Univerzita Pardubice
Fakulta elektrotechniky a informačních technologií

# Neuronová síť typu transformer pro extrakci vlastností ze signálu

Diplomová práce

2024 Vojtěch Smetana

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2023/2024

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

| | |
|---|---|
| Jméno a příjmení: | **Bc. Vojtěch Smetana** |
| Osobní číslo: | **I22193** |
| Studijní program: | **N0714A150005 Automatické řízení** |
| Téma práce: | **Neuronová síť typu transformer pro extrakci vlastností ze signálu** |
| Zadávající katedra: | **Katedra řízení procesů** |

## Zásady pro vypracování

Postup: Cílem práce je návrh a implementace extraktoru relevantních vlastností z vícekanálového signálu založeného na umělé neuronové síti typu transformer. Student v rámci práce navrhne vlastní topologii umělé neuronové sítě a ve zvoleném frameworku tuto topologii implementuje. Schopnost extrakce relevantních vlastností bude ověřena na minimálně třech fyzikálně odlišných typech signálu.
Teoretická část: Stručná rešerše problematiky umělých neuronových sítí a jejich využití pro extrakci relevantních vlastností. Podrobná rešerše topologií typu transformer. Popis frameworků pro implementaci neuronových sítí. Popis typu signálu vybraných pro realizaci praktické části práce.
Praktická část: Návrh a implementace extraktoru relevantních vlastností z vícekanálového signálu. Tvorba datasetů. Trénovací experimenty. Vyhodnocení kvality extrakce.

Rozsah pracovní zprávy: **cca. 60 stran**
Rozsah grafických prací:
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

HAYKIN, Simon S., c2009. *Neural networks and learning machines*. 3rd ed. New York: Prentice Hall. ISBN 9780131471399.
VOLNÁ, Eva, 2014. *Umělá inteligence: rozpoznávání vzorů v dynamických datech*. Praha: BEN – technická literatura. ISBN 9788073004972.
CHOLLET, François, 2021. *Deep Learning with Python, Second Edition*. Shelter Island, NY: MANNING PUBN. ISBN 9781617296864.

Vedoucí diplomové práce: **doc. Ing. Petr Doležel, Ph.D.**
Katedra řízení procesů

Datum zadání diplomové práce: **8. listopadu 2023**
Termín odevzdání diplomové práce: **17. května 2024**

L.S.

**Ing. Zdeněk Němec, Ph.D.** v.r.
děkan

**Ing. Daniel Honc, Ph.D.** v.r.
vedoucí katedry

Prohlašuji:

Práci s názvem Neuronová síť typu transformer pro extrakci dat ze signálu jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 9. 5. 2025

Vojtěch Smetana

# PODĚKOVÁNÍ

Na tomto místě bych chtěl srdečně poděkovat svému vedoucímu, prof. Ing. Petr Doležel, Ph.D., za trpělivé a odborné vedení. Vždy mi poskytl cenné rady a pomohl nasměrovat další kroky mé práce. Vaše podpora a zpětná vazba výrazně přispěly k jejímu úspěšnému dokončení. Děkuji!

# ANOTACE

*Cílem práce je návrh a implementace extraktoru relevantních vlastností z vícekanálového signálu založeného na umělé neuronové síti typu transformer. V rámci práce bude navržena vlastní topologie umělé neuronové sítě a ve zvoleném frameworku bude tato topologie implementována. Schopnost extrakce relevantních vlastností bude ověřena na minimálně třech fyzikálně odlišných typech signálu.*

# KLÍČOVÁ SLOVA

*Neuronová síť, Model typu transformer, Zpracování signálu, Hluboké učení, Vícekanálové signály, Extrakce vlastností*

# TITLE

***Neural network of type transformer for feature extraction from a signal***

# ANNOTATION

*The objective of this work is to design and implement an extractor of relevant properties from a multi-channel signal based on a neural network of type transformer. The work will be focused on the design of a unique topology of an artificial neural network and its implementation in a chosen chosen framework. The ability to extract relevant properties will be verified on at least three different types of signal.*

# KEYWORDS

*Neural network, Transformer model, Signal processing, Deep learning, Multichannel signals, Feature extraction*

# OBSAH

# SEZNAM ILUSTRACÍ A TABULEK

# SEZNAM ZKRATEK A ZNAČEK

| | |
|---|---|
| AI | Umělá inteligence *(Artificial Intelligence)* |
| ASR | Automatické rozpoznávání řeči *(Automatic Speech Recognition)* |
| CER | Character Error Rate – míra chybovosti při rozpoznávání řeči |
| CNN | Konvoluční neuronová síť (*Convolutional Neural Network*) |
| ECG / EKG | Elektrokardiogram / Elektrokardiografie |
| F1 | F1 skóre – harmonický průměr přesnosti a úplnosti |
| F1-like | Upravené F1 skóre se zohledněním tolerančního intervalu |
| FS | Vzorkovací frekvence *(Sampling Frequency)* |
| LSTM | Long Short-Term Memory – typ rekurentní neuronové sítě |
| MAE | Mean Absolute Error – střední absolutní chyba |
| MSE | Mean Squared Error – Střední kvadratická chyba |
| P/S fáze | Primární (P) a sekundární (S) seismická vlna |
| PTB-XL | Databáze elektrokardiogramů z Physikalisch-Technische Bundesanstalt |
| QRS | Charakteristický tvar EKG signálu reprezentující depolarizaci komor |
| RNN | Rekurentní neuronová síť *(Recurrent Neural Network)* |
| STEAD | Seismic Transfer and Earthquake Arrival Database |

# ÚVOD

Sekvenční data hrají zásadní roli v celé řadě aplikačních oblastí – od monitorování geofyzikálních jevů a biologických procesů až po rozpoznávání lidské řeči. Tato data jsou typicky vícerozměrná, časově závislá a často zatížená šumem, což klade vysoké nároky na jejich zpracování a interpretaci. V posledních letech došlo k výraznému pokroku ve využití hlubokého učení, zejména modelů založených na architektuře transformerů, které se osvědčily zejména v oblasti zpracování přirozeného jazyka. Jejich schopnost modelovat dlouhodobé závislosti a paralelně zpracovávat vstupy je činí atraktivními i pro další domény.

Cílem této práce je navrhnout, implementovat a experimentálně ověřit univerzální transformerovou architekturu, která je schopna efektivně extrahovat reprezentace z různých typů vícekánálových časových signálů. Práce se zaměřuje na tři specifické modality: seismická data, elektrokardiogramy (EKG) a řečové signály. Pro každou z těchto oblastí byla vytvořena sada úloh – klasifikačních i regresních – které reprezentují prakticky významné scénáře, jako je detekce zemětřesení, diagnostika srdečních poruch nebo přepis řeči z reálných prostředí.

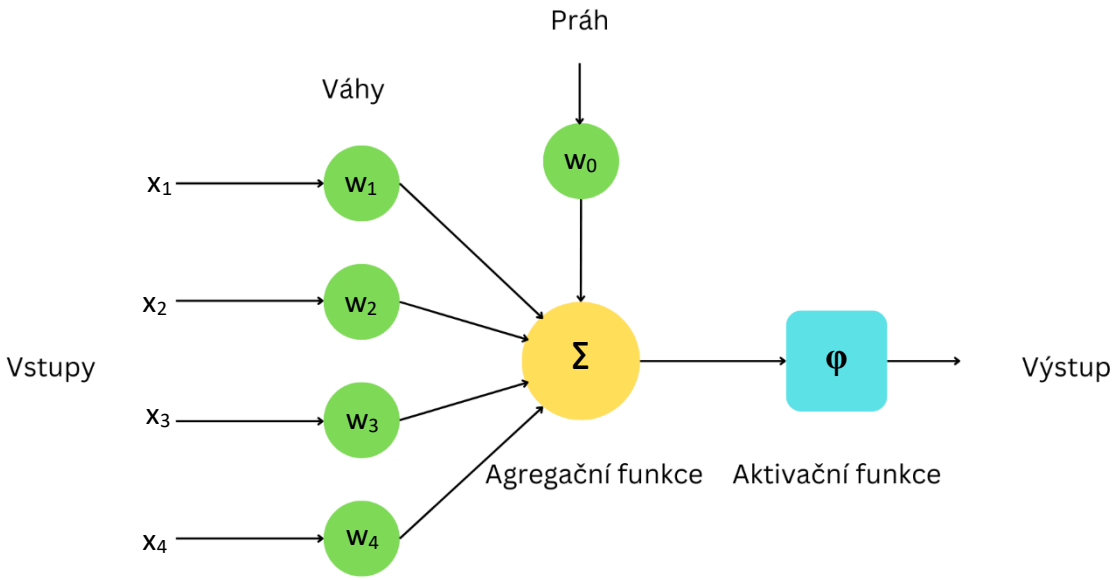
Navržený systém je rozdělen do tří fází: self-supervised předtrénování na rekonstrukční úloze, následný fine-tuning na konkrétní úlohy a detailní vyhodnocení pomocí relevantních metrik. Součástí práce je i porovnání s tradičními modely (CNN, RNN), analýza vlivu hloubky encoderu a diskuse o možnostech dalšího zlepšení.

Tato práce si klade za cíl nejen ověřit praktickou použitelnost transformerové architektury napříč modalitami, ale také přispět k diskusi o univerzálních přístupech ke zpracování signálů, které překračují hranice jednotlivých domén.

# 1 Neuronové sítě a extrakce vlastností

Umělé neuronové sítě (angl. Neural Networks – NN) jsou jednou z klíčových technologií v oblasti strojového učení a umělé inteligence (UI). Inspirace pro jejich struktury vychází ze struktur biologických neuronů, které tvoří lidský mozek. Tyto modely jsou navrženy tak, aby se byly schopny učit z dat a byly schopny obecně generalizovat, tedy aplikovat nabyté znalosti na nové problémy.

První matematický model neuronu byl představen McCullochem a Pittsem (1943), který popisoval neurony jako binární jednotky schopné logických operací. Neurony mají typicky několik vážených vstupů, práh neuronu a jediný výstup. Neurony jsou typicky uspořádány do vrstev, které jsou vzájemně propojeny a tvoří neuronové sítě. Na základě možnosti konfigurace vah a prahů jednotlivých neuronů podle trénovacích vzorů je poté možné danou neuronovou síť učit. Model takového neuronu, jak ho představili McCulloch a Pitts je na obrázku 1.1.




Obr. 1.1: Model umělého neuronu

Během dalších desetiletí výzkum pokračoval přes různé přístupy, jako je perceptron (Rosenblatt, 1958), který byl schopen učit se lineární klasifikaci, a algoritmus zpětného šíření chyby (Rumelhart et al., 1986), který otevřel cestu pro trénování hlubokých neuronových sítí. V poslední dekádě zažívají neuronové sítě dynamický rozvoj díky rostoucímu výpočetnímu výkonu, dostupnosti velkých dat a novým algoritmům, jako je Adam optimizer (Kingma and Ba, 2015) nebo metody regulace, například dropout (Srivastava et al., 2014), které pomáhají omezit přetrénování hlubokých modelů.

## 1.1 Principy učení neuronových sítí

### 1.1.1 Gradient descent

Gradient descent je jedním ze základních optimalizačních algoritmů používaných při učení neuronových sítí. Princip jeho fungování spočívá v iterativní aktualizaci vah modelu s cílem minimalizovat chybovou funkci, která měří rozdíl mezi předpovědí modelu a skutečnými hodnotami. Při každé iteraci se vypočítá gradient – tj. směr, ve kterém se funkce mění nejrychleji – a váhy jsou následně upraveny v opačném směru tohoto gradientu. Klíčovým parametrem je učící míra, která určuje velikost jednotlivých kroků při aktualizaci vah. Příliš nízká hodnota vede k pomalé konvergenci, zatímco příliš vysoká může způsobit nestabilitu a nedosažení globálního minima. Gradient descent se dá aplikovat v několika variantách, jako jsou standardní, stochastický nebo mini-batch gradient descent, přičemž každá varianta nabízí jiné kompromisy mezi rychlostí učení a přesností výpočtu gradientu.

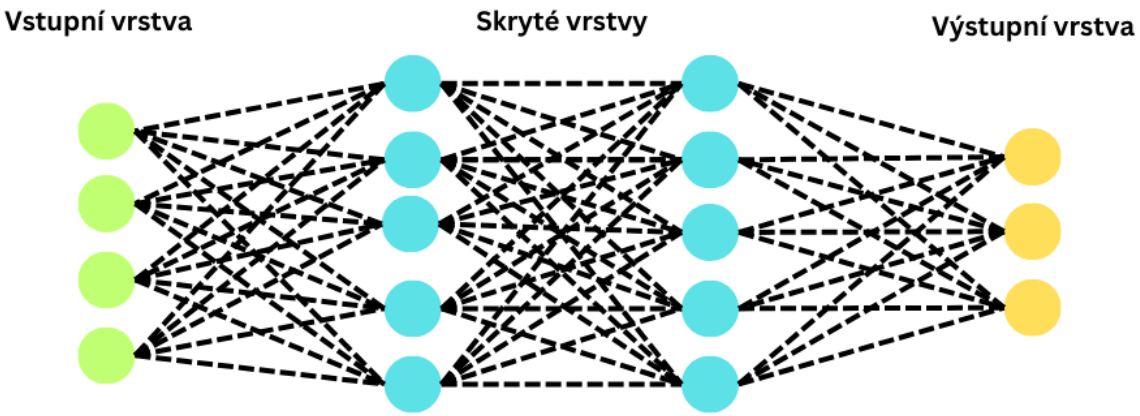
### 1.1.2 Backpropagation

Backpropagation je algoritmus, který umožňuje efektivní výpočet gradientů potřebných pro úpravu vah v neuronové síti a je úzce spojen s metodou gradient descent. Po průchodu dat sítí v tzv. feedforward fázi je na konci sítě vypočtena chyba, která se následně šíří zpětně skrze všechny vrstvy pomocí řetězového pravidla. Tím se pro každou váhu určí, jaký měla vliv na výslednou chybu, což je nezbytné pro přesnou aktualizaci těchto vah. Díky této metodě se umožňuje modelu adaptivně se učit z chyb, což výrazně zvyšuje efektivitu tréninku i u hlubokých a složitých sítí. Kombinace backpropagation a gradient descentu tak tvoří základ moderního učení neuronových sítí, jelikož umožňuje systémově a přesně optimalizovat modely tak, aby byly schopny zachytit a generalizovat složité vzory a nelinearity obsažené v datech.

## 1.2 Typy neuronových sítí

### 1.2.1 Dopředné neuronové sítě (Feedforward Neural Networks)

Dopředné sítě představují nejjednodušší typ neuronových sítí, ve kterých informace proudí jednosměrně ze vstupní vrstvy přes skryté vrstvy až do výstupní vrstvy. Tyto sítě jsou vhodné zejména pro základní klasifikační a regresní úlohy, jako je například predikce cen akcií nebo rozpoznávání obrazů. Díky své jednoduché architektuře jsou snadno pochopitelné a implementovatelné. Podle Bishopa (1995) představují dopředné sítě základní nástroj pro zpracování vzorců v rámci statistického rozpoznávání. Jejich nevýhodou je však omezená schopnost analyzovat složitější datové struktury nebo sekvenční data.



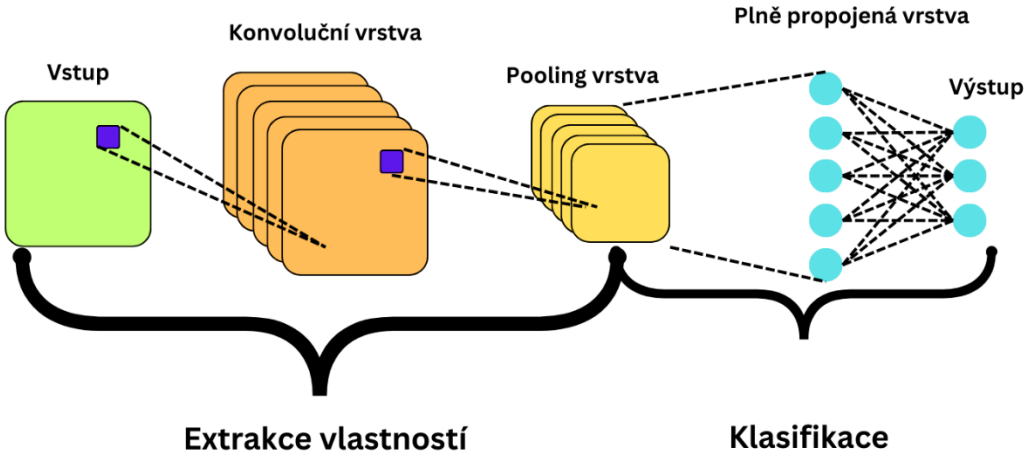

Obr. 1.2: Architektura dopředné neuronové sítě

### 1.2.2 Konvoluční neuronové sítě (Convolutional Neural Networks, CNN)

Konvoluční neuronové sítě jsou specializované na analýzu obrazových a prostorových dat. LeCun et al. (2015) ukazují, jak CNN umožňují efektivní extrakci prostorových vzorců díky svým konvolučním vrstvám, které identifikují specifické vzory, jako jsou hrany nebo textury. CNN se často využívají v oblasti počítačového vidění, například pro rozpoznávání objektů v obrazech (Kim, 2014). Typická architektura CNN zahrnuje následující vrstvy:

- **Konvoluční vrstvy:** Slouží k extrakci lokálních vzorů, jako jsou hrany nebo textury, pomocí kernelů (filtrů), které jsou optimalizovány během tréninku.
- **Pooling vrstvy:** Snižují rozměrnost dat při zachování klíčových informací. Často se využívají metody jako max-pooling nebo average-pooling.

- **Plně propojené vrstvy:** Transformují extrahované rysy na výstupy a umožňují finalizaci úloh, jako je klasifikace nebo regresní predikce.



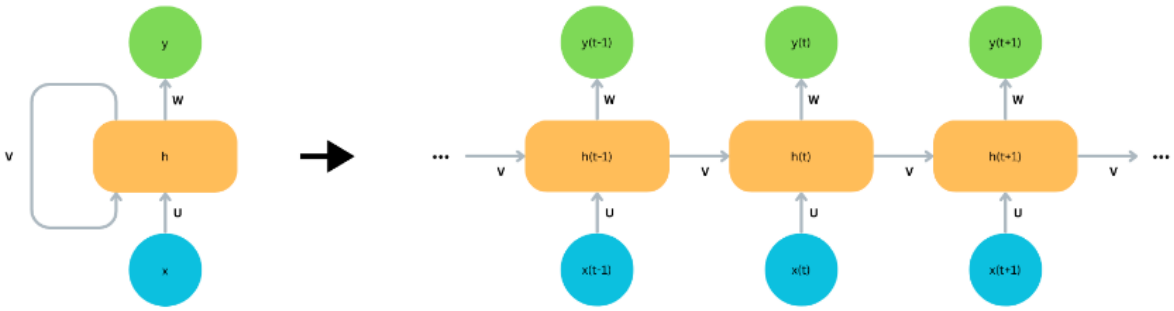

Obr. 1.3: Architektura konvoluční neuronové sítě

### 1.2.3 Rekurentní neuronové sítě (Recurrent Neural Networks, RNN)

RNN jsou především určeny pro práci se sekvenčními daty, jako je text, časové řady nebo zvukové signály. Díky svým rekurentním vlastnostem dokážou zohlednit kontext předchozích vstupů, což je klíčové při analýze datových sekvencí. Jak uvádějí Hochreiter a Schmidhuber (1997), RNN umožňují modelování sekvenčních závislostí díky svým skrytým stavům, které uchovávají informace o předchozích krocích. To je užitečné pro úlohy, jako je analýza textu nebo časových řad.

Varianta Long Short-Term Memory (LSTM), jak popisuje Graves (2012), řeší problémy se zachováváním dlouhodobých závislostí. LSTM implementuje paměťové buňky a brány (input, forget, output gates), které rozhodují, jaké informace si síť zachová nebo zapomene.

Gated Recurrent Units (GRU) jsou varianta rekurentních neuronových sítí, která byla navržena k řešení problémů s dlouhodobými závislostmi, které trápí tradiční RNN. GRU využívají dvě brány: aktualizační a resetovací, které určují, jakým způsobem se uchovávají a zapomínají informace mezi jednotlivými časovými kroky. Tento zjednodušený přístup oproti LSTM sítím snižuje výpočetní náročnost, ale stále efektivně zachovává dlouhodobé závislosti v sekvencích (Cho et al., 2014).




Obr. 1.4: Architektura rekurentní neuronové sítě

### 1.2.4 Transformery

Transformery jsou moderní architekturou zaměřenou na efektivní paralelní zpracování sekvencí dat. Jsou široce využívány v oblasti zpracování přirozeného jazyka (NLP) a dalších aplikacích, kde je potřeba zohlednit kontext v dlouhých sekvencích. Tento typ sítě bude podrobněji vysvětlen v kapitole 2.

## 1.3 Typy úloh pro neuronové sítě

V závislosti na typu úlohy se neuronová síť přizpůsobuje a učí identifikovat různé aspekty dat. Existují tři hlavní typy úloh, pro které jsou neuronové sítě často aplikovány: klasifikace, regrese a clustering (shlukování).

### 1.3.1 Shlukování (Clustering)

Shlukování je metoda, která se používá k rozdělení dat na skupiny, přičemž objekty ve stejné skupině jsou si navzájem podobné. V případě neuronových sítí se často používají autoenkodéry nebo rekurentní neuronové sítě k extrakci vzorců, které následně umožňují shlukování podobných vzorků. Tento proces je užitečný v případech, kdy je potřeba najít přirozené struktury v datech bez předchozích anotací, například při analýze obrazů, zvukových signálů nebo textu (Xu a Wunsch, 2005; Bengio et al., 2007).

### 1.3.2 Klasifikace

Klasifikační úloha se obvykle používá, když je cílem přiřadit vzorky dat do určitých kategorií. Například v rozpoznávání obrazů neuronová síť analyzuje obrazové vzory a rozhoduje, zda se na obrázku nachází kočka, pes, nebo jiný objekt. Modely, jako jsou konvoluční neuronové sítě (CNN), jsou velmi úspěšné v tomto typu úkolu, protože dokážou extrahovat a kombinovat různé vzory na několika úrovních detailů (Krizhevsky et al., 2017; Simonyan a Zisserman, 2015).

### 1.3.3 Regrese

Regresní úloha se používá, když je cílem předpovědět kontinuální hodnotu. Příklad zahrnuje predikci ceny nemovitosti na základě různých atributů, jako je lokalita, velikost domu nebo počet pokojů. Neuronové sítě pro regresi obvykle využívají hlubší vrstvy, které umožňují modelu zjistit komplexní vztahy mezi vstupy a výstupy, což je obzvlášť užitečné v oblasti ekonomických a environmentálních predikcí (Goodfellow et al., 2016).

### 1.3.4 Extrakce vlastností

Extrakce vlastností představuje proces identifikace a reprezentace klíčových charakteristik obsažených ve vstupních datech. Tato úloha může být realizována dvěma základními přístupy: manuální extrakcí příznaků a automatickou extrakcí příznaků pomocí hlubokých neuronových architektur, jako je například transformer.

Manuální extrakce vlastností zahrnuje explicitní definici a ruční výběr relevantních příznaků odborníkem. Tento přístup se uplatňuje zejména tehdy, jsou-li klíčové charakteristiky dat dobře známé, snadno definovatelné a jejich identifikace nevyžaduje příliš velké úsilí. Naopak automatická extrakce, která využívá schopnosti modelů hlubokého učení, je vhodnější v situacích, kdy by manuální identifikace a výběr příznaků byly příliš složité, časově náročné či dokonce nemožné.

Transformery představují efektivní příklad automatické extrakce vlastností, neboť jejich struktura založená na self-attention mechanismu umožňuje modelu dynamicky a adaptivně odhalovat složité a dlouhodobé vztahy v datech. Díky globálnímu charakteru attention mechanismu jsou transformery schopny zachytit jak jemné lokální detaily, tak i rozsáhlé globální závislosti. Tato schopnost je obzvláště užitečná pro analýzu sekvenčních a vícedimenzionálních dat, jako jsou časové řady, audio signály nebo zdravotnická data, kde je ruční specifikace relevantních příznaků obtížná.

Automatická extrakce vlastností pomocí transformerů je široce využívána pro analýzu dat v úlohách klasifikace, regrese a detekce anomálií. Díky schopnosti automaticky identifikovat relevantní vzory bez nutnosti ručního zásahu se výrazně zvyšuje efektivita, flexibilita a přesnost analýzy. Tato vlastnost je zásadní zejména v aplikacích průmyslového charakteru, ve zdravotnictví, ve finančních analýzách nebo v přírodovědných disciplínách, kde rozsáhlé manuální předzpracování dat není praktické či efektivní (Mikolov et al., 2013).

## 1.4 Typy učení v neuronových sítích

V oblasti strojového učení existují různé přístupy, jakým způsobem modely trénovat z dat. V závislosti na povaze úlohy a dostupných datech se vybírá odpovídající typ učení. Hlavními typy učení jsou:

### 1.4.1 Učení s učitelem (Supervised Learning)

Učení s učitelem je nejběžnější a nejznámější typ učení. Tento přístup spočívá v trénování modelu na historických datech, která obsahují vstupy a odpovídající správné výstupy (popisky, labely). Model se během trénování učí najít vzory mezi vstupy a výstupy, aby byl schopen správně předpovědět výstupy pro nové, neviděné vstupy (Goodfellow et al., 2016, s. 103–105). Příkladem může být úloha klasifikace, kde je z označených obrázků koček a psů cílem modelu naučit se rozpoznávat, zda obrázek obsahuje kočku nebo psa. Příkladem použití může být použití pro klasifikaci, např. rozpoznávání objektů na obrázcích (Krizhevsky et al., 2017, s. 84–87.) nebo pro regresi např. pro predikce ceny nemovitosti, jak uvádí Bishop (2006, s. 214–218).

### 1.4.2 Učení bez učitele (Unsupervised Learning)

Učení bez učitele je proces, při kterém model pracuje s daty, která neobsahují explicitní výstupy nebo labely. Cílem je odhalit strukturu nebo vzory v těchto datech. Model se učí zjišťovat podobnosti a rozdíly mezi jednotlivými datovými vzorky a organizovat je do skupin nebo dalších reprezentací (Bengio et al., 2007, s. 155). Tento typ učení se používá v úlohách, kde není možné nebo není efektivní anotovat všechna data. Příkladem použití může být segmentace zákazníků na základě jejich chování (Xu and Wunsch, 2005, s. 647) nebo redukce dimenzionality (např. PCA – Principal Component Analysis), kde se hledají nejdůležitější faktory v datech (Jolliffe, 2002, s. 67–71).

### 1.4.3 Autoenkodéry

Autoenkodéry jsou typem neuronových sítí využívaných v učení bez učitele, jejichž hlavním úkolem je naučit se kompaktní, latentní reprezentaci vstupních dat. Model se skládá ze dvou částí – enkodéru, který transformuje vstupní data do nižší dimenze, a dekodéru, jenž se snaží rekonstruovat původní data ze získané latentní reprezentace. Tento princip umožňuje automaticky odhalit nejdůležitější rysy dat a využít je například pro následnou klasifikaci, detekci anomálií či další analýzu. Díky své flexibilitě mohou být autoenkodéry dále rozšířeny o techniky, jako je přidání šumu (denoising autoenkodér) či použití učitelného CLS tokenu, což výrazně zlepšuje jejich schopnost zachytit složité časové závislosti a mezi-kanálové korelace,

zejména u dat s vysokou dimenzionalitou, jako jsou vícekanálové seizmické signály (Hinton a Salakhutdinov, 2006, s. 504–507).

### 1.4.4 Zpětnovazební učení (Reinforcement Learning)

Zpětnovazební učení je typ učení, který se zaměřuje na učení prostřednictvím interakce s prostředím. Model (agent) provádí akce a za každou akci dostává zpětnou vazbu ve formě odměn nebo trestů. Cílem je najít politiku (sérii akcí), která maximalizuje celkovou odměnu během interakcí s prostředím (Sutton and Barto, 2018, s. 43–45). Tento přístup je užitečný pro úkoly, kde je potřeba rozhodovat v sekvenci kroků a učit se z následků těchto rozhodnutí. Úloha byla s výhodou využita např. v modelu AlphaGo, kde se agent učí hrát hru Go, a dokonce v ní poráží nejlepší lidské hráče (Silver et al., 2017, s. 354–355) nebo v robotice, kde se robot naučí pohybovat a vykonávat úkoly v reálném světě (Kober et al., 2013, s. 1239–1242).

### 1.4.5 Kombinované učení (Semi-supervised Learning)

Semi-supervised learning učení spojuje aspekty učení s učitelem a bez učitele. V tomto přístupu model trénuje na malé množině označených dat a velké množině neoznačených dat. Tento typ učení je užitečný, když je obtížné nebo nákladné získat velké množství anotovaných dat, ale máme k dispozici velké množství neoznačených dat (Chapelle et al., 2010, s. 17–19). Semi-supervised learning využívá strukturu a vzory v neoznačených datech k vylepšení výkonu modelu. Tento typ učení lze použít ke zlepšení výkonu klasifikátorů, kdy je k dispozici pouze malé množství označených dat a mnoho neoznačených dat např. ve zpracování textu nebo analýze obrazů, jak ukazují Zhu a Goldberg (2009, s. 5–6).

Tato kapitola představila základní principy neuronových sítí, jejich architektury a úlohy, které mohou řešit. Byly popsány různé typy sítí, včetně konvolučních a rekurentních modelů, a jejich schopnosti pracovat s různými datovými strukturami. Zvláštní pozornost byla věnována úloze extrakce vlastností, která je klíčová pro tuto práci. Na tuto obecnou část navazuje podrobný rozbor architektury transformer, která v posledních letech přinesla zásadní pokrok v oblasti zpracování sekvenčních dat.

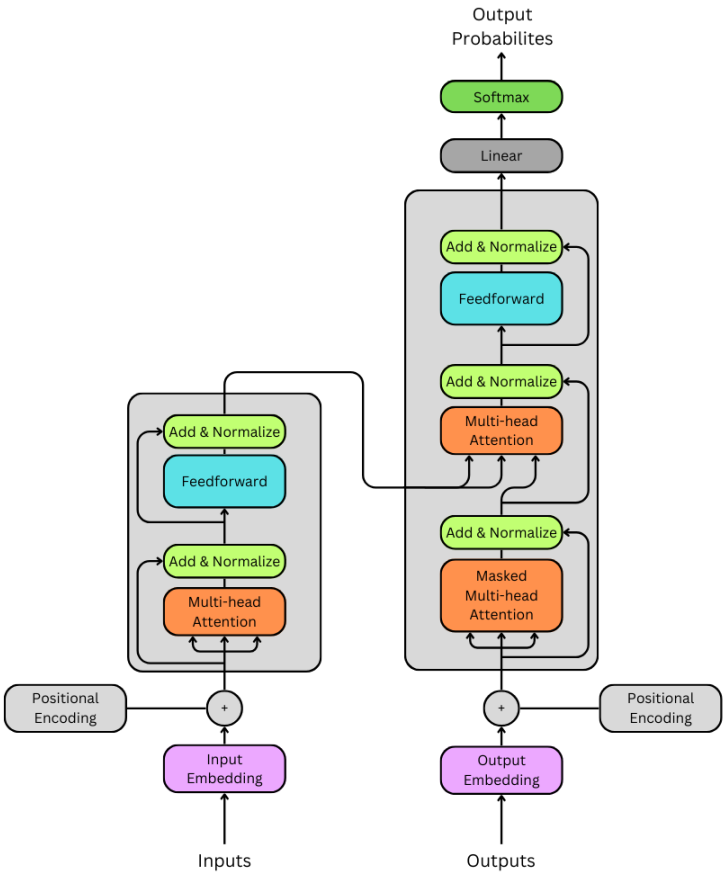
# 2 Modely typu transformer

Transformer architektura byla vyvinuta především s cílem zlepšit zpracování sekvenčních dat v porovnání s tradičními modely jako jsou RNN a LSTM. Tyto starší architektury měly problémy s dlouhými sekvencemi, jelikož zpracovávaly data sekvenčně, což vedlo k problémům s gradientním rozpadem a ztrátou dlouhodobé závislosti v datech

(Hochreiter & Schmidhuber, 1997). Transformer byl navržen tak, aby eliminoval potřebu sekvenčního zpracování a umožnil paralelní zpracování celé sekvence. Tento přístup nejenom že zrychlil trénování, ale také usnadnil práci s dlouhými sekvencemi, což je pro mnoho úloh, jako je strojový překlad nebo generování textu, klíčové (Vaswani et al., 2017).

## 2.1 Struktura transformer modelu

Transformer architektura se skládá z enkodéru a dekodéru, které mají zásadní úlohu v celkovém zpracování dat. Enkodér přijímá vstupní sekvenci a přetváří ji do latentního prostoru, který obsahuje všechny relevantní informace o datech. Tento latentní prostor pak slouží jako základ pro dekodér, který generuje výstupy na základě informací z enkodéru. Mechanismy jako self-attention a cross-attention, které jsou součástí jak enkodéru, tak dekodéru, umožňují modelům soustředit se na relevantní části vstupních a výstupních dat. Self-attention mechanismus zajišťuje, že každý prvek v sekvenci může věnovat pozornost všem ostatním prvkům ve stejné sekvenci, což modelu poskytuje globální kontext. Cross-attention zase pomáhá dekodéru se zaměřit na relevantní části výstupu z enkodéru při generování konečných výsledků (Vaswani et al., 2017). Obrázek 2.1 zobrazuje strukturu transformeru.




Obr. 2.1: Struktura transformer modelu

Díky těmto mechanismům je transformer schopen efektivně zpracovávat dlouhé sekvence a využívat paralelní zpracování, což výrazně urychluje trénování. To je jedna z hlavních výhod této architektury oproti předchozím přístupům. Navíc díky schopnosti modelu zohlednit globální kontext a vzory v datech jsou transformer modely schopné dosahovat vynikajících výsledků v široké škále úloh zpracování přirozeného jazyka, jako je strojový překlad, generování textu, sumarizace nebo analýza sentimentu (Devlin et al., 2019).

### 2.1.1 Enkodér

Enkodér v architektuře transformer představuje klíčovou komponentu pro zpracování vstupních dat a extrakci relevantních vlastností. Skládá se z několika po sobě jdoucích vrstev, přičemž každá z těchto vrstev obsahuje dvě hlavní části: mechanismus self-attention (pozornosti) a plně propojenou neuronovou síť (feedforward neural network).

Mechanismus self-attention umožňuje každé pozici v rámci vstupní sekvence dynamicky vážit důležitost ostatních částí vstupních dat na základě jejich relevance. Díky tomu je transformer schopen efektivně zachytit jak lokální, tak zejména globální závislosti napříč celou sekvencí. Tento mechanismus je obzvláště výhodný pro sekvenční data, jako jsou časové řady, text nebo signály, jelikož umožňuje modelu flexibilně se zaměřit na nejdůležitější části dat bez omezení tradiční rekurentní struktury, která může mít problémy s dlouhodobými závislostmi.

Každá vrstva enkodéru je vybavena normalizací vrstev (layer normalization) a zbytkovými spojeními (residual connections). Tato spojení usnadňují trénink hlubokých sítí tím, že minimalizují problémy jako mizení (vanishing) nebo explodování (exploding) gradientů.

Pro efektivní práci se sekvenčními daty využívá transformer poziční kódování (positional encoding), které modelu explicitně poskytuje informaci o pořadí jednotlivých prvků v sekvenci. Důvodem je, že samotný mechanismus pozornosti není citlivý vůči pořadí vstupních tokenů. Poziční kódování tedy zajistí, že transformer dokáže rozlišit pořadí vstupů a efektivně pracovat se sekvenčními daty.

V kontextu zpracování vícekanálových sekvenčních dat, jako jsou seismické signály, elektrokardiogramy nebo řečové nahrávky, se ukazuje, že často stačí využít pouze enkodér architektury transformer bez nutnosti dekodéru. Tento přístup je efektivní, neboť enkodér je schopen samostatně extrahovat komplexní a globálně relevantní vlastnosti ze vstupních dat, které lze následně využít pro různé downstreamové úlohy, jako jsou klasifikace nebo regrese (Vaswani et al., 2017).

### 2.1.2 Dekodér

Dekodér se používá k generování výstupu na základě zpracovaných informací z enkodéru. Stejně jako enkodér, se dekodér skládá z několika vrstev, přičemž každá vrstva obsahuje tři hlavní komponenty: mechanismus pozornosti (self-attention), mechanismus pozornosti mezi enkodérem a dekodérem (encoder-decoder attention) a plně propojenou neuronovou síť (feedforward neural network). Mechanismus pozornosti mezi enkodérem a dekodérem umožňuje dekodéru soustředit se na relevantní části vstupních dat z enkodéru při generování výstupu. Každá vrstva dekodéru má také zbytková spojení a normalizaci, což pomáhá k efektivnímu trénování modelu. Stejně jako v enkodéru, i v dekodéru se používá poziční kódování pro zachování informací o pořadí tokenů. Dekodér generuje výstup postupně, jeden token po druhém, a při generování každého nového tokenu se soustředí na relevantní části předchozích tokenů, čímž je schopen produkovat koherentní a relevantní výstupy na základě kontextu, který byl získán z enkodéru.

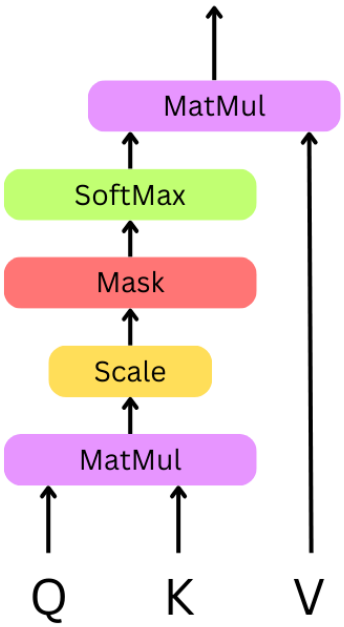
Při trénování transformeru se v dekodéru často používá technika maskování, která zajišťuje, že model nebude moci vidět budoucí slova nebo části sekvencí při generování výstupu. Tento mechanismus je nezbytný pro autoregresivní generování textu nebo sekvencí, kde model musí předpovědět slovo na základě předchozích slov, nikoli budoucích. Maskování zajišťuje, že při predikci určitého slova model používá pouze ta předchozí slova v sekvenci, což odpovídá procesu, který by měl být během generování použit v praxi.

Maskování v dekodéru je realizováno tak, že do matice pozornosti (attention matrix) se přidávají hodnoty, které blokují přístup k pozicím v sekvenci, které mají být "neviditelné". To je často implementováno jako metoda, která nahradí všechny hodnoty v těchto pozicích velmi negativními čísly, což způsobí, že jejich softmax váha je prakticky nulová.

Tento mechanismus je obzvláště důležitý při trénování modelů pro úkoly, jako je strojový překlad nebo generování textu, kde model musí generovat výstup krok za krokem a zároveň zohledňovat závislosti pouze na předchozím kontextu (Vaswani et al., 2017).

### 2.1.3 Self-attention Mechanismus

Základní operací v rámci self-attention je tzv. Scaled dot-product attention, který umožňuje modelu vážit různé části vstupní sekvence podle jejich relevance pro aktuální pozici v sekvenci. Tento mechanismus spočívá v tom, že pro každý token (nebo pozici) ve vstupu se spočítá podobnost mezi "query" (dotazem) a všemi ostatními "keys" (klíči). Výsledek této podobnosti určuje, jak velkou váhu bude mít daný token při generování výstupu. Tento proces je následně škálován podle dimenze klíčů, aby se předešlo příliš velkým hodnotám a stabilizoval trénink. Po tomto škálování se na hodnoty aplikuje softmax funkce, která normalizuje výsledky na pravděpodobnosti. Nakonec jsou tyto váhy použity k vytvoření váženého součtu hodnot "values" (V), což umožňuje modelu rozhodnout, které části vstupní sekvence jsou pro generování výstupu nejrelevantnější. Operace je zobrazena na obr. 2.2.




Obr. 2.2: Scaled Dot-product attention podle Vaswani et al. (2017)

Rovnice pro scaled dot-product attention je následující:

$$Attention(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.1}$$

kde $Q$ (queries) je matice dotazů,

$K$ (keys) – matice klíčů,

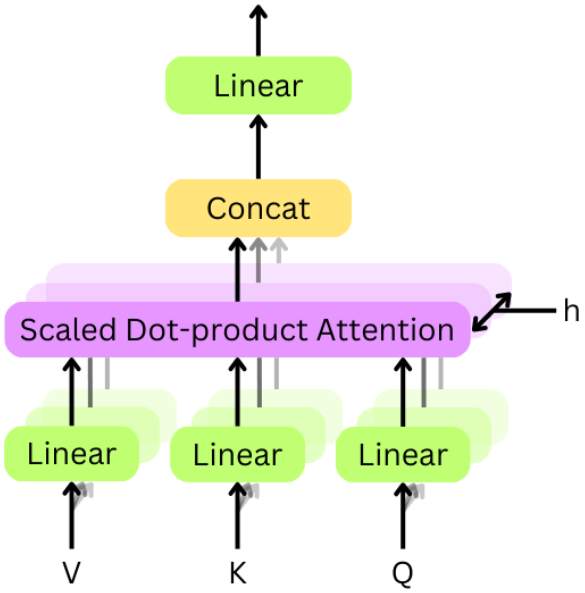
$V$ (values) – matice hodnot,

$d_k$ – rozměr klíčů,

softmax – funkce aplikovaná na jednotlivé řádky výsledné matice pro normalizaci na pravděpodobnosti.

### 2.1.4 Multi-head Attention

Multi-head attention je klíčovým prvkem architektury transformerů, který umožňuje modelu zachytit různé aspekty vztahů mezi prvky vstupní sekvence. Na rozdíl od klasického mechanismu self-attention, který provádí pouze jeden výpočet pozornosti, multi-head attention rozděluje vstupní data do několika nezávislých "hlav". Každá hlava využívá vlastní sadu transformací k vytvoření dotazů (queries), klíčů (keys) a hodnot (values) a provádí samostatný výpočet self-attention. Tento přístup umožňuje každé hlavě zaměřit se na odlišné charakteristiky sekvence, jako jsou krátkodobé a dlouhodobé závislosti nebo různé syntaktické a sémantické vazby mezi prvky. Výstupy jednotlivých hlav se následně spojí (concatenate) a projdou lineární transformací, která zajišťuje, že výsledná reprezentace kombinuje informace z různých perspektiv, čímž zvyšuje kapacitu modelu a jeho schopnost porozumět komplexním vztahům v datech. Tento mechanismus je zvláště efektivní při zpracování sekvenčních dat, jako je přirozený jazyk, kde jednotlivé části vstupu mohou mít odlišnou důležitost v různých kontextech. Koncept je zobrazen na obr. 2.3. (Vaswani et al., 2017; Khan et al., 2021).




Obr. 2.3: Multihead Attention podle Vaswani et al. (2017)

### 2.1.5 Poziční kódování (Positional Encoding)

Poziční kódování je nezbytnou součástí transformerů, protože jejich architektura nemá inherentní znalost o pořadí vstupních prvků, na rozdíl od rekurentních nebo konvolučních neuronových sítí. Pomocí pozičního kódování se do vstupní reprezentace přidávají informace o relativní nebo absolutní pozici prvků ve vstupní sekvenci. V NLP tyto kódy umožňují modelu porozumět pořadí slov v textu, což je klíčové pro správnou analýzu a interpretaci. Podobně při aplikaci transformerů na časově sekvenční data, jako jsou signály zpracovávané v doménách

akustiky, EEG nebo senzorických měření, poziční kódování zajistí, že model dokáže správně zohlednit časovou strukturu vstupu. Tradiční přístup používá sinusové a kosinusové funkce různých frekvencí, což umožňuje modelu extrapolovat na délky sekvencí, které během tréninku neviděl. V oblasti časově sekvenčních dat však mohou být poziční informace rozšířeny o doménově specifické přístupy, jako je kódování založené na časových znacích nebo využití frekvenčních spekter, což může dále zlepšit schopnost modelu zpracovávat složité časové závislosti. Matematicky:

$$PE(pos, 2i) = \sin\frac{pos}{10000^{\frac{2i}{d_{model}}}}, \tag{2.2}$$

$$PE(pos, 2i+1) = \cos\frac{pos}{10000^{2i/d_{model}}}, \tag{2.3}$$

kde *pos* je pozice v sekvenci,

$d_{model}$ – dimenze embeddingu.

## 2.2 Trénink a optimalizace

Trénink transformeru zahrnuje použití pokročilých optimalizačních metod, přičemž jednou z nejběžněji používaných metod je **Adam optimizer** (Adaptive Moment Estimation). Adam je rozšířením metody Stochastic Gradient Descent (SGD), která zohledňuje jak průměrné gradienty (momentum), tak i jejich variabilitu (momentum druhého řádu). To pomáhá optimalizátoru efektivněji upravovat váhy během trénování, což vede k rychlejší konvergenci a lepšímu výkonu, zejména v případě složitých modelů, jako jsou transformery. Zároveň lze pro efektivnější trénování transformerů využívat proměnlivou rychlost učení (learning rate), což může ještě zrychlit konvergenci modelu během trénování.

### 2.2.1 Dropout

Dropout je regularizační technika využívaná při tréninku neuronových sítí, která snižuje riziko přetrénování tím, že náhodně "vypíná" určitou část neuronů během jednotlivých tréninkových iterací. Tento přístup nutí síť, aby se nespoléhala pouze na určité neuronové cesty, ale rozvíjela robustnější a obecnější reprezentace dat. Konkrétně se dropout aplikuje s určitou pravděpodobností (například 0.5), což znamená, že přibližně polovina neuronů je náhodně deaktivována, a tím se snižuje závislost na specifických váhách. Výsledkem je model, který lépe generalizuje na nová, neviděná data, což je klíčové pro dosažení vyšší stability a přesnosti při predikci. Dropout se tak stává nedílnou součástí moderních optimalizačních technik, které přispívají k efektivnímu a robustnímu tréninku hlubokých neuronových sítí.

### 2.2.2 Adam optimizer

Adam používá dvě hlavní složky pro každý parametr modelu:

1. **První moment (m)**, což je exponenciálně vážený průměr gradientů. Tento moment pomáhá zajišťovat stabilitu a "setrvačnost" gradientu během trénování.
2. **Druhý moment (v)**, což je exponenciálně vážený průměr kvadrátů gradientů. Tento druhý moment umožňuje modelu lépe se vyrovnávat s nevyváženými gradienty a zajistit lepší adaptaci na různorodé charakteristiky dat.

Adam se tedy dynamicky přizpůsobuje rychlosti učení pro každý parametr na základě těchto momentů, což je výhodné pro trénování složitých modelů, jako je transformer, kde jsou parametry často velmi velké a gradienty se mohou lišit v závislosti na specifikách dat.

V praxi Adam optimizer pomáhá modelu rychleji konvergovat k optimálním parametrům, což je zvláště důležité při trénování složitých modelů, jako jsou transformery. Například, při trénování na textových datech Adam umožňuje modelu adaptivně upravovat váhy pro různé sloupce a vrstvy modelu, což vede k rychlejšímu a stabilnějšímu procesu učení. Tento přístup je však využíván nejen v NLP, ale i v dalších oblastech, jako je analýza časově sekvenčních dat, kde je potřeba zohlednit historické závislosti mezi časovými kroky. V takových případech Adam zajišťuje, že model se dokáže efektivně přizpůsobit i dynamickým změnám ve vstupních datech (Kingma a Ba, 2014; Reddi et al., 2018).

### 2.2.3 Learning rate scheduler

Learning rate schedulers jsou techniky používané k adaptivnímu měnění hodnoty učícího se kroku během trénování modelu. Učení s pevně nastaveným učícím krokem může být neefektivní, protože se rychlost učení nemění v závislosti na fázi trénování. Místo toho lze použít různé typy schedulerů, které umožňují dynamické přizpůsobování hodnoty učícího kroku.

Volba vhodného scheduleru závisí na typu úlohy, dostupných výpočetních prostředcích a dalších faktorech. Snižování rychlosti učení v průběhu trénování může pomoci modelu konvergovat k lepším výsledkům a předejít problémům, jako je přetrénování nebo pomalá konvergence (Losch a Hakkani-Tür, 2018; Gonzalez et al., 2020).

Nejběžnějšími typy learning rate schedulerů jsou:

1. **Step Decay**

Tento scheduler snižuje hodnotu učícího kroku po pevně stanovených intervalech, což znamená, že každých N epoch klesne rychlost učení o faktor. Tento přístup je jednoduchý

a efektivní v případech, kdy je potřeba snížit rychlost učení, jakmile se model přibližuje k optimálnímu řešení.

Rychlost učení $\eta$ se mění podle vzorce:

$$\eta(t) = \eta_0 \cdot \gamma^{\lfloor t/T_{\text{step}} \rfloor}, \tag{2.4}$$

kde $\eta_0$ je počáteční hodnota učícího kroku,

$\gamma$ – je faktor snižování,

$T_{\text{step}}$ – je interval epoch, po kterém se rychlost učení sníží.

2. **Exponential Decay**

Exponenciální snižování snižuje učící krok exponenciálně s časem, což znamená, že rychlost učení klesá stále rychleji, jak trénování pokračuje. Tento přístup se používá pro jemné ladění modelu v pozdějších fázích trénování.

Vzorec pro tento přístup je:

$$\eta(t) = \eta_0 \cdot e^{-\lambda t}, \tag{2.5}$$

kde $\eta_0$ je počáteční učící krok,

$\lambda$ – rychlost poklesu,

$t$ – aktuální epocha.

3. **Cosine Annealing**

Cosine annealing postupně snižuje učící krok podle kosinové funkce, což znamená, že rychlost učení začíná vysokou a postupně klesá na velmi nízkou hodnotu, přičemž tento proces probíhá po určitém počtu epoch. Tento přístup je efektivní v případě, kdy je potřeba postupně zjemnit učení, aby model mohl efektivněji najít globální optimum.

Vzorec pro tento scheduler je:

$$\eta(t) = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})\left(1 + \cos\left(\frac{t}{T}\pi\right)\right), \tag{2.6}$$

kde $\eta_{\max}$ je maximální učící krok,

$\eta_{\min}$ – minimální krok,

$T$ – celkový počet epoch.

4. **Warm-up**

Warm-up scheduler začíná s nízkou hodnotou učícího kroku a postupně jej zvyšuje až do stanovené hodnoty, než začne aplikovat další způsob snižování učícího kroku, jako je step decay nebo cosine annealing. Warm-up je užitečný zejména na začátku trénování, kdy je model ještě daleko od optimálního řešení a může být náchylný k příliš agresivnímu učení, což může vést k nestabilitě.

Výpočet učení během warm-upu může vypadat následovně:

$$\eta(t) = \eta_0 \cdot \frac{t}{T_{\text{warm-up}}}, \tag{2.7}$$

kde $T_{\text{warm-up}}$ je počet epoch pro warm-up fázi.

5. **Learning rate on plateau**

Learning rate on plateau je technika, která dynamicky snižuje hodnotu učícího kroku, když se zlepšení výkonu modelu během trénování zastaví nebo výrazně zpomalí. Místo pevného schématu snižování, jako je step decay či exponenciální pokles, tento přístup monitoruje metriky (například validační ztrátu) a pokud se tyto metriky nezlepšují po stanoveném počtu epoch, automaticky redukuje učící krok o předem definovaný faktor. Takové snížení rychlosti učení pomáhá modelu unikat z potenciálních lokálních minim a zároveň zabraňuje přetrénování, protože menší učící krok umožňuje jemnější ladění parametrů v pozdějších fázích tréninku. Použití learning rate on plateau je výhodné zejména v situacích, kdy model dosáhne dočasného stagnování, což indikuje, že aktuální rychlost učení je příliš vysoká na to, aby umožnila další efektivní zlepšení. Tento adaptivní přístup tak optimalizuje proces tréninku tím, že automaticky reaguje na změny v chování modelu a přispívá k dosažení lepší konvergence.

## 2.3 Aplikace transformeru

Transformer se ukázal jako velmi univerzální model, který našel široké uplatnění v mnoha oblastech. Například v oblasti strojového překladu se stal základem moderních systémů, jako je Google Translate či DeepL, kde je využíván k přesnému překladu textů mezi různými jazyky. Tento model dokáže lépe zachytit složité jazykové vzory a kontext, což vede k přesnějším překladům ve srovnání s tradičními metodami. Tuto skutečnost potvrzuje práce autorů Vaswani et al. (2017), kteří představili původní architekturu transformeru, jež se stala revolucí v oblasti zpracování přirozeného jazyka.

Další významnou aplikací transformeru je generování textu. Modely jako GPT (Generative Pretrained Transformer), vyvinuté OpenAI, využívají transformer pro generování textu na základě zadaného promptu. Podle Browna et al. (2020) se tyto modely vyznačují schopností vytvářet koherentní a relevantní texty na základě malého množství vstupních dat, což ukazuje jejich schopnost "rozumět" širokému spektru témat.

Transformer našel uplatnění i v oblasti zpracování obrazu, konkrétně v modelu Vision Transformer (ViT), který je používán pro klasifikaci obrázků. Jak uvádí Dosovitskiy et al. (2020), ViT ukazuje, že transformer může být efektivní i pro úkoly, které byly tradičně

doménou konvolučních neuronových sítí (CNN), což otevřelo nové možnosti pro analýzu vizuálních dat.

V oblasti zpracování zvuku se transformer aplikoval na analýzu zvukových signálů, například při rozpoznávání řeči. Modely jako wav2vec 2.0, jak popisují Baevski et al. (2020), využívají transformer pro efektivní extrakci rysů z audio signálů a jejich následné přetvoření na text.

K predikci dat z časových řad se uplatnil model Informer, který podle Zhua et al. (2021) využívá mechanismus pozornosti pro efektivní práci s velkými časovými daty.

Využití transformerů se rozšířilo také na zpracování vícekanálových časových dat, což zahrnuje například signály z EEG, multikanálových senzorů nebo fyziologických měření. Díky mechanismu pozornosti dokáže transformer efektivně modelovat jak prostorové, tak časové závislosti mezi jednotlivými kanály. Tento přístup umožňuje detailnější analýzu a přesnější extrakci relevantních rysů z těchto složitých dat. Příkladem je studie LI et al. (2023), která představuje model Convolutional Transformer s mechanismem multi-head attention. Tento model kombinuje výhody konvolučních neuronových sítí a transformerů a uplatňuje se při klasifikaci fází spánku na základě vícekanálových signálů. Studie zdůrazňuje, že tato architektura umožňuje efektivní propojení a analýzu datových korelací mezi různými kanály, což z ní činí univerzální nástroj pro analýzu fyziologických signálů.

V této kapitole byla vysvětlena struktura modelu transformer, včetně jeho klíčových komponent – mechanismu pozornosti, multi-head attention, pozičního kódování a způsobu trénování. Popsány byly také důvody, proč se tato architektura ukázala jako výjimečně účinná pro učení dlouhodobých závislostí. V následující kapitole bude pozornost přesunuta na praktické aspekty využití neuronových sítí, konkrétně na softwarové frameworky, které umožňují jejich implementaci a trénování.

# 3 Frameworky

Framework je softwarové prostředí, které poskytuje předem definované nástroje, knihovny a rozhraní, umožňující vývojářům snadno implementovat a experimentovat s různými algoritmy a modely, včetně neuronových sítí. Díky těmto abstrakcím a modulárním komponentám se odpadá potřeba psát základní matematické operace a správu datových struktur od nuly, což značně zjednodušuje proces vývoje. Frameworky, jako například TensorFlow či PyTorch, nabízejí nejen hotové funkce pro implementaci neuronových sítí, ale také robustní nástroje pro ladění, vizualizaci a monitorování tréninkového procesu. Tímto způsobem

podporují rychlý vývoj a experimentování, přičemž usnadňují integraci optimalizačních algoritmů a dalších pokročilých metod. Celkově lze říct, že framework představuje stavební kámen moderního softwarového vývoje, který zajišťuje efektivitu, opakovatelnost a konzistenci implementovaných řešení.

## 3.1 Využití

V procesu vývoje neuronových sítí hraje framework klíčovou roli nejen při samotné implementaci modelů, ale také v optimalizaci využití výpočetních zdrojů a podpoře experimentální práce. Díky své flexibilitě umožňuje framework snadnou integraci s různými hardwarovými platformami, jako jsou GPU a TPU (tensor processing unit), což značně zvyšuje výkon a škálovatelnost výpočtů. Uživatelé tak mají možnost rychle testovat nové architektury a optimalizovat modely bez zdlouhavého nastavování a konfigurace. Standardizované rozhraní frameworků navíc v akademickém i pracovním prostředí usnadňuje spolupráci v týmech, sdílení kódu a opakování experimentů, což je klíčové pro vývoj a ověřování nových přístupů v oblasti strojového učení. V konečném důsledku frameworky zvyšují produktivitu a kvalitu výsledných modelů, což přispívá k celkovému pokroku v oblasti neuronových sítí.

Při výběru frameworku je třeba zvážit i řadu kritérií, která zajistí jeho vhodnost pro konkrétní projekt. Mezi ty patří například uživatelská přívětivost a kvalita dokumentace, což usnadňuje jak samotný vývoj, tak pozdější údržbu a rozšiřování aplikace. Dále je důležitá flexibilita a možnost integrace s dalšími knihovnami a nástroji, což umožňuje rychlé experimentování a iteraci modelů. Nezanedbatelným kritériem zůstává také výkon frameworku, jeho schopnost efektivně pracovat s rozsáhlými datovými sadami, a kompatibilita s různými výpočetními platformami, které dohromady přispívají k robustnosti a dlouhodobé udržitelnosti vyvíjených řešení.

## 3.2 Frameworky používané pro neuronové sítě

Mezi nejvýznamnější frameworky pro implementaci neuronových sítí patří TensorFlow, PyTorch a Keras. Tyto knihovny tvoří základní stavební kámen většiny současných výzkumných i produkčních aplikací v oblasti hlubokého učení.

**TensorFlow**, vyvinutý společností Google, je robustní open-source platforma, která umožňuje efektivní implementaci a škálování modelů. Díky podpoře výpočtů na GPU a TPU je vhodný jak pro výzkumné experimenty, tak pro produkční nasazení složitých modelů. Jeho rozsáhlý ekosystém zahrnující TensorBoard (pro vizualizaci tréninkových metrik), TensorFlow Serving (pro nasazení modelů) a TFLite (pro mobilní zařízení) přispívá k jeho popularitě. Dále

má bohatou dokumentaci a velmi aktivní komunitu, což usnadňuje řešení problémů v oblasti vývoje a nasazení strojového učení.

**PyTorch**, vyvíjený firmou Meta (dříve Facebook), si získal velkou oblibu především díky svému dynamickému výpočetnímu grafu (define-by-run). Tento přístup umožňuje flexibilní, intuitivní a čitelnou konstrukci neuronových sítí, což výrazně usnadňuje ladění, ladění chyb a prototypování nových architektur. PyTorch je velmi oblíbený ve výzkumných kruzích, kde je vyžadována experimentální svoboda a přímá kontrola nad výpočtem. Významným přínosem PyTorche je také jeho jednoduchá integrace s knihovnami jako NumPy, Scikit-learn nebo Pandas, a podpora paralelního výpočtu na GPU bez nutnosti explicitních deklarací. Díky knihovnám jako TorchVision, TorchAudio a PyTorch Lightning lze navíc velmi snadno vytvářet projekty v oblasti počítačového vidění, zpracování zvuku či přehledně spravovat tréninkový cyklus modelu.

V této práci byl zvolen právě framework **PyTorch**, a to především pro jeho flexibilitu, přehlednost a schopnost snadno implementovat vlastní architektury modelů, což je klíčové pro návrh specifického vícekanálového extraktoru vlastností. V praktické části této diplomové práce bude PyTorch využit pro celý tréninkový a vyhodnocovací proces, včetně předzpracování dat, návrhu architektury transformeru a ladění modelových hyperparametrů. Jeho podpora GPU výpočtů zároveň umožňuje efektivní trénink i na běžném desktopovém hardwaru s jednou grafickou kartou.

**Keras**, původně samostatná knihovna a nyní součást TensorFlow, představuje vysoce abstraktní rozhraní, které zjednodušuje tvorbu neuronových sítí pomocí intuitivního a modulárního API. Je oblíbený mezi začátečníky a vývojáři, kteří ocení jednoduchost a rychlost tvorby prototypů bez nutnosti hluboké znalosti nízkoúrovňových operací. I když nabízí menší flexibilitu oproti PyTorchi, je ideální pro aplikace, kde není třeba detailní kontrola nad výpočetním tokem.

Společně tyto frameworky pokrývají široké spektrum potřeb – od rychlého experimentování a návrhu nových architektur až po robustní a efektivní nasazení modelů v produkčním prostředí. Výběr konkrétního nástroje závisí zejména na charakteru projektu, požadavcích na rychlost vývoje, dostupných výpočetních prostředcích a preferencích vývojářů.

# 4 Vyhodnocování kvality u neuronových sítí

Prvním a zásadním krokem při vyhodnocování kvality neuronových sítí je správné rozdělení dostupných dat do tří základních množin: tréninkové, validační a testovací. Tréninková množina slouží k učení modelu, kde se pomocí algoritmů, jako je gradient descent

a backpropagation, optimalizují váhy sítě. Validační množina je využívána k ladění hyperparametrů a monitorování výkonu během tréninku, což umožňuje včasné odhalení problémů, jako je přetrénování (overfitting). Nakonec testovací množina poskytuje objektivní zhodnocení modelu na neviděných datech, čímž se ověřuje schopnost modelu generalizovat získané znalosti na reálné případy.

Při vyhodnocování kvality neuronových sítí je nutné zaměřit se na metody a postupy, které umožňují měřit a interpretovat výkonnost vytrénovaných modelů. V první řadě je nezbytné rozlišit mezi interním a externím hodnocením. Interní evaluace využívá metriky jako jsou ztrátová funkce (loss), přesnost (accuracy), precision, recall či F1-skóre u klasifikačních modelů, případně střední kvadrát chyby (MSE), střední absolutní chybu (MAE) nebo koeficient determinace ($R^2$) u regresních modelů. Kromě těchto základních metrik hraje důležitou roli také analýza matic záměn (confusion matrix), ROC křivky a oblast pod křivkou (AUC), které pomáhají odhalit slabiny modelu, například při nerovnováze tříd. Základním krokem Díky těmto metodám lze identifikovat jevy jako přetrénování (overfitting) či nedotrénování (underfitting) a na jejich základě upravovat architekturu nebo ladit hyperparametry modelu.

Externí evaluace se pak zaměřuje na aplikaci modelu v reálných podmínkách a jeho adaptabilitu na nové situace. Kromě standardních metrik se často využívá křížové validace (cross-validation), která poskytuje robustnější odhad modelové generalizace a odhaluje variabilitu výsledků při různých náhodných děleních dat. Důležitým aspektem je rovněž interpretabilita modelu, která pomáhá pochopit, jaké rysy vstupních dat mají největší vliv na výslednou předpověď. V současnosti se rovněž testuje robustnost modelů vůči adverzariálním útokům, kdy se záměrně generují vstupy s malými, avšak cílenými úpravami, aby se otestovala odolnost sítě. Celkově vyhodnocování kvality neuronových sítí představuje komplexní proces, který kombinuje kvantitativní měření s kvalitativní analýzou, a je klíčovým prvkem pro úspěšný vývoj a nasazení modelů strojového učení v praxi.

Kapitola poskytla přehled nejpoužívanějších frameworků pro vývoj neuronových sítí, zejména TensorFlow, Keras a PyTorch. Byla zdůrazněna flexibilita a výhody PyTorche, který byl v této práci použit díky své podpoře dynamických výpočetních grafů a intuitivnímu ladění modelů. Po vymezení teoretického a nástrojového zázemí se dále zaměříme na samotnou implementaci navrženého řešení, počínaje popisem použitých datových sad.

## 4.1 Metriky pro evaluaci kvality

Pro objektivní a transparentní hodnocení modelů se pro úlohy umělé inteligence používají následující standardní metriky:

**Přesnost (Accuracy)** Přesnost udává poměr správně klasifikovaných případů vůči celkovému počtu klasifikovaných případů. Vypočítává se jako:

$$\text{Přesnost} = \frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + TN + FP + FN}}, \tag{4.1}$$

kde TP (True Positive) jsou správně klasifikované pozitivní případy,

TN (True Negative) – správně klasifikované negativní případy,

FP (False Positive) – chybně klasifikované pozitivní případy,

FN (False Negative) – chybně klasifikované negativní případy.

Přesnost je vhodná zejména u vyvážených datasetů, avšak může být zavádějící při výrazně nevyvážených datech.

**F1 skóre**

F1 skóre kombinuje preciznost (precision) a citlivost (recall). Tato metrika je zvlášť vhodná pro nevyvážené datasety a poskytuje komplexnější pohled na výkonnost klasifikátoru:

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{4.2}$$

kde

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{4.3}$$

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{4.4}$$

**AUC (Area Under ROC Curve)**

AUC představuje plochu pod ROC (Receiver Operating Characteristic) křivkou. ROC křivka graficky znázorňuje vztah mezi citlivostí (True Positive Rate, TPR) a mírou falešně pozitivních klasifikací (False Positive Rate, FPR) při různých prahových hodnotách. Hodnota AUC se pohybuje od 0 do 1, kde:

- **AUC = 0,5** znamená, že klasifikátor má výkon srovnatelný s náhodným tipováním.
- **AUC blížící se 1** označuje velmi dobrý klasifikátor schopný efektivně rozlišit pozitivní a negativní případy.
- **AUC pod 0,5** signalizuje horší výkon než náhodné tipování a znamená nesprávné přiřazování tříd.

$$\text{TPR (citlivost)} = \frac{TP}{TP + FN}, \tag{4.5}$$

$$\text{FPR} = \frac{FP}{FP + TN}. \tag{4.6}$$

ROC křivka vzniká změnou prahu klasifikace a vykreslením závislosti mezi citlivostí (TPR) a mírou falešně pozitivních klasifikací (FPR). ROC křivka poskytuje komplexní pohled na chování modelu napříč různými prahy a umožňuje najít optimální bod rozhodování podle specifických požadavků na minimalizaci chyby určitého typu.

**MSE (Mean Squared Error)**

Mean Squared Error (MSE) se používá při regresních úlohách a měří průměrnou kvadratickou odchylku mezi skutečnými hodnotami a hodnotami predikovanými modelem:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2, \tag{4.7}$$

kde $y_i$ je skutečná hodnota,

$\hat{y_i}$ – predikovaná hodnota modelem,

$n$ – počet vzorků.

Nízká hodnota MSE značí vyšší přesnost predikcí.

## 4.2 Ztrátové funkce

Ztrátová funkce (loss function) představuje základní složku trénovacího procesu neuronové sítě, neboť kvantifikuje rozdíl mezi predikcí modelu a skutečnou hodnotou (label). Optimalizační algoritmus poté iterativně upravuje váhy modelu tak, aby byla hodnota této funkce minimalizována. V této práci bylo použito několik různých ztrátových funkcí v závislosti na povaze konkrétní úlohy a typu výstupu modelu:

- **Binary Cross-Entropy (BCE)**

Použita pro binární klasifikaci, např. při rozlišování mezi normálním a abnormálním EKG záznamem nebo zemětřesením a šumem v seismických datech. V základní podobě má tvar:

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)], \tag{4.8}$$

kde $y_i \in \{0,1\}$ je skutečná třída,

$p_i \in \{0,1\}$ – predikovaná pravděpodobnost.

Pro nevyvážená data byla použita vážená verze BCEWithLogitsLoss, která navíc dovnitř vkládá sigmoid aktivaci a zavádí vahový parametr pos_weight.

- **Křížová entropie s měkkým značením (KL divergence + entropy regularizace)**

Pro úlohy přesné lokalizace časových událostí (např. pickování fází nebo detekce QRS komplexů) byla použita kombinace klasické CrossEntropyLoss s KLDivLoss (Kullback a Leibler, 1951), která porovnávala predikované rozdělení pravděpodobnosti s předem připravenými „měkkými“ (soft) labely ve formě Gaussových křivek. Tato metoda umožňuje lépe zohlednit nejistotu v přesné poloze událostí.

KL divergence mezi predikovaným rozdělením $q$(x) a skutečným rozdělením $p$(x) je definována jako:

$$\mathcal{L}_{\mathrm{K}} = \sum_x p(x) \log\left(\frac{p(x)}{q(x)}\right). \tag{4.9}$$

Celková ztráta byla pak kombinací všech KL divergence a CrossEntropyLoss.

- **Smooth L1 loss (Huberova ztráta)**

Tato funkce byla využita při rekonstrukčních úlohách během self-supervised předtrénování. Je robustní vůči odlehlým hodnotám a kombinuje výhody MSE a MAE, což ji činí vhodnou pro úlohy, kde model rekonstruuje části signálu nebo spektrogramu (Goodfellow et al., 2016)

$$\mathcal{L}_{\mathrm{SmoothL1}}(x) = \begin{cases} 0{,}5x^2, & x < 1 \\ |x| - 0{,}5, & x \geq 1 \end{cases}, \tag{4.10}$$

kde $x$ je rozdíl mezi predikcí a cílem.

- **CTC Loss (Connectionist Temporal Classification)**

Použita při rozpoznávání řeči. Tato ztrátová funkce umožňuje trénování modelu bez nutnosti přesného zarovnání mezi vstupním signálem a výstupní sekvencí znaků, což je ideální pro proměnlivou délku slov a jejich výskyt v čase (Graves et al., 2006).

$$\mathcal{L}_{CTC} = -\log P\,(y|x), \tag{4.11}$$

kde $x$ je vstupní sekvence (např. mel-spektrogram),

$y$ – cílový text.

Každá z těchto funkcí byla zvolena s ohledem na povahu úlohy a typ výstupu, který měl model generovat. Jejich správná volba a případné kombinace výrazně ovlivnily úspěšnost tréninku a dosažené výsledky.

Následující kapitola se zaměří na praktické aspekty zpracování dat, konkrétně na předzpracování vícero typů časových signálů před jejich vstupem do neuronových sítí.

# 5 Předzpracování dat pro extrakci vlastností

Předzpracování dat je klíčovým krokem mezi získáním surových multikanálových časových dat a jejich následnou analýzou pomocí strojového učení. Tento proces začíná správnou interpretací a dekódováním binárních souborů, čímž se extrahují jednotlivé kanály a další důležité informace. Následně se provádí normalizace, přičemž se často využívá kanálová min–max normalizace, která sjednocuje rozsahy hodnot napříč kanály a zabraňuje tak nepřiměřeným rozdílům, které by mohly ovlivnit další analýzu. Předzpracování tedy transformuje komplexní a vysoce dimenzionální surová data do formy, která efektivně reprezentuje klíčové vlastnosti signálu, a tím umožňuje přesnější a robustnější modelování z různých typů dat.

## 5.1 Parsing a normalizace

Parsing a normalizace představují první fázi předzpracování surových dat, ve které je nutné data správně interpretovat a připravit pro další analýzu. Dekódování binárních souborů umožňuje převést surová data do čitelných a strukturovaných formátů, kde jsou jasně identifikovány jednotlivé kanály, hlavičky a datové bloky. Následná normalizace, zejména kanálová min–max normalizace, převádí hodnoty každého kanálu do jednotného rozsahu (například [0, 1]). Tento postup zajišťuje, že rozdíly v amplitudách, které mohou vzniknout vlivem odlišného zesílení nebo měřicích podmínek, neovlivní výsledky následných analýz, a zároveň přispívá k lepší stabilitě a rychlejší konvergenci při trénování modelů.

## 5.2 Segmentace signálů

Segmentace signálů spočívá v rozdělení dlouhých záznamů na menší, časově konzistentní úseky, tzv. okna, což usnadňuje další analýzu a extrakci příznaků. Klíčovým aspektem této metody je volba vhodné délky oken, která musí být dostatečně dlouhá na to, aby zachytila podstatné dynamické vlastnosti signálu, a zároveň dostatečně krátká, aby umožnila detailní analýzu. Často se využívají překryvy mezi sousedními okny, které zajišťují, že se nepřijdou o důležité informace na hranicích segmentů a signál je kontinuálně pokryt. Překryvy navíc pomáhají vyhlazovat přechody mezi okny, což zvyšuje konzistenci dat a usnadňuje identifikaci klíčových vzorců, jako jsou změny v amplitudě či frekvenční charakteristiky. Správná volba velikosti oken přispívá k efektivitě modelu.

## 5.3 Extrakce příznaků

Extrakce příznaků je krok, při kterém se z předzpracovaných segmentů dat získávají klíčové informace potřebné pro další analýzu a modelování. Tento proces zahrnuje výpočet statistických ukazatelů, jako jsou průměr, rozptyl, medián, zkosení a špičatost, které poskytují

základní popis distribuce hodnot v jednotlivých oknech signálu. Statistické příznaky mohou být důležité pro identifikaci anomálií či změn ve sledovaných procesech. Kromě toho se extrahují také frekvenční charakteristiky, získané například Fourierovou transformací, které odhalují dominantní frekvence, energetické spektrum a fázové posuny. V některých případech jsou tyto příznaky dále standardizovány, aby měly normovanou škálu, což zajišťuje stabilnější a rychlejší konvergenci při trénování modelů strojového učení. Výsledkem této extrakce příznaků jsou poté příznaky (labely), které ke každému oknu (segmentu surových dat) přidávají dodatečné kontextuální informace, a které mohu být následně použity pro trénování neuronových sítí.

## 5.4 Relevantní vlastnosti

V oblasti strojového učení, zejména při práci s komplexními daty, jako jsou multikanálové seizmické signály, je extrakce relevantních znaků klíčová pro dosažení kvalitních výsledků. Tradiční přístup spočívá v ručním výběru vlastností na základě odborných znalostí. Mezi takové vlastnosti patří například amplituda, energie signálu, spektrální obsah a statistické charakteristiky jako průměr, rozptyl, šikmost či špičatost. Tyto ručně navržené znaky umožňují expertnímu pracovníkovi interpretovat výsledky a přímo propojit naměřené hodnoty s fyzikálními jevy či geologickými strukturami (Yilmaz, 2001).

Na druhé straně, auto enkodéry umožňují automatickou extrakci latentních znaků přímo z dat. Například použitím transformerové architektury může model zachytit složité časové závislosti a mezi-kanálové korelace, aniž by bylo třeba explicitně definovat, které znaky jsou relevantní. Latentní reprezentace, získané prostřednictvím autoenkodéru, nejsou tak přímo interpretovatelné jako ručně volené znaky, avšak často obsahují komplexní a jemné informace, které manuální přístup opomíjí (Goodfellow et al., 2016; Vaswani et al., 2017).

Kombinace obou přístupů může přinést významné výhody. Ručně inženýrské vlastnosti poskytují jasnou interpretaci a mohou sloužit jako silný základ pro klasické metody analýzy, zatímco latentní znaky získané z autoenkodéru přinášejí datově řízený pohled, který je schopen zachytit nelineární a komplexní vztahy v datech. Taková kombinace umožňuje vytvořit robustnější modelovací rámec, který je lépe přizpůsobený reálným podmínkám seizmických měření a může vést k lepším výsledkům v úlohách, jako je klasifikace nebo detekce anomálií (Bishop, 2006; Goodfellow et al., 2016; Vaswani et al., 2017).

## 5.5 Rozdělení dat

Každá datová sada se pro použití v hlubokých modelech typicky rozdělí na 60–80 % trénovacích vzorů, 10–20 % validačních a 10–20 % testovacích. Trénovací sada slouží k učení modelu, validační sada k ladění hyperparametrů a detekci přeučení a testovací sada k finálnímu,

nezaujatému hodnocení. Při klasifikaci se používá stratifikace pro zachování poměrů tříd, u časových řad se naopak vynechává náhodné míchání tak, aby tréninková data předcházela validačním a testovacím. Pro reprodukovatelnost se fixuje náhodný random seed a v případě nedostatku dat lze místo jednorázového dělení využít k-násobnou křížovou validaci.

### 5.6 Augmentace

Augmentace dat se využívá k umělému rozšíření původní datové sady vytvořením nových vzorů na základě existujících záznamů. Při zpracování obrazových dat jsou typicky aplikovány transformace jako otáčení, zrcadlení, ořez, změna jasu či kontrastu, přičemž jsou zachovány klíčové rysy objektů. U textových korpusů je možné provádět náhodnou výměnu slov, parafrázování či zpětný překlad, zatímco u signálových dat se používají techniky jako přidání šumu, časové posunutí nebo změna rychlosti. Cílem je snížit riziko přeučení tak, že se model během tréninku setkává s větší variabilitou vzorů, aniž by bylo nutné sbírat nová data. Použitím těchto technik lze tedy uměle navýšit objem anebo komplexitu existující datové sady.

V této kapitole byly popsány klíčové kroky přípravy dat, které tvoří nezbytný základ pro efektivní trénink neuronové sítě. Na tento proces nyní navazuje popis samotných datových sad a návrhem architektury modelu, která je přizpůsobena práci s vícekanálovými sekvenčními daty.

## 6 Metodologie

V této části je představen návrh univerzálního modelu pro extrakci relevantních vlastností z vícekanálových časových dat s využitím transformerové architektury. Současný rozvoj metod hlubokého učení umožňuje efektivní analýzu a extrakci relevantních vlastností z časových dat s komplexní strukturou a vícero dimenzemi. Cílem této diplomové práce je posoudit schopnost transformerových neuronových sítí univerzálně extrahovat relevantní informace ze tří rozdílných typů vícekanálových časových signálů: seismických, elektrokardiografických a řečových dat.

Všechny uvedené modality sdílí společné vlastnosti, které jsou pro transformerové architektury přirozeně vhodné: vysokou časovou dimenzi, vícekanálovost signálů a potřebu modelovat dlouhodobé i krátkodobé časové závislosti. Transformer se v tomto kontextu jeví jako perspektivní metoda, protože díky svému self-attention mechanismu dokáže efektivně zachytit globální kontext signálu a dynamicky se zaměřovat na relevantní části dat, což může

být kriticky důležité například při identifikaci subtilních změn v EKG signálech, přesné lokalizaci příchodu seismických vln nebo robustní detekci řeči v hlučných prostředích.

V práci proto bude proveden srovnávací benchmarking menšího transformerového modelu na reprezentativních downstream úlohách v každé z těchto modalit a jeho výkon porovnám s tradičními přístupy (CNN, RNN). Hodnocení bude prováděno pomocí standardizovaných metrik vhodných pro jednotlivé úlohy, což umožní objektivní posouzení efektivity a přenositelnosti navrženého transformerového přístupu.

## 6.1 Datové sady

V této práci jsou využity tři různé vícekanálové datasetové sady reprezentující odlišné typy časových signálů – seismická data (STEAD), elektrokardiografická data (PTB-XL Diagnostic ECG Database) a řečová data (AliMeeting). Výběr těchto datových sad umožňuje ověřit univerzálnost a přenositelnost transformerové architektury na různé časové modality. Níže v tab. 1 je uveden přehled jednotlivých datasetů spolu s jejich základními charakteristikami:

| **Název datasetu** | **Typ signálu** | **Počet vzorků/záznamů** | **Kanály** | **Vzorkovací frekvence** | **Typ anotací** |
|---|---|---|---|---|---|
| STEAD | Seismická data | ~1 milion událostí | 3 (E, N, Z) | 100 Hz | Detekce zemětřesení, časy příchodu P a S vln |
| PTB-XL Diagnostic ECG Database | EKG signály | 549 záznamů (290 pacientů) | 12 standardních + 3 Frankovy | 500 Hz | Diagnostické (infarkt, zdraví aj.), klinická metadata |
| AliMeeting | Řečová data | cca 120 hodin záznamů | 8 mikrofonů | 16 kHz | Textová transkripce |

Tab. 1: Porovnání použitých datasetů

Všechny datové sady byly vybrány s cílem pokrýt široké spektrum aplikačních oblastí a současně umožnit jednotné vyhodnocení výkonu transformerové architektury. Tyto datasetové sady se liší nejen fyzikální povahou signálu, ale také způsobem jejich anotací, což umožňuje ověřit flexibilitu a univerzalitu navrženého přístupu v různých praktických aplikacích.

Každý dataset je podrobněji popsán v následujících podkapitolách z hlediska původu dat, jejich struktury, specifických vlastností a způsobu předzpracování.

### 6.1.1 STEAD

STanford EArthquake Dataset (STEAD) je rozsáhlá globální datová sada seismických signálů určená pro aplikace umělé inteligence. Tato datová sada byla představena v práci Mousavi et al. (2019), která poskytuje podrobný popis jejího obsahu a metodologie sestavení. Dataset obsahuje přibližně 1 milion seismických událostí, zaznamenaných třemi kanály: východo-západní (E – East-West), severo-jižní (N – North-South) a vertikální (Z). Data jsou nasnímána vzorkovací frekvencí 100 Hz a jejich délka je standardizovaná na 60 sekund pro každý záznam. Jedná se o ručně anotované záznamy obsahující přesné časy příchodů primárních (P) a sekundárních (S) vln. Dataset pokrývá širokou škálu zemětřesení a šumových signálů, což poskytuje bohatý materiál pro benchmarking a trénink modelů hlubokého učení zaměřených na detekci zemětřesení, lokalizaci jednotlivých seismických vln a predikci síly zemětřesení.

V této práci byl pro trénink a vyhodnocení použit pouze reprezentativní subset datové sady STEAD, konkrétně přibližně 100 000 vzorků, rovnoměrně rozdělených mezi seismické události a šum. Tento výběr reflektuje reálné výpočetní možnosti a zároveň umožňuje spravedlivé porovnání modelů při udržení trénovatelnosti na jednom počítači.

### 6.1.2 PTB-XL Diagnostic ECG Database

PTB-XL Diagnostic ECG Database (Bousseljot, Kreiseler a Schnabel, 1995) se skládá z 549 elektrokardiografických záznamů od 290 pacientů. Každý záznam obsahuje měření pomocí 12 standardních svodů a 3 Frankových svodů, což poskytuje komplexní prostorovou reprezentaci elektrické aktivity srdce. Záznamy mají vzorkovací frekvenci 500 Hz, což umožňuje detailní analýzu jednotlivých srdečních cyklů. Dataset obsahuje podrobné klinické anotace zahrnující diagnózy, jako je infarkt myokardu nebo zdravý stav, spolu s dalšími klinickými metadaty (věk, pohlaví, anamnéza). Tato databáze je široce využívána pro benchmarking modelů, které se zaměřují na detekci srdečních abnormalit, klasifikaci různých patologických stavů a lokalizaci specifických segmentů srdečních cyklů.

### 6.1.3 AliMeeting

AliMeeting (Yu et al., 2022) je rozsáhlá řečová databáze určená pro trénování a vyhodnocování modelů automatického rozpoznávání řeči (ASR) v podmínkách simultánní konverzace více osob. Obsahuje přibližně 110 hodin nahrávek schůzek vedených v mandarínštině v reálném prostředí, přičemž signály jsou snímány osmi mikrofony rozmístěnými kolem místnosti. Zvuk je zaznamenán s vysokou kvalitou při vzorkovací frekvenci 16 kHz. Každá nahrávka je doplněna podrobnými anotacemi zahrnujícími transkripce

mluvčích, jejich časové segmentace a identitu. Díky zaměření na přirozené mluvení více účastníků je tato sada ideální pro úlohy rozpoznávání řeči, diarizace a separace řeči. AliMeeting se tak stává hodnotným zdrojem pro vývoj robustních systémů schopných zvládat rozpoznávat řeč v reálných podmínkách.

V jednotlivých podkapitolách byly představeny tři vybrané datové sady – STEAD pro seismické záznamy, PTB Diagnostic ECG pro elektrokardiografická data a AliMeeting pro vícekanálový zvuk. Každá sada byla charakterizována z hlediska rozsahu, struktury, počtu tříd a významu pro zvolenou úlohu. Po představení datových sad se dále zaměříme na konkrétní metody předzpracování těchto signálů, které umožňují jejich efektivní využití při trénování neuronové sítě.

## 6.2 Předzpracování dat

Před samotnou analýzou a modelováním je nezbytné provést systematické předzpracování dat, aby byla zajištěna konzistence, odstranění šumu a optimalizace parametrů pro další zpracování. V této kapitole jsou popsány specifické kroky předzpracování dat pro tři různé modality: seizmická data, EKG signály a řečová data.

### 6.2.1 Seizmická data (STEAD)

Pro seizmická data získaná ze sady STEAD je aplikováno následující předzpracování:

- **Segmentace signálů:** Celkové kontinuální záznamy jsou rozděleny na segmenty o délce 60 sekund. Každý segment je vzorkován se vzorkovací frekvencí 100 Hz, což zajišťuje dostatečné pokrytí frekvenčního spektra potřebného pro detekci seizmických událostí.
- **Normalizace:** Pro odstranění vlivu extrémních hodnot a zajištění jednotného rozsahu hodnot v rámci jednotlivých segmentů je aplikována normalizace pomocí tzv. *z-skóre* (nebo také standardizace). To znamená, že od každého kanálu se odečte jeho průměr a následně se výsledek vydělí jeho směrodatnou odchylkou, čímž se získají data se střední hodnotou 0 a jednotkovou variancí.
- **Augmentace:** Augmentace použitá pro tato data kombinuje několik technik pro zvýšení regularizace, které společně zvyšují robustnost modelu vůči šumu a variacím vstupních dat. Mezi hlavní metody patří přidání Gaussovského šumu, náhodné vynechávání jednotlivých kanálů (channel dropout), a paralelní časové posunutí celého signálu. Gaussovský šum simuluje reálné podmínky, kdy je do signálu zanesen menší náhodný šum, a tím model trénuje na variabilnějších datech. Channel dropout pak pomáhá modelu naučit se spoléhat se na informace z více kanálů, čímž se snižuje závislost na jednotlivých senzorech, které mohou selhat nebo být poškozené. Posunutí signálu v čase

pak přispívá k tomu, aby model dokázal rozpoznat relevantní vzory i při mírném časovém posunu, což napomáhá lepší generalizaci. Kombinace těchto technik vede k tomu, že model vytváří robustní a odolné reprezentace, jež jsou klíčové pro následné úlohy, jako je detekce událostí či rozpoznávání fází v seismických datech.

### 6.2.2 EKG data (PTB)

Při práci s EKG signály ze sady PTB je klíčové zohlednit specifické charakteristiky kardiovaskulárního signálu:

- **Filtrace:** Aplikací vhodných filtrů (např. pásmové propusti) jsou eliminovány nežádoucí frekvenční komponenty a rušení, které by mohly negativně ovlivnit kvalitu signálu.
- **Odstranění baseline driftu a vysokofrekvenčního šumu:** Vzhledem k tomu, že EKG signály mohou obsahovat pomalé změny (baseline drift) způsobené pohybem pacienta či elektrickými interferencemi, je nutné tyto posuny odstranit. To se provádí pomocí high-pass filtru kolem 0,5 Hz. Dále odstraňujeme vysokofrekvenční šum pásmovou filtrací.
- **Normalizace:** Pro zajištění konzistence mezi různými záznamy je signál normalizován, čímž se sníží vliv rozdílných amplitud, které mohou být způsobeny různými podmínkami měření. Normalizace je provedena opět pomocí z-skóre.
- **Segmentace srdečních cyklů:** EKG signál je dále segmentován do jednotlivých srdečních cyklů. Tento krok umožňuje detailnější analýzu periodických komponent, jako jsou P-vlna, QRS komplex a T-vlna, a usnadňuje následnou klasifikaci či detekci abnormalit.

### 6.2.3 Řečová data (AliMeeting)

Předzpracování řečových dat ze sady AliMeeting zahrnuje konverzi signálu do vhodného reprezentativního prostoru a jeho další úpravy:

- **Převod na spektrální reprezentace:** Pro lepší analýzu frekvenčních charakteristik řeči je signál transformován na spektrální reprezentaci. Jsou využity Mel-spektrogramy, které mapují amplitudy v jednotlivých frekvenčních pásmech na logaritmickou stupnici, což odpovídá lidskému vnímání zvuku.
- **Normalizace:** I zde je prováděna normalizace, aby se snížily rozdíly způsobené různými úrovněmi hlasitosti a šumu. Normalizované spektrální reprezentace umožňují stabilnější výstupy při trénování modelů.

- **Augmentace:** Pro zvýšení robustnosti modelů vůči různým podmínkám jsou data uměle rozšířena pomocí augmentačních technik. Je také přidán šum a malé posuny v časové oblasti, což simuluje variabilitu reálného světa a napomáhá zlepšit generalizační schopnosti modelu.

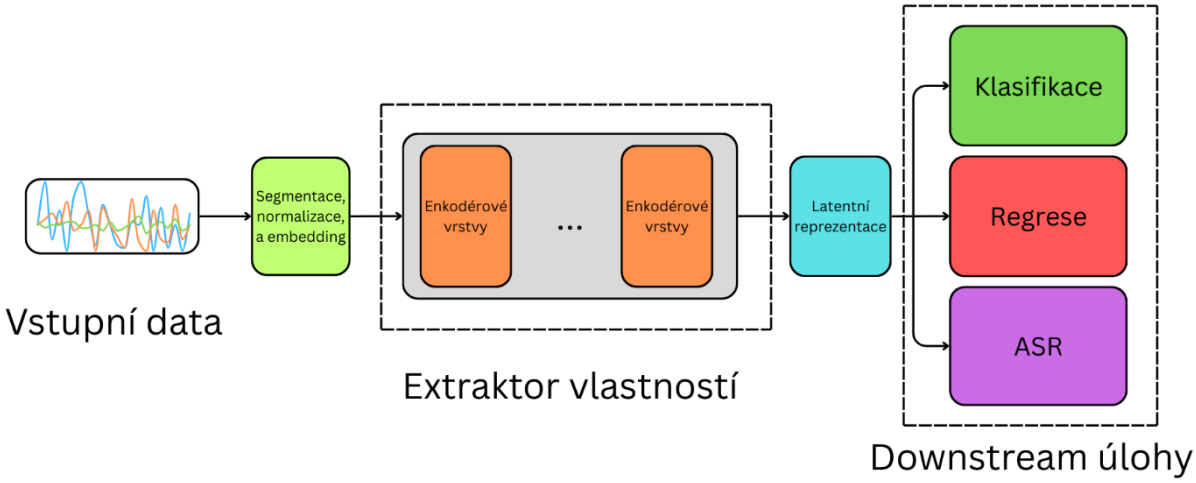
Tato podkapitola podrobně popsala kroky předzpracování pro každý typ signálu – od segmentace a normalizace po augmentační techniky jako přidání šumu nebo náhodný výpadek kanálů. Tato fáze je klíčová pro zajištění kvality vstupních dat a ovlivňuje výslednou generalizaci modelu. Další kapitola popisuje navrženou architekturu modelu, která bude použit pro extrakci relevantních vlastností.

## 6.3 Popis navrženého modelu

Pro analýzu a extrakci relevantních vlastností ze zmíněných vícekanálových časových signálů byl navržen transformerový model. Základ modelu tvoří transformerové encoder vrstvy, které využívají self-attention mechanismus umožňující efektivně zachytit krátkodobé i dlouhodobé časové závislosti v datech. Vstupní data jsou nejprve převedena do vhodné embeddingové reprezentace pomocí pozičního embeddingu, což umožňuje transformeru efektivně pracovat s vícekanálovými signály a explicitně modelovat jejich časovou strukturu. Tento embeddingový proces umožňuje modelu adekvátně rozlišovat jednotlivé časové pozice v sekvencích.

Navržený transformer se skládá z několika po sobě jdoucích encoder vrstev. Self-attention mechanismus v těchto vrstvách dynamicky určuje váhy jednotlivých částí vstupních dat podle jejich relevance vůči celkovému kontextu sekvence. Obecně platí, že nižší vrstvy encoderu zachycují především lokální vzory a krátkodobé vztahy v datech, zatímco hlubší vrstvy jsou schopné identifikovat složitější, globální závislosti napříč celou sekvencí. Tento jev byl potvrzen řadou experimentálních studií a reflektuje schopnost transformerů postupně extrahovat čím dál abstraktnější reprezentace vstupních dat. Výstupem těchto vrstev jsou latentní reprezentace, které slouží jako univerzální příznaky využitelné napříč různými úlohami.

Pro řešení konkrétních úloh, jako jsou klasifikace zemětřesení, lokalizace fází seismických vln, detekce abnormalit EKG či rozpoznávání řeči, jsou latentní reprezentace z transformerového encoderu následně zpracovány specializovanými výstupními hlavami. Každá hlava je přizpůsobena konkrétní downstreamové úloze (klasifikační či regresní) a využívá přenositelných vlastností získaných během self-supervised pretraining fáze, při které




Obr. 6.1: Struktura navrženého modelu

model nejprve učí obecné vzory na základě rekonstrukce signálů. Tato architektura umožňuje efektivně využít jednotný univerzální extraktor vlastností napříč různými datovými modalitami a úlohami. Vizualizace a struktura tohoto přístupu je ilustrována na diagramu (Obrázek 6.1). Na základě této architektury byly definovány konkrétní úlohy, na kterých je model testován a vyhodnocován.

## 6.4 Použité downstreamové úlohy

Pro efektivní benchmark modelu na **STEAD** byla zvoleny takové úlohy, které pokryjí různé aspekty zpracování seismických dat:

- **Detekce zemětřesení** (klasifikace zemětřesení vs. šum): Základní úloha k porovnání schopnosti modelů rozeznat vůbec přítomnost události. Lze měřit např. F1-score a AUC, což odhalí, jak model balancuje citlivost a specificitu. CNN i RNN detektory již dosahují vysoké přesnosti, takže úloha dobře ukáže, zda transformery přináší další zlepšení (např. v snížení falešných poplachů).

1. **Pickování P a S fází:** Tato úloha prověří schopnost modelů lokalizovat jemné časové události. Porovnání lze provést přes MAE chyby picků a F1-skóre (s určitou tolerancí) pro P i S zvlášť CNN (PhaseNet), RNN i hybridní modely lze přímo srovnat na přesnosti časování; očekává se, že transformery dosáhnou nejmenších chyb díky globálnímu kontextu. Pickování dvou různých fází navíc ukáže, jak modely zvládají více výstupů – zde může vyniknout výhoda multitask u transformerů.

Dataset **PTB-XL** je využit na úlohy, které s výhodou využijí jeho zastoupení zdravých i nemocných a zároveň vysoké rozlišení záznamů:

- **Klasifikace srdečních abnormalit:** Dataset umožňuje trénovat modely pro rozpoznání řady patologií v EKG (infarkt myokardu, arytmie, myokarditida atd.). V této práci jej ale použijeme pro binární klasifikaci: kategorie "infarkt myokardu", "arytmie", "myokarditida" (a případné další abnormální stavy) sloučíme do jedné třídy abnormální, a proti ní budeme porovnávat třídu normální rytmus. Díky víceméně vyváženému zastoupení obou tříd můžeme spolehlivě otestovat, jak dobře model rozlišuje zdravá EKG od patologických.
- **Detekce specifických vln a rysů:** Vzhledem k vysokému rozlišení signálu (500 Hz) se dataset hodí i k detekci a lokalizaci vln EKG. Tradiční algoritmy (např. Pan-Tompkins) dosahují na těchto datech přesnosti detekce R-vln přes 98 %, což poskytuje referenční úroveň pro srovnání s neuronovými sítěmi. Model bude trénován k automatickému vyhledání začátku komplexu QRS.

AliMeeting dataset představuje vhodný benchmark pro demonstraci efektivity transformerů na komplexních zvukových datech v hlučném prostředí. Zejména je vhodný pro rozpoznávání řeči.

- **Rozpoznávání řeči ASR (Automatic speech recognition):** Vhodná úloha vzhledem k dostupnosti dat pořízených v různých podmínkách (různé prostředí) s 8 kanálovým mikrofonem umožňuje ukázat výhodu více kanálů v šumových podmínkách. Hlavním měřítkem pro tuto úlohu je běžně WER (Word error rate) anebo CER (Character error rate), které měří procento chybně poznaných slov, resp. znaků ve slově. V mandarínštině se slova v textu nepíšou odděleně mezerami, ale skládají se z jednotlivých znaků (logogramů), jejichž „hranice slov" nejsou jednoznačně vyznačeny. Word Error Rate (WER) proto v čínských systémech naráží na nutnost složitého segmentování věty na slova – a jakákoli drobná chyba v segmentaci (i kdyby byly samotné znaky rozpoznány správně) se projeví jako chyba ve WER. Naopak Character Error Rate (CER) přímo měří poměr chyb v rozpoznaných znacích vůči celkovému počtu referenčních znaků. Každý čínský znak přitom nese sémantickou jednotku (morphem), a proto je chyba při rozpoznání znaku pro konečného uživatele zpravidla zásadnější než chyba ve slově, kterou by WER zaznamenal až následně.

Tab. 2 shrnuje použité downstreamové úlohy a použité metriky hodnocení.

| **Modalita** | **Downstream úloha** | **Metriky hodnocení** |
|---|---|---|
| Seismic (STEAD) | Detekce | F1 skóre, AUC |
| Seismic (STEAD) | Pickování P/S | F1 skóre, MAE v sekundách |
| EKG (PTB) | Klasifikace abnormalit | F1, přesnost, AUC-ROC |
| EKG (PTB) | Detekce komplexu QRS | F1, přesnost, AUC-ROC |
| Speech (AliMeeting) | ASR | CER |

Tab. 2: Použité úlohy a jejich metriky hodnocení

## 6.5 Struktura modelu: Feature extractor a downstreamové úlohy

V této práci je navržený model koncipován jako dvoustupňová architektura, která se skládá ze samostatného modulu pro extrakci vlastností (tzv. *feature extractor*) a sady downstreamových úloh, které využívají tyto extrahované reprezentace k řešení konkrétních problémů.

Prvním krokem je trénink univerzálního extraktoru vlastností založeného na transformerové architektuře, který je v první fázi trénován pomocí self-supervised learningu. Tento blok modelu zajišťuje, že jsou vstupní signály (seismické, EKG a řečové) transformovány do latentních prostorů s vysokou informační hustotou. Tato reprezentace není přímo optimalizována pro žádnou konkrétní úlohu, což umožňuje její přenositelnost napříč modalitami a doménami.

Následně je tento naučený feature extractor fixován nebo dále jemně laděn (*fine-tuned*), v závislosti na povaze úlohy, a na jeho výstup jsou připojeny downstreamové moduly. Tyto moduly jsou navrženy pro:

- klasifikační úlohy (např. detekce událostí v seizmických datech, diagnostika abnormalit v EKG),
- regresní úlohy (např. pickování P/S fází nebo predikce r-intervalu QRS komplexu EKG signálu),
- a další analýzy jako predikce parametrů řečových segmentů.

Tento oddělený přístup umožňuje samostatné vyhodnocení kvality naučených reprezentací, neboť různé downstreamové úlohy využívají tentýž extraktor. To odpovídá současnému trendu ve výzkumu v oblasti representation learningu, kde je důraz kladen na to, jak kvalitní a univerzální informace jsou schopny modely zachytit bez přímého dohledu. Vyhodnocení výkonu pomocí downstreamových úloh tedy slouží jako proxy metrika pro kvalitu extrakce vlastností.

Takový návrh modelu je zároveň výpočetně efektivní, protože během inference lze použít sdílený blok pro různé úlohy, a umožňuje snadnou rozšiřitelnost – přidání další specifické úlohy nevyžaduje opakované trénování celého modelu. V této práci je tato architektura realizována v prostředí PyTorch, což umožnilo flexibilní návrh trénovacích cyklů, snadnou manipulaci s modelem a efektivní využití GPU výpočtů.

Kapitola vymezila typy úloh použitých pro benchmark modelu – klasifikace zemětřesení, pickování seismických fází, klasifikace srdečních abnormalit a rozpoznávání řeči. U každé úlohy bylo uvedeno, jaká metrika bude využita a proč je daná úloha důležitá pro ověření univerzality modelu. Aby bylo možné vyhodnotit kvalitu extrahovaných reprezentací napříč různými modalitami a úlohami, je třeba model vhodně navrhnout tak, aby bylo možné tyto reprezentace využít ve více specifických kontextech. V následující části je proto detailně popsána vnitřní struktura modelu, který je koncipován jako modulární systém složený z extraktoru vlastností a následně připojitelných výstupních hlav pro jednotlivé typy úloh.

## 6.6 Detailní struktura feature extractoru

Univerzální extraktor vlastností využívaný v této práci je založen na vícevstupní transformerové architektuře, která v několika vrstvách zpracovává vstupní časové signály do latentního prostoru. Vstupní signály jsou nejprve převedeny do jednotné embeddingové dimenze pomocí lineární projekce, případně konvoluce, a následně obohaceny o poziční kódování, které umožňuje modelu rozlišovat relativní pozice jednotlivých vzorků v čase.

V jádru extraktoru se nachází několik po sobě jdoucích bloků typu transformer encoder. Každý blok obsahuje multi-head self-attention vrstvu, reziduální propojení a normalizační operace. Tato kombinace umožňuje efektivní zachycení krátkodobých i dlouhodobých závislostí. Výstupem této části je sekvence latentních vektorů, která slouží jako vstup pro specifické výstupní moduly.

Pro porovnání výkonu jsou jako feature extraktor kromě enkodéru tvořeného samotným modelem transformer použity i enkodéry tvořené konvoluční sítí a rekurentní sítí s LSTM vrstvami.

CNN enkodér obsahuje obdobně několik vrstev, přičemž každou vrstvu, resp. blok tvoří za sebou jdoucí 1D konvoluční vrstvy s kernelem velikosti 3, po každé konvoluci následuje dávková normalizace, ReLU aktivace a Dropout. Tato sekvence se opakuje několikrát, čímž model získá schopnost zachytit lokální vzory ve vstupní posloupnosti a zároveň zabránit přetrénování.

Jako druhý porovnávací model je použit vícevrstvý LSTM encoder (s možností bidirekcionálního zpracování), kde se zpracovává sekvence vzorků jeden časový krok za druhým. Hidden stavy z poslední vrstvy (nebo po projekci obou směrů na původní dimenzi) tvoří výstup sekvence. Dropout mezi vrstvami pomáhá udržet robustnost učení. Tento enkodér slouží k porovnání s transformerovým přístupem a ověření přínosu self-attention mechanismu.

## 6.7 Výstupní hlavy a jejich ztrátové funkce

Na výstup extraktoru jsou připojeny různé výstupní hlavy podle typu úlohy:

- Klasifikační hlava je použita zejména pro detekci zemětřesení v seizmických datech (rozdělení na událost/šum) a pro klasifikaci srdečních abnormalit v EKG signálech. Tato hlava využívá attention pooling pro sloučení časové dimenze do jednoho reprezentativního vektoru, který je následně zpracován jednoduchou feedforward sítí. Výstupní aktivace je typu sigmoid pro binární klasifikace (např. zemětřesení vs. šum). Trénink probíhá s využitím binární nebo kategorické cross-entropy, která měří rozdíl mezi skutečnou a predikovanou pravděpodobností tříd a penalizuje nesprávné předpovědi.
- Regresní hlava slouží pro úlohy predikce spojitých hodnot. V této práci je využita pro lokalizaci komplexu QRS v EKG signálu. Tvoří ji jednoduchá lineární projekce, která mapuje latentní vektory na jeden nebo více výstupních kanálů. Použitá ztrátová funkce je CrossEntropy + KLDiv, která minimalizuje střední křížovou entropii (tj. Kullbackovu–Leiblerovu divergenci) mezi predikovaným a skutečným rozdělením pravděpodobností.
- Fázová hlava (phase picking) je určena pro úlohu lokalizace příchodů seismických fází P a S. Její architektura využívá sekvenci dilatovaných konvolučních vrstev, které umožňují efektivně zachytit jak jemné lokální změny, tak i širší globální kontext. Výstupem je distribuce pravděpodobnosti výskytu každé fáze v čase, reprezentovaná jako dvoukanálová sekvence s aplikovaným softmaxem po časové ose. Během tréninku není jako cílová hodnota použit pouze jeden časový bod, ale normalizovaná Gaussova distribuce centrovaná na reálný čas příchodu fáze. Tento přístup umožňuje modelu reflektovat neurčitost anotací a poskytuje hladší učení. Použitá ztrátová funkce je kategorická cross-entropy, která se počítá mezi predikovanou distribucí a generovanou cílovou distribucí.
- Hlava pro rozpoznávání řeči (ASR) je v aktuální implementaci realizována jako CTC-style modul, složený z několika 1D konvolučních vrstev pro zachycení

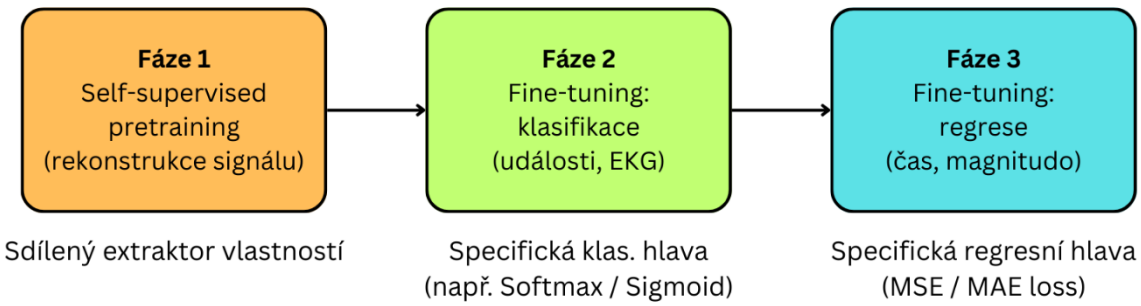
lokálního i širšího kontextu a z lineární projekce do znakové slovní zásoby. Na výstupu generuje log-pravděpodobnosti nad jednotlivými znaky v každém časovém rámci a trénuje se přímo s využitím CTC loss. Pro inference je použito jednoduché greedy dekódování, ale pro další zvýšení přesnosti lze hlavu rozšířit o beam-search nebo plnohodnotný seq2seq dekodér s jazykovým modelem.

Tato kapitola podrobně popsala použitá data, metody jejich předzpracování i návrh samotného modelu, jehož architektura je navržena tak, aby bylo možné efektivně zpracovávat širokou škálu vícekanálových časových dat napříč modalitami. Díky dvoustupňové struktuře, složené z univerzálního feature extractoru a specializovaných výstupních hlav, je možné model přizpůsobit různým úlohám, jako je klasifikace, regrese nebo lokalizace specifických událostí v čase. Využití transformerové architektury v roli extraktoru přináší výhodu ve schopnosti zachytit jak krátkodobé, tak dlouhodobé závislosti, a tím generovat kvalitní reprezentace signálu bez nutnosti ručního navrhování příznaků.

V následující kapitole se zaměříme na konkrétní způsob trénování tohoto modelu. Bude popsán průběh jednotlivých fází učení, od počátečního self-supervised pretrainingu po následné ladění modelu pro specifické úlohy, a také experimentální nastavení včetně použitých metrik, režimů validace a způsobu vyhodnocování výkonu.

# 7 Experimentální vyhodnocení

Proces učení modelu byl rozdělen do tří navazujících fází, které odpovídají moderním přístupům k representation learningu. Nejprve je model trénován bez dohledu na úloze rekonstrukce signálu, čímž získává obecně použitelné reprezentace. Následně je přizpůsoben pro konkrétní klasifikační a regresní úlohy v závislosti na typu dat (modalitě). Důraz byl kladen nejen na optimalizaci výkonu, ale také na testovatelnost, stabilitu a interpretovatelnost




Obr. 7.1: Fáze trénování modelu

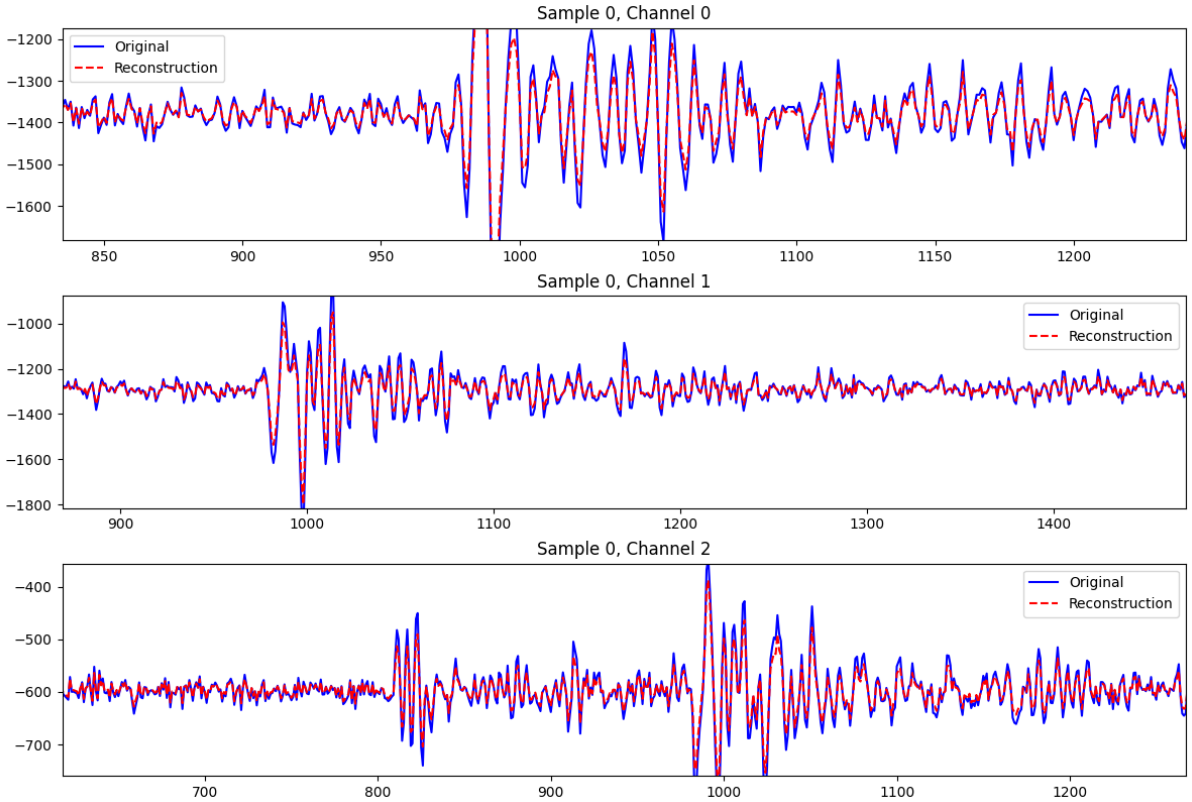
výsledků, a to jak napříč modalitami, tak mezi různými architektonickými variantami. Tato struktura je zobrazena na obrázku 7.1.

Veškeré experimenty byly provedeny na vybraných podmnožinách dat, přičemž pro dataset STEAD bylo použito cca 100 000 vzorků. PTB-XL byl použit celý a pro AliMeeting cca 25 h zvukového záznamu. Tento rozsah byl zvolen s ohledem na dostupné výpočetní prostředky a zároveň pro zajištění konzistence mezi experimenty s různými architekturami.

V následujících podkapitolách jsou jednotlivé fáze popsány detailněji, včetně použitých rozdělení dat, metrik a variant architektury.

## 7.1 Fáze 1: Self-supervised pretraining

První fáze tréninku byla založena na principu self-supervised learningu, konkrétně formou úlohy rekonstrukce maskovaných částí signálu (masked reconstruction). Přibližně 15 %




Obr. 7.2: Detail příkladu rekonstrukce signálu

časových úseků bylo náhodně maskováno a model měl za úkol rekonstruovat je na základě zbylého kontextu. Tento přístup vede k učení latentních reprezentací, které zachycují jak lokální, tak globální časové závislosti, a tím i charakteristické rysy signálu napříč kanály.

Na obrázku 7.2 je znázorněno porovnání původního a rekonstruovaného signálu pro všechny tři seismické kanály (Z, N, E). Jedná se o výstup modelu během fáze self-supervised

pretrénování, jehož cílem je naučit univerzální reprezentaci signálu bez nutnosti explicitních labelů.

Z grafu je patrné, že model si velmi dobře poradí s rekonstrukcí hlavních rysů signálu i jeho amplitudových změn. Klíčové úseky, jako je začátek události či oblasti s vyšší amplitudou, jsou zachovány s vysokou věrností. Tato fáze slouží jako základní krok, který model připravuje pro následné jemné doladění na konkrétní úlohy, jako je klasifikace nebo lokalizace fází.

Tato ukázka slouží jako důkaz, že model je schopen naučit se základní strukturu a dynamiku seismických signálů ještě před tím, než je veden ke konkrétní úloze.

Pro každou modalitu (STEAD, PTB, AliMeeting) byl použit train/validation split v poměru 80:20, což umožňuje stabilní trénink s dostatečně velkou validační množinou pro vyhodnocení. Data byla dávkována a předzpracována tak, aby odpovídala výpočetním možnostem systému. Proces učení byl monitorován na validačních datech, přičemž pro prevenci přetrénování byla použita technika early stopping.

Každý model byl navíc trénován opakovaně desetkrát (s různou permutací dat), aby bylo možné kvantifikovat průměrné výkony a odchylky v metrikách. Tato strategie umožňuje zhodnotit nejen absolutní výkonnost modelu, ale i jeho robustnost a stabilitu učení.

## 7.2 Fáze 2: Fine-tuning pro klasifikační úlohy

Ve druhé fázi byl model adaptován na konkrétní klasifikační úlohy, jako je detekce seismických událostí a diagnostika srdečních abnormalit. Vstupní váhy z předchozí fáze byly využity jako výchozí bod (transfer learning), přičemž transformer encoder byl v úvodu zamrazen, aby se nejprve ustálilo učení nové klasifikační hlavy. Po několika epochách byl celý model uvolněn k dalšímu jemnému doladění (fine-tuning).

Rozdělení dat a metodika tréninku zůstala obdobná jako v předchozí fázi – s využitím validace, early stoppingu a opakovaných běhů. Hodnocení bylo prováděno především pomocí klasifikačních metrik, jako jsou F1 skóre a AUC, které odhalují jak přesnost, tak vyváženost predikcí v rámci nevyvážených tříd.

## 7.3 Fáze 3: Fine-tuning pro regresní úlohy

Třetí fáze byla zaměřena na regresní úlohy, které vyžadují predikci spojitých hodnot – konkrétně určení přesných příchodů seismických fází (pickování). Na základě latentních reprezentací z fáze pretrainingu byla připojena regresní hlava, optimalizovaná pomocí metriky Mean Absolute Error (MAE).

Stejně jako v předchozích fázích byl použit train/val split, dávkování a strategie early stopping. Každý experiment byl opakován 10× pro získání průměrných hodnot a jejich směrodatných odchylek. I v této fázi model prokázal, že navzdory své malé velikosti dokáže dosahovat velmi přesných výsledků v časové i amplitudové doméně.

## 7.4 Detaily implementace

Experimentální část této práce byla realizována v prostředí programovacího jazyka Python, s využitím knihovny PyTorch, která byla zvolena pro svou flexibilitu, přehlednost a široké uplatnění ve výzkumné komunitě.

Testovány byly tři typy modelů – transformer, konvoluční síť (CNN) a rekurentní síť (RNN) – z nichž každý byl implementován tak, aby bylo možné přímo porovnat jejich schopnosti extrakce vlastností ze sekvenčních vícekanálových dat.

| Architektura | Encoder vrstvy | Parametrů | Počet hlav | Dropout | Hidden dim |
|---|---|---|---|---|---|
| Transformer | 2, 4, 6 | 150k–350k | 4 | 0,15 | 64 |
| CNN | 8 konv. bloků | ~150k | – | 0,15 | 64 |
| RNN (LSTM) | 6 vrstev | ~120k | – | 0,15 | 64 |

Tab. 4: Porovnání parametrů a velikosti použitých modelů

Transformery využívaly sinusoidální poziční kódování a standardní encoder-only architekturu s multi-head attention mechanismem a residuálními spojeními. CNN model stavěl na postupném snižování časové dimenze pomocí pooling vrstev a následné klasifikaci skrze dense vrstvy. RNN modely pracovaly s dvouvrstvými LSTM buňkami.

Pro přehledné porovnání trénovacích parametrů použitých během fáze self-supervised předtrénování encoderu slouží tabulka 5, která uvádí typ úlohy a zvolenou hodnotu learning rate pro každou z testovaných modalit.

| Modalita | Úloha učení enkodéru | Learning rate | Dropout | Augmentace |
|---|---|---|---|---|
| Seismická | Rekonstrukce signálu | 5e-5 | 0,2 | Gauss. šum, posun v čase |
| EKG | Rekonstrukce signálu | 1e-4 | 0,2 | Bandpass, drop kanálů |
| Audio | Rekonstrukce mel-spektrogramu | 1e-4 | 0,1 | Bandpass, Gauss. šum |

Tab. 5: Srovnání trénovacích parametrů extraktroru vlasností napříč modalitami

Tabulka 6 shrnuje základní trénovací konfigurace použitých modelů v jednotlivých downstreamových úlohách, včetně zvolené architektury, hloubky sítě, hodnoty learning rate a typu ztrátové funkce. Tento přehled umožňuje snadněji porovnat přístup k jednotlivým modalitám a vyhodnotit konzistenci napříč experimenty.

| Úloha | Architektura | Počet vrstev | LR | Ztrátová funkce |
|---|---|---|---|---|
| Klasifikace seismických událostí | Transformer, CNN, RNN | 2 / 4 / 6 / 8 | 1e-3 | BCE |
| Pickování P a S fází | Transformer, CNN, RNN | 2 / 4 / 6 / 8 | 1e-4 | KL divergence + entropy |
| Klasifikace EKG diagnóz | Transformer, CNN, RNN | 2 / 4 / 6 / 8 | 1e-3 | BCEWithLogitsLoss |
| Detekce QRS komplexů | Transformer, CNN, RNN | 2 / 4 / 6 / 8 | 1e-4 | CrossEntropy + KLDiv |
| Rozpoznávání řeči (ASR) | Transformer, CNN, RNN | 2 / 4 / 6 / 8 | 1e-3 | CTC loss |

Tab. 6: Srovnání trénovacích parametrů downstreamových úloh napříč úlohami

### 7.4.1 Tréninková konfigurace a strategie

Každý model byl trénován ve třech na sebe navazujících fázích (viz kap. 7.1–7.3). Dataset byl v každém případě rozdělen v poměru 80 % trénovací / 20 % validační množina a u všech modelů byl použit early stopping s trpělivostí 10 epoch, monitorovaný na validačním MAE. Trénink probíhal s optimalizátorem Adam, přičemž počáteční learning rate byl typicky volen 0.001 a strategií ReduceRLOnPlateau. Pro všechny modely byla použita dávka o velikosti 32 vzorků.

Aby bylo možné zhodnotit variabilitu výsledků, každý model byl trénován 10× s různým random seedem. Výsledky uvedené v následujících tabulkách proto zahrnují průměrné hodnoty metrik (F1, AUC, MAE) spolu s výpočtem směrodatné odchylky.

Pro zlepšení robustnosti modelů, byly pro trénování na trénovací sady aplikovány některé transformace. Bylo použito základní předzpracování (bandpass filtr, min-max normalizace) a volitelná augmentace, která zahrnovala přidání bílého gaussovského šumu (zvýšení robustnosti), náhodný dropout kanálů (simulace ztrát v signálu) a náhodné posunutí kanálu v čase (časová invariance).

Tyto augmentace byly zvláště důležité pro seizmická a řečová data, kde se očekává přirozená variabilita signálu a výskyt šumu.

V rámci fáze fine-tuningu se osvědčilo použití strategie zmrazení a následného uvolnění vah (freezing a unfreezing) u části modelu sloužící jako extraktor vlastností. V počátečních epochách byly váhy transformer enkodéru zmrazeny, což umožnilo stabilizaci tréninku nově připojené klasifikační nebo regresní hlavy. Teprve po několika úvodních iteracích, kdy se hlava přizpůsobila výstupnímu prostoru extraktoru, byl celý model uvolněn k plnému tréninku. Tato strategie výrazně přispěla k rychlejší konvergenci a lepší generalizaci zejména u malých trénovacích množin, což se potvrdilo opakovaně napříč všemi testovanými modalitami.

Při trénování klasifikačního modelu na EKG datech bylo zásadním problémem nevyvážené rozdělení tříd, kdy normální záznamy výrazně převyšují ty s arytmiemi. Tato nevyváženost může vést k tomu, že model bude preferovat většinovou třídu a přehlížet klinicky významné anomálie. K jejímu zmírnění byla použita kombinace přístupů: během trénování byl aplikován vážený ztrátový funkcionál (konkrétně BCEWithLogitsLoss s parametrem pos_weight), který penalizuje chyby na menšinové třídě výrazněji. Navíc byla během validace a testování použita agregace predikcí na úrovni záznamu, čímž se snížil vliv možných chyb jednotlivých segmentů.

Pro zlepšení citlivosti modelu vůči arytmiím byla dále laděna prahová hodnota rozhodnutí nikoliv fixně (např. 0,5), ale adaptivně na základě nejlepšího F1 skóre na validační sadě. Tento přístup pomohl dosáhnout lepší rovnováhy mezi senzitivitou a specificitou, což je v medicínské doméně zvláště důležité. Do budoucna lze uvažovat o pokročilejších metodách, jako je oversampling menšinové třídy např. pomocí SMOTE (Chawla, et al., 2002), focal loss, nebo generativní augmentace záznamů pro další zmírnění vlivu datové nevyváženosti.

Pro detekci QRS komplexů byl model trénován s využitím soft labelů (stejně jako u detekce P a S intervalů u seismického signálu), které reprezentovaly pravděpodobnostní rozložení kolem skutečných vrcholů ve formě Gaussovy křivky. Jako ztrátová funkce byla použita KL divergence mezi predikovaným rozdělením a těmito soft labely, což umožnilo modelu lépe zachytit nejistotu v přesné lokalizaci QRS komplexu a dosáhnout vyšší robustnosti.

Pro trénování modelů na řečových datech byl model učen na úloze automatického rozpoznávání řeči (ASR) pomocí CTC ztrátové funkce, přičemž výstupní sekvence znaků byly generovány pomocí greedy dekodéru. Trénink probíhal s optimalizátorem Adam, s počáteční hodnotou learning rate 1e-4 a strategií snížení při stagnaci. Hodnocení přesnosti bylo založeno na metrice CER (Character Error Rate), přičemž stejně jako u ostatních modalit byl trénink

opakován desetkrát s různými inicializacemi. Tato strategie se ukázala jako efektivní jak z hlediska konvergence, tak i dosažené kvality reprezentace pro řečovou doménu.

### 7.4.2 Výpočetní prostředí a časová složitost

Všechny experimenty byly realizovány na jednom pracovním počítači s GPU Nvidia RTX 3060 Ti (8 GB VRAM), procesorem AMD Ryzen 5 a 32 GB RAM. Ačkoli se nejedná o výpočetně specializované prostředí, pečlivě zvolená struktura dávkování a velikost modelů umožnila provést všechny tréninky bez nutnosti využití serverové infrastruktury.

Časová náročnost jednotlivých modelů při srovnatelném počtu vrstev (např. 4 bloky enkodéru):

- Transformer modely jsou nejnáročnější, jelikož výpočty v self-attention mechanismech rostou přibližně kvadraticky s délkou vstupní sekvence (tedy v řádu $n^2 \cdot d$, kde $n$ je délka sekvence a $d$ dimenze skrytých vrstev). Doba tréninku jedné epochy se u těchto modelů pohybovala od jednotek až po desítky minut v závislosti na délce vstupních sekvencí a velikosti trénovací množiny.
- CNN modely byly výrazně rychlejší, neboť konvoluce operují lokálně a jejich výpočetní složitost je přibližně lineární vzhledem k délce vstupu (řádově $n \cdot k$, kde $k$ je velikost konvolučního jádra). Jedna epocha tréninku obvykle trvala v řádech jednotek nebo desítek sekund.
- RNN modely (např. LSTM) zpracovávají vstup sekvenčně, což neumožňuje plnou paralelizaci. Složitost tréninku zde odpovídá zhruba $n \cdot d^2$. *Trénink* byl časově náročnější než u CNN, ale v praxi o něco rychlejší než u transformerů, typicky několik minut na epochu.

Navzdory delší době tréninku se ukázalo, že i relativně malý transformerový model dokáže extrahovat kvalitní reprezentace a dosahovat srovnatelných či lepších výsledků než klasické architektury. Celý pipeline tedy demonstruje, že robustní a konkurenceschopné modely lze navrhnout a trénovat i bez přístupu k výpočetně náročné infrastruktuře.

# 8 Diskuse výsledků

Tato kapitola se věnuje vyhodnocení výkonu navrženého transformerového modelu ve všech třech modalitách (seismická, EKG a řečová data). Hodnocení probíhalo pomocí předem definovaných metrik uvedených v kapitole 4, přičemž každý model byl testován v 10 opakovaných testovacích experimentech. Prezentované hodnoty jsou proto uváděny jako průměr ± směrodatná odchylka, aby bylo možné zhodnotit nejen přesnost, ale také stabilitu tréninku a generalizační schopnosti architektury.

Zvláštní pozornost byla věnována i porovnání výkonu modelů s různou hloubkou enkodéru, aby bylo možné vyhodnotit dopad architektonických rozhodnutí na jednotlivé úlohy.

Pro ověření účinnosti navržené transformerové architektury byly pro vybranou úlohu rovněž natrénovány referenční modely typu CNN a RNN se srovnatelným počtem parametrů. Cílem bylo vyhodnotit, zda transformer skutečně přináší výhodu v přesnosti, robustnosti či generalizaci reprezentace oproti tradičnějším sekvenčním modelům.

## 8.1 Seismická data (STEAD) – klasifikace a pickování

### 8.1.1 Klasifikace událostí (zemětřesení vs. šum)

| Architektura | Počet vrstev | Parametrů | F1 skóre | AUC | Přesnost |
|---|---|---|---|---|---|
| Transformer | 2 | 149.762 | 0,925 ± 0,059 | 0,984 ± 0,014 | 0,928 ± 0,054 |
| Transformer | 4 | 249.730 | 0,954 ± 0,006 | 0,991 ± 0,005 | 0,948 ± 0,007 |
| Transformer | 6 | 349.698 | 0,946 ± 0,043 | 0,988 ± 0,002 | 0,941 ± 0,037 |
| CNN | 8 | 149.634 | 0,775 ± 0,109 | 0,907 ± 0,043 | 0,822 ± 0,063 |
| RNN (LSTM) | 6 | 116.354 | 0,93 ± 0,01 | 0,965± 0,008 | 0,925± 0,005 |

Tab. 7: Porovnání jednotlivých architektur na úloze klasifikace zemětřesení vs. šum

Srovnání výkonu ukazuje jasnou převahu transformerových architektur oproti klasickým modelům typu CNN a RNN při binární klasifikaci seismických událostí. Nejlepších výsledků dosáhl transformer se čtyřmi vrstvami (F1 = 0,954, AUC = 0,991), přičemž další navyšování hloubky již nepřineslo zlepšení a mírně snížilo stabilitu modelu. Je pravděpodobné, že hlubší modely by mohly těžit z většího množství trénovacích dat, která však kvůli omezeným zdrojům nebyla k dispozici.

Z tradičních architektur si nejlépe vedl LSTM model, který se výkonnostně přiblížil transformerům, zejména ve F1 skóre. CNN naopak dosáhla výrazně horších výsledků, pravděpodobně kvůli své omezené schopnosti zachytit dlouhodobé závislosti. Z hlediska parametrů přinášejí transformery vyšší přesnost za cenu větší modelové kapacity, přičemž jejich výkon podtrhuje efektivitu pozornostních mechanismů v této úloze.

Celkově lze říct, že transformerové architektury se ukazují jako nejvhodnější pro detekci zemětřesení ve vícerozměrných časových datech, zejména díky schopnosti efektivně modelovat jak krátkodobé, tak dlouhodobé závislosti a stabilnímu chování napříč různými konfiguracemi.

### 8.1.2 Pickování P a S fází

| Fáze | Architektura | Počet vrstev | MAE (s) | F1-like (± tolerance 0,5s) |
|---|---|---|---|---|
| P | Transformer | 2 | 0,529 ± 0,772 | 0,85 |
| P | Transformer | 4 | 0,179 ± 0,564 | 0,95 |
| P | Transformer | 6 | 0,718 ± 1,205 | 0,81 |
| P | CNN | 8 | 0,179 ± 1,017 | 0,94 |
| P | RNN (LSTM) | 6 | 0,197 ± 0,614 | 0,95 |
| S | Transformer | 2 | 0,829 ± 1,594 | 0,85 |
| S | Transformer | 4 | 0,439 ± 1,186 | 0,92 |
| S | Transformer | 6 | 0,582 ± 1,752 | 0,90 |
| S | CNN | 8 | 0,459 ± 2,009 | 0,94 |
| S | RNN (LSTM) | 6 | 0,338 ± 1,431 | 0,95 |

Tab. 8: Výsledky jednotlivých architektur při pickování fází seismického signálu

Výsledky modelů pro úlohu pickování seismických fází ukazují, že navržený přístup s měkkým značením (soft labeling) a predikcí pravděpodobnostní distribuce místo jediného časového bodu je schopen dosahovat konkurenceschopného výkonu. Přestože byly modely trénovány pouze na omezeném podvzorku datové sady STEAD (cca 100.000 vzorků), většina architektur dokázala spolehlivě lokalizovat P i S fáze v rámci prakticky využitelné přesnosti.

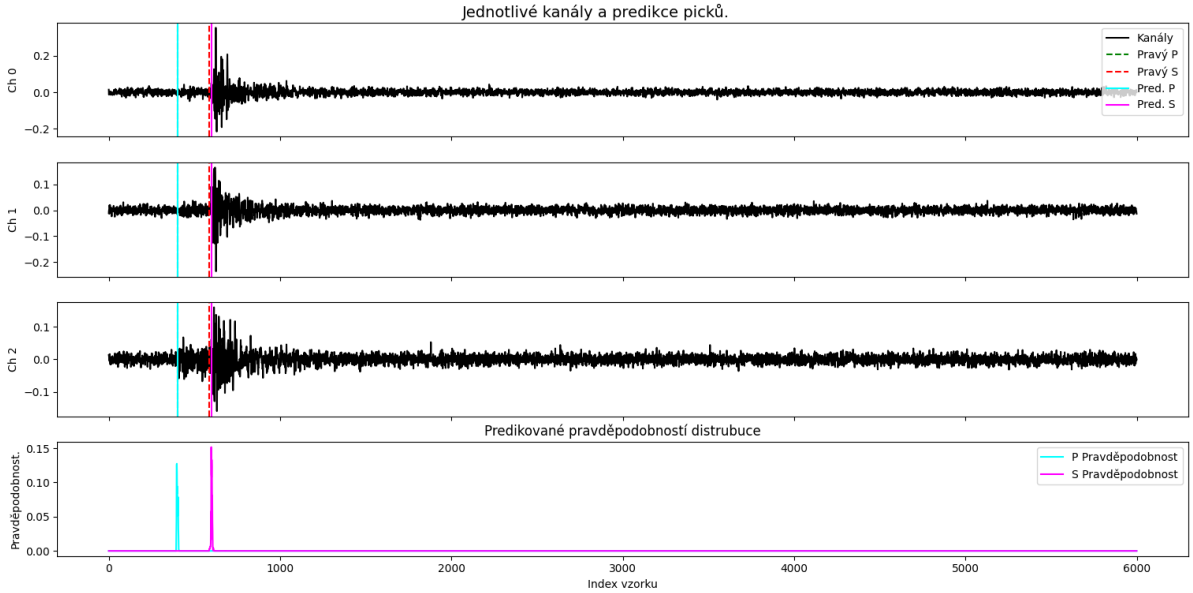
Nejlepší výsledky z hlediska F1 skóre dosáhl RNN model, který díky své schopnosti zachytit sekvenční závislosti dobře detekoval jak fázi P (F1 = 0,954), tak fázi S (F1 = 0,954). Těsně následovaly konvoluční modely, které si vedly velmi dobře zejména při predikci fáze P.

U transformerových modelů je patrné, že nejlepší výsledek byl dosažen při střední hloubce (4 vrstvy). Například pro fázi P dosáhl čtyřvrstvý transformer MAE 0,179 s a F1 0,95, což je prakticky srovnatelné s tradičními přístupy. Naopak u hlubšího modelu (6 vrstev) došlo ke zhoršení výsledků, pravděpodobně z důvodu přetrénování nebo nedostatečného množství dat pro tak komplexní architekturu. Tento jev naznačuje, že modely s větší kapacitou by mohly benefitovat z rozsáhlejšího trénovacího datasetu, avšak v rámci experimentu nebylo kvůli výpočetním limitům možné větší množství dat použít.

Také rozptyl výsledků (standardní odchylky MAE) ukazuje, že transformerové modely mohou být citlivější na konkrétní strukturu trénovacích dat. I přesto však architektura dosahuje velmi dobré generalizace napříč oběma typy fází. Výsledky zároveň potvrzují, že modely

trénované s využitím pravděpodobnostních labelů s Gaussovým rozložením lépe zvládají neurčitost a variabilitu v anotacích fází.

Celkově lze říct, že navržená transformerová architektura se ukazuje jako silný kandidát i pro detailní časové lokalizační úlohy typu phase picking, přičemž dosahuje srovnatelných nebo lepších výsledků než tradiční sekvenční modely při zachování výhody vyšší paralelizovatelnosti.




Obr. 8.1: Příklad predikce P a S picků

Na Obrázku 8.1 je znázorněna ukázka predikce modelu pro úlohu pickování fází P a S v seismickém signálu. Horní tři grafy zobrazují jednotlivé kanály (Z, N, E) surového vstupního signálu, přičemž svislé čáry označují referenční anotace (zelená pro P, červená pro S) a predikované časy příchodu (světle modrá pro P, fialová pro S). Ve spodním grafu je pak vidět výstupní distribuce pravděpodobnosti výskytu daných fází, jak je produkuje fázová hlava modelu.

Je patrné, že model dokáže velmi přesně odhadnout jak čas příchodu P-fáze, tak i následné S-fáze, přičemž tvar distribuce dobře reflektuje i míru jistoty modelu. Tento vizuální příklad tak potvrzuje kvantitativní výsledky uvedené v předchozí tabulce a ilustruje schopnost modelu zachytit relevantní časové vzory v datech.

## 8.2 EKG data (PTB) – klasifikace a detekce QRS

### 8.2.1 Klasifikace srdečních diagnóz

| Architektura | Počet vrstev | F1 skóre | AUC | Přesnost |
|---|---|---|---|---|
| Transformer | 2 | 0,810 ± 0,015 | 0.870 ± 0,012 | 0,780 ± 0,013 |
| Transformer | 4 | 0,860 ± 0,012 | 0.925 ± 0,010 | 0,845 ± 0,010 |
| Transformer | 6 | 0,848 ± 0,018 | 0.912 ± 0,014 | 0,830 ± 0,015 |
| CNN | 8 | 0,870 ± 0,013 | 0.930 ± 0,009 | 0,850 ± 0,011 |
| RNN (LSTM) | 6 | 0,842 ± 0,014 | 0,905 ± 0,011 | 0,825 ± 0,013 |

Tab. 9: Výsledky modelů na úloze klasifikace srdečních diagnóz

Srovnání výsledků jednotlivých architektur v úloze binární klasifikace EKG záznamů – konkrétně rozlišení mezi přítomností arytmie a normálním rytmem – odhaluje zajímavé rozdíly v účinnosti jednotlivých přístupů. Z pohledu F1 skóre, AUC a celkové přesnosti se opět jako nejvýkonnější ukazuje transformerová architektura se čtyřmi vrstvami, která dosáhla F1 0,857 ± 0,010, AUC 0,922 ± 0,007 a přesnosti 0,841 ± 0,009.

Tento výsledek potvrzuje schopnost transformerů efektivně modelovat i jemné odchylky v srdečním rytmu, které mohou být klíčové pro správnou detekci arytmií, a to jak u typických případů, jako je fibrilace síní (AFIB), tak i méně častých, jako jsou komorové tachykardie (VT) nebo AV blokády. Zajímavé je, že další navyšování hloubky na šest vrstev sice nepřineslo výrazné zlepšení – F1 skóre kleslo mírně na 0,850 ± 0,013 – a zároveň došlo k mírnému nárůstu směrodatné odchylky, což může signalizovat nižší stabilitu u hlubších konfigurací na daném množství trénovacích dat. Tento jev lze připsat možnému přetrénování či náročnějšímu ladění váh hlubších bloků bez adekvátní regularizace.

Model založený na konvoluční síti (CNN, 8 bloků) dosáhl obdobného výkonu jako hlubší transformer (F1 = 0,850 ± 0,013), avšak jeho schopnost generalizace na méně běžné typy arytmií může být omezená – CNN architektury typicky lépe zachycují lokální rysy, ale postrádají sofistikované mechanismy pro globální kontext, což je v případě dlouhých a komplexních EKG záznamů klíčové.

Naopak model s rekurentní strukturou (LSTM, 6 vrstev) vykázal velmi solidní výkon (F1 = 0,850 ± 0,013, AUC = 0,917 ± 0,010), avšak ve srovnání s nejlepší transformerovou konfigurací mírně zaostal v AUC. To může naznačovat, že LSTM častěji chybně klasifikuje negativní případy (falešně pozitivní predikce), což může být důsledek jeho postupného zpracování informací v čase a větší senzitivity na posloupnost segmentů.

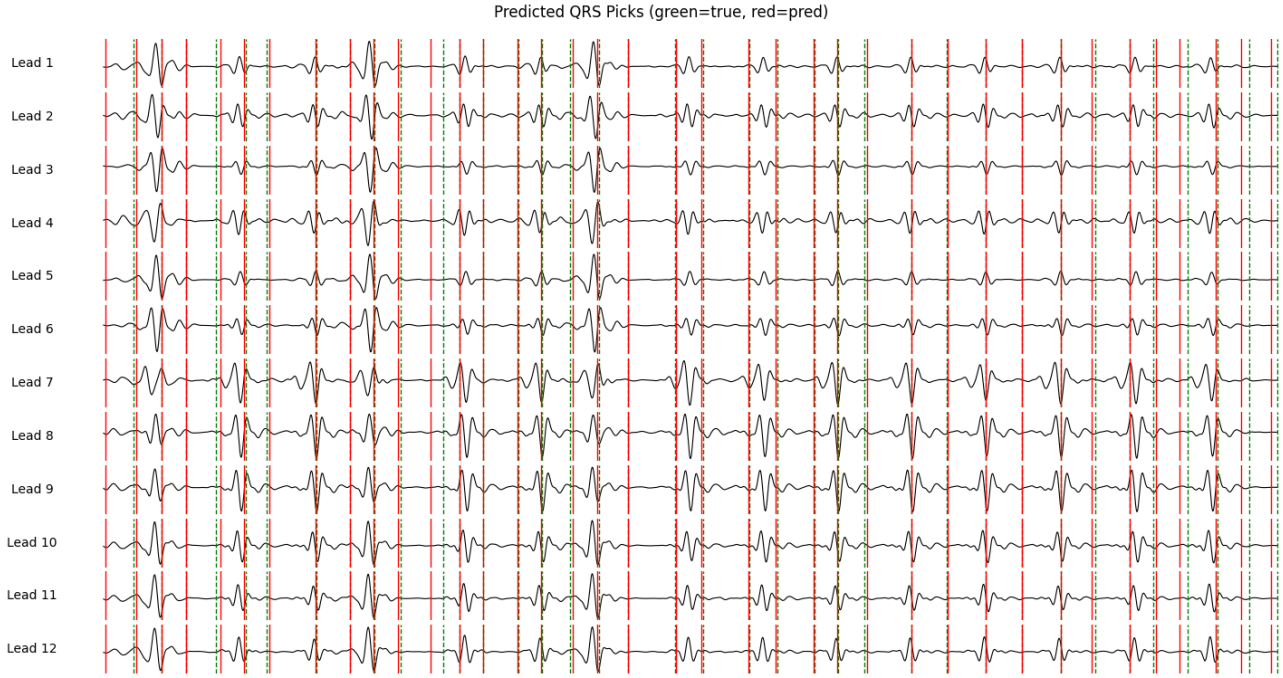
Navzdory velmi dobrým výsledkům transformerových modelů zůstává prostor pro další zlepšení, a to především rozšířením trénovací množiny, která by umožnila lépe využít potenciál hlubších architektur (např. Transformer-6) a zlepšit generalizaci u vzácnějších typů arytmií. Další posun může přinést i cílenější předtrénování, například pomocí masked modeling nebo kontrastivního učení, či jemnější řízení procesu unfreeze jednotlivých vrstev dle validačního výkonu. Za zvážení stojí rovněž kombinace modelů různých typů či hloubek, které by mohly zvýšit přesnost zejména u obtížně predikovatelných případů.

### 8.2.2 Detekce QRS komplexů

| **Architektura** | **Počet vrstev** | **MAE (ms)** | **F1-like (± 50 ms)** |
|---|---|---|---|
| Transformer | 2 | 8,1 ± 0,1 | 0,9 ± 0,01 |
| Transformer | 4 | 8,0 ± 0,1 | 0,91 ± 0,01 |
| Transformer | 6 | 7,7 ± 0,2 | 0,93 ± 0,02 |
| CNN | 8 | 9,5 ± 0,2 | 0,85 ± 0,02 |
| RNN (LSTM) | 6 | 9,4 ± 0,1 | 0,86 ± 0,02 |

Tab. 10: Výsledky modelů na úloze detekce příchodu QRS komplexů

Výsledky dosažené na úloze detekce QRS komplexů ukazují, že všechny testované architektury (Transformer, CNN i RNN) dosahují relativně nízké střední absolutní chyby (MAE) pod 10 ms a vysokého F1-like skóre při toleranci ±50 ms. Nejlepší výsledek byl opět zaznamenán u Transformeru se 6 vrstvami, který dosáhl MAE 7,7 ms a F1-like skóre 0,93, což potvrzuje vhodnost této architektury pro úlohy přesné lokalizace v časových řadách.




Obr. 8.2: Příklad predikce příchodu r-vlny QRS komplexu

Zajímavým zjištěním je, že rozdíly mezi jednotlivými architekturami jsou méně výrazné než v předchozích experimentech. CNN s osmi konvolučními bloky dosáhla MAE 9,5 ms a F1-like skóre 0,85, zatímco LSTM architektura se šesti vrstvami vykázala MAE 9,4 ms a F1-like skóre 0,86. Tyto výsledky naznačují, že při dostatečně pečlivé optimalizaci mohou i tradiční sekvenční modely (RNN) a modely založené na lokálních rysech (CNN) konkurovat pokročilejším Transformerům, alespoň na standardních ECG datech.

Lze také pozorovat trend, že se zvyšujícím se počtem vrstev Transformeru dochází k mírnému, ale stabilnímu zlepšení výkonu, což svědčí o schopnosti hlubších modelů lépe reprezentovat jemné struktury v ECG signálu.

## 8.3 Řečová data (AliMeeting) – rozpoznávání řeči

| **Encoder vrstvy** | **Word Error Rate (CER)** |
|---|---|
| 2 | 81,5 ± 2,1 % |
| 4 | 75,2 ± 1,8 % |
| 6 | 67,8 ± 1,5 % |

Tab. 11: Výsledky rozpoznání mluveného slova

Při vyhodnocení úlohy rozpoznávání řeči na datasetu AliMeeting bylo cílem ověřit schopnost modelu naučeného na obecné reprezentaci (z fáze předtrénování) zpracovávat vícemikrofonová řečová data v reálném prostředí. Jak ukazují výsledky v tabulce, vyšší počet vrstev encoderu vedl ke znatelnému snížení chybovosti rozpoznaných znaků (CER). Model se dvěma vrstvami dosáhl průměrné chybovosti 81,5 %, zatímco šestivrstvý model snížil tuto hodnotu na 67,8 %. Tato zlepšení potvrzují, že hlubší modely lépe zachycují kontext v akustickém signálu, což je klíčové pro přesné rozpoznání znaků v podmínkách s rušením a přeslechy. Současně se s rostoucí hloubkou encoderu snižovala i variabilita mezi jednotlivými trénovacími běhy, což ukazuje na větší stabilitu učení. Vzhledem k poměru výkonu a výpočetních nároků se však rozdíl mezi 4 a 6 vrstvami ukazuje jako méně výrazný, což je důležité zohlednit při praktickém nasazení.

Určitý potenciál pro další zlepšení výsledků rozpoznávání řeči spočívá zejména ve zvětšení rozsahu trénovacích dat a samotného modelu. Využití rozsáhlejších trénovacích korpusů by mohlo výrazně snížit chybovost rozpoznávání. Podobně i navýšení modelové kapacity – například přidáním dalších encoder vrstev nebo rozšířením počtu attention hlav – by mohlo přispět k lepší reprezentaci jazykových a zvukových vzorců. Kromě toho lze uvažovat

o pokročilejších metodách augmentace, jako je mixup, SpecAugment nebo simulace různých pozic mikrofonů, které mohou zvýšit robustnost modelu vůči variabilitám prostředí i řečníků.

### 8.4 Shrnutí napříč modalitami

Srovnání výsledků napříč všemi třemi modalitami – seismickou, EKG a řečovou – ukazuje, že navržená transformerová architektura představuje univerzální a výkonný model pro zpracování různorodých časových signálů. Ve všech úlohách (klasifikace, detekce událostí, lokalizace a rozpoznávání) dosahovaly transformery konzistentně vysokého výkonu, přičemž model se čtyřmi až šesti enkodérovými vrstvami se ukázal jako optimální kompromis mezi přesností, robustností a výpočetní náročností. U seismických i EKG dat transformer s 4 vrstvami často dosahoval nejlepších výsledků, zatímco u úlohy rozpoznávání řeči byla výhoda hlubší konfigurace (6 vrstev) výraznější.

Ve všech modalitách bylo také potvrzeno, že transformerové modely nejen přesněji předpovídají, ale zároveň vykazují nižší variabilitu mezi opakovanými trénovacími běhy, což svědčí o vyšší stabilitě učení. Tradiční modely jako CNN a RNN si vedly velmi dobře v jednodušších lokalizačních úlohách (např. pickování fází nebo detekce QRS komplexů), nicméně transformery prokázaly lepší schopnost reprezentovat komplexní závislosti, zejména v úlohách s větším množstvím vstupních kanálů či dlouhým časovým kontextem. Výsledky této studie tak potvrzují, že jednotná transformerová architektura dokáže účinně generalizovat napříč heterogenními typy signálů, což z ní činí silného kandidáta pro univerzální zpracování časových dat v multimodálních aplikacích.

# ZÁVĚR

Tato diplomová práce se zabývala návrhem a vyhodnocením univerzální transformerové architektury pro extrakci relevantních vlastností z různých typů vícekanálových časových signálů. V rámci návrhu byla vytvořena modulární architektura skládající se ze sdíleného extraktoru vlastností a specializovaných výstupních hlav pro různé typy downstreamových úloh. Model byl trénován a testován na třech odlišných modalitách – seismických, EKG a řečových datech – čímž byla ověřena jeho schopnost generalizace napříč doménami.

Výsledky experimentální části ukazují, že i při použití poměrně malého modelu je transformerová architektura je schopna dosahovat vysoké přesnosti a stability napříč všemi třemi zvolenými modalitami – seismickými, EKG a řečovými daty. Modely se čtyřmi

enkodérovými vrstvami dosahovaly opakovaně nejlepších výsledků ve smyslu kompromisu mezi výkonem a robustností. Ve většině případů navíc transformer předčil tradiční architektury typu CNN a RNN, zejména díky své schopnosti efektivně zachytit dlouhodobé časové závislosti.

Z praktického hlediska lze navržený systém využít v široké škále aplikací. V oblasti seismologie může sloužit pro včasnou detekci zemětřesení a přesné určování příchodů fází. V kardiologii může pomoci s automatizovanou analýzou EKG záznamů a detekcí arytmií. V oblasti zpracování řeči má potenciál pro robustní přepis mluveného slova v hlučných prostředích s více mluvčími. Modularita celého systému navíc umožňuje snadné rozšíření na další typy biosignálů a fyzikálních dat.

Přestože výsledky ukazují vysokou účinnost navrženého řešení, otevřené směry pro další výzkum zahrnují rozšíření trénovacích dat o další heterogenní modality, jako jsou biosignály (EEG, EMG), průmyslové senzory či finanční časové řady; zkoumání optimálního škálování architektury transformátoru včetně počtu enkodérů a attention hlav pro specifické úlohy; aplikaci metod komprese modelu, jako je kvantizace, knowledge distillation či pruning, pro nasazení v zařízeních s omezenými zdroji; implementaci pokročilých technik augmentace a regularizace, například SpecAugment, mixup či simulace reálných šumových podmínek; rozvoj self-supervised pretraining s moderními pretextovými úlohami (contrastive learning, maskované modelování) cílenými na různé domény; využití transfer learningu a few-shot learningu pro rychlou adaptaci na nové datové domény; a konečně důkladné hodnocení modelů z hlediska latence a energetické náročnosti pro reálné nasazení v průmyslových a zdravotnických systémech. Navržený přístup tak představuje robustní základ pro další rozvoj univerzálních modelů pro zpracování sekvenčních dat.

# POUŽITÁ LITERATURA


BAEVSKI, A., ZHOU, Y., MOHAMED, A., AULI, M. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Advances in Neural Information Processing Systems*, 2020, roč. 33, s. 12449–12460.

BENGIO, Yoshua, LAMBLIN, Patrice, POPOVICI, Dan a LAROCHELLE, Hugo. Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems*. 2007, s. 153–160.

BISHOP, C. M. Neural Networks for Pattern Recognition. *Oxford University Press*, 1995. ISBN 978-0-19-853864-6.

BISHOP, C. M. Pattern Recognition and Machine Learning. Berlin: *Springer*, 2006. ISBN 978-0387310732.

BOUSSELJOT, R.; KREISELER, D.; SCHNABEL, A. Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet. *Biomedizinische Technik*, 1995, 40(S1), s. 317–318. Dostupné z: https://www.physionet.org/physiobank/database/ptbdb/.

BROWN, T., MANN, B., RYDER, N., SUBBIAH, M., KAPLAN, J. D., DHARIWAL, P., NEELAKANTAN, A., SHYAM, P., SASTRY, G., ASKELL, A., AGARWAL, S., HERBLAIN, M., HENIGHAN, T., CHOWDHERY, A., GRACE, C., DIVER, A., et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, 2020, roč. 33, s. 1877–1901.

DEVLIN, J., CHANG, M. W., LEE, K. a TOUTANOVA, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT 2019*. 2019. DOI: 10.48550/arXiv.1810.04805. Dostupné z: https://arxiv.org/abs/1810.04805.

DOSOVITSKIY, A., BEYER, L., KOLTUN, V., ANTONIOU, T., ZHAO, Y., ZHAI, X., KRAHENBUHL, P. Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. 2020. arXiv preprint arXiv:2010.11929.

GONZALEZ, R., MANTOVANI, G. a WANG, D. A Comprehensive Survey of Learning Rate Schedulers in Deep Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, roč. 31, č. 12, s. 4793–4813. DOI: 10.1109/TNNLS.2019.2952264.

GOODFELLOW, Ian, BENGIO, Yoshua a COURVILLE, Aaron. *Deep Learning*. Cambridge: MIT Press, 2016. ISBN 978-0262035613.

GRAVES, A. Supervised Sequence Labelling with Recurrent Neural Networks. *Springer*, 2012. ISBN 978-3-642-24797-2.

GRAVES, A.; FERNÁNDEZ, S.; GÓMEZ, F.; SCHMIDHUBER, J. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, Pittsburgh, PA, USA, 2006, s. 369–376. Dostupné z: https://doi.org/10.1145/1143844.1143891.

HINTON, G.E. a SALAKHUTDINOV, R.R. Reducing the dimensionality of data with neural networks. *Science*, 2006, 313(5786), s. 504–507. DOI: 10.1126/science.1127647. Dostupné z: https://doi.org/10.1126/science.1127647.

HOCHREITER, S. a SCHMIDHUBER, J. Long short-term memory. *Neural Computation*. 1997, roč. 9, č. 8, s. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

HOCHREITER, S. a SCHMIDHUBER, J. Long Short-Term Memory. *Neural Computation*, 1997, roč. 9, č. 8, s. 1735–1780. DOI: https://doi.org/10.1162/neco.1997.9.8.1735.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 2002, 16, s. 321–357. Dostupné z: https://doi.org/10.1613/jair.953.

CHO, K., BOLUKBASI, T., CUN, Y. et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, s. 1724–1734. Dostupné z: https://arxiv.org/abs/1406.1078

JOLLIFFE, Ian T. *Principal Component Analysis*. 2. vyd. New York: *Springer*, 2002. ISBN 978-0387954424.

KHAN, S., NASEER, M., HAYAT, M., ZAMIR, S. W., KHAN, F. S. a SHAH, M. *Transformers in Vision: A Survey*. ACM Computing Surveys. 2019. DOI: 10.1145/3505244. Dostupné z: https://arxiv.org/abs/2101.01169.

KIM, Y. Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, s. 1746–1751. DOI: 10.3115/v1/D14-1181.

KINGMA, D. P. a BA, J. Adam: A Method for Stochastic Optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*. 2014. DOI: 10.48550/arXiv.1412.6980. Dostupné z: https://arxiv.org/abs/1412.6980.

KINGMA, D.P. a BA, J. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. San Diego, CA. 2015. Dostupné z: https://arxiv.org/abs/1412.6980.

KOBER, Jens, BAGNELL, J. Andrew a PETERS, Jan. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*. 2013, roč. 32, č. 11, s. 1238–1274. DOI: 10.1177/0278364913495721.

KRIZHEVSKY, Alex, SUTSKEVER, Ilya a HINTON, Geoffrey E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 2017, roč. 60, č. 6, s. 84–90. DOI: 10.1145/3065386.

KULLBACK, S. a LEIBLER, R. A. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 1951, 22(1), s. 79–86. Dostupné z: https://doi.org/10.1214/aoms/1177729694

LECUN, Y., BENGIO, Y. a HINTON, G. Deep learning. *Nature*. 2015, roč. 521, č. 7553, s. 436–444. DOI: 10.1038/nature14539.

LI, X., ZHANG, Y., WANG, C., LIU, S., ZHANG, Y. Convolutional Transformer with Domain Adversarial Learning for Multi-Channel Sleep Stage Classification. In *Proceedings of the 2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2023, s. 1234–1239. DOI: 10.1109/CISP-BMEI60920.2023.10373231.

LOSCH, J. a HAKKANI-TÜR, D. Learning Rate Schedulers: A Review. *Proceedings of the 31st International Conference on Neural Information Processing (ICONIP 2018)*. 2018. Springer. DOI: 10.1007/978-3-030-03203-1_64.

MCCULLOCH, W.S. a PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*. 1943, roč. 5, č. 4, s. 115–133. DOI: 10.1007/BF02478259. Dostupné z: https://link.springer.com/article/10.1007/BF02478259.

MIKOLOV, Tomas, CHEN, Kai, CORRADO, Greg a DEAN, Jeffrey. Efficient estimation of word representations in vector space. *Proceedings of the International Conference on Learning Representations (ICLR)*. 2013. Dostupné z: https://arxiv.org/abs/1301.3781.

MOUSAVI, S.M.; SHENG, Y.; ZHU, W.; BEROZA, G.C. STanford EArthquake Dataset (STEAD): A Global Data Set of Seismic Signals for AI. *IEEE Access*, 2019. DOI: 10.1109/ACCESS.2019.2947848. Dostupné z: https://doi.org/10.1109/ACCESS.2019.2947848.

REDDI, S. J., KALE, S. a KUMAR, S. On the Convergence of Adam and Beyond. *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*. 2018. DOI: 10.1109/ICASSP.2018.8461973. Dostupné z: https://openreview.net/forum?id=ryQu7f-RZ.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*. 1958, roč. 65, č. 6, s. 386–408. DOI: 10.1037/h0042519. Dostupné z: https://www.ling.upenn.edu/courses/cogs501/Rosenblatt1958.pdf.

RUMELHART, D.E., HINTON, G.E. a WILLIAMS, R.J. Learning representations by back-propagating errors. *Nature*. 1986, roč. 323, č. 6088, s. 533–536. DOI: 10.1038/323533a0. Dostupné z: https://www.nature.com/articles/323533a0.

SILVER, David et al. Mastering the game of Go without human knowledge. *Nature*. 2017, roč. 550, č. 7676, s. 354–359. DOI: 10.1038/nature24270.

SIMONYAN, Karen a ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. *Proceedings of the International Conference on Learning Representations (ICLR)*. 2015. Dostupné z: https://arxiv.org/abs/1409.1556.

SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I. a SALAKHUTDINOV, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research.* 2014, roč. 15, č. 1, s. 1929–1958. Dostupné z: https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf.

SUTTON, Richard S. a BARTO, Andrew G. *Reinforcement Learning: An Introduction.* 2. vyd. Cambridge: MIT Press, 2018. ISBN 978-0262039246. DOI: 10.5555/3312046.

VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A., KAISER, Ł., a POLOSUKHIN, I. Attention is all you need. *Proceedings of NeurIPS 2017*. Dostupné z: https://arxiv.org/abs/1706.03762.

XU, Rui a WUNSCH, Donald. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*. 2005, 16(3), s. 645–678. DOI: 10.1109/TNN.2005.845141.

YILMAZ, O.: Seismic Data Analysis. Houston: *Society of Exploration Geophysists*, 2001. Dostupné z: https://library.seg.org/doi/book/10.1190/1.9781560803910.

YU, F.; ZHANG, S.; FU, Y.; XIE, L.; ZHENG, S.; DU, Z.; HUANG, W.; GUO, P.; YAN, Z.; MA, B.; XU, X.; BU, H.M2MeT: The ICASSP 2022 Multi-Channel Multi-Party Meeting Transcription Challenge. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022. IEEE.https://www.openslr.org/119/.

ZHU, H., ZHOU, H., ZHANG, Z., DAI, N., ZHANG, W., LI, J., LEE, R. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *AAAI Conference on Artificial Intelligence*, 2021, roč. 35, č. 12, s. 11106–11115. DOI: https://doi.org/10.1609/aaai.v35i12.17324.

ZHU, W. a BEROZA, G. C. PhaseNet: A Deep-Neural-Network-Based Seismic Arrival Time Picking Method. arXiv preprint arXiv:1803.03211, 2018. Dostupné z: https://arxiv.org/abs/1803.03211.

ZHU, Xiaojin a GOLDBERG, Andrew B. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*. 2009, roč. 3, č. 1, s. 1–130. DOI: 10.2200/S00196ED1V01Y200906AIM006.

# SEZNAM PŘÍLOH

Příloha A: CD

**Příloha A**

**Příloha k diplomové práci**

Neuronová síť typu transformer pro extrakci vlastností ze signálu

Vojtěch Smetana

# CD

# OBSAH