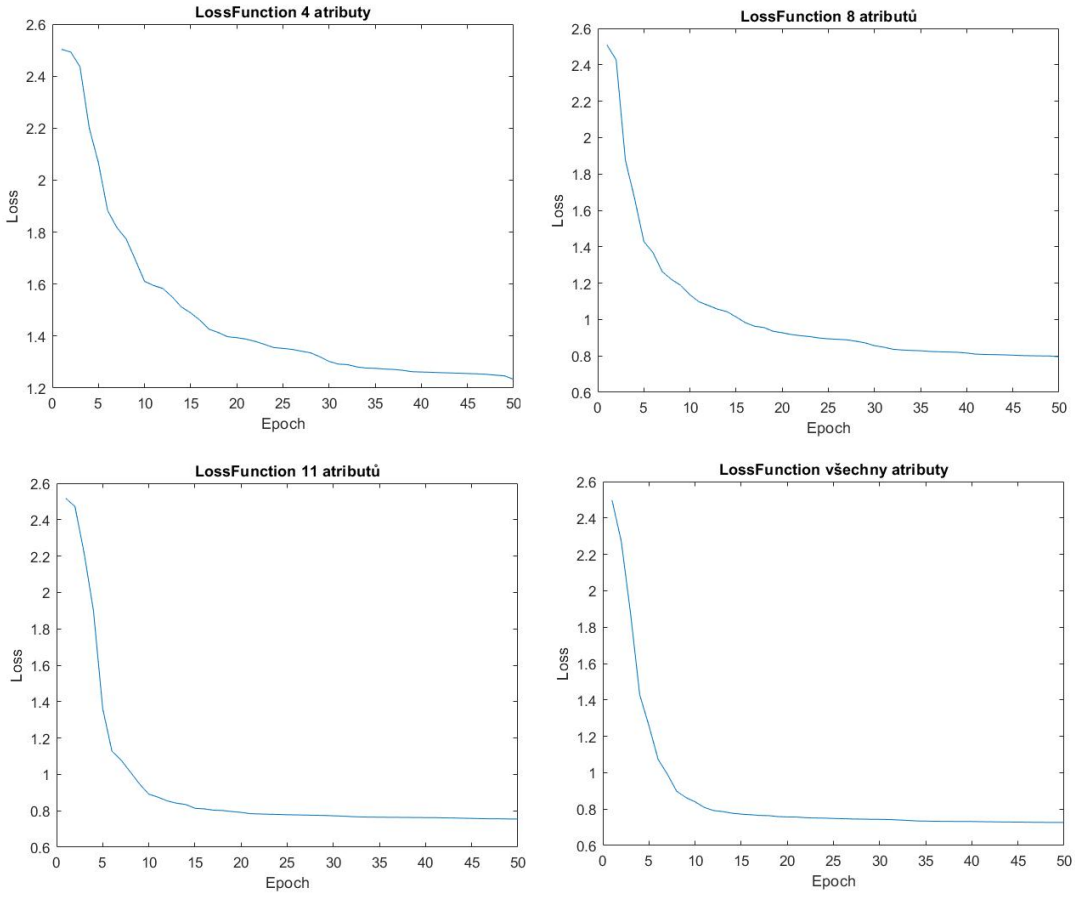
3.4 Počet atributů (features)

Jak již bylo uvedeno v kapitole Data, bylo vybráno konkrétních 11 atributů. Pro ověření správnosti tohoto výběru byla provedena analýza vlivu počtu a typu atributů

na výslednou úspěšnost. Z učících křivek vykreslený při použití 4 atributů (NUMBER_OF_PKT, PKT_RATE, PKT_AVG_SIZE, PKT_DELAY) vidíme, že loss funkce stabilizuje pomalu a taky celková úspěšnost není lepší oproti ostatním. Přidání dalších informací o další 4 atributy se učící křivka ustálí a zvedne celkovou úspěšnost. Přidáním informací o zdrojovém uzlu a cílovém síťovém uzlu, tedy na počet 11, je zaručen stabilní výsledek. Při použitích všech 26 atributů, které máme k dispozici, se učící křivka stabilizuje sice velmi rychle, ale v grafu úspěšnosti můžeme vidět, že došlo k přetrénování učícího algoritmu, a navíc i k zbytečnému nárůstu výpočetní složitosti.



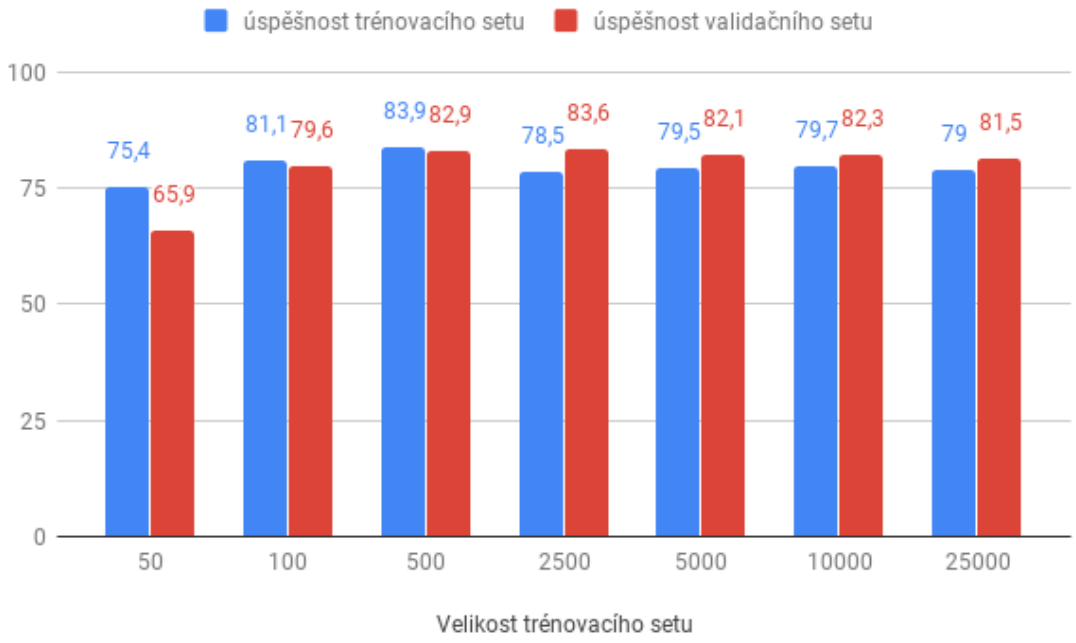
Obr. 8 Porovnání úspěšnosti trénovacího a validačního datasetu při změně počtu atributů



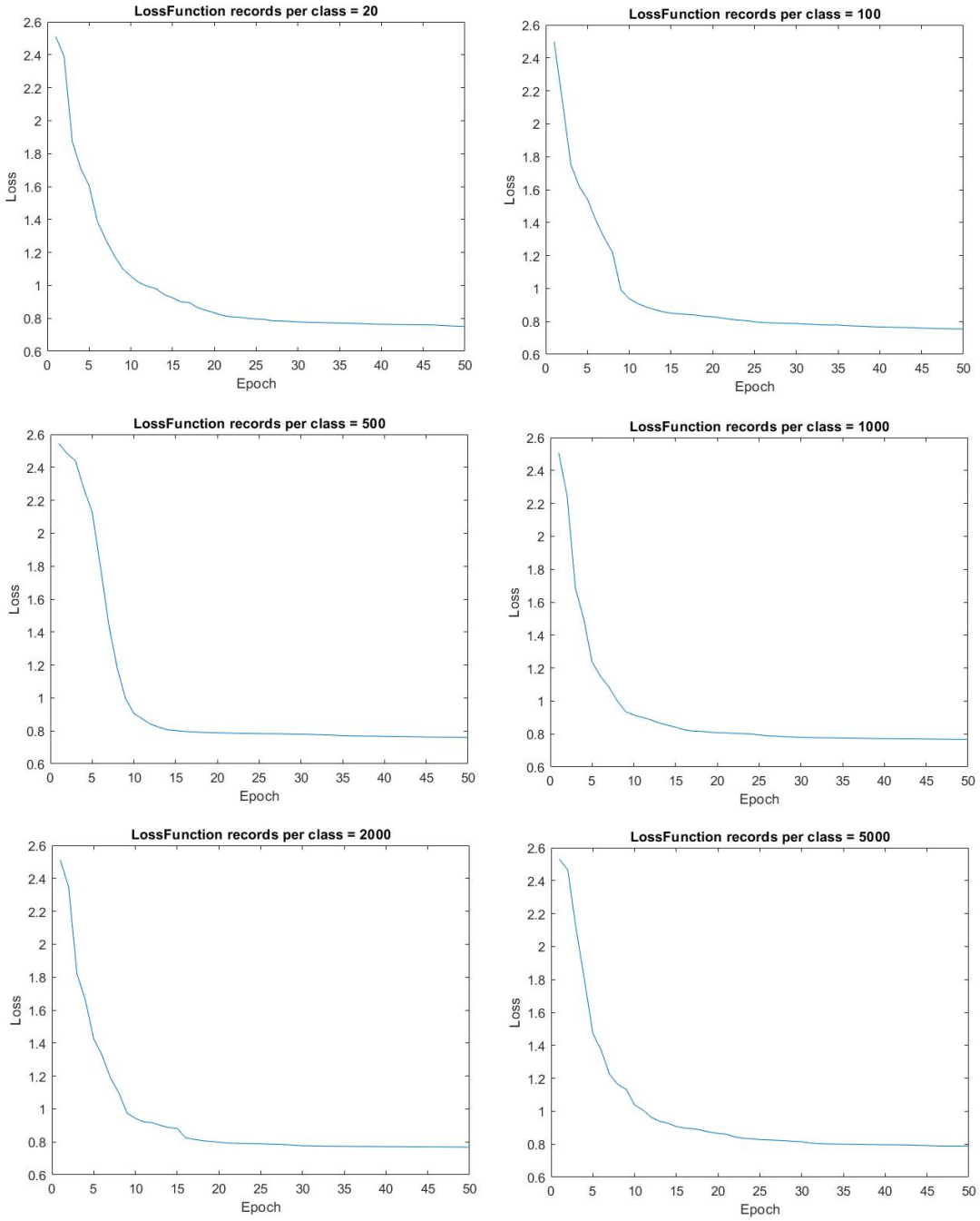
Obr. 9 Vykreslení učících křivek při změně počtu atributů

3.5 Velikost trénovacího datasetu

Jako defaultní počet záznamů na třídu pro evaluaci hypotéz bylo vybráno 500, tedy 2 500 záznamů. Jak je vidět z grafu na Obr. 10 volba 500 záznamů na třídu má nejvyšší úspěšnost a i učící křivka, viz Obr. 11, je dostatečně stabilní.



Obr. 10 Porovnání úspěšnosti trénovacího a validačního datasetu při změně velikosti trénovacího datasetu



Obr. 11 Vykreslení učících křivek při změně velikosti datasetu

4 Implementace

Učící algoritmus byl vytvořen na základě zdrojových kódů ze cvičení předmětu Neuronové sítě v aplikacích. Algoritmus je napsán v prostředí MATLAB a součástí implementace je i skript v jazyce Perl pro definování, filtrování a generování datasetů ve formátu .csv.
Stručný popis souborů a jejich funkcionalit:

- index.m – řídící soubor projektu
- featureNormalization.m – normalizace vstupních dat pomocí Mean normalization
- loadData.m – načtení a selekce vstupních dat
- nnCostFunction.m – implementace učícího algoritmu ze cvičení
- predict.m – testování úspěšnosti neuronové sítě
- randInitializeWeights.m – pomocná funkce pro prvotní generování vah
- sigmoid.m – aktivační funkce sigmoid
- pomocné soubory pro zobrazovaní grafu
- datasetFilter.pl – generování datasetu

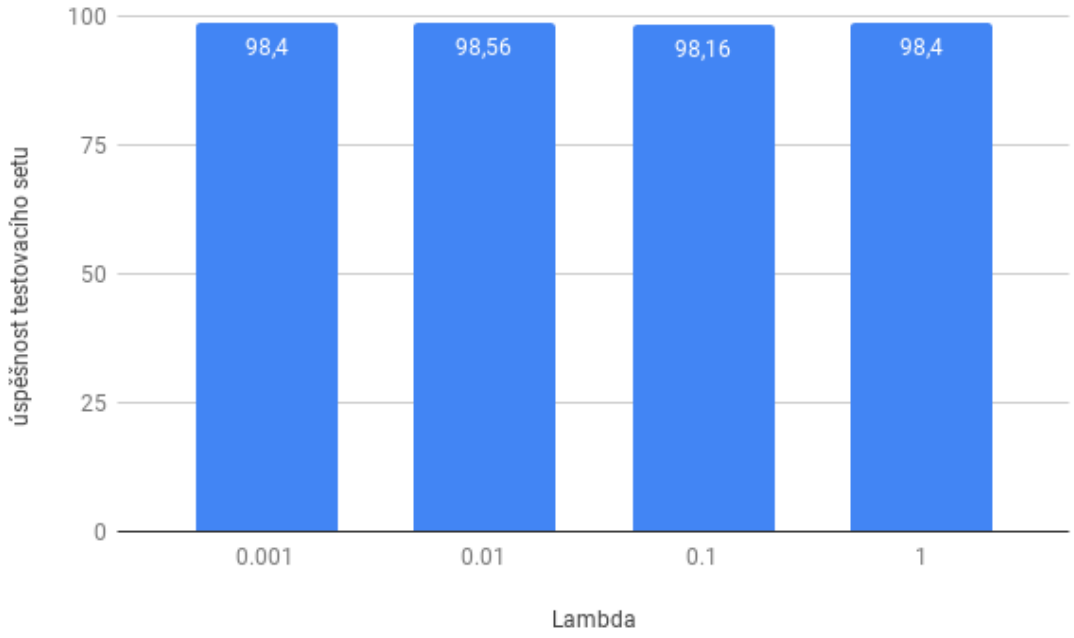
5 Vyhodnocení celkové úspěšnosti

Pro celkové vyhodnocení úspěšnosti neuronové sítě byl použit dataset vytvořený ze všech dostupných záznamů, kde trénovací dataset obsahoval 70% celkového počtu záznamů a validační a testovací po 15%. Celkově jsme tedy získali datasety s následujícím počtem záznamů pro jednotlivé klasifikační třídy:

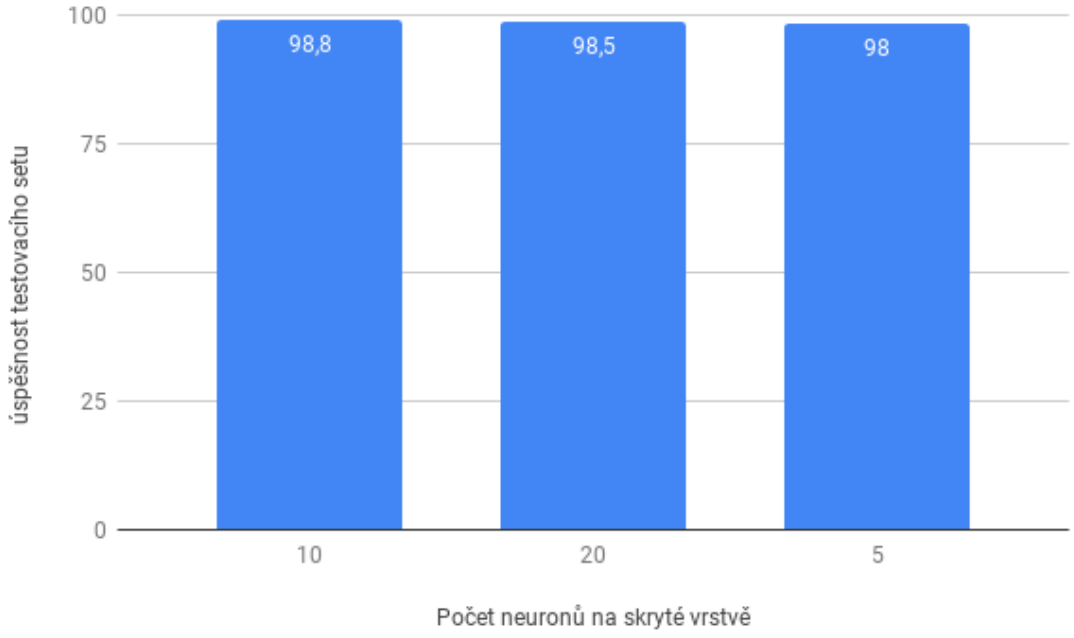
	Train	**Validation**	**Test**
Normal	1355171	290393	290393
UDP-Flood	140940	30201	30201
Smurf	8813	1888	1888
SIDDOS	4665	999	999
HTTP-FLOOD	2877	616	616

Tab. 2 Počet záznamů na jednotlivé třídy Train – Validation – Test

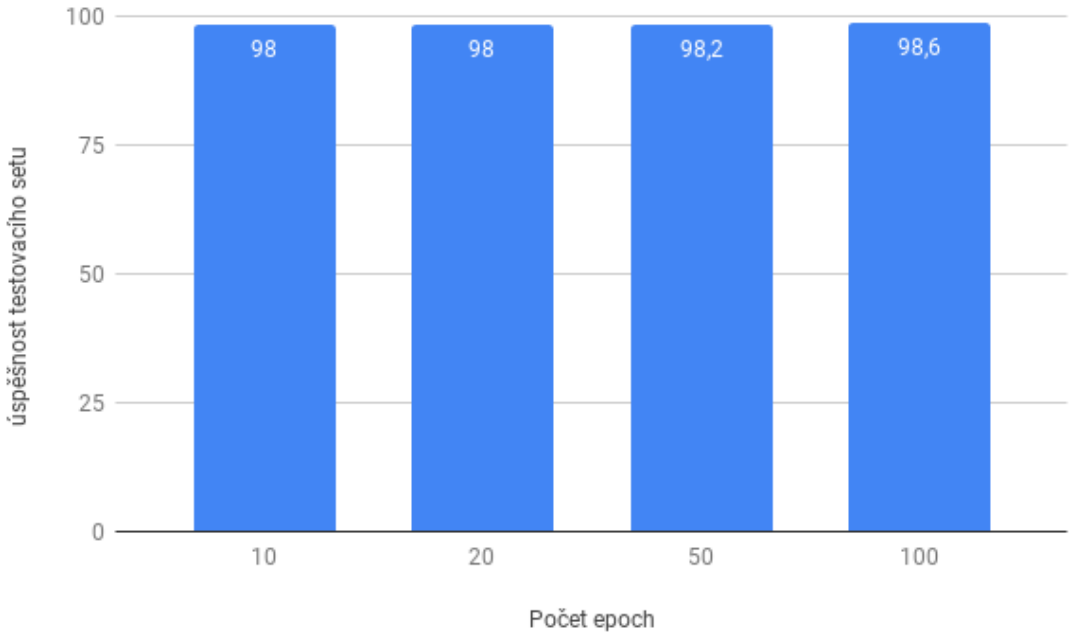
Pro výběr modelu a zvolení výchozích hodnot jsme ještě jednou zkusili definovat některé parametry, ale celková úspěšnost testovacího datasetu jenom mírně kolísala nad celkovou úspěšností větší než 98%.



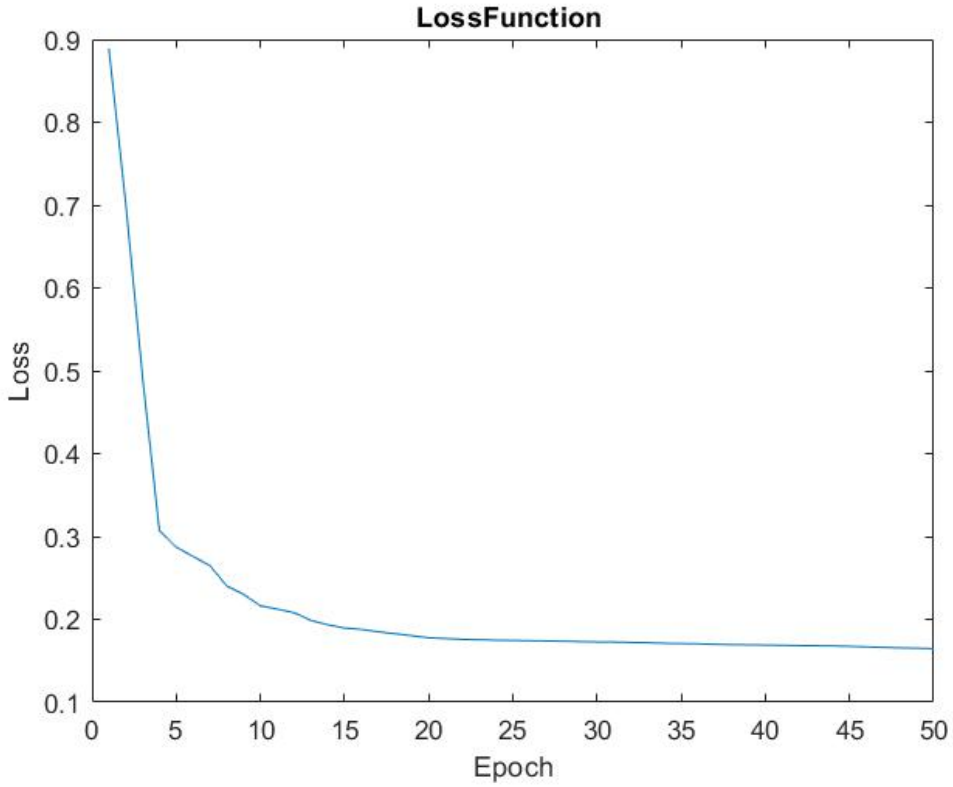
Obr. 12 Vliv parametru Lambda na celkovou úspěšnost



Obr. 13 Vliv počtu neuronů na skryté vrstvě na celkovou úspěšnost



Obr. 14 Vliv počtu epoch na celkovou úspěšnost



Obr. 15 Vykreslení učící křivky při Lambda = 0,1; počet skrytých neuronů = 10; počet epoch = 50; počet atributů = 11; 70% datasetu