

## EFEKTY UCZENIA SIĘ I TREŚCI PROGRAMOWE ZAJĘĆ

Kierunek: **Informatyka**

Poziom studiów: **studia drugiego stopnia**

Nazwa zajęć: **Analiza danych sportowych**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. Zna tematykę badawczą związaną z analizą i wnioskowaniem z dziedzinowych źródeł danych oraz potrafi przygotować środowisko programistyczne do przetwarzania danych tekstowych, obrazów i wideo oraz wykonywać w nim podstawowe operacje.

**w zakresie umiejętności:**

1. Potrafi budować repozytoria danych cyfrowych – w tym pobierać je wybranymi metodami, analizować i budować strukturę danych do przetwarzania.

2. Potrafi stosować metody czyszczenia i integracji danych pochodzących z różnych źródeł.

3. Potrafi stosować zaawansowane algorytmy przetwarzania danych, statystyki oraz prowadzić proces dedukcyjny.

4. Potrafi stosować oraz interpretować metryki związane ze „skutecznością” zdarzeń w spotkaniach sportowych w różnych dyscyplinach.

5. Potrafi wykonywać segmentację i rozpoznawanie obrazów.

6. Potrafi analizować materiały wideo w celu śledzenia obiektów.

7. Potrafi wykrywać i rozpoznawać tekst na obrazach.

8. Potrafi dobierać, stosować oraz modyfikować biblioteki wizualizacji danych w zależności od potrzeb analityka danych.

9. Zna i potrafi stosować najnowsze osiągnięcia w zakresie analizy danych sportowych, np. w celu sugerowania decyzji w czasie rzeczywistym na podstawie danych historycznych.

**Treści programowe dla zajęć:**

Wprowadzenie do przetwarzania danych sportowych: przygotowanie środowiska programistycznego (Azure) – przegląd modułów; podstawowe operacje związane z pozyskiwaniem i zapisywaniem danych, budowanie repozytoriów.

Omówienie istotnych cech opisujących zdarzenia sportowe w poszczególnych dyscyplinach (i wybranych ligach) – koszykówka – NBA, hokej na lodzie – NHL, piłka nożna i tenis. Analiza dostępnych repozytoriów danych: tekstowych, plików video, posegmentowanych obrazów.

Metody czyszczenia oraz integracji danych pochodzących z różnych (niezależnych) źródeł.

Przygotowanie danych do wnioskowania, zastosowanie metod statystycznych, w tym szeregów czasowych.

Metryki analizy danych o zdarzeniach sportowych. Omówienie oraz wybrane implementacje: metryki stosowane w sportach zespołowych, metryki rozproszenia graczy podczas gier zespołowych, metryki charakteryzujące zawodników (m. in. Entropia Shannona, podłużne i boczne przemieszczenia w sytuacji meczowej, entropia Kołmogorowa, eksploracja przestrzenna), metryki stosowane w ocenie taktyk drużyn sportowych.

Segmentacja i rozpoznawanie obrazów: segmentacja obrazów, rozpoznawanie wybranych cech obiektów, klasyfikacja obrazów.

Analiza wideo: przepływ optyczny, śledzenie obiektów, analiza równoważnych klas zdarzeń.

Ręczne oraz półautomatyczne tagowanie zdarzeń sportowych w zakresie zarówno danych graficznych, jak i danych video. Wykrywanie i rozpoznawanie tekstu.

Wizualizacja danych ze zdarzeń sportowych oraz analizy zachowania/zaangażowania zawodników podczas treningów i zawodów sportowych. Odwzorowanie boisk, symulacje oraz agregacja zdarzeń podobnych.

Przedstawienie wybranej interpretacji danych sportowych w postaci opracowania analitycznego. Zastosowanie wybranych narzędzi i bibliotek do budowy systemu rekomendacyjnego.

Nazwa zajęć: **Warsztat kompetencji miękkich**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie umiejętności:**

1. potrafi zwiększyć efektywność osobistą.

2. umie rozpoznawać swój potencjał: mocne i słabe strony.

3. zna i stosuje w praktyce narzędzia poprawiania efektywności zespołu, z którym pracuje.

4. rozpoznaje style myślenia oraz funkcjonowania.

5. skutecznie poprawia swoje relacje społeczne.
6. bardziej efektywnie współpracuje w zespole; docenia znaczenie pracy zespołowej.
7. skuteczniej komunikuje się z otoczeniem - współpracownikami, przełożonymi, bliskimi w życiu prywatnym
8. rozumie wybrane teorie i narzędzia z obszaru psychologii i socjologii.
9. zna metodologię i znaczenie inwentarzy osobowości oraz testów psychologicznych.
10. rozumie znaczenie kompetencji miękkich w życiu zawodowym i osobistym.
11. potrafi przeprowadzić autoprezentację i profesjonalne wystąpienie publiczne.
12. udziela profesjonalnej informacji zwrotnej. Rozumie znaczenie informacji zwrotnej w motywowaniu ludzi.
13. stosuje podstawowe narzędzia z obszaru coachingu i mentoringu.
14. potrafi wyciągać wnioski z porażek w celu podejmowania trafnych decyzji w przyszłości.

**w zakresie kompetencji społecznych:**

1. potrafi analizować własny styl funkcjonowania.
2. potrafi wytłumaczyć wybrane teorie i narzędzia z obszaru psychologii i socjologii.

**Treści programowe dla zajęć:**

Kompetencje społeczne. Kompetencje miękkie i twarde. Kompetencje przyszłości (2030). Leadership. Warsztat team-buildingu.  
Warsztat kompetencji interpersonalnych.  
Warsztat efektywności osobistej.  
Sztuka feedbacku. Model FUKO.  
Coaching. Implementacja najważniejszych narzędzi.  
Mentoring w pracy zawodowej.  
Autoprezentacja i wystąpienia publiczne.  
Sztuka błędzenia – jak wyciągać wnioski z porażek i niepowodzeń w życiu zawodowym (prywatnym).  
Model pogłębionych relacji międzyludzkich.

Nazwa zajęć: **Systemy dialogowe**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. Zna podstawowe rodzaje systemów dialogowych i potrafi wskazać występujące pomiędzy nimi różnice.
2. Zna i potrafi scharakteryzować strukturę semantyczną dialogu.
3. Zna i potrafi opisać architekturę systemu dialogowego ukierunkowanego na wykonanie zadania.
4. Zna i potrafi opisać architekturę systemu dialogowego end-to-end.

**w zakresie umiejętności:**

1. Potrafi zaimplementować prostego chatbota.
2. Potrafi pozyskiwać dane do budowy systemu dialogowego.
3. Potrafi napisać gramatykę dla parsera semantycznego.
4. Potrafi zastosować techniki uczenia maszynowego do prowadzenia analizy semantycznej wypowiedzi.
5. Potrafi konstruować reguły zarządzania dialogiem.
6. Potrafi zastosować techniki uczenia maszynowego do zarządzania dialogiem.
7. Potrafi posługiwać się technikami generowania tekstu.
8. Zna techniki ujednoznaczniania wypowiedzi w systemach ukierunkowanych na wykonywanie wielu zadań równocześnie.
9. Potrafi ocenić jakość systemu dialogowego z wykorzystaniem powszechnie przyjętych metod i metryk.
10. Potrafi zaimplementować system dialogowy w architekturze end-to-end.

**w zakresie kompetencji społecznych:**

1. Jest gotowy do przeprowadzenia analizy wymagań na potrzeby budowy systemu dialogowego.

**Treści programowe dla zajęć:**

Rodzaje systemów dialogowych. Systemy dialogowe ukierunkowane na wykonywanie zadań oraz chatboty  
Analiza wymagań, jakie powinien spełniać system dialogowy. Metody pozyskiwania danych do budowy systemów dialogowych.  
Eksperymenty typu "Czarnoksiężnik z Oz".  
Charakterystyka struktury semantycznej dialogu: akty mowy, ramy i sloty. Poprawki, wtręty i żądania resetu. Anotowane korpusy dialogów.  
Architektura systemu dialogowego ukierunkowanego na wykonanie zadania.

Parsing semantyczny z wykorzystaniem gramatyk.  
Parsing semantyczny z wykorzystaniem technik uczenia maszynowego.  
Zarządzanie dialogiem z wykorzystaniem reguł.  
Zarządzanie dialogiem z wykorzystaniem technik uczenia maszynowego.  
Generowanie odpowiedzi.  
Ujednoznacznianie wypowiedzi w systemach ukierunkowanych na wykonywanie wielu zadań równocześnie. Inteligentne asystenty osobiste.  
Ewaluacja systemów dialogowych – metody i metryki.  
Systemy dialogowe end-to-end. Zbiory danych, agenty, środowiska wykonania i symulatory.

Nazwa zajęć: **Zaawansowane algorytmy**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. zna i rozumie zaawansowane algorytmy sortowania i wyszukiwania.
2. zna i rozumie algorytmy generowania obiektów kombinatorycznych.
3. zna i rozumie zaawansowane algorytmy grafowe.
4. zna i rozumie wybrane algorytmy geometryczne.
5. zna i rozumie wybrane algorytmy tekstowe.
6. zna i rozumie wybrane algorytmy teorioliczne.

**w zakresie umiejętności:**

1. potrafi stosować zróżnicowane techniki algorytmiczne.
2. potrafi oceniać i porównywać efektywność algorytmów.
3. potrafi implementować zaawansowane algorytmy w wybranym języku programowania.
4. potrafi samodzielnie dalej kształcić się w zakresie algorytmiki.

**w zakresie kompetencji społecznych:**

1. jest gotów do pogłębiania zrozumienia istotności algorytmiki w praktycznych zastosowaniach.

**Treści programowe dla zajęć:**

Złożoność obliczeniowa algorytmów (notacja asymptotyczna). Podstawowe techniki algorytmiczne (metoda dziel i zwyciężaj, programowania dynamicznego, zachłanna, z nawrotami). Podstawowe struktury danych (stos, kolejka, kolejka priorytetowa).

Wyszukiwanie i sortowanie (sortowanie w czasie liniowym, statystyki pozycyjne).

Generowanie obiektów kombinatorycznych (podzbiory, permutacje, podziały zbiorów i liczb). Algorytmy pozycyjne i antypozycyjne.

Algorytmy grafowe: sortowanie topologiczne grafu, najkrótsze ścieżki (algorytm Dijkstry, Bellmana-Forda, Floyda-Warshalla, Johnsona, A\*), minimalne drzewa rozpinające (algorytm Prima, Kruskala), kod Prüfera, problemy przepływowe (metoda Forda-Fulkersona, metoda przedprzepływowa), skojarzenia w grafach dwudzielnych, problem stabilnych małżeństw.

Algorytmy geometryczne (położenie punktu względem prostej, płaszczyzny i wielokąta, przecinanie się odcinków, otoczka wypukła).

Algorytmy tekstowe: wyszukiwanie wzorca (algorytm Rabina-Karpa, Knutha-Morrisa-Pratta), odległość edycyjna, palindromy.

Algorytmy teorioliczne (rozszerzony algorytm Euklidesa, rozwiązywanie liniowych równań modularnych, testowanie pierwszości).

Nazwa zajęć: **Gry kombinatoryczne**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. Potrafi przedstawić proste gry kombinatoryczne w postaci ekstensywnej (drzewa gry).

**w zakresie umiejętności:**

1. Potrafi zastosować na drzewie gry analizę wstecz.
2. Potrafi rozstrzygnąć, kto ma strategię wygrywającą lub nieprzegrywającą w prostych grach, korzystając z metody kradzieży strategii i metody ruchów odpowiadających. Potrafi oceniać poprawność strategii.
3. Umie ocenić złożoność obliczeniową prostych algorytmów związanych z poszukiwaniem dobrej strategii w grach kombinatorycznych.
4. Umie przedstawiać tok swojego rozumowania w sposób zrozumiały dla słuchaczy.

**Treści programowe dla zajęć:**

Definicja gry kombinatorycznej. Przykłady gier kombinatorycznych, w tym HEX, szachy, kółko i krzyżyk. Drzewo gry, wartość gry. Analiza wstecz drzewa gry. Twierdzenie o strategii nieprzegrywającej. Ocena złożoności obliczeniowej naiwnych metod analizy drzewa gry.

Techniki pomagające w ocenie gry: metoda kradzieży strategii, metoda ruchów odpowiadających. Analiza poprawności strategii.  
Narzędzia teoriografowe w grach. Gra Shannona, gry ramseyowskie, wielowymiarowe gry w kółko i krzyżyk.  
Gry NIM.  
Gry kombinatoryczne a złożoność obliczeniowa i hipoteza  $P \neq NP$ . Przykłady gier NP-zupełnych, PCSPACE-zupełnych, EXPTIME-zupełnych.  
Algorytmy alfa-beta analizy drzewa gry. Deep Blue i inne programy szachowe.  
Przeszukiwanie Monte Carlo drzewa gry (MCST). AlphaGo.

Nazwa zajęć: **Zaawansowany UX**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. metodę Design Thinking do projektowania produktów cyfrowych.
2. heurystyki Nielsena i inne prawa UX; konieczność tworzenia produktów intuicyjnych, użytecznych i estetycznych.

**w zakresie umiejętności:**

1. przeprowadzić badania jakościowe z użytkownikami i rozumie ich znaczenie w procesie tworzenia produktów cyfrowych; przeanalizować wyniki badań i wykorzystać je do ulepszenia tworzonego produktu.
2. stworzyć user flow i wire frames; przeprowadzić testy użyteczności na stworzonych prototypach LO-FI; przeanalizować wyniki testów i wykorzystać je do ulepszenia tworzonego produktu.
3. wykorzystać funkcjonalności Figmy w tworzeniu prototypów HI-FI.
4. wykorzystać zasady kompozycji oraz zasady tworzenia podstawowych komponentów UI do stworzenia estetycznego projektu UI.

**w zakresie kompetencji społecznych:**

1. zaprezentowania swojego projektu, umotywowania podjętych decyzji projektowych, uzasadnienia swojego punktu widzenia i przekonania do niego odbiorcy produktu lub usługi; udzielenia konstruktywnej informacji zwrotnej; projektowania narzędzia cyfrowego, które odpowiada na potrzeby użytkowników, jest intuicyjne, użyteczne i estetyczne.

**Treści programowe dla zajęć:**

Metoda Design Thinking, podstawowe pojęcia z obszaru UX/UI designu i wprowadzenie do UX research.

UX research, heurystyki Nielsena i dostępność.

Analiza danych z badań, odkrywanie problemów, definiowanie wyzwań, ideacja.

Prototypowanie, wire frames, user flows i wprowadzenie do testowania.

Pierwsza iteracja na bazie wyników testów, wprowadzenie do Figmy.

Prototypowanie w figmie, kompozycja, elementy UI.

Kompetencje miękkie w UX designie.

Nazwa zajęć: **Polityka cyberbezpieczeństwa. Cybernetyczne działania wojenne**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. rozumie znaczenie cyberbezpieczeństwa dla całościowego problemu procesu zapewnienia odpowiedniego poziomu bezpieczeństwa państwa w XXI w.
2. rozumie możliwość wykorzystania przez obce państwa lub struktury niepaństwowe domeny cyber do działań szpiegowskich względem własnych zasobów informacji, a także zasobów znajdujących się po stronie instytucji/firmy, w której pracuje.
3. rozumie zróżnicowane postawy względem cyberbezpieczeństwa definiowane przez system polityczny danego państwa oraz formy zabezpieczenia jego interesów narodowych.
4. rozumie potrzebę przekładania własnych działań zawodowych i pozazawodowych na stworzenie odpowiedniego poziomu cyberbezpieczeństwa oraz odporności systemu obronnego państwa.
5. rozumie wymóg poznawania innych niż techniczne aspekty cyberbezpieczeństwa.
6. rozumie potrzebę współpracy z osobami zajmującymi się cyberbezpieczeństwem z różnych perspektyw naukowych oraz zawodowych.
7. zna podstawowe pojęcia oraz zjawiska z zakresu bezpieczeństwa i cyberbezpieczeństwa oraz potrafi wskazać cechy cyberprzestrzeni.
8. zna "miękkie" wyzwania i zagrożenia dla polityki cyberbezpieczeństwa.
9. rozumie istotę "twardych" wyzwań i zagrożeń dla polityki bezpieczeństwa oraz potrafi scharakteryzować ich przykłady, w tym wskazać znaczenie czy następstwa.

**w zakresie umiejętności:**

1. zna formy i metody zwalczania cyberzagrożeń; potrafi ocenić ich skuteczność wskazując mocne i słabe strony; umie prognozować nowe potencjalne zagrożenia.

**w zakresie kompetencji społecznych:**

1. zauważa możliwość wykorzystania przez obce państwa lub struktury niepaństwowe domeny cyber do działań szpiegowskich oraz wojskowych wymierzonych w jego państwo.

2. zauważa rolę i znaczenie domeny cyber dla całościowego systemu obronnego państwa w warunkach nowych sposobów prowadzenia działań zbrojnych i uderzeń niekinetycznych w toku operacji hybrydowych.

3. potrafi wskazać przyczyny, przejawy oraz następstwa rewolucji informatycznej; rozumie ich kontekst społeczny oraz technologiczny, a także powiązania ze sferą bezpieczeństwa.

4. potrafi wytłumaczyć formy i przykłady ich stosowania.

**Treści programowe dla zajęć:**

Nowa domena działań, czyli jak państwa oraz sojusze obronne postrzegają cyberprzestrzeń w ujęciu wywiadów oraz wojska.

Kryptologia i rozwój SIGINT-u jako kluczowego źródła pozyskiwania danych wywiadowczych w XX i XXI w., czyli o tym, czy cyberszpiegostwo staje się niezbędne.

Cyberprzestrzeń pełnoprawną domeną prowadzenia działań zbrojnych, czyli jak współczesne konflikty zbrojne przechodzą do cyberprzestrzeni – od Iraku 1991 do Górskiego Karabachu 2020.

C4ISTR, czyli gdzie cyberbezpieczeństwo zaczyna decydować o współczesnych sukcesach oraz porażkach sił zbrojnych.

Chińsko-amerykańska „wojna” o cyberprzestrzeń, czyli jak kształt rywalizacji mocarstw globalnych jest definiowany względami cyberbezpieczeństwa.

Państwo Izrael, czyli przykład państwa, w którym wojsko i służby specjalne rozumieją strategiczne potrzeby oraz możliwości wykorzystania domeny cyber na Bliskim Wschodzie.

Cyberprzestrzeń i WRE czyli jak zrodził się rosyjski pomysł na ograniczenie dysproporcji względem państw NATO.

Wprowadzenie do polityki bezpieczeństwa i cyberbezpieczeństwa.

Rewolucja informatyczna i główne jej tendencje.

"Miękkie" zagrożenia i wyzwania dla polityki cyberbezpieczeństwa (np. cyfrowa przepaść, cyberszpiegostwo, hacking itp.).

"Twarde" zagrożenia i wyzwania dla polityki cyberbezpieczeństwa (np. cyberwojny, cyberterroryzm itp.).

Zwalczanie cyberzagrożeń oraz prognoza ich przyszłej ewolucji.

**Nazwa zajęć: Język angielski C11**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie umiejętności:**

1. potrafi w sposób przystępny przedstawić fakty z zakresu informatyki, porozumiewać się przy użyciu różnych technik w środowisku zawodowym oraz w innych środowiskach w języku angielskim.

2. potrafi przygotować dokumentację, opracowania i raporty w języku angielskim.

3. potrafi przygotować prace akademickie - pisemne w języku angielskim, dotyczące zagadnień teoretycznych i praktycznych informatyki.

4. potrafi posługiwać się językiem obcym na poziomie C1 Europejskiego Systemu Opisu Kształcenia Językowego oraz zna język angielski w stopniu umożliwiającym czytanie ze zrozumieniem dokumentacji oraz artykułów dotyczących informatyki.

5. zna i potrafi stosować słownictwo dotyczące informatyki.

**Treści programowe dla zajęć:**

Praca podczas zajęć (nad wspólnym zadaniem, wymiana zdobytych informacji, zadania typu information gap).

Tłumaczenie na język angielski oraz zwrótnie na język polski fragmentów dokumentacji i opracowań z wykorzystaniem cech stylu naukowego.

Mini-prezentacje podczas każdego zajęcia prezentujące wyniki zadania domowego oraz wyniki pracy w grupie (in-class). Prezentacja zaliczeniowa na koniec semestru podsumowujące wyniki pracy podczas semestru na samodzielnie wybrany i opracowany temat z dziedziny informatyki.

Praca z artykułami (IT) oraz artykułami naukowymi (IT-study, case study) dotyczącymi informatyki.

Praca ze słownictwem specjalistycznym z użyciem tekstów oraz prezentacji audio/video o zróżnicowanej tematyce i stopniu trudności. Wykonywanie ćwiczeń pisemnych i ustnych aktywizujących zapamiętywanie słownictwa poprzez jego użycie w analogicznym kontekście.

**Nazwa zajęć: Bezpieczeństwo systemów komputerowych**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. zna podstawowe metody rozwiązywania problemów bezpieczeństwa stosowane w złożonych systemach informatycznych.
2. przestrzega podstawowych zasad bezpieczeństwa przy używaniu i implementacji systemów informatycznych.
3. jest świadomy prawnych wymogów stawianych przed bezpiecznymi systemami informatycznymi.
4. posiada wiedzę o narzędziach stosowanych do zapewniania bezpieczeństwa systemów komputerowych.

**w zakresie umiejętności:**

1. potrafi wykorzystywać odpowiednio dobrane metody rozwiązywania problemów bezpieczeństwa stosowane w złożonych systemach informatycznych.
2. potrafi ocenić daną implementację pod kątem potencjalnych zagrożeń i zaproponować metody ich ograniczeń.
3. potrafi dobierać odpowiednie narzędzia stosowane do zapewniania bezpieczeństwa systemów komputerowych.

**Treści programowe dla zajęć:**

Podstawowe pojęcia kryptologii i inżynierii bezpieczeństwa, poufność danych, terminologia, proste szyfry.

Integralność danych, tworzenie używalnych kopii zapasowych. Strategie działań odtworzeniowych.

Tryby szyfrowania szyfrów symetrycznych, konsekwencje doboru trybów szyfrowania. Szyfrowanie dysków w locie na przykładzie programu Veracrypt. Plausible deniability.

Klasyfikacja malware, techniki tworzenia wirusów. Zaawansowane mechanizmy używane w złośliwym oprogramowaniu na przykładzie Stuxnet. Ochrona antywirusowa.

SPAM i UCE. Sposoby przeciwdziałania na poziomie serwerów i klientów.

Bezpieczne przechowywanie haseł. Zabezpieczanie systemów hasłami na przykładzie http simple authentication.

Atak typu buffer overflow na integralność pamięci na przykładzie stack smashing.

Atak SQL injection na prosty program. Sposoby obrony przed atakami po stronie klienta bazy danych i silników bazodanowych.

**Nazwa zajęć: Praktyczne zastosowania chmury obliczeniowej**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. Rozumie hierarchię modeli chmury obliczeniowej
2. Rozumie model IaaS chmury obliczeniowej
3. Rozumie model PaaS chmury obliczeniowej
4. Rozumie model FaaS/Serverless chmury obliczeniowej
5. Rozumie model SaaS chmury obliczeniowej
6. Zna typy chmury obliczeniowej
7. Rozumie znaczenie kompleksowej publicznej chmury obliczeniowej

**w zakresie umiejętności:**

1. Zna podstawowy obliczeń w chmurze
2. Zna dostawców chmury w modelu PaaS
3. Potrafi wykorzystać interfejs CLI do konfiguracji zasobów chmury
4. Potrafi przygotować i wdrożyć prostą aplikację w modelu PaaS
5. Potrafi zautomatyzować proces wdrażania systemu w modelu PaaS
6. Zna dostawców chmury w modelu IaaS
7. Potrafi skonfigurować sieć prywatną w chmurze
8. Potrafi skonfigurować zasoby dyskowe w chmurze
9. Potrafi obsługiwać narzędzie cloud-init
10. Potrafi wykorzystać chmurę do obliczeń na kartach graficznych (GPU)
11. Potrafi wykorzystać interfejs webowy do konfiguracji zasobów chmury
12. Potrafi skonfigurować system równoważenia obciążenia
13. Potrafi skonfigurować automatyczne skalowanie infrastruktury w chmurze
14. Zna usługi dostępne w modelu SaaS
15. Potrafi wykorzystać zarządzaną w modelu SaaS bazę danych w chmurze
16. Zna dostępne usługi w modelu FaaS
17. Potrafi zaimplementować webserwis w modelu Serverless

18. Rozumie potrzebę ujednoczenia i automatyzacji procesu przydzielania zasobów (IaaS)
19. Potrafi wykorzystać chmurę obliczeniową do budowy złożonego systemu informatycznego
20. Potrafi wykorzystać chmurę do obliczeń wymagających dużej ilości zasobów
21. Potrafi przeprowadzić analizę kosztów infrastruktury chmurowej w różnych modelach

**Treści programowe dla zajęć:**

Wprowadzenie do obliczeń w chmurze, hierarchia modeli chmury obliczeniowej: IaaS, PaaS, FaaS, SaaS, dostawcy i typy chmury.

Przedstawienie modelu Platform as a Service (PaaS), umówienie dostawców (np. Heroku), wykorzystanie interfejsu CLI do konfiguracji zasobów, przygotowanie i wdrożenie prostej aplikacji w modelu PaaS, automatyzacja procesu wdrażania poprzez integrację z Git.

Przedstawienie modelu Infrastructure as a Service (IaaS), omówienie dostawców (np. Hetzner, Linode), wykorzystanie CLI do konfiguracji zasobów, konfiguracja sieci prywatnej, zasobów dyskowych, obsługa narzędzia cloud-init, wykorzystanie chmury do obliczeń na kartach graficznych (GPU).

Omówienie kompleksowej publicznej chmury obliczeniowej (np. AWS, Azure), konfiguracja z wykorzystaniem przeglądarki: sieci prywatnej, zasobów dyskowych, systemu równoważenia obciążenia (load balancer), automatycznego skalowania infrastruktury.

Przedstawienie modelu Software as a Service (SaaS), umówienie dostępnych usług, omówienie modelu na podstawie zarządzanej bazy danych w chmurze (np. AWS DynamoDB).

Przedstawienie modelu Serverless/Function as a Service (FaaS), omówienie dostępnych usług (np. AWS Lambda, Azure Functions), implementacja REST API z modelu Serverless.

Projekt obejmujący przygotowanie złożonego systemu informatycznego wdrożonego w całości w chmurze i korzystającego z pełni jej możliwości. Omówienie problemu ujednoczenia i automatyzacji procesu przydzielania zasobów (Infrastructure as a Code, IaC).

nowa bezsensowna treść

Nazwa zajęć: **Przetwarzanie równoległe i strumieniowe**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna narzędzia do edycji i tworzenia kodu oraz śledzenia wywołania programów wielowątkowych.
2. rozumie pojęcie programowania równoległego i współbieżnego oraz zna problemy z nimi związane.
3. zna pojęcia brokera danych, kolejki, rozproszonej kolejki i umie wskazać przykłady ich realizacji.
4. rozumie różnice między przetwarzaniem strumieniowym a przetwarzaniem batchowym oraz zna ich zastosowania.
5. zna podstawowe biblioteki przetwarzania strumieniowego .
6. zna pojęcie okna czasowego.
7. zna podstawowe programowe metody zarządzania wątkami.
8. zna kolekcje bezpieczne wielowątkowo.

**w zakresie umiejętności:**

1. potrafi przełożyć problemy opisane językiem naturalnym na problemy przetwarzania wielowątkowego oraz zaprojektować algorytm rozwiązujący zadany problem.
2. potrafi krytycznie ocenić poprawność algorytmów i rozwiązań działających w środowisku wielowątkowym.
3. potrafi zaimplementować proste przetwarzanie danych przy pomocy bibliotek przetwarzania strumieniowego.
4. potrafi wykorzystać pojęcie okna czasowego dla danych czasowych.
5. potrafi napisać własny egzekutor wątków.
6. umie skutecznie wykorzystać w praktyce kolekcje bezpieczne wielowątkowo.

**Treści programowe dla zajęć:**

Przygotowanie środowiska programistycznego, wybór edytora, repozytorium kodu, pojęcie wątku, przetwarzania równoległego. Sugerowana ścieżka: język Java, środowisko IntelliJ, repozytorium Git, zarządzanie zależnościami Maven.

Programowanie funkcyjne i strumieniowe, metody map, filter, struktury danych, debuggowanie programów.

Implementacja wątków w Java, Wait, Notify, Synchronizacja, Semafor

Anomalie związane z przetwarzaniem równoległym, zagłódzenie, race condition, zakleszczenie

Przetwarzanie zadań na wątkach – zadanie projektowe - struktura i implementacja, wykorzystywane narzędzia.

Projekt I - sprawdzenie

Pule wątków, egzekutory, bariery, liczniki, blokady.

Własne implementacje egzekutorów, przykłady z sieci, zastosowania, wydajność

Algorytmy równoległe  
Projekt II - sprawdzenie  
Dane sensoryczne, serwer danych, broker danych  
Apache Flink – wprowadzenie, podstawowe pojęcia  
Apache Flink – okna czasowe  
Apache Flink – zaawansowane elementy  
Projekt III - sprawdzenie

Nazwa zajęć: **Bezpieczeństwo informatyczne**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna teoretyczne i praktyczne aspekty kryptologii oraz narzędzia i metody zapewniające bezpieczeństwo komputerowe projektowanych i zarządzanych systemów informatycznych.

**w zakresie umiejętności:**

1. potrafi korzystać z narzędzi zapewniających bezpieczeństwo informatyczne oraz demonstrujących jego potencjalne słabe punkty.

**w zakresie kompetencji społecznych:**

1. jest świadomy problemów prawnych i zagadnień etycznych związanych z atakami na systemy informatyczne.

**Treści programowe dla zajęć:**

Terminologia kryptologii, definicje bezpieczeństwa. Ewolucja metod łamania zabezpieczeń od czasów historycznych do współczesnych.

Szyfry symetryczne. Ataki na szyfry. Kryteria oceny siły szyfru. Sposoby budowania szyfrów. Schemat Feistela.

Funkcje jednokierunkowe, ataki na nie i zastosowania. Redukcja trudności ataków i kompromisy między złożonością pamięciową a czasową. Rainbow tables - metody konstrukcji i użycia.

Losowość. Generatory pseudolosowe. Niebezpieczeństwa systemów o małej entropii. Sprzętowe generowanie danych losowych. Zastosowania.

Uwierzytelnianie, hasła, kryteria doboru, strategie łamania haseł przez hardware (FPGA/ASIC). Key stretching. Challenge-response, hasła jednorazowe, schemat Lamporta. Biometria, karty chipowe.

Zarządzanie kluczami, uzgadnianie kluczy, protokół Diffie-Hellmana, infrastruktura klucza publicznego, certyfikaty, unieważnianie kluczy, wykluczanie użytkowników, podział sekretów.

Typowe schematy ataków programistycznych, buffer overflow, SQL injection, cross-site scripting, reply attack, atak z rozmnożeniem, reflection attack.

Prawidłowe traktowanie danych od użytkownika - podział na tainted/clean. Urządzenia czarne i czerwone. Sprzętowa ochrona przed atakami łamiącymi barierę dane/kod. Wirtualizacja.

Social engineering i ataki łamiące abstrakcję modelu bezpieczeństwa. Wirusy, wormy, denial of service, spam, phishing. Podstęp klawiatury, monitora. Sposoby ochrony.

Inwigilacja. Dobre praktyki tworzenia systemów inwigilacji. Inwigilacja wizualna. DORI. Sprzętowe ograniczenia współczesnych urządzeń.

Przechowywanie i bezpieczne niszczenie danych. Standardy dla różnych nośników. Szyfrowanie systemu plików - tryby szyfrowania XEX, XTS. Urządzenia tamper proof i tamper resistant.

Informacje uzyskane pośrednio (side-channel attacks). Analiza poboru mocy (power analysis). Ulot elektromagnetyczny (tempest), ulot optyczny (optical tempest). Urządzenia czarne i czerwone.

Nazwa zajęć: **Inżynieria wiedzy**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. Zna pojęcia związane z zarządzaniem wiedzą.

2. Zna wybrane historyczne narzędzia i metody przeznaczone do reprezentacji i organizacji wiedzy.

3. Zna obszary zastosowania języków informacyjnych oraz różne sposoby organizacji wiedzy, w tym różne typy relacji pomiędzy pojęciami.

4. Zna sposoby definiowania i formalizacji ontologii. Zna wybrane metody i narzędzie wykorzystywane przy budowie i przetwarzaniu ontologii.

5. Zna sposoby definiowania grafów wiedzy (ang. knowledge graph).

6. Zna pojęcia związane z Linked data oraz zna metody i narzędzie wykorzystywane do budowy zasobów tego typu. Zna obszary zastosowania Linked data we współczesnej gospodarce.

7. Zna cykl życia produktów opartych na zasobach wiedзовych oraz zna zasady uzyskania zasobów tego typu wysokiej jakości.



8. Zna strukturę leksykalnych baz danych typu wordnet oraz ontologii typu wordnet. Zna podstawowe zasady budowy zasobów tego typu.

**w zakresie umiejętności:**

1. Potrafi wskazać obszary zastosowania wybranych metod reprezentacji i organizacji wiedzy.
2. Umie wykorzystać język logiki w reprezentacji wiedzy.
3. Potrafi wskazać obszary zastosowania wybranych języków informacyjnych z uwzględnieniem kontekstu i potrzeb budowanego systemu informatycznego opartego o wiedzę.
4. Potrafi wskazać obszary zastosowania ontologii, ich struktury oraz narzędzia wykorzystywane do ich budowy.
5. Potrafi wskazać obszary zastosowania grafów wiedzy oraz rozstrzygać o postaci zasobów w tej postaci.
6. Potrafi budować ontologie w wybranym formacie oraz przy pomocy odpowiednich metod i narzędzi.
7. Potrafi wskazać obszary wykorzystania Linked data i budować zasoby tego typu.
8. Potrafi wskazać różnice przy budowie leksykalnych baz danych typu wordnet oraz ontologii typu wordnet
9. Potrafi wskazać obszary zastosowania i narzędzia wykorzystywane przy budowie leksykalnych baz danych typu wordnet oraz ontologii typu wordnet.

**Treści programowe dla zajęć:**

Wprowadzenie do zarządzania wiedzą (dane, informacja, wiedza, składnia, semantyka, taksonomie, ontologie).

Przegląd wybranych historycznych narzędzi i metod przeznaczonych do reprezentacji i organizacji wiedzy (sieci semantyczne, ramy, grafy konceptualne) i ich wykorzystanie w systemach eksperckich. Implementacja przykładowej bazy wiedzy w języku logiki.

Języki informacyjne w organizowaniu informacji (systemy klasyfikacyjne, słowniki, tezaury, tagowanie, metadane w opisywaniu treści) i przykładowe zasoby tego typu (Geonames, Getty AAT). Różne typy relacji (synonimia, polihierarchiczność, polirelacyjność, podobieństwo, powiązania). Analiza wybranych obszarów zastosowań.

Ontologie, różne sposoby ich definiowania i zapisu. Wykorzystanie w organizacji informacji i zarządzaniu wiedzą. Przykładowe ontologie (Cidoc CRM, Sumo, CYC, ontologie dziedzinowe). Języki zapisu i przetwarzania ontologii (RDF, OWL, SPARQL). Budowa grafów wiedzy (knowledge graph) i obszary ich wykorzystania we współczesnej gospodarce opartej na wiedzy. Implementacja przykładowej ontologii.

Linked data jako przykład zastosowania inżynierii wiedzy i zarządzania wiedzą.

Cykl życia produktów opartych na zasobach wiedzowych. Uzyskanie zasobów wiedzowych wysokiej jakości.

Leksykalne bazy danych typu wordnet i ich przykłady (WordNet, plWordnet). Ontologie typu wordnet jako przykład polihierarchicznego i polirelacyjnego systemu zarządzania wiedzą na przykładzie ontologii PMAH. Analiza wybranych obszarów zastosowań i narzędzi wykorzystywanych przy przetwarzaniu zasobów tego typu.

Nazwa zajęć: **Inżynieria uczenia maszynowego**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. Zna specyfikę rozwoju modeli uczenia maszynowego.
2. Zna i rozumie podstawowe pojęcia i koncepcje związane z procesem ciągłej integracji.
3. Zna popularne biblioteki uczenia maszynowego i różnice między nimi.
4. Rozumie podstawę działania kontenerów w systemie operacyjnym Linux i zna ich zastosowania.
5. Rozumie, na czym polega proces ciągłej integracji, potrafi podać przykład takiego procesu.
6. Zna popularne narzędzia do kontroli danych, parametrów i wyników eksperymentów uczenia maszynowego.

**w zakresie umiejętności:**

1. Potrafi przeprowadzić trening i ewaluację prostego modelu uczenia maszynowego przy użyciu wybranej biblioteki.
2. Potrafi przygotować dane trenujące model uczenia maszynowego i podzielić je na podzbiory i rozumie na czym polega ewaluacja krzyżowa.
3. Potrafi korzystać z narzędzia konteneryzacji Docker.
4. Potrafi utworzyć i skonfigurować zadanie w systemie ciągłej integracji Jenkins.
5. Potrafi zintegrować system ciągłej integracji z systemem kontroli wersji.
6. Potrafi stworzyć proces ciągłej integracji modelu uczenia maszynowego w systemie ciągłej integracji Jenkins.

7. Potrafi stworzyć wykres wizualizujący wyniki działania modelu uczenia maszynowego.
8. Potrafi zintegrować proces wizualizacji wyników ewaluacji modelu uczenia maszynowego z procesem ciągłej integracji w środowisku Jenkins.
9. Potrafi zastosować wybrane narzędzie do kontroli danych, parametrów i wyników eksperymentów uczenia maszynowego.
10. Potrafi zintegrować wybrane narzędzie do kontroli danych, parametrów i wyników eksperymentów uczenia maszynowego z narzędziem do wizualizacji wyników. Potrafi zinterpretować wyniki przedstawione w tym narzędziu.
11. Potrafi zintegrować wszystkie elementy kompletnego systemu ciągłej integracji uczenia maszynowego i używać tego systemu do rozwoju modeli uczenia maszynowego.

**Treści programowe dla zajęć:**

Wprowadzenie podstawowych koncepcji ciągłej integracji, konteneryzacji i uczenia maszynowego. Przedstawienie specyfiki procesów rozwoju, testowania i integracji modeli uczenia maszynowego. Przegląd najpopularniejszych bibliotek używanych w uczeniu maszynowym. Sposoby podziału danych trenujących i ewaluacji modeli uczenia maszynowego. Przygotowanie danych i bazowego rozwiązania dla wybranego problemu uczenia maszynowego. Konteneryzacja przy użyciu narzędzia Docker. Stworzenie obrazu Docker zawierającego środowisko potrzebne do przeprowadzania prostego eksperymentu uczenia maszynowego dla wybranego wcześniej problemu. Wprowadzenie do ciągłej integracji. Przedstawienie istniejących środowisk ciągłej integracji. Zapoznanie z możliwościami i zastosowaniami ciągłej integracji na przykładzie systemu Jenkins. Stworzenie pierwszego zadania w systemie Jenkins uruchamiające prosty eksperyment uczenia maszynowego. Połączenie kilku zadań Jenkins ze sobą w całość automatyzującą wielostopniowy proces uczenia maszynowego. Wizualizacja wyników eksperymentów za pomocą biblioteki do tworzenia wykresów, np. Matplotlib, stworzenie zadania Jenkins realizującego wizualizację. Przegląd narzędzi do kontroli danych, parametrów i wyników eksperymentów: DVC, Sacred, MLFlow. Kontrola parametrów i wyników eksperymentów uczenia maszynowego przy pomocy wybranego narzędzia. Wizualizacja wyników eksperymentów uczenia maszynowego przy pomocy wybranego narzędzia. Integracja kompletnego procesu trenowania i ewaluacji modelu uczenia maszynowego przy użyciu poznanych wcześniej narzędzi. Podsumowanie wyników pracy podczas kursu.

**Nazwa zajęć: Zaawansowane struktury danych**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. zna i rozumie zasady działania i zastosowania kopców dwumianowych i kopców Fibonacciego.
2. zna i rozumie zasady działania i zastosowania zrównoważonych drzew poszukiwań binarnych i samoorganizujących się drzew poszukiwań binarnych.
3. zna i rozumie zasady działania i zastosowania B-drzew.
4. zna i rozumie zasady działania i zastosowania drzew van Emde-Boasa.
5. zna i rozumie zasady działania i zastosowania drzew poszukiwań pozycyjnych.
6. zna i rozumie zasady działania i zastosowania drzew sufiksowych i grafów podstów.
7. zna i rozumie zasady działania i zastosowania tablic z haszowaniem.

**w zakresie umiejętności:**

1. potrafi wykorzystywać zróżnicowane struktury danych do rozwiązywania problemów algorytmicznych.
2. potrafi oceniać i porównywać efektywność operacji na wybranych strukturach danych.
3. potrafi implementować zaawansowane struktury danych w wybranym języku programowania.
4. potrafi samodzielnie dalej kształcić się w zakresie struktur danych.

**w zakresie kompetencji społecznych:**

1. jest gotów do pogłębiania zrozumienia istotności wyboru odpowiednich struktur danych w praktycznych zastosowaniach.

**Treści programowe dla zajęć:**

Elementarne struktury danych (stos, kolejka, lista). Własności drzew. Kopiec binarny, drzewo poszukiwań binarnych.

Analiza kosztu zamortyzowanego.

Kopce dwumianowe, kopce Fibonacciego.

Zrównoważone drzewa poszukiwań binarnych (drzewa AVL, drzewa czerwono-czarne).  
Samoorganizujące się drzewa poszukiwań binarnych.  
B-drzewa.  
Drzewa van Emde Boasa.  
Drzewa poszukiwań pozycyjnych (RST, trie, Patricia).  
Drzewa sufiksowe, grafy podstów.  
Tablice z haszowaniem.

Nazwa zajęć: **Złożoność obliczeniowa**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. zna podstawowe pojęcia dotyczące języków, problemów oraz schematów kodowania.
2. zna definicję deterministycznej maszyny Turinga.
3. zna definicję niedeterministycznej maszyny Turinga.
4. zna główne daty w rozwoju teorii złożoności.
5. zna podstawowe klasy algorytmów oraz ich własności.
6. zna definicje klas P i NP oraz rozumie związki między nimi.
7. zna definicje transformacji wielomianowej oraz problemu NP-zupełnego.
8. zna twierdzenia Cooka oraz rozumie jego konsekwencje.
9. zna sformułowania głównych problemów NP-zupełnych.
10. zna definicje transformacji pseudowielomianowej oraz problemu silnie NP-zupełnego.
11. zna sformułowania głównych problemów silnie NP-zupełnych.
12. zna pojęcia słabej i silnej NP-zupełności, ich wzajemne relacje oraz ich konsekwencje.
13. zna związki między NP-zupełnością a NP-trudnością.
14. zna główne metody dowodzenia NP-zupełności i NP-trudności.
15. zna definicje algorytmu aproksymacyjnego oraz schematu aproksymacyjnego oraz główne metody ich konstrukcji, potrafi podać przykłady takich algorytmów i schematów.

**w zakresie umiejętności:**

1. umie napisać proste programu dla modelu deterministycznej maszyny Turinga.
2. potrafi wykorzystać model niedeterministycznej maszyny Turinga w rozumowaniach.
3. potrafi określić przynależność konkretnego algorytmu do jednej z podstawowych klas algorytmów.
4. potrafi zastosować definicję transformacji wielomianowej oraz problemu NP-zupełnego w rozumowaniach.
5. potrafi zastosować definicje transformacji pseudowielomianowej i problemu silnie NP-zupełnego w rozumowaniach.
6. potrafi zastosować główne metody dowodzenia NP-zupełności i NP-trudności w odniesieniu do wybranych problemów.

**Treści programowe dla zajęć:**

Podstawowe pojęcia i definicje. Pojęcie problemu, problemy nierozstrzygalne, problemy algorytmiczne, problemy optymalizacyjne, problemy decyzyjne, problemy przeszukiwania, przykłady problemów różnych typów. Schematy kodowania, kodowanie unarne, kodowanie binarne, reguły dotyczące schematów kodowania, instancje problemów. Deterministyczna i niedeterministyczna maszyna Turinga, inne główne teoretyczne modele obliczeń, związki między różnymi modelami obliczeń. Pojęcie języka, redukcje języków, rozpoznawania języków, języki a problemy.

Główne daty w rozwoju teorii złożoności. Problemy Hilberta. Twierdzenie Gödla o niezupełności arytmetyki Peano. Powstanie teoretycznych modeli obliczeń i teorii obliczalności. Nierozstrzygalność problemu stopu maszyny Turinga. Znaczenie prac Cobhama, Hartmanisa, Stearnsa i Edmondsa dla teorii złożoności. Twierdzenie Cooka. Hipoteza  $P = ? NP$ . Lista 21 problemów Karpa. Teoria złożoności współcześnie.

Główne klasy algorytmów i ich własności. Algorytmy deterministyczne i niedeterministyczne. Algorytmy dokładne i przybliżone. Algorytmy optymalne i suboptymalne. Algorytmy wielomianowe, pseudowielomianowe i wykładnicze. Algorytmy offline i online. Złożoność obliczeniowa i pamięciowa algorytmu. Złożoność algorytmu a jego efektywność w praktyce.

Klasy P i NP. Problemy wielomianowe, definicja klasy P. Problemy weryfikowalne w wielomianowym czasie, definicja klasy NP. Przykłady problemów z klas P i NP. Związki między klasami P i NP. Hipoteza  $P = ? NP$ .

Transformacje wielomianowe i słaba NP-zupełność. Transformacja wielomianowa, definicja problemu NP-zupełnego. Przykłady problemów NP-zupełnych. Metody dowodzenia NP-zupełności.

Twierdzenie Cooka. Problem SAT. Główne odmiany problemu SAT. Twierdzenie Cooka. Konsekwencje twierdzenia Cooka.

Główne problemy słabo NP-zupełne. Problem 3-wymiarowego skojarzenia. Problem pokrycia wierzchołkowego. Problem podziału zbioru. Problem cyklu Hamiltona. Problem klikli. Problem plecakowy. Problem minimalizacji długości uszeregowania.

Transformacje pseudowielomianowe i silna NP-zupełność. Algorytmy pseudowielomianowe dla problemów NP-zupełnych, definicja problemu silnie NP-zupełnego. Przykłady problemów silnie NP-zupełnych.

Główne problemy silnie NP-zupełne. Problem 3-podziału. Problem wędrującego komiwojażera.

Konsekwencje NP-zupełności. Konsekwencje słabej i silnej NP-zupełności.

NP-trudność i jej związki z NP-zupełnością. Problemy słabo i silnie NP-trudne. Słaba (silna) NP-zupełność a słaba (silna) NP-trudność.

Metodyka rozwiązywania problemów NP-trudnych. Badanie wielomianowych przypadków szczególnych problemów NP-trudnych. Przykłady przypadków szczególnych wybranych problemów NP-trudnych. Stosowanie algorytmów dokładnych dla problemów NP-trudnych. Stosowanie algorytmów pseudowielomianowych dla problemów NP-trudnych. Stosowanie algorytmów heurystycznych i metaheurystycznych dla problemów NP-trudnych.

Algorytmy aproksymacyjne i schematy aproksymacyjne. Definicja algorytmu aproksymacyjnego, współczynnika najgorszego przypadku algorytmu offline i współczynnika wydajności algorytmu online. Ogólne zasady analizy najgorszego przypadku i analizy wydajności. Przykłady algorytmów aproksymacyjnych dla wybranych problemów NP-trudnych. Definicja schematu aproksymacyjnego. Schematy aproksymacyjne typu PTAS oraz FPTAS. Metody konstrukcji schematów aproksymacyjnych. Przykłady schematów aproksymacyjnych dla wybranych problemów NP-trudnych. Klasy APX, PTAS oraz FPTAS.

**Nazwa zajęć: Zaawansowana animacja komputerowa**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. zna zagadnienia związane z dynamiką hierarchicznych układów brył.
2. zna zagadnienia związane z animacją modeli hierarchicznych.
3. zna zagadnienia związane z animacją ruchu człowieka.
4. zna zagadnienia związane z animacją płynów.

**w zakresie umiejętności:**

1. potrafi zaimplementować algorytmy związane z dynamiką hierarchicznych układów brył.
2. potrafi zaimplementować algorytmy związane z animacją modeli hierarchicznych.
3. potrafi zaimplementować algorytmy związane z animacją ruchu człowieka.
4. potrafi zaimplementować algorytmy związane z animacją płynów.

**Treści programowe dla zajęć:**

Dynamika hierarchicznych układów brył: układy z więzami, algorytm Featherstone'a, wymuszanie więzów miękkich i twardych (minimalizacja energii, więzy przestrzennoczasowe)

Animacja modeli hierarchicznych: notacja Denavita-Hartenberga, sformułowanie zagadnienia kinematyki odwrotnej w języku położeń (prosty przykład obliczeniowy), zagadnienie kinematyki odwrotnej w języku prędkości (Jakobian), uogólniony operator odwrotny dla macierzy, metody numeryczne rozwiązywania równania wymagającego odwracania macierzy niekwadratowej

Animacja ruchu człowieka I: rejestrowanie ruchu (kalibracja kamer, odtwarzanie położenia w trzech wymiarach), dopasowywanie szkieletu, przetwarzanie danych z systemów rejestrowania ruchu, zastosowanie kinematyki odwrotnej

Animacja ruchu człowieka II: warstwowy model postaci (szkielet (artykulacja), mięśnie (zmiana kształtu), skóra (wygląd)), model ręki (animacja sięgania i chwytania), animacja chodu (brak stabilności statycznej, wykorzystanie danych empirycznych, zastosowanie kinematyki i dynamiki do animacji szczegółów ruchu), animowanie ubrań

Animacja płynów I: Symulacje oparta na cząstkach struktury danych (Dane systemu cząstek, Przykład systemu cząstek, Wyszukiwanie sąsiadów, Wyszukiwanie pobliskich cząstek, Buforowanie sąsiadów), Metoda SPH (Jądro, Model danych, Interpolacja, Gęstość, Operatory różniczkowe, Siła gradientu ciśnienia, Siły grawitacji i oporu, Przypadek płynu nieściśliwego, Obsługa kolizji)

Animacja płynów II: Symulacje oparta na siatce. Struktury danych (Rodzaje siatek, Dane związane z siatką. Operatory różniczkowe (Różnica skończona, Gradient, Dywergencja, Rotacja, Laplace'a). Symulacja cieczy (Obsługa kolizji, Metody FDM, FEM, FVM, Adwekcja (Metoda Semi-Lagrange'a). Grawitacja, Lepkość, Ciśnienie i nieściśliwość. Symulacja dymu (Siła wyporu, Adwekcja i dyfuzja). Płyn z powierzchnią (Definiowanie powierzchni na siatkach, Powierzchnie swobodne).

Animacja płynów III: Symulacje hybrydowe. Metoda Particle In Cell, Metoda Fluid Implicit Particle, Metoda Particle Level Set, Metoda Vortex Particle, Metoda Lattice Boltzmann

**Nazwa zajęć: Inżynieria wsteczna złośliwego oprogramowania**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. Rozumie, czym są cechy statyczne plików, jak je sprawdzić i zinterpretować.
2. Zna popularne metody obfuskacji kodu.
3. Wie, w jakim celu wykonuje się analizę złośliwego oprogramowania. Ma świadomość zagrożenia, jakie stanowi złośliwe oprogramowanie dla funkcjonowania społeczeństwa i gospodarki.
4. Wie, z jakich technik utrudniających analizę korzystają twórcy złośliwego oprogramowania oraz jak je omijać.

**w zakresie umiejętności:**

1. Potrafi wytłumaczyć, czym jest złośliwe oprogramowanie, przedstawić podstawowe techniki wykorzystywane do jego analizy.
2. Potrafi stworzyć własne laboratorium do pracy ze złośliwym oprogramowaniem.
3. Potrafi przeprowadzić analizę dynamiczną w bezpiecznym środowisku.
4. Potrafi korzystać z debugera.
5. Potrafi korzystać z deasemblera.
6. Zna podstawy języka assemblera.
7. Potrafi przedstawić zasadę działania różnych typów złośliwego oprogramowania.
8. Potrafi wykonać analizę ruchu sieciowego.
9. Potrafi wykonać analizę złośliwych skryptów.
10. Potrafi wykonać analizę złośliwych dokumentów pakietu Microsoft Office oraz PDF.
11. Potrafi przeprowadzić analizę zrzutu pamięci.

**Treści programowe dla zajęć:**

Wprowadzenie do analizy złośliwego oprogramowania: Czym jest złośliwe oprogramowanie?.

Konfiguracja laboratorium do przeprowadzania analizy. Metody i techniki analizy.

Analiza statyczna: Narzędzia i techniki. Open source intelligence.

Analiza dynamiczna: Narzędzia i techniki monitorujące działanie złośliwego oprogramowania.

Interakcja ze złośliwym oprogramowaniem.

Podstawy analizy kodu: Korzystanie z debugera. Popularne metody obfuskacji kodu.

Cele analizy: Cykl zapewniania bezpieczeństwa w przedsiębiorstwach. Rola analityków złośliwego oprogramowania. Indicators of Compromise (IOCs)

Analiza kodu języka niskiego poziomu: Korzystanie z deasemblera. Podstawy języka assemblera.

Popularne techniki wykorzystywane przez złośliwe oprogramowanie: Rootkits. Keyloggers. Downloaders. HTTP C2 channels.

Przechwytywanie ruchu sieciowego: iNetSim, iptables, Wireshark.

Interakcja ze złośliwymi stronami internetowymi: tor, wget, curl, CapTipper, NetworkMiner. Deobfuskacja skryptów.

Self-defending malware: wykrywanie oraz omijanie zabezpieczeń utrudniających analizę.

Analiza złośliwych dokumentów: Pliki pakietu Microsoft Office. Pliki PDF.

Analiza pamięci. Czym jest i jakie korzyści daje analiza pamięci? Podstawy korzystania z pakietu Volatility.

**Nazwa zajęć: Modelowanie języka**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. Zna podstawowe techniki przetwarzania języka naturalnego (tokenizacja, lowercasing, itp.) i potrafi je zastosować. Potrafi „czyścić” dane tekstowe. Rozumie sposób reprezentacji tekstu w komputerze i problemy związane z wysoką wymiarowością reprezentacji liczbowej tekstu.
2. Zna definicję modelu językowego, sposobów jego ewaluacji i zastosowania.
3. Potrafi wytłumaczyć sposób działania statystycznego modelu językowego. Rozumie pojęcie n-gramu. Rozumie sposób trenowania statystycznego modelu językowego.
4. Rozumie podstawy działania sieci neuronowych. Potrafi stworzyć prostą sieć neuronową przy wykorzystaniu biblioteki języka programowania.
5. Zna modele typu „word2vec”. Potrafi wskazać zastosowania tych modeli, ich zalety i wady. Zna pojęcie reprezentacji numerycznej tekstu („embedding”) i potrafi wyjaśnić różnice reprezentacji rzadkiej i gęstej.
6. Potrafi wykorzystywać gotowe modele typu „word2vec”.
7. Rozumie sposób działania neuronowego modelu językowego opartego o prostą sieć „feed-forward”.
8. Potrafi zaimplementować i wytrenować prosty model językowy oparty o sieć „feed-forward”.

9. Rozumie sposób działania rekurencyjnego neuronowego modelu językowego. Zna mechanizm uwagi.
10. Potrafi zaimplementować i wytrenować rekurencyjny neuronowy model językowy.
11. Rozumie sposób działania neuronowego modelu językowego opartego o sieć typu „transformer”.
12. Potrafi zaimplementować i wytrenować model językowy oparty o sieć typu „transformer” przy użyciu gotowych bibliotek.
13. Zna różne algorytmy trenowania modelu językowego typu transformer („masked language model”, „next sentence prediction”). Zna pojęcia „pretraining”, „fine-tuning”, „transfer learning”.

**w zakresie umiejętności:**

1. Potrafi stworzyć statystyczny model językowy bez użycia gotowych bibliotek. Potrafi stworzyć statystyczny model językowy przy pomocy gotowych bibliotek. Umie ewaluować modele językowe.
2. Potrafi wykorzystać wytrenowany model językowy typu „transformer” dla problemu klasyfikacji.
3. Zna alternatywne sposoby ewaluacji neuronowych modeli językowych na podstawie benchmarków „GLUE” i „SuperGLUE”. Potrafi wskazać różnice pomiędzy modelami językowymi typu „transformer”: „BERT”, „RoBERTa”, rodzina modeli „GPT”.
4. Potrafi generować tekst przy użyciu modelu językowego typu transformer.

**Treści programowe dla zajęć:**

Wprowadzenie do przetwarzania języka naturalnego. Przedstawienie podstawowych metod przetwarzania języka naturalnego. Zaznajomienie z problemami wynikającymi z pracy z tekstem. Zastosowanie metod przetwarzania tekstu w praktyce na podstawie implementacji klasyfikatora tekstu opartego o TF-IDF i regresję logistyczną.

Modele językowe i ich ewaluacja

Statystyczne modele językowe – sposób działania, trenowania, zalety i wady.

Trenowanie statystycznego modelu językowego bez użycia gotowych bibliotek. Ewaluacja wytrenowanego modelu językowego

Trenowanie statystycznego modelu językowego z wykorzystaniem biblioteki „KenLM”.

Wprowadzenie do sieci neuronowych.

Implementacja prostej sieci neuronowej przy wykorzystaniu biblioteki „pytorch”

Wprowadzenie modeli typu „word2vec” – algorytm trenowania, zalety, wady.

Wykorzystanie modeli typu „word2vec” w praktyce.

Modele językowe oparte o sieć „feed-forward”.

Implementacja modelu językowego opartego o sieć „feed-forward”.

Modele językowe oparte o rekurencyjne sieci neuronowe, sieci typu „GRU” i „LSTM”. Mechanizm uwagi.

Implementacja i trenowanie modelu językowego opartego o sieć rekurencyjną przy użyciu biblioteki „pytorch”. Implementacja i trenowanie modelu językowego opartego o sieć rekurencyjną z mechanizmem uwagi przy użyciu biblioteki „pytorch”.

Modele językowe oparte o sieci neuronowe, sieci typu transformer – wprowadzenie.

Implementacja i trenowanie modelu językowego opartego o sieć typu „transformer” przy użyciu biblioteki „pytorch”.

Modele językowe oparte o sieci neuronowe, sieci typu transformer – metody zaawansowane, zastosowania.

Klasyfikacja tekstu za pomocą sieci typu „transformer” przy wykorzystaniu biblioteki fairseq.

Modele językowe oparte o sieci neuronowe typu transformer- ewaluacja, alternatywne sposoby ewaluacji modeli. Omówienie różnych modeli typu „transformer” („BERT”, „RoBERTa”, rodzina „GPT”).

Generowanie tekstu przy użyciu modelu językowego GPT2.

Nazwa zajęć: **Seminarium magisterskie 1 - zdalne**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna podstawy metodologii prowadzenia badań w obszarze danego kierunku.
2. zna metody projektowania rozwiązań oraz modyfikacji istniejących rozwiązań problemów badawczych na potrzeby wykorzystania ich w gospodarce

**w zakresie umiejętności:**

1. potrafi wykorzystać pogłębioną wiedzę z zakresu wybranych obszarów informatyki do rozwiązywania problemów stawianych podczas zajęć.
2. potrafi identyfikować problemy badawcze.
3. potrafi dobierać właściwe narzędzia badawcze, a także projektować warsztat badawczy.
4. potrafi projektować rozwiązania lub modyfikacje istniejących rozwiązań.

5. potrafi przygotować i zaprezentować krótkie opracowanie wybranego problemu w sposób zrozumiały dla innych uczestników; potrafi redagować spójną i logiczną wypowiedź z wykorzystaniem poprawnej i terminologii.

6. potrafi poszerzać swoją wiedzę poprzez samodzielne poszukiwania w istniejących opracowaniach naukowych.

7. potrafi samodzielnie rozwiązać problem teoretyczny lub praktyczny.

**w zakresie kompetencji społecznych:**

1. jest gotów do pogłębiania wiedzy z zakresu wybranych obszarów informatyki związanych z problematyką seminarium.

2. jest gotów do identyfikowania nowych problemów badawczych.

**Treści programowe dla zajęć:**

Prezentacja przez prowadzącego seminarium wyników badań prowadzonych aktualnie w grupach badawczych.

Prezentacja przez studentów opracowań wybranych problemów i projektów badawczych zgodnych z potencjalną tematyką pracy magisterskiej.

Określenie celu pracy dyplomowej oraz zakresu planowanych do wykonania w ramach niej badań.

Przygotowanie warsztatu badawczego oraz konspektu pracy magisterskiej.

Nazwa zajęć: **Seminarium magisterskie 2 - stacjonarne**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna podstawy metodologii prowadzenia badań w obszarze danego kierunku.

2. zna metody projektowania rozwiązań oraz modyfikacji istniejących rozwiązań problemów badawczych na potrzeby wykorzystania ich w gospodarce

**w zakresie umiejętności:**

1. potrafi wykorzystać pogłębioną wiedzę z zakresu wybranych obszarów informatyki do rozwiązywania problemów stawianych podczas zajęć.

2. potrafi identyfikować problemy badawcze.

3. potrafi dobierać właściwe narzędzia badawcze, a także projektować warsztat badawczy.

4. potrafi projektować rozwiązania lub modyfikacje istniejących rozwiązań.

5. potrafi przygotować i zaprezentować krótkie opracowanie wybranego problemu w sposób zrozumiały dla innych uczestników; potrafi redagować spójną i logiczną wypowiedź z wykorzystaniem poprawnej i terminologii.

6. potrafi poszerzać swoją wiedzę poprzez samodzielne poszukiwania w istniejących opracowaniach naukowych.

7. potrafi samodzielnie rozwiązać problem teoretyczny lub praktyczny.

**w zakresie kompetencji społecznych:**

1. jest gotów do pogłębiania wiedzy z zakresu wybranych obszarów informatyki związanych z problematyką seminarium.

2. jest gotów do identyfikowania nowych problemów badawczych.

**Treści programowe dla zajęć:**

Prezentacja przez studentów opracowań wybranych problemów i projektów badawczych zgodnych z zadaniami realizowanymi w pracy magisterskiej.

Raportowanie i prezentowanie wykonania badań teoretycznych związanych z tematyką pracy magisterskiej.

Raportowanie i prezentowanie wykonania prac rozwojowych związanych z tematyką pracy magisterskiej.

Dostarczenie ustalonego fragmentu pracy magisterskiej.

Nazwa zajęć: **Wykrywanie ataków na systemy komputerowe**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna w pogłębionym stopniu problematykę sieci komputerowych.

2. zna problematykę zabezpieczeń aktywnych.

3. rozumie podstawowe zasady działania protokołów HTTP, SPDY oraz QUIC; rozumie problemy związane z wydajnym wykorzystaniem sieci komputerowych do udostępniania treści.

4. zna zaawansowane aspekty konfiguracji systemów IPS.

5. zna wybrane, zaawansowane metody pozyskiwania wiedzy o atakach sieciowych
6. zna narzędzia wyszukiwania błędów w oprogramowaniu.
7. zna i rozumie wybrane parametry sprzętu i systemu operacyjnego, istotne w procesie wydajnego wykrywania zagrożeń.
8. zna wybrane narzędzia służące do phishingu i ochrony przed phishingiem, rozumie ich działanie i ograniczenia.
9. zna i rozumie problem podszywania się w sieci.
10. zna i rozumie miary skuteczności systemów oraz granice ich wydajności.

**w zakresie umiejętności:**

1. potrafi przygotować złożony warsztat pracy z wykorzystaniem maszyn wirtualnych i dostępnego oprogramowania.
2. potrafi rozróżnić różne typy pasywnych zabezpieczeń sieciowych i zna podstawy konfiguracji sieci.
3. rozumie idee i sposoby zabezpieczeń serwerów http oraz potrafi analizować logi bezpieczeństwa wybranego serwera.
4. potrafi przeprowadzić rozpoznanie sieci z wykorzystaniem zarówno narzędzi kontaktowych, jak i bezkontaktowych.
5. zna i potrafi stosować wybrane oprogramowanie do przeprowadzania testów penetracyjnych.
6. zna narzędzia i potrafi tworzyć reguły wybranych narzędzi ochrony antywirusowej i analizy plików.
7. zna narzędzia i potrafi tworzyć reguły wybranych narzędzi ochrony antyspamowej.
8. potrafi pisać reguły systemów wykrywania i przeciwdziałania atakom.

**w zakresie kompetencji społecznych:**

1. zna podział i powody ataków na systemy komputerowe.

**Treści programowe dla zajęć:**

Krótkie wprowadzenie do sieci komputerowych, protokół TCP/IPv4 i TCP/IPv6 (TCP/UDP/ICMP).  
Przypomnienie i uszczegółowienie wiedzy zdobytej na studiach inżynierskich.  
Przygotowanie środowiska pracy, instalacja oprogramowania na maszynach wirtualnych  
Zapory sieciowe, serwery proxy (rodzaje, funkcjonalność, zasady działania i możliwości), lokalizacja sensorów (problemy), ukrywanie zabezpieczeń  
Systemy pasywne i aktywne (IDS, IPS), przykładowe systemy (Snort – Cisco, Suricata i inne), programy „antywirusowe” (wprowadzenie), wymagania dla systemów IPS  
Protokół http – problemy z protokołem, nowe pomysły (SPDY, QUIC), wpływ na bezpieczeństwo  
Zabezpieczanie serwerów HTTP (reguły ModSecurity, czytanie logów)  
Metody skanowania sieci (narzędzia bezkontaktowe i kontaktowe)  
Konfiguracja Snort/Suricata, preprocesory, interpretacja wyników programów  
Narzędzia do przeprowadzania testów penetracyjnych, etapy ataku, narzędzia, wykorzystanie Metasploita, przykładowe ataki  
Sposoby pozyskiwania danych o atakach (honeypots, podejrzane zachowania, anomalie), sygnatury ataków  
Podstawowe powody włamań i naruszeń bezpieczeństwa  
Wykrywanie problemów w oprogramowaniu (podstawy np. Valgrinda), zabezpieczenia sprzętowe systemów operacyjnych  
Narzędzia analizy przesyłanych danych (np. program ClamAV i połączenie go z IPS, pisanie reguł do ClamAV), Safe Browsing.  
Narzędzia wykrywania i analizy SPAMu, techniki  
Pisanie reguł systemów IPS, dodatkowe źródła reguł  
Techniczne aspekty wykrywania ataków (ustawienia systemu operacyjnego, kart sieciowych, przechwytywania pakietów), Algorytmy wyszukiwania wzorców (w łańcuchach).  
Ataki phishingowe (narzędzia do organizacji ataków i sposoby wykrywania)  
Spoofing, projekty typu HoneyPot, skanowanie portów i ochrona  
Miary skuteczności systemów wykrywających ataki, próby wykrywania ataków metodami sztucznej inteligencji, ograniczenia systemów wykrywania

Nazwa zajęć: **Kryptografia**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna i rozumie współczesne podstawowe protokoły kryptograficzne i metody szyfrowania danych.
2. zna i rozumie zagadnienia związane z integralnością i uwierzytelnianiem danych.

**w zakresie umiejętności:**

1. potrafi korzystać z klasycznych protokołów kryptograficznych w komunikacji danych.



2. potrafi szyfrować, sprawdzać integralność i uwierzytelniać dane przy pomocy metod kryptograficznych.

**w zakresie kompetencji społecznych:**

1. rozumie konsekwencje stosowania technik kryptograficznych.

**Treści programowe dla zajęć:**

Podstawowe protokoły kryptograficzne. Symetryczny protokół szyfrowania. Doskonały schemat szyfrowania.

Bezpieczeństwo obliczeniowe. EAV-secure schemat szyfrowania. Pseudolosowość.

Ataki z wybranym tekstem jawnym. CPA-secure schemat szyfrowania. Szyfrowanie strumieniowe.

Integralność i uwierzytelnianie danych. Message Authentication Codes. CCA-secure schemat szyfrowania.

Szyfrowanie uwierzytelnione. Funkcje hashujące. HMAC.

Grupy skończone. Problemy obliczeniowe w grupach skończonych. Protokoły uzgadniania klucza sesyjnego.

EAV, CPA oraz CCA bezpieczny schemat szyfrowania z kluczem publicznym.

Szyfrowanie hybrydowe oraz KEM/DEM.

Podpisy cyfrowe.

Kanały podprogowe.

Techniki kryptograficzne.

Nazwa zajęć: **Wirtualna i rozszerzona rzeczywistość**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna zasady tworzenia oprogramowania rozszerzonej rzeczywistości.

2. zna ograniczenia w procesie tworzenia oprogramowania rozszerzonej rzeczywistości.

3. zna zasady tworzenia oprogramowania wirtualnej rzeczywistości.

4. zna ograniczenia w procesie tworzenia oprogramowania rozszerzonej rzeczywistości.

**w zakresie umiejętności:**

1. potrafi tworzyć aplikacje rozszerzonej rzeczywistości.

2. potrafi tworzyć aplikacje wirtualnej rzeczywistości.

**Treści programowe dla zajęć:**

Inicjalizacja projektu i wprowadzenie do AR Foundation

Obsługa chmur punktów, płaszczyzn

Obsługa znaczników

Obsługa map i nawigacji

Wykrywanie postaci/twarzy

Inicjalizacja projektu, integracja gogli z Unity

Obsługa kontrolerów ruchowych

Interakcja z obiektami

Zaawansowana interakcja z obiektami (chwytnie dwoma rękoma, symulowanie ciężaru, snap zone)

Metody poruszania się po świecie

Obsługa wykrywania dłoni

Profilowanie aplikacji

Nazwa zajęć: **Projekt badawczo-rozwojowy 2**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna i rozumie narzędzia i metody wykorzystywane podczas prac nad projektem badawczo-rozwojowym.

**w zakresie umiejętności:**

1. potrafi zastosować znajomość aspektów społecznych pracy zespołowej w projekcie badawczo-rozwojowym.

2. potrafi dostarczać częściowe rezultaty prac wykonywanych w ramach projektu-badawczego.

3. potrafi zaimplementować użyteczny system informatyczny, będący wynikiem prac badawczo-rozwojowych.

4. potrafi zapewnić wysoką jakość systemu informatycznego będącego wynikiem prac badawczo-rozwojowych.

5. potrafi dokumentować architekturę i działanie systemu informatycznego będącego wynikiem prac badawczo-rozwojowych.

6. potrafi przeprowadzić proces testowania systemowego.

7. potrafi przeprowadzić proces testowania użyteczności.
  8. potrafi przygotować i zaprezentować demonstrację systemu działającego w warunkach operacyjnych
- w zakresie kompetencji społecznych:**

1. jest gotów do poprawy jakości systemu informatycznego będącego wynikiem prac badawczo-rozwojowych w przedsiębiorstwie w celu zwiększenia jego wartości.
2. jest gotów do dbania o dokumentację architektury i działania systemu informatycznego będącego wynikiem prac badawczo-rozwojowych w przedsiębiorstwie w celu utrzymania i/lub zwiększania jego wartości.

**Treści programowe dla zajęć:**

Analiza aspektów pracy zespołowej w ramach przedmiotu „Projekt badawczo-rozwojowy 1”. Analiza realizacji obowiązków poszczególnych członków zespołu implementującego system informatyczny w ramach przedmiotu Projekt Badawczo-Rozwojowy 1. Organizacja zespołu projektowego uwzględniające wyniki ww. analiz. Podział ról w zespole projektowym.

Retrospektywa zadań wykonanych w ramach przedmiotu „Projekt badawczo-rozwojowy 1”. Zaplanowanie zadań do wykonania w celu osiągnięcia VII poziomu gotowości technologicznej.

Testowanie systemu w warunkach operacyjnych. Testowanie akceptacyjne. Lokalizacja i usuwanie defektów w kodzie. Przeprowadzenie analizy użyteczności systemu.

Przygotowanie demonstracji systemu działającego w warunkach operacyjnych. Wykonanie demonstracji na forum interesariuszy systemu.

Przeprowadzenie procesu testowania systemowego: sporządzenie planu testów, opracowanie przypadków testowych, przeprowadzenie testów, sporządzenie raportu z testów.

Przeprowadzenie procesu testowania użyteczności: sporządzenie planu testów, zdefiniowanie typów użytkowników, opracowanie person, opracowanie zadań do wykonania podczas testowania, zaplanowanie scenariuszy, przeprowadzenie testów użyteczności, zdefiniowanie obszarów problemowych, sporządzenie raportu testów.

Opracowanie finalnej wersji systemu informatycznego, będącego wynikiem projektu badawczo-rozwojowego na podstawie raportu z testów użyteczności. Opracowanie dokumentacji systemu. Przekazanie końcowej wersji systemu informatycznego do testowania akceptacyjnego. Przekazanie dokumentacji systemu jego interesariuszom. Modyfikacja systemu na podstawie raportu z testów akceptacyjnych.

Nazwa zajęć: **Sztuczna empatia**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. Zna podstawowe pojęcia związane z empatią. Potrafi określić poziom swojej empatii. Zauważa celowość stosowania empatii zarówno w życiu codziennym jak i w systemach informatycznych.
2. Zna podstawy etyki tworzenia sztucznej inteligencji. Potrafi wytyczyć cele badań i rozwoju psychologicznego sztucznej inteligencji.
3. Potrafi odróżnić naśladownictwo od nauki w systemach informatycznych. Zna podstawowe metody implementacji empatii w robotyce.
4. Potrafi wskazać cele i zalety implementacji empatii w systemach dialogowych. Zna metody implementacji zachowań empatycznych w systemach dialogowych.
5. Zna aktualne i istniejące zastosowania systemów wykorzystujących sztuczną empatię. Potrafi określić cele i zalety stosowania sztucznej empatii w marketingu.
6. Zna zastosowania sztucznej empatii w medycynie.
7. Zna metody rozpoznawania emocji. Potrafi podzielić je w zależności od sygnału wejściowego.
8. Zna mechanizmy zachowań stadnych i potrafi określić ich zastosowanie w robotach społecznych, np. w roju.

**w zakresie umiejętności:**

1. Potrafi określić wymagania użytkownika dotyczące zarządzania emocjami w grach komputerowych.
2. Potrafi zaplanować zastosowanie odpowiedniego mechanizmu rozpoznawania emocji.
3. Potrafi zaplanować rozszerzenie prostego istniejącego systemu dialogowego o mechanizmy umożliwiające modelowanie emocji.
4. Potrafi stworzyć słownik emocji, zaplanować zachowania agenta oraz dokonać mapowania emocji na zachowania agenta.
5. Potrafi zaimplementować zachowania empatyczne.
6. Potrafi zaimplementować graficzną reprezentację emocji.
7. Potrafi testować system informatyczny.
8. Potrafi zaprezentować i scharakteryzować wykonany system informatyczny.

**w zakresie kompetencji społecznych:**

1. Rozumie aspekty społeczne rozwijania sztucznej inteligencji.

**Treści programowe dla zajęć:**

Podstawy empatii: Podstawowe pojęcia związane z empatią. Empatia kognitywna. Neuronauka. Skrócony test ilorazu empatii. Neurologiczne podstawy zjawiska empatii. Rodzaje empatii.

Empatia a prawdziwa Sztuczna Inteligencja (SI): Empatia a doskonała SI. Test Turinga. Hierarchia emocji. Zasady z Asimolar. Perspektywy na przyszłość. Nauka poprzez empatię. Odtwarzanie modelu rozwoju dziecka.

Empatia w robotyce: Naśladownictwo i mimikra. Systemy wizyjne i uczenie maszynowe. Sieci neuronowe. Proste roboty naśladowujące i uczące się: Affetto i WAMEOBA

Empatia w systemach dialogowych: Chatboty i wirtualni asystenci. Analiza języka mówionego i pisanego. Metody implementacji emocji w systemach dialogowych. Emocjonalni wirtualni rozmówcy.

Empatia w grach: Zagadnienie immersji. Realistyczni NPCs. Awatary. Analiza wymagań użytkownika. Analiza wymagań gracza.

Empatia w zastosowaniach praktycznych: Zastosowania sztucznej empatii w motoryzacji, marketingu i naukach społecznych. Rozpoznawanie emocji w marketingu i jego wykorzystanie w celu zwiększenia sprzedaży. Spersonalizowane reklamy. Nastrój.

Empatia w medycynie: Uszkodzenia mózgu a empatia. Zastosowania w niwelowaniu skutków Alzheimerera. Gra FearNot! Empatia emocjonalna i empatia kognitywna.

Rozpoznawanie emocji: Metody i sposoby rozpoznawania emocji. Affectiva.

Behawioryzm: Zachowania stadne i społeczne, w tym implementowane w robotyce. Rój i jego dynamika. Test wiedzy z zakresu zajęć 1-9.

Tworzenie empatycznego bota I: Zapoznanie się z bazowym systemem, na którym rozwijany będzie projekt empatycznego bota. Przygotowanie bazowego system oraz zaplanowanie postaci oraz metod implementacji modułów umożliwiających rozpoznanie emocji i zachowania empatyczne. Praca w zespołach.

Tworzenie empatycznego bota II: Tworzenie słownika emocji, zbioru zachowań, mapowanie emocji na odpowiednie zachowania. Planowanie szczegółowych rozwiązań oraz komponentów koniecznych do implementacji modułu rozpoznania emocji i zachowań empatycznych. Praca w zespołach.

Tworzenie empatycznego bota III: Implementacja zachowań i zaprojektowanych rozwiązań. Praca w zespołach.

Tworzenie empatycznego bota IV: Implementacja wybranej reprezentacji graficznej emocji. Praca w zespołach.

Tworzenie empatycznego bota V: Testowanie wdrożonych rozwiązań i całego systemu. Praca w zespołach.

Podsumowanie zajęć: Stworzenie prezentacji multimedialnej nt. powstałego bota. Przygotowanie bota do prezentacji. Przedstawienie botów. Prezentacje multimedialne.

Nazwa zajęć: **Inteligencja obliczeniowa**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. Zna matematyczne podstawy inteligencji obliczeniowej
2. Zna standardowe metody inteligencji obliczeniowej

**w zakresie umiejętności:**

1. Umie zaprogramować algorytmy inteligencji obliczeniowej
2. Potrafi przygotować dane na potrzeby algorytmów inteligencji obliczeniowej

**Treści programowe dla zajęć:**

Wprowadzenie do inteligencji obliczeniowej.

Współczesne kierunki badań i zastosowań inteligencji obliczeniowej.

Logika i zbiory rozmyte

Algorytmy genetyczne i ewolucyjne

Sztuczne sieci neuronowe

Projekt wykorzystujący inteligencję obliczeniową

Nazwa zajęć: **Uczenie głębokie**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. zna i rozumie klasyczne modele uczenia maszynowego.
2. zna i rozumie metody uczenia sieci neuronowych.
3. zna i rozumie architektury głębokich sieci neuronowych ze szczególnym uwzględnieniem zastosowań w widzeniu komputerowym i przetwarzaniu języka naturalnego.

**w zakresie umiejętności:**

1. potrafi zaimplementować klasyczne modele uczenia maszynowego.
2. potrafi zaimplementować, trenować i optymalizować modele głębokich sieci neuronowych w zastosowaniach widzenia komputerowego i przetwarzania języka naturalnego.

**w zakresie kompetencji społecznych:**

1. jest gotów do pogłębiania wiedzy dotyczącej wpływu sztucznej inteligencji na inne dziedziny życia społecznego.
2. jest gotów do śledzenia najnowszych osiągnięć z dziedziny głębokiego uczenia prezentowanych na konferencjach naukowych i w czasopiśmie naukowych.

**Treści programowe dla zajęć:**

Metoda kNN i klasyfikacja liniowa.

SVM i regresja logistyczna, regularyzacja.

Optymalizacja: metoda gradientowa, stochastyczna, warianty SGD takie jak RMSprop i Adam.

Algorytm propagacji wstecznej.

Perceptron i sieci wielowarstwowe.

Wprowadzenie do sieci spłotowych.

Opis wybranych sieci spłotowych: AlexNet, VGG, GoogLeNet, ResNet.

Sieci spłotowe w praktyce, rozszerzanie (augmentacja) zbioru danych, dane syntetyczne.

Model „worka słów” - zastosowanie regresji liniowej i logistycznej jako prostych sieci neuronowych dla tekstu, zastosowanie sieci spłotowych dla tekstu.

Zanurzenia słów - model word2vec.

Przetwarzanie danych sekwencyjnych za pomocą rekurencyjnych sieci neuronowych.

Sieci LSTM i GRU.

Sieci seq2seq, zadania etykietowania sekwencji.

Atencja w sieciach rekurencyjnych, zastosowanie w tłumaczeniu maszynowym.

Pre-trenowanie modeli, model ELMo i BERT.

Nazwa zajęć: **Zaawansowane metody renderingu**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna podstawowe równania renderingu oraz potencjału oraz metody ich rozwiązywania.
2. zna metody Monte Carlo i Quasi Monte Carlo wykorzystywane w algorytmach dla globalnego oświetlenia.
3. zna zagadnienie Volume Rendering.

**w zakresie umiejętności:**

1. potrafi konstruować i implementować uniwersalne metody renderingu.

**Treści programowe dla zajęć:**

Wyprowadzenie podstawowych równań: renderingu (różne przypadki) oraz potencjału, pierwsze z nich opisuje globalne metody oświetlenia, drugie pozwala rozwiązywać pierwsze jedynie w obszarze widocznym w obrazie.

Metody rozwiązywania równań renderingu: iteracyjne metody macierzowe, numeryczne metody rozwiązywania równań całkowych.

Metody Monte Carlo i Quasi Monte-Carlo: generatory pseudolosowe próbkowanie funkcji BRDF.

Algorytmy Monte Carlo dla globalnego oświetlenia - śledzenie fotonów, dwukierunkowe śledzenie ścieżki.

Volume Rendering.

Nazwa zajęć: **Matematyczne fundamenty informatyki**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna pojęcia wektora, macierzy, przestrzeni liniowej wektorów oraz liniowej niezależności i bazy wektorów.
2. zna sposób obliczania wyznacznika oraz jego własności.
3. zna pojęcie wektorów ortogonalnych.
4. zna pojęcie rozkładu warunkowego i brzegowego oraz twierdzenie Bayesa.
5. zna metody estymacji punktowej i przedziałowej.

6. zna pojęcie funkcji straty, miary jakości modelu i pojęcie regularyzacji.
7. zna idee algorytmów Monte Carlo Markov Chains (MCMC) oraz podstawowe próbniki.
8. zna warianty algorytm największego spadku: algorytm największego spadku z momentum, stochastyczny algorytm największego spadku.
9. zna algorytm wstecznej propagacji błędu.
10. zna metodę mnożników Lagrange'a.
11. zna metodę programowania liniowego (wraz z twierdzeniem o dualności) oraz kwadratowego.
12. zna pojęcie grafu i jego podstawowe własności.
13. zna pojęcie propagacji ataku w sieci komputerowej oraz sposoby wykorzystywania uczenia maszynowego w zapewnianiu bezpieczeństwa sieci.

**w zakresie umiejętności:**

1. potrafi wykonywać operacje algebraiczne takie jak dodawanie i mnożenie macierzy i wektorów, mnożenie wektora przez macierz; potrafi opisać macierz o wybranych własnościach.
2. potrafi znaleźć odwrotność zadanej macierzy; rozwiązuje układy równań liniowych wielu zmiennych; znajduje wyznaczniki z nimi stowarzyszone i potrafi określić liczbę rozwiązań w zależności od wyznacznika.
3. wyznacza wielomian charakterystyczny macierzy, wartości własne; oblicza wektory własne oraz bazę takich wektorów
4. umie obliczyć normę, iloczyn skalarny dwóch wektorów; potrafi obliczyć odległość dwóch punktów w metryce euklidesowej; potrafi dokonać rzutowania jednego wektora względem drugiego.
5. potrafi dokonać rozkładu macierzy (Choleskiego, Singular Value Decomposition).
6. potrafi obliczyć pochodną funkcji i wyznaczać szereg Taylora; dla funkcji wielu zmiennych umie obliczyć pochodne cząstkowe i gradient funkcji.
7. potrafi wyznaczyć spłot funkcji (konwolucję) i jego dyskretny odpowiednik.
8. umie wyznaczyć dystrybuantę i gęstość rozkładu zmiennej losowej; potrafi rozróżnić wybrane rozkłady prawdopodobieństwa; umie zastosować twierdzenie Bayesa we wnioskowaniu; potrafi obliczyć wartość oczekiwaną i wariancję zmiennej losowej.
9. potrafi zbudować model statystyczny; potrafi szacować parametry za pomocą metody najmniejszych kwadratów i metody największej wiarygodności.
10. potrafi testować wybrane hipotezy statystyczne.
11. umie przeprowadzić wnioskowanie bayesowskie.
12. potrafi zaimplementować algorytm największego spadku.
13. potrafi scharakteryzować problem optymalizacji wypukłej.
14. potrafi wykonać algorytm sympleks oraz algorytm interior point.
15. potrafi opisać praktyczne przykłady grafów i wyznaczyć ich cechy charakterystyczne.
16. potrafi zamodelować propagację ataku w sieci komputerowej z uwzględnieniem jej specyficznej topologii; umie wykorzystać techniki uczenia maszynowego do maksymalizowania wybranych parametrów bezpieczeństwa sieci.
17. potrafi wykorzystać poznany aparat pojęciowy z zakresu algebry do sformułowania i rozwiązania problemu redukcji wymiaru za pomocą metody składowych głównych; umie ująć swoje wnioski płynące z obliczeń w postaci raportu i potrafi zaprezentować swój model w sposób popularny.
18. potrafi wykorzystać zaawansowane techniki statystyczne i analityczne do rozwiązywania wybranego problemu uczenia maszynowego za pomocą sieci neuronowych; umie przygotować dokumentację techniczną swojego modelu oraz przeprowadzić dyskusję dotyczącą wybranych parametrów i architektury.

**w zakresie kompetencji społecznych:**

1. jest gotów do formułowania pytań w celu zamodelowania propagacji ataku w sieci komputerowej.
2. jest gotów do ujęcia swoich wniosków płynących z obliczeń w postaci raportu i do zaprezentowania swojego modelu w sposób popularny.
3. jest gotów do przygotowania raportu dotyczącego swoich obserwacji i do ich właściwego zakomunikowania grupie.

**Treści programowe dla zajęć:**

Wprowadzenie do algebry liniowej. Zdefiniowanie pojęcia wektora, macierzy transformacji, liniowej niezależności wektorów oraz bazy. Opisanie podstawowych typów macierzy: macierz diagonalna, górnotrójkątna. Wprowadzenie do operacji na macierzach – dodawanie, mnożenie, mnożenie macierzy przez wektor i skalar.

Wyznaczanie macierzy z opisu transformacji liniowej. Praktyczne ćwiczenia w rachunkach na macierzach, dyskusja optymalnych metod wykonywania rachunków na macierzach wraz z praktyczną implementacją.

Rozwiązywanie układów równań liniowych wraz z dyskusją liczby rozwiązań. Sprowadzanie macierzy do postaci całkowicie zredukowanej i wyznaczanie przestrzeni rozwiązań układu liniowego. Definicja wyznacznika oraz jego podstawowe własności. Znajdowanie odwrotności macierzy metodą eliminacji Gaussa.

Dyskusja efektywnych metod rozwiązywania układów równań liniowych. Znajdowanie rozwiązań za pomocą algorytmu eliminacji Gaussa. Przykłady dla małej liczby zmiennych i implementacja metody w przypadku problemu o bardzo dużej liczbie zmiennych.

Zdefiniowanie wielomianu charakterystycznego i minimalnego macierzy oraz wartości własnych. Opis algorytmu wyznaczania wektorów własnych zadanej wartości własnej. Zdefiniowanie bazy wektorów własnych zadanej przestrzeni.

Wyznaczanie wielomianów charakterystycznych i minimalnych dla wybranych macierzy i operatorów liniowych. Implementacja algorytmu wyznaczania bazy wektorów własnych

Wprowadzenie pojęcia iloczynu skalarnego wektorów oraz normy wektora. Przedstawienie pojęcia metryki euklidesowej. Definicja rzutu ortogonalnego oraz projekcji.

Obliczanie wektora ortogonalnego do danego wektora. Wyznaczanie rzutu ortogonalnego na daną podprzestrzeń liniową. Implementacja rzutowania wraz z dyskusją przykładów.

Opis faktoryzacji macierzy metodą Cholesky'ego wraz z opisem algorytmu. Opis metody diagonalizacji macierzy symetrycznych. Wprowadzenie do rozkładu macierzy metodą SVD i porównanie z innymi metodami.

Implementacja wybranych rozkładów (SVD, Cholesky, diagonalizacja) z wykorzystaniem opisu algorytmów z wykładu. Porównanie i dyskusja szybkości wybranych metod oraz wskazanie niestabilności numerycznej wybranych algorytmów.

Definicja pochodnej funkcji i pochodnej cząstkowej. Wprowadzenie pojęcia szeregu Taylora dla funkcji jednej i wielu zmiennych. Definicja gradientu funkcji. Definicja spłotu.

Obliczanie pochodnych funkcji elementarnych oraz reguły obliczania pochodnych dla iloczynu, sumy i ilorazu funkcji oraz funkcji złożonej. Wyznaczanie gradientu funkcji wektorowej. Wyznaczanie spłotów funkcji.

Definicja zmiennej losowej, dystrybucyjności oraz gęstości rozkładu. Zdefiniowanie rozkładu wielu zmiennych oraz rozkładu warunkowego i brzegowego. Prezentacja wybranych rozkładów prawdopodobieństwa. Omówienie twierdzenia Bayesa i jego znaczenia we wnioskowaniu. Definicja wartości oczekiwanej i wariancji zmiennej losowej, przykłady wyznaczania. Definicja kowariancji.

Wyznaczanie rozkładów empirycznych. Porównanie rozkładu empirycznego z rozkładem teoretycznym. Wylizanie wartości oczekiwanych i wariancji dla zmiennej losowej oraz dla próby. Porównanie uzyskanych wyników. Wyznaczanie macierzy kowariancji.

Definicja modelu statystycznego. Przedstawienie podstaw estymacji punktowej i przedziałowej. Wprowadzenie do metod estymacji: metoda najmniejszych kwadratów i metoda największej wiarygodności. Pokazanie praktycznych przykładów zastosowań.

Wyznaczanie estymatorów punktowych i przedziałowych z danych. Wyznaczenie parametrów w modelu regresji liniowej za pomocą metody najmniejszych kwadratów, implementacja rozwiązania. Implementacja bootstrapowych przedziałów ufności.

Przedstawienie problemu testowania hipotez. Definicja błędów pierwszego i drugiego rodzaju. Definicja poziomu istotności testu oraz pwartości. Omówienie przykładowych procedur testowych: test t dla jednej i dwóch prób, test niezależności oraz test zgodności. Przedstawienie idei testów permutacyjnych.

Praktyczne testowanie hipotez dla rzeczywistych zbiorów danych. Implementacja permutacyjnego testu t.

Definicja podstawowych funkcji straty stosowanych w uczeniu maszynowym. Omówienie związku funkcji straty z miarą jakości modelu. Wprowadzenie idei regularyzacji i jego znaczenia w optymalizacji. Wprowadzenie do statystyki bayesowskiej. Definicja rozkładu a priori i a posteriori. Wprowadzenie do algorytmów MCMC, próbnik Metropolisa-Hastingsa oraz próbnik Gibbsa.

Implementacja różnych funkcji straty w uczeniu parametrów regresji liniowej. Implementacja próbnika Metropolisa-Hastingsa.

Omówienie algorytmu największego spadku oraz pojęcia momentum. Omówienie stochastycznego algorytmu największego spadku. Omówienie algorytmu wstecznej propagacji błędów.

Implementacja algorytmów największego spadku do estymacji parametrów regresji liniowej. Wykorzystanie algorytmu wstecznej propagacji błędów do uczenia parametrów sieci neuronowych.

Zdefiniowanie problemu optymalizacji wypukłej. Omówienie metody mnożników Lagrange'a. Przedstawienie przykładów na rzeczywistych problemach.

Wykorzystanie metody mnożników Lagrange'a do rozwiązania praktycznych problemów optymalizacji z ograniczeniami.

Omówienie metody programowania liniowego oraz programowania kwadratowego. Podanie twierdzenia o dualności i omówienie jego konsekwencji. Przedstawienie algorytmów sympleks oraz interior point.

Wykorzystanie algorytmów sympleks i interior point do rozwiązania praktycznych problemów optymalizacji. Implementacja algorytmu simpleks.

Omówienie pojęcia grafu, drzewa, sieci i ich podstawowych własności.

Wskazanie podstawowych przykładów grafów, opis topologii sieci (LAN, WAN) w terminach dużych grafów. Ćwiczenie wyznaczania podstawowych własności grafów.

Modelowanie cyberbezpieczeństwa sieci za pomocą pojęcia grafów. Opis propagacji ataku na sieć komputerów. Modelowanie bezpieczeństwa sieci z wykorzystaniem sieci neuronowej i algorytmu uczącego opartego na uczeniu maszynowym.

Praktyczna analiza sieci o małej liczbie wierzchołków pod kątem jej bezpieczeństwa, z uwzględnieniem wyboru topologii i metod ochrony komputerów. Dyskusja algorytmu uczącego dobierającego wagowo środki bezpieczeństwa.

Implementacja metody składowych głównych z wykorzystaniem poznanych metod algebraicznych.

Implementacja sieci neuronowej z wykorzystaniem poznanych metod optymalizacyjnych.

**Nazwa zajęć: Elementy językoznawstwa: od głoski do fake newsa**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie umiejętności:**

1. potrafi scharakteryzować przedmiot językoznawstwa.
2. potrafi wskazać wybrane problemy związane z użyciem terminologii anglojęzycznej w pracy informatyka.
3. potrafi nazwać oraz omówić najważniejsze etapy rozwoju językoznawstwa oraz scharakteryzować współczesne dyscypliny lingwistyczne.
4. potrafi nakreślić najistotniejsze zagadnienia dotyczące typologii języków świata ze szczególnym uwzględnieniem języków indoeuropejskich.
5. potrafi określić przedmiot fonologii i fonetyki oraz zdefiniować pojęcie fonemu; potrafi scharakteryzować metody badawcze wykorzystywane w fonetyce.
6. potrafi wskazać podstawowe organy aparatu artykulacyjnego.
7. potrafi wskazać przykładowe sposoby wyrażania znaczenia gramatycznego i kategorii gramatycznych w języku polskim.
8. potrafi dokonać podziału wyrazu na morfemy oraz scharakteryzować pojęcie części mowy.
9. potrafi dokonać analizy składniowej zdania prostego.
10. potrafi omówić podstawowe relacje semantyczne.
11. potrafi dokonać analizy słotwórczej wyrazu i rozróżnia typy frazeologizmów.
12. potrafi omówić podstawowe rodzaje tłumaczeń oraz związane z nimi strategie translacyjne.
13. potrafi omówić fundamentalne zagadnienia z zakresu makro- i mikrostruktury słowników.
14. potrafi przeprowadzić podstawową analizę pragmatyczną wypowiedzi.
15. potrafi redagować wybrane teksty użytkowe (korespondencja).
16. potrafi wykonać projekt badawczy z wykorzystaniem jego zasobów.
17. potrafi przeprowadzić tego rodzaju badanie za pomocą narzędzi poznanych w trakcie kursu.
18. potrafi korzystać z narzędzi oferowanych przez wyszukiwarkę Google, bazę Google Books i środowisko Google Ngram Viewer podczas realizowania prostych scenariuszów badań z zakresu lingwistyki, historii i kulturologii (badania z zakresu folkloru współczesnego dotyczące m.in. legend miejskich, pogłosek, memów czy fake newsów).

**w zakresie kompetencji społecznych:**

1. zna definicje pojęcia „znak” i „znaczenie” oraz zna funkcje pełnione przez znaki.
2. zna podstawowe zagadnienia z zakresu kultury języka i poprawności językowej.
3. zna podstawowe symbole wykorzystywane w transkrypcji fonetycznej IPA.
4. zna terminy „gramatyka”, „znaczenie gramatyczne”, „kategoria gramatyczna”.
5. zna podstawowe pojęcia z zakresu morfologii.
6. zna podstawowe pojęcia z zakresu składni.
7. zna podstawowe pojęcia i zjawiska z zakresu leksykologii i semantyki.
8. zna podstawowe terminy z zakresu morfonologii, słotwórstwa i frazeologii.
9. zna podstawowe zagadnienia z zakresu przekładoznawstwa.
10. ma podstawową wiedzę na temat leksykografii jako dyscypliny językoznawczej, zna typologię słowników.
11. ma podstawową wiedzę na temat dziejów, przedmiotu i kierunków badawczych pragmatyki językowej i teorii aktów mowy.

12. zna podstawowe zagadnienia dotyczące gatunków tekstu i zasady pisania tekstów użytkowych.
13. potrafi scharakteryzować typy korpusów językowych; potrafi używać wybranych narzędzi komputerowych stosowanych w pracy z korpusami tekstowymi.
14. potrafi używać narzędzi systemu Odkrywka.
15. ma podstawową wiedzę o badaniach o charakterze lingwochronologicznym i fotodokumentacyjnym.

#### **Treści programowe dla zajęć:**

Język jako przedmiot językoznawstwa: Zakres terminu „język naturalny”. Pochodzenie języka. Znak i znaczenie, rodzaje znaków. Język jako system znaków. Funkcje pełnione przez znaki. Znak a układ komunikacyjny. Ćwiczenia z zakresu praktycznych aspektów używania języka w pracy informatyka: Poprawność językowa. Terminologia anglojęzyczna. Zastosowanie języka angielskiego podczas pracy w środowisku wielojęzycznym.

Zarys historii badań językoznawczych i działy lingwistyki: Prenaukowa refleksja nad językiem. Językoznawstwo historyczno-porównawcze. Koncepcja Wilhelma von Humboldta. Strukturalizm językoznawczy. Generatywizm. Kognitywizm. Językoznawstwo szczegółowe, konfrontatywne i ogólne. Językoznawstwo synchroniczne i diachroniczne.

Klasyfikowanie języków świata: Klasyfikacja genetyczna, geograficzna, typologiczna, socjalna. Rodziny językowe. Omówienie języków indoeuropejskich. Uniwersalia językowe.

Fonologiczny system języka: Przedmiot fonologii i fonetyki. Pojęcie fonemu. System fonologiczny. Prozodia. Działy fonetyki i metody badawcze stosowane w fonetyce. Aparat artykulacyjny. Transkrypcja fonetyczna.

Gramatyczny system języka: Gramatyka. Znaczenie gramatyczne i forma gramatyczna. Sposoby wyrażania znaczenia gramatycznego (afiksacja, wymiany głoskowe, supletywizm, reduplikacja, intonacja, szyk wyrazów, przedimki). Pojęcie kategorii gramatycznej (liczba, rodzaj, przypadek, czas, osoba, tryb, strona).

Gramatyczny system języka: Morfologia. Przedmiot morfologii. Morfem. Morf. Allomorf. Zagadnienie części mowy.

Gramatyczny system języka: Składnia. Przedmiot składni. Zdanie jako podstawowa jednostka składni. Konstrukcje składniowe.

Leksykalno-semantyczny system języka: System leksykalno-semantyczny. Leksykologia i semantyka. Etymologia. Leksem a wyraz. Ewolucja znaczeniowa leksemu. Metafora. Metonimia. Podstawowe relacje semantyczne.

Pośrednie płaszczyzny językowe: Morfonologia. Słowotwórstwo. Frazeologia i frazematyka. Wymiana głoskowa. Analiza słowotwórcza wyrazów. Rozróżnianie typów frazeologizmów.

Przekładoznawstwo: Pojęcie tłumaczenia. Zarys teorii przekładu. Rodzaje tłumaczeń. Rola tłumacza w procesie tłumaczenia. Powszechne problemy i błędy występujące w procesie tłumaczenia. Oprogramowanie stosowane w procesie tłumaczenia.

Leksykografia: Miejsce leksykografii wśród dyscyplin lingwistycznych. Słownik a encyklopedia. Typologia słowników. Jednostka opisu słownikowego i jednostka języka. Makro- i mikrostruktura słownika. Budowa artykułu hasłowego. Przegląd najważniejszych słowników języka polskiego i angielskiego.

Pragmatyka językowa: Pragmatyka językowa a semantyka. Lokucja, illokucja, perlokucja. Zarys teorii aktów mowy. Klasyfikacje aktów mowy. Implikatury konwersacyjne i maksymy uprzejmości. Etykieta językowa. Ćwiczenia z zakresu ustalania celu oraz siły illokucyjnej i perlokucyjnej wypowiedzi, jak również identyfikowania grzecznościowych aktów mowy. Analiza form grzecznościowych w wybranych językach

Tekst i gatunek tekstu: Pojęcie tekstu i dyskursu. Metody analizy tekstu. Wzorzec gatunkowy i wzorzec tekstu. Ćwiczenia z zakresu rozpoznawania i charakteryzowania różnych typów tekstu oraz redagowania tekstów użytkowych w języku polskim należących do różnych stylów i odmian języka.

Językoznawstwo korpusowe: Typy korpusów. Narzędzia komputerowe stosowane w badaniach korpusowych. Przegląd wybranych korpusów tekstowych. Badania prowadzone przy pomocy systemu Odkrywka: korzystanie z wyszukiwarki, tworzenie wykresów częstości, tworzenie Dossier. Wyszukiwarka Google i baza Google Books a badania chronologiczno-frekwencyjne. Narzędzie Ngram Viewer. Omówienie i realizacja projektu badawczego wykorzystującego internetowe zasoby tekstowe.

Lingwochronologizacja i lingwochronografia. Obiekt opisu. Parametry czasowe. Historia badań lingwochronologicznych (m-LCH i TLCH). Granice lingwochronologiczne. Redatacje. Fotodokumentacja.

Nazwa zajęć: **Uczenie głębokie w przetwarzaniu tekstu**



**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. Zna neuronowe modele języka oparte na architekturze Transformer. Potrafi wskazać różnicę pomiędzy modelami opartymi na architekturze enkoder, dekoder oraz enkoder-dekoder.
2. Zna oraz potrafi wskazać cech charakterystyczne neuronowych modeli języka: BERT, RoBERTa, GPT-2, T5, T0, OPT.
3. Zna oraz potrafi wskazać cechy charakterystyczne wielojęzycznych neuronowych modeli języka: XLM, XLM-RoBERTa, mT5, ByT5, XGLM.
4. Zna problem długości sekwencji w tekście oraz potrafi wskazać sposoby na rozwiązanie tego problemu. Zna neuronowe modele języka: Longformer, LongT5 oraz FlashAttention.
5. Zna różnicę pomiędzy modelami dziedzinowymi i ogólnymi (jednojęzyczne oraz wielojęzyczne) oraz umie wskazać wady i zalety modeli.
6. Zna różnicę pomiędzy podejście few-shot, one-shot oraz zero-shot learning.

**w zakresie umiejętności:**

1. Zna oraz potrafi wskazać sposoby rozwiązania problemu "rzadkich słów" – Byte Pair Encoding (model GPT-2) oraz SentencePiece (model T5).
2. Potrafi wytrenować oraz dostroić neuronowy model języka oraz zna pojęcia "pretraining" oraz "fine-tuning".
3. Potrafi wykorzystać neuronowy model języka w problemie klasyfikacji oraz zadaniu typu QA (Pytanie-Odpowiedź).
4. Potrafi wykorzystać neuronowy model języka w trybie: few-shot, one-shot oraz zero-shot learning.

**Treści programowe dla zajęć:**

Rozwiązanie problemu "rzadkich słów" z wykorzystaniem algorytmu Byte Pair Encoding oraz uniagramowego modelu języka.

Charakterystyka architektury Transformer, modeli opartych na architekturze enkoder, dekoder oraz enkoder-dekoder.

Charakterystyka neuronowych modeli języka: BERT, RoBERTa, GPT-2, T5, T0, OPT.

Charakterystyka neuronowych modeli języka: XLM, XLM-RoBERTa, mT5, ByT5, XGLM.

Rozwiązywanie problemu długości sekwencji, wykorzystanie neuronowych modeli języka: Longformer, LongT5 oraz FlashAttention.

Charakterystyka dziedzinowych, ogólnych (jednojęzycznych oraz wielojęzycznych) modeli.

Trenowanie oraz dostrojenie neuronowych modeli języka z wykorzystaniem biblioteki transformers.

Wykorzystanie podejścia few-shot, one-shot oraz zero-shot learning z neuronowymi modelami języka.

Rozwiązanie zadania klasyfikacji oraz typu QA (Pytania-Odpowiedź) z wykorzystaniem neuronowych modeli języka.

Nazwa zajęć: **Narzędzia matematyki w informatyce**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. Potrafi dostrzec związki między pojęciami algebry liniowej (np. wektorami i wartościami własnymi macierzy) a własnościami grafowymi.

**w zakresie umiejętności:**

1. Potrafi zbudować model matematyczny prostych zjawisk rzeczywistych przy pomocy narzędzi teorii grafów lub skończonych łańcuchów Markowa.
2. Umie przewidzieć wynik symulacji procesów Markowa, korzystając z twierdzeń ergodycznych.
3. Umie przedstawiać tok swojego rozumowania w sposób zrozumiały dla słuchaczy.

**Treści programowe dla zajęć:**

Interpretacja zjawisk rzeczywistych w języku teorii grafów. Macierz przyległości grafu (prostego), wartości i wektory własne tej macierzy. Własności spektrum macierzy symetrycznych.

Odczytywanie własności grafowych na podstawie spektrum macierzy przyległości, np. informacji o spójności, regularności, dwudzielności, cyklach i spacerach zadanej długości.

Laplasjan grafu. Metody spektralne szacowania trudnych obliczeniowo parametrów grafowych: liczby niezależności, liczby chromatycznej, największego cięcia w grafie. Wyznaczanie liczby drzew rozpiętych metodą algebraiczną.

Definicja (skończonego) łańcucha Markowa. Macierz przejścia, skierowany graf łańcucha Markowa. Algebraiczne i probabilistyczne metody wyznaczania rozkładu łańcucha Markowa po zadanej liczbie kroków.

Symulacja wielu kroków łańcucha Markowa. Rozkłady stacjonarne, twierdzenie ergodyczne.

Odwracalne łańcuchy Markowa. Błądzenie na grafach, czas pokrycia. Generowanie struktur losowych metodą Monte Carlo na łańcuchach Markowa (MCMC).

Zastosowania łańcuchów Markowa i MCMC do łamania szyfrów podstawieniowych, generowania tekstów, generowania muzyki, tworzenia rankingu Page'a.

Nazwa zajęć: **Seminarium magisterskie 1 - stacjonarne**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. zna podstawy metodologii prowadzenia badań w obszarze danego kierunku.
2. zna metody projektowania rozwiązań oraz modyfikacji istniejących rozwiązań problemów badawczych na potrzeby wykorzystania ich w gospodarce

**w zakresie umiejętności:**

1. potrafi wykorzystać pogłębioną wiedzę z zakresu wybranych obszarów informatyki do rozwiązywania problemów stawianych podczas zajęć.
2. potrafi identyfikować problemy badawcze.
3. potrafi dobrać właściwe narzędzia badawcze, a także projektować warsztat badawczy.
4. potrafi projektować rozwiązania lub modyfikacje istniejących rozwiązań.
5. potrafi przygotować i zaprezentować krótkie opracowanie wybranego problemu w sposób zrozumiały dla innych uczestników; potrafi redagować spójną i logiczną wypowiedź z wykorzystaniem poprawnej i terminologii.
6. potrafi poszerzać swoją wiedzę poprzez samodzielne poszukiwania w istniejących opracowaniach naukowych.
7. potrafi samodzielnie rozwiązać problem teoretyczny lub praktyczny.

**w zakresie kompetencji społecznych:**

1. jest gotów do pogłębiania wiedzy z zakresu wybranych obszarów informatyki związanych z problematyką seminarium.
2. jest gotów do identyfikowania nowych problemów badawczych.

**Treści programowe dla zajęć:**

Prezentacja przez prowadzącego seminarium wyników badań prowadzonych aktualnie w grupach badawczych.

Prezentacja przez studentów opracowań wybranych problemów i projektów badawczych zgodnych z potencjalną tematyką pracy magisterskiej.

Określenie celu pracy dyplomowej oraz zakresu planowanych do wykonania w ramach niej badań.

Przygotowanie warsztatu badawczego oraz konspektu pracy magisterskiej.

Nazwa zajęć: **Projekt badawczo-rozwojowy 1**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. zna i rozumie narzędzia i metody wykorzystywane podczas prac nad projektem badawczo-rozwojowym.

**w zakresie umiejętności:**

1. potrafi zastosować znajomość aspektów społecznych pracy zespołowej w projekcie badawczo-rozwojowym.
2. potrafi wdrożyć cechy innowacyjnego projektu informatycznego w projekcie badawczo-rozwojowym.
3. potrafi wizualizować system informatyczny za pomocą makiety dynamicznej.
4. potrafi pozyskiwać inwestorów dla projektu badawczo-rozwojowego.
5. potrafi uczestniczyć w projekcie prowadzonym we współpracy z inwestorem i użytkownikami.
6. potrafi przygotować środowisko pracy w projekcie badawczo-rozwojowym.
7. potrafi dostarczać częściowe rezultaty prac wykonywanych w ramach projektu-badawczego.
8. potrafi zaimplementować użyteczny system informatyczny, będący wynikiem prac badawczo-rozwojowych.
9. potrafi zapewnić wysoką jakość systemu informatycznego będącego wynikiem prac badawczo-rozwojowych.
10. potrafi dokumentować architekturę i działanie systemu informatycznego będącego wynikiem prac badawczo-rozwojowych.
11. potrafi przygotować prezentację dla potencjalnych inwestorów projektu badawczo-rozwojowego
12. potrafi opracować artefakty zgodne z metodyką PMBOK
13. potrafi zdefiniować pryncypia, tematy i role zgodnie z metodyką PRINCE2
14. potrafi zastosować zasady bezpieczeństwa aplikacji do projektu badawczo-rozwojowego

**w zakresie kompetencji społecznych:**

1. jest gotów do przygotowywania dla interesariuszy biznesowych dokumentacji architektury i działanie systemu informatycznego będącego wynikiem prac badawczo-rozwojowych.

### **Treści programowe dla zajęć:**

Stworzenie zespołu projektowego uwzględniające społeczne aspekty pracy zespołowej. Wykonanie ćwiczeń mających na celu weryfikację współpracy w zespole. Podział ról w zespole projektowym.

Opracowanie koncepcji projektu badawczo-rozwojowego. Konsultowanie koncepcji projektu z przedstawicielami podmiotów gospodarczych, instytucji administracyjnych lub jednostek badawczych. Opracowanie wizji systemu informatycznego realizującego koncepcję projektu. Konsultowanie wizji systemu informatycznego z jego interesariuszami.

Przygotowanie środowiska prac projektowych: system kontroli wersji, narzędzia do planowania i kontroli wykonywanych zadań. Opracowanie makiety dynamicznej dla Minimum Value Product.

Opracowanie prezentacji dla inwestorów projektu badawczo-rozwojowego.

Opracowanie artefaktów zgodnych z metodyką Scrum (dziennik projektu, dziennik sprintu). Wykonanie zaplanowanych zadań programistycznych. Zaplanowanie kolejnych zadań do wykonania.

Opracowanie artefaktów zgodnych z metodyką PMBOK: zakres projektu, siatka podziału zadań, wykres Gantta. Wykonanie zaplanowanych zadań programistycznych. Zaplanowanie kolejnych zadań do wykonania.

Odniesienie pojęć związanych z metodyką PRINCE2 do prowadzonego projektu: pryncypia, tematy i role. Wykonanie zaplanowanych zadań programistycznych. Zaplanowanie kolejnych zadań do wykonania.

Zarządzanie bezpieczeństwem aplikacji.

Określenie zależności projektu. Zastosowanie zaleceń bezpieczeństwa OWASP. Wykonanie ćwiczeń związanych z identyfikacją podatności aplikacji.

Przeprowadzenie testowania prototypu na poziomie systemowym i akceptacyjnym. Zlokalizowanie i naprawa defektów w kodzie źródłowym.

Opracowanie demonstracyjnej wersji działającego prototypu systemu. Opracowanie dokumentacji prototypu systemu. Przekazanie demonstracyjnej wersji prototypu do testowania akceptacyjnego.

Przekazanie dokumentacji prototypu jego interesariuszom.

Nazwa zajęć: **Wykrywanie incydentów**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. Zna metodyki zarządzania incydentami i przedstawić ich etapy.
2. Zna metody i technologie wykorzystywane do wykrywania ataków cybernetycznych.
3. Rozumie techniki rekonesansu i techniki ich wykrywania.
4. Rozumie charakterystykę ruchu Command & Control i możliwości jego identyfikacji.
5. Zna techniki eksfiltracji danych i metody ich wykrycia.
6. Zna charakterystykę ataków w środowisku chmurowym i możliwości ich identyfikacji.

**w zakresie umiejętności:**

1. Potrafi określić cechy incydentu bezpieczeństwa.
2. Potrafi przygotować raport opisujący incydent bezpieczeństwa.
3. Potrafi wykorzystać wywiad zagrożeń do wykrywania incydentów.
4. Potrafi odwzorować incydent wykorzystując modele zagrożeń.
5. Potrafi scharakteryzować technologię SIEM i jej zastosowanie w wykrywaniu incydentów.
6. Potrafi zidentyfikować ataki na zewnętrzne usługi sieciowe.
7. Potrafi rozpoznać i dokonać analizy ataków typu phishing.
8. Potrafi przeprowadzić podstawową analizę złośliwych dokumentów elektronicznych.
9. Potrafi zidentyfikować próby eskalacji uprawnień.
10. Potrafi przedstawić metody utrwalania dostępu i sposoby na ich rozpoznanie.
11. Potrafi przedstawić przykłady technik ruchu poprzecznego i metody ich detekcji.

**Treści programowe dla zajęć:**

Wprowadzenie do wykrywania incydentów bezpieczeństwa: Charakterystyka incydentu bezpieczeństwa. Porównanie metodyk zarządzania incydentami.

Modelowanie i wywiad zagrożeń: Wykrywanie incydentów w oparciu o wywiad zagrożeń oraz wykorzystanie modeli „Cyber Kill Chain” i Mitre „Att&ck” do przedstawienia cyklu ataku cybernetycznego.

Metody i systemy wykrywania ataków: Zapoznanie z podstawowymi metodami wykrywania ataków. Porównanie możliwości systemów IDS, EDR, WAF oraz UBA.

Technologia SIEM: Zapoznanie się z możliwościami wykorzystania rozwiązań SIEM do identyfikacji ataków cybernetycznych.

Ataki na perymetr sieci: Przegląd wybranych technik ataków na zewnętrzne usługi sieciowe i aplikacje webowe oraz ich detekcji.

Ataki typu phishing i złośliwe dokumenty: Analiza wybranych wariantów ataków wykorzystujących pocztę elektroniczną i złośliwe dokumenty elektroniczne.

Rekonosans: Wykrywanie aktywności cyberprzestępców związanych z rozpoznaniem otoczenia skompromitowanych systemów informatycznych.

Ruch Command & Control: Poznanie wybranych metod komunikacji z serwerami C2 i opracowanie sposobu ich identyfikacji.

Eskalacja uprawnień: Przedstawienie metod podnoszenia uprawnień oraz ich detekcji

Techniki utrwalania dostępu: Prezentacja sposobów na zapewnienie trwałego dostępu do skompromitowanych systemów i ich identyfikację.

Techniki ruchu poprzecznego: Omówienie kluczowych technik poruszania się po skompromitowanej sieci oraz generowanych przez nie śladów.

Eksfiltracja danych: Poznanie technik eksfiltracji danych oraz metod ich detekcji.

Ataki w chmurze: Metody wykrywania incydentów w środowisku chmurowym.

Nazwa zajęć: **Komputerowe wspomaganie tłumaczenia**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. Zna podstawowe techniki komputerowego wspomaganie tłumaczenia przy użyciu pamięci tłumaczeń.

2. Zna zaawansowane techniki użycia pamięci tłumaczeń oraz kierunki badań w tej dziedzinie.

**w zakresie umiejętności:**

1. Potrafi zaimplementować działający ekstraktor terminologii używając różnych technik rozwiązania problemu.

2. Potrafi zbudować słownik dziedzinowy używając samodzielnie zaprojektowanych algorytmów.

3. Potrafi używać wyrażeń regularnych do zadań związanych ze wspomaganie pracy tłumacza.

4. Zna kryteria użyteczności systemów tłumaczenia automatycznego oraz potrafi dokonać ewaluacji tych systemów.

5. Potrafi używać technik web scrapingu do pozyskiwania zasobów lingwistycznych.

6. Posiada wiedzę i umiejętności w dziedzinie automatycznego urownoleglania tekstów.

7. Potrafi wykonać ewaluację przy użyciu oprogramowania typu key logger.

8. Zna techniki korekty pisowni, potrafi zaimplementować własny korektor pisowni.

9. Zna algorytmy korekty gramatycznej tekstu, potrafi zaimplementować własny korektor gramatyczny na podstawie znanych rozwiązań.

10. Potrafi skonstruować własny system komputerowego wspomaganie tłumaczenia.

**Treści programowe dla zajęć:**

Zapoznanie z podstawowymi pojęciami wspomaganie tłumaczenia. Testowanie działania pamięci tłumaczeń w popularnych programach do wspomaganie tłumaczenia

Zaawansowane użycie pamięci tłumaczeń - ICE matching, fuzzy matching. Implementacja modułu pamięci tłumaczeń w oparciu o bibliotekę Lucene.

Techniki automatycznego zarządzania terminologią, w tym ekstrakcja terminologii. Ewaluacja narzędzi do ekstrakcji fraz i terminologii.

Klasyfikacja dziedzinowa terminologii – techniki automatyczne i półautomatyczne. Tworzenie słowników dziedzinowych.

Preprocessing i postprocessing tłumaczonych tekstów – automatyczne wstawianie elementów formatujących, konwersje dat i liczb. Zastosowanie wyrażeń regularnych do postprocessingu.

Tłumaczenie automatyczne jako technika wspomaganie tłumaczenia. Ewaluacja jakości oraz możliwości wykorzystania tłumaczenia automatycznego do wspomaganie tłumaczenia.

Web scraping - pozyskiwanie danych na potrzeby wspomaganie tłumaczenia. Praktyczne ćwiczenia z web scrapingu.

Urownoleglanie jako technika tworzenia pamięci tłumaczeń. Eksperymenty z wykorzystaniem dostępnych narzędzi do urownoleglania.

Techniki badania wydajności procesu tłumaczenia - key logging, eye tracking. Uruchomienie keyloggera (Translog) i analiza wyników.

Algorytmy automatycznej korekty pisowni. Ewaluacja/implementacja korektora pisowni.

Automatyczna korekta gramatyczna tekstu. Ewaluacja wybranych narzędzi do korekty gramatycznej (Grammarly, MS Word).

Projekt własnego mechanizmu wspomaganie tłumaczenia. Implementacja i ewaluacja wybranej techniki wspomaganie tłumaczenia.

Nazwa zajęć: **Algorytmy przybliżone i metaheurystyczne**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. zna pojęcie enumeratywnego algorytmu dokładnego.
2. zna pojęcie algorytmu podziału i ograniczeń.
3. zna pojęcie algorytmu przybliżonego.
4. zna pojęcie schematu aproksymacyjnego.
5. zna pojęcie i własności wybranych algorytmów heurystycznych.
6. zna pojęcie i własności wybranych algorytmów przeszukiwania lokalnego.
7. zna pojęcie i własności wybranych algorytmów metaheurystycznych.
8. zna zasady przeprowadzania eksperymentów obliczeniowych.

**w zakresie umiejętności:**

1. potrafi zastosować algorytm enumeratywny do rozwiązywania wybranych problemów optymalizacji kombinatorycznej oraz zaimplementować go w języku programowania.
2. potrafi zastosować algorytm podziału i ograniczeń do rozwiązywania wybranych problemów optymalizacji kombinatorycznej oraz zaimplementować go w wybranym języku programowania.
3. potrafi zastosować algorytm przybliżony do rozwiązywania wybranych problemów optymalizacji kombinatorycznej oraz zaimplementować go w wybranym języku programowania.
4. potrafi zastosować schemat aproksymacyjny do rozwiązania wybranych problemów optymalizacji kombinatorycznej.
5. potrafi zastosować algorytmy heurystyczne do rozwiązywania wybranych problemów optymalizacji kombinatorycznej oraz zaimplementować je w wybranym języku programowania.
6. potrafi zastosować algorytmy przeszukiwania lokalnego do rozwiązywania wybranych problemów optymalizacji kombinatorycznej oraz zaimplementować je w wybranym języku programowania.
7. potrafi zastosować algorytmy metaheurystyczne do rozwiązywania wybranych problemów optymalizacji kombinatorycznej oraz zaimplementować w wybranym języku programowania.
8. potrafi przeprowadzić eksperyment obliczeniowy dla wybranych algorytmów heurystycznych i metaheurystycznych.

**Treści programowe dla zajęć:**

Algorytmy dokładne: algorytmy generowania permutacji i podzbiorów, algorytmy enumeratywne, algorytm podziału i ograniczeń.  
Algorytmy przybliżone (aproksymacyjne), współczynnik najgorszego przypadku, schematy aproksymacyjne: PTAS, FPTAS.  
Algorytmy heurystyczne: konstrukcyjne, zachłanne. Zasady przeprowadzania analizy eksperymentalnej, generowanie instancji testowych, eksperymenty obliczeniowe.  
Algorytmy przeszukiwania lokalnego: iteracyjnego poprawiania, największego spadku.  
Algorytmy metaheurystyczne: przeszukiwanie tabu, algorytmy genetyczne, algorytmy ewolucyjne.

**Nazwa zajęć: Uczenie głębokie w widzeniu komputerowym**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. zna problematykę dziedziny widzenia komputerowego.
2. zna architektury modeli głębokich sieci neuronowych wykorzystywanych w widzeniu komputerowym.

**w zakresie umiejętności:**

1. potrafi tworzyć, trenować i wykorzystywać architektury głębokich sieci neuronowych do zadań widzenia komputerowego.

**w zakresie kompetencji społecznych:**

1. jest świadomy wyzwań i problemów społecznych stojących za stosowaniem głębokich sieci neuronowych w widzeniu komputerowym

**Treści programowe dla zajęć:**

Podpisywanie obrazów (image captioning)

Przeniesienie stylu obrazu (image style transfer)

Autoencodery

Generowanie obrazów - VAE, GAN, CGAN DCGAN, WGAN, ACGAN, LSGAN

Wykrywanie obiektów - YOLO

Wzmacnianie i segmentacja obrazów - FCC, Autoencodery, Mask-RCNN, Faster R-CNN

Wzmacnianie obrazów - GAN, sztuczne dane, VAE, DANN, zastosowanie do kierowania pojazdem

Klasyfikacja wideo - RNN, LSTM

AI Painter - iGAN, GauGAN

Generowanie twarzy o wysokiej wierności (high-fidelity face generation) - proGAN, StyleGan

Podejście self-attention w generowaniu obrazów - SAGAN, BigGAN

Zamiana tekstu w obraz (text to image) - StackGAN, AttnGAN  
Retargeting wideo - pose transfer, motion transfer  
Rendering neuronowy - rekonstrukcja obiektów 3D

Nazwa zajęć: **Podstawy bezpieczeństwa komputerowego**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. zna popularne wektory ataku na infrastrukturę lokalną lub zdalną.

**w zakresie umiejętności:**

1. potrafi wykryć użycie i zapobiegać wystąpieniu popularnych wektorów ataków na infrastrukturę lokalną lub zdalną.

2. potrafi prowadzić przegląd kodu w celu wyeliminowania faktycznych i potencjalnych podatności.

3. potrafi czerpać informacje o bezpieczeństwie komputerowym z właściwych źródeł.

4. potrafi właściwie przeciwdziałać potencjalnym incydentom w odniesieniu do bieżącego stanu zagrożeń.

**w zakresie kompetencji społecznych:**

1. rozumie potrzebę ustawicznego podnoszenia wiedzy i umiejętności z zakresu bezpieczeństwa komputerowego.

2. rozumie kulturowe i socjologiczne uwarunkowania bezpieczeństwa komputerowego.

**Treści programowe dla zajęć:**

Identyfikacja i zapobieganie podatnościom. Najbardziej rozpoznawalne grupy problemów. Symptomy podatności i standardowe rozwiązania. Przeprowadzenie eksperymentów związanych z identyfikacją problemów bezpieczeństwa.

Przegląd narzędzi do wykrywania podatności. Omówienie popularnych rozwiązań dla danego zastosowania. Przeprowadzenie eksperymentów polegający na wykorzystaniu podatności.

Systemy monitoringu infrastruktury: Wyzwania. Przegląd produktów i rozwiązań. Przeprowadzenie eksperymentów związanych z użyciem wybranych rozwiązań.

Systemy wykrywania wtargnięć i ich raportowania. Przegląd rozwiązań. Przeprowadzenie eksperymentów związanych z użyciem wybranych rozwiązań.

Nazwa zajęć: **Seminarium magisterskie 2 - zdalne**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. zna podstawy metodologii prowadzenia badań w obszarze danego kierunku.

2. zna metody projektowania rozwiązań oraz modyfikacji istniejących rozwiązań problemów badawczych na potrzeby wykorzystania ich w gospodarce

**w zakresie umiejętności:**

1. potrafi wykorzystać pogłębioną wiedzę z zakresu wybranych obszarów informatyki do rozwiązywania problemów stawianych podczas zajęć.

2. potrafi identyfikować problemy badawcze.

3. potrafi dobrać właściwe narzędzia badawcze, a także projektować warsztat badawczy.

4. potrafi projektować rozwiązania lub modyfikacje istniejących rozwiązań.

5. potrafi przygotować i zaprezentować krótkie opracowanie wybranego problemu w sposób zrozumiały dla innych uczestników; potrafi redagować spójną i logiczną wypowiedź z wykorzystaniem poprawnej i terminologii.

6. potrafi poszerzać swoją wiedzę poprzez samodzielne poszukiwania w istniejących opracowaniach naukowych.

7. potrafi samodzielnie rozwiązać problem teoretyczny lub praktyczny.

**w zakresie kompetencji społecznych:**

1. jest gotów do pogłębiania wiedzy z zakresu wybranych obszarów informatyki związanych z problematyką seminarium.

2. jest gotów do identyfikowania nowych problemów badawczych.

**Treści programowe dla zajęć:**

Prezentacja przez studentów opracowań wybranych problemów i projektów badawczych zgodnych z zadaniami realizowanymi w pracy magisterskiej.

Raportowanie i prezentowanie wykonania badań teoretycznych związanych z tematyką pracy magisterskiej.

Raportowanie i prezentowanie wykonania prac rozwojowych związanych z tematyką pracy magisterskiej.

Dostarczenie ustalonego fragmentu pracy magisterskiej.

**Nazwa zajęć: Seminarium magisterskie 3 - stacjonarne**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna podstawy metodologii prowadzenia badań w obszarze danego kierunku.
2. zna metody projektowania rozwiązań oraz modyfikacji istniejących rozwiązań problemów badawczych na potrzeby wykorzystania ich w gospodarce

**w zakresie umiejętności:**

1. potrafi wykorzystać pogłębioną wiedzę z zakresu wybranych obszarów informatyki do rozwiązywania problemów stawianych podczas zajęć.
2. potrafi identyfikować problemy badawcze.
3. potrafi dobierać właściwe narzędzia badawcze, a także projektować warsztat badawczy.
4. potrafi projektować rozwiązania lub modyfikacje istniejących rozwiązań.
5. potrafi przygotować i zaprezentować krótkie opracowanie wybranego problemu w sposób zrozumiały dla innych uczestników; potrafi redagować spójną i logiczną wypowiedź z wykorzystaniem poprawnej i terminologii.
6. potrafi poszerzać swoją wiedzę poprzez samodzielne poszukiwania w istniejących opracowaniach naukowych.
7. potrafi samodzielnie rozwiązać problem teoretyczny lub praktyczny.

**w zakresie kompetencji społecznych:**

1. jest gotów do pogłębiania wiedzy z zakresu wybranych obszarów informatyki związanych z problematyką seminarium.
2. jest gotów do identyfikowania nowych problemów badawczych.

**Treści programowe dla zajęć:**

Prezentacja przez studentów opracowań wybranych problemów i projektów badawczych zgodnych z zadaniami realizowanymi w pracy magisterskiej.

Raportowanie i prezentowanie wykonania badań teoretycznych związanych z tematyką pracy magisterskiej.

Raportowanie i prezentowanie wykonania prac rozwojowych związanych z tematyką pracy magisterskiej.

Dostarczenie kolejnych rozdziałów pracy magisterskiej.

Dostarczenie końcowej wersji pracy magisterskiej.

**Nazwa zajęć: Programowanie na GPU**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna i rozumie metody oraz problemy związane z obliczeniami równoległymi na GPU.

**w zakresie umiejętności:**

1. potrafi tworzyć aplikacje do obliczeń równoległych na GPU z wykorzystaniem bibliotek OpenCL.
2. potrafi tworzyć aplikacje do obliczeń równoległych na GPU z wykorzystaniem bibliotek CUDA.

**w zakresie kompetencji społecznych:**

1. jest świadomy problematyki badawczej stojącej za zagadnieniem przetwarzania równoległego na GPU.

**Treści programowe dla zajęć:**

Współczesne architektury GPU

OpenCL: Platform Model, Execution Model, Programming Model (Kernels), Memory model (host/device)

OpenCL – proste przykłady

OpenCL – profilowanie i debugowanie

Kolizje w Bullet Physics – OpenCL

Fast Subdivision w Open Subdiv – OpenCL

C++Amp – integracja C++ i OpenCL

WebCL – OpenCL w aplikacjach przeglądarkowych

CUDA – proste przykłady

CUDA – wątki, strumienie

CUDA – profilowanie i debugowanie

Integracja Pythona i CUDA (Numba, CuPy)

Symulacja płynów – CUDA

Zastosowania naukowe – CUDA

**Nazwa zajęć: Wizualizacja danych**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. Zna podstawowe typy wykresów wykorzystywane w analizie danych.

**w zakresie umiejętności:**

1. Potrafi przygotować wykresy korzystając z języka R.
2. Potrafi wykorzystać bibliotekę ggplot2 do przygotowania wykresów statycznych.
3. Potrafi wykorzystać poznane wykresy do sformułowania i wizualizacji problemu analizy rzeczywistego zbioru danych za pomocą biblioteki ggplot2.
4. Potrafi wykorzystać różne biblioteki do tworzenia wykresów interaktywnych.
5. Potrafi prezentować dane na mapach.
6. Potrafi wykorzystać poznane biblioteki do formułowania i wizualizacji problemu analizy rzeczywistego zbioru danych; umie przedstawić swoje wnioski płynące z wizualizacji w postaci raportu i zaprezentować je w sposób popularny.
7. Zna podstawowe typy dashboardów. Potrafi je przygotować z wykorzystaniem biblioteki shiny.
8. Potrafi wykorzystać poznane biblioteki do tworzenia dashboardów

**w zakresie kompetencji społecznych:**

1. Umie wyciągnąć wnioski płynące z wizualizacji i przedstawić je w postaci raportu w sposób popularny.
2. Umie przedstawić swoje wnioski płynące z wizualizacji w postaci raportu i zaprezentować je w sposób popularny.

**Treści programowe dla zajęć:**

Podstawowe typy wykresów wykorzystywane w analizie danych.

Biblioteka ggplot2.

Przygotowanie projektu wykorzystującego możliwości biblioteki ggplot2. Przygotowanie prezentacji omawiającej wyniki projektu wykorzystującego bibliotekę ggplot2. Prezentacja projektów wykorzystujących bibliotekę ggplot2.

Wykresy interaktywne. Biblioteki: plotly, NVD3, MorrisJS.

Przedstawianie danych na mapach. Biblioteki: leaflet oraz maps.

Przygotowanie projektu prezentującego możliwości interaktywnej wizualizacji danych oraz przedstawiania danych na mapach. Przygotowanie prezentacji omawiającej wyniki projektu wykorzystującego wykresy interaktywne oraz mapy. Prezentacja projektów wykorzystujących wykresy interaktywne oraz mapy.

Dashboardy. Biblioteka shiny.

Przygotowanie projektu prezentującego wyniki raportów za pomocą dashboardów. Przygotowanie prezentacji omawiającej projekty wykorzystujące dashboardy. Prezentacja projektów wykorzystujących dashboardy.

**Nazwa zajęć: Przygotowanie do projektu badawczo-rozwojowego**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. rozumie społeczne aspekty pracy zespołowej w projekcie badawczo-rozwojowym.
2. zna specyfikę zarządzania projektem badawczo-rozwojowym.
3. zna zasady zarządzania za pomocą metodyk zwinnych
4. zna zasady zarządzania za pomocą metodyki PMBOK
5. zna zasady zarządzania za pomocą metodyki PRINCE2
6. zna i rozumie proces ciągłej integracji
7. zna i rozumie procesy ewaluacji i estymacji jakości systemu uczenia maszynowego
8. zna zagrożenia związane z bezpieczeństwem aplikacji komputerowych
9. zna podstawowe metody ataków w sieciach komputerowych
10. zna podstawowe metody testowania aplikacji komputerowych
11. zna aspekty użyteczności aplikacji komputerowych
12. zna parametry i metryki jakości systemu informatycznego

**w zakresie umiejętności:**

1. potrafi określić cechy innowacyjnego projektu informatycznego.
2. potrafi wizualizować system informatyczny za pomocą makiety dynamicznej.
3. potrafi pozyskiwać inwestorów dla projektu badawczo-rozwojowego,
4. umie przygotować prezentację koncepcji projektu badawczo-rozwojowego.
5. potrafi publicznie prezentować koncepcję projektu badawczo-rozwojowego.
6. potrafi uczestniczyć w projekcie prowadzonym we współpracy z inwestorem i użytkownikami.



7. potrafi dostarczać częściowe rezultaty prac wykonywanych w ramach projektu-badawczego.
8. potrafi opracować specyfikację zakresu systemu informatycznego.
9. potrafi przeprowadzić proces testowania w różnych metodykach programowania
10. umie zorganizować proces testowania integracyjnego i systemowego.
11. potrafi zapewnić użyteczność systemu informatycznego, będącego wynikiem prac badawczo-rozwojowych.
12. potrafi uruchomić procesy weryfikujące jakość systemu informatycznego.
13. potrafi przedstawić cele i działanie systemu informatycznego jego interesariuszom.
14. potrafi przygotować i przeprowadzić demonstrację systemu informatycznego.
15. potrafi przygotować poster i uczestniczyć w sesji plakatowej związanej z projektem badawczo-rozwojowym
16. potrafi korzystać z systemów ciągłej integracji
17. potrafi korzystać z systemów ciągłej ewaluacji
18. potrafi pisać bezpieczne aplikacje komputerowe
19. potrafi zapobiegać podstawowym atakom w sieciach komputerowych

**w zakresie kompetencji społecznych:**

1. jest gotów do przedstawiania celów i działania systemów informatycznych w celu pozyskiwania inwestorów.
2. jest gotów do współpracy z zespołem projektowym

**Treści programowe dla zajęć:**

Pojęcie projektu badawczo-rozwojowego.

Działalność badawczo-rozwojowa. Cechy projektu badawczo-rozwojowego. Poziomy gotowości technologicznej. Cechy produktu zaawansowanych technologii (ang. high tech product). Czynniki sukcesu rynkowego produktu zaawansowanych technologii.

Pozyskiwanie kapitału na projekt badawczo-rozwojowy.

Typy inwestorów. Etapy procesu inwestycyjnego. Zespół, produkt i rynek jako kryteria oceny ryzyka inwestycyjnego. Pojęcie "Unique Value Proposition". Pojęcie "Minimum Viable Product". Zasady tworzenia prezentacji dla inwestorów.

Transfer wiedzy do gospodarki.

Pojęcie innowacji. Typy innowacji. Własność intelektualna w dziedzinie oprogramowania. Sposoby wdrażania wyników projektów do gospodarki. Badanie rynku odbiorców. Tworzenie "Business Canvas Model".

Prezentacja koncepcji projektu badawczo-rozwojowego.

Publiczne prezentacje w wykonaniu studentów na forum inwestorów. Ocena prezentacji przez wykładowcę i inwestorów. Dyskusja i wnioski.

Metodyki zwinne w tworzeniu oprogramowania.

Manifest Agile. Manifest zespołu projektowego. Pryncypia metodyk zwinnych. Metodyka Scrum. Przykłady błędnego stosowania metodyk zwinnych.

Zarządzanie projektem informatycznym. Metodyka PMBOK.

Zarządzanie projektem informatycznym: Kryzys oprogramowania i jego przyczyny. Pojęcie projektu informatycznego. Interesariusze projektu. Parametry projektu. Trójkąt zakresu.

Metodyka PMBOK: Fazy i cykl życia projektu. Iteracyjność w procesie zarządzania projektem informatycznym. Obszary wiedzy.

Metodyka PRINCE2.

Geneza powstania metodyki PRINCE2. Pryncypia. Tematy. Role. Komitet sterujący i poziomy zarządzania. Procesy. Środowisko. Dokumentacja. Wady, zalety i wskazówki w stosowaniu metodyki PRINCE2.

Ciągła integracja i ciągła ewaluacja w systemach uczenia maszynowego.

Dobre praktyki w tradycyjnej inżynierii oprogramowania: ciągła integracja i ciągłe wdrażanie. Problemy ciągłej integracji w systemach uczenia maszynowego. Tworzenie systemów opartych na uczeniu maszynowym. Ewaluacja systemów uczenia maszynowego. Metryki ewaluacji. Przykład systemu ewaluacji. Estymacja jakości systemu uczenia maszynowego.

Bezpieczeństwo aplikacji.

Zagadnienie cyberbezpieczeństwa

Testowanie aplikacji zaawansowanych technologii

Zagadnienia użyteczności

Jakość systemu informatycznego

Przygotowywanie demonstracji posterowej

Demonstracja posterowa

**Nazwa zajęć: Seminarium magisterskie 3 - zdalne**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. zna podstawy metodologii prowadzenia badań w obszarze danego kierunku.
2. zna metody projektowania rozwiązań oraz modyfikacji istniejących rozwiązań problemów badawczych na potrzeby wykorzystania ich w gospodarce

**w zakresie umiejętności:**

1. potrafi wykorzystać pogłębioną wiedzę z zakresu wybranych obszarów informatyki do rozwiązywania problemów stawianych podczas zajęć.
2. potrafi identyfikować problemy badawcze.
3. potrafi dobierać właściwe narzędzia badawcze, a także projektować warsztat badawczy.
4. potrafi projektować rozwiązania lub modyfikacje istniejących rozwiązań.
5. potrafi przygotować i zaprezentować krótkie opracowanie wybranego problemu w sposób zrozumiały dla innych uczestników; potrafi redagować spójną i logiczną wypowiedź z wykorzystaniem poprawnej i terminologii.
6. potrafi poszerzać swoją wiedzę poprzez samodzielne poszukiwania w istniejących opracowaniach naukowych.
7. potrafi samodzielnie rozwiązać problem teoretyczny lub praktyczny.

**w zakresie kompetencji społecznych:**

1. jest gotów do pogłębiania wiedzy z zakresu wybranych obszarów informatyki związanych z problematyką seminarium.
2. jest gotów do identyfikowania nowych problemów badawczych.

**Treści programowe dla zajęć:**

Prezentacja przez studentów opracowań wybranych problemów i projektów badawczych zgodnych z zadaniami realizowanymi w pracy magisterskiej.

Raportowanie i prezentowanie wykonania badań teoretycznych związanych z tematyką pracy magisterskiej.

Raportowanie i prezentowanie wykonania prac rozwojowych związanych z tematyką pracy magisterskiej.

Dostarczenie kolejnych rozdziałów pracy magisterskiej.

Dostarczenie końcowej wersji pracy magisterskiej.

**Nazwa zajęć: Język angielski C12**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie umiejętności:**

1. potrafi w sposób przystępny przedstawić fakty z zakresu informatyki, porozumiewać się przy użyciu różnych technik w środowisku zawodowym oraz w innych środowiskach w języku angielskim.
2. potrafi przygotować dokumentacje, opracowania i raporty w języku angielskim.
3. potrafi przygotować wystąpienia ustne w języku angielskim, dotyczące zagadnień teoretycznych i praktycznych informatyki.
4. potrafi posługiwać się językiem obcym na poziomie C1 Europejskiego Systemu Opisu Kształcenia Językowego oraz zna język angielski w stopniu umożliwiającym czytanie ze zrozumieniem dokumentacji oraz artykułów dotyczących informatyki.
5. zna i potrafi stosować słownictwo dotyczące informatyki.

**Treści programowe dla zajęć:**

Praca podczas zajęć (nad wspólnym zadaniem, wymiana zdobytych informacji, zadania typu information gap).

Tłumaczenie na język angielski oraz zwrótnie na język polski fragmentów dokumentacji i opracowań z wykorzystaniem cech stylu naukowego.

Mini-prezentacje podczas każdego zajęcia prezentujące wyniki zadania domowego oraz wyniki pracy w grupie (in-class). Prezentacja zaliczeniowa na koniec semestru podsumowująca wyniki pracy podczas semestru na samodzielnie wybrany i opracowany temat z dziedziny informatyki.

Praca z artykułami (IT) oraz artykułami naukowymi (IT-study, case study) dotyczącymi informatyki.

Praca ze słownictwem specjalistycznym z użyciem tekstów oraz prezentacji audio/video o zróżnicowanej tematyce i stopniu trudności. Wykonywanie ćwiczeń pisemnych i ustnych aktywizujących zapamiętywanie słownictwa poprzez jego użycie w analogicznym kontekście.

**Nazwa zajęć: Zaawansowane modelowanie geometryczne**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna zagadnienia krzywych i powierzchni niejawnych stosowanych w programowaniu gier komputerowych.
2. zna zagadnienia numerycznej analizy izogeometrycznej stosowanej w programowaniu gier komputerowych.
3. zna sposoby modelowania proceduralnego wykorzystywanych w programowaniu gier komputerowych.
4. zna metody modelowania za pomocą baz hierarchicznych, w tym falkowych, stosowanych w programowaniu gier komputerowych.

**w zakresie umiejętności:**

1. potrafi wykorzystać w programowaniu gier komputerowych krzywe i powierzchnie niejawne.
2. potrafi wykorzystać w programowaniu gier komputerowych metody numerycznej analizy izogeometrycznej.
3. potrafi wykorzystać w programowaniu gier komputerowych metody modelowania proceduralnego i modelowania.
4. potrafi wykorzystać w programowaniu gier komputerowych metody modelowania za pomocą baz hierarchicznych, w tym falkowych.

**Treści programowe dla zajęć:**

Podstawy matematyczne krzywych i powierzchni niejawnych: funkcje i odwzorowania, odwracalność i gładkość, metoda poziomicy, podrozmaitości.

Przestrzenne struktury danych dla krzywych i powierzchni niejawnych: metody podziału obiektu (stratyfikacja, dekompozycja komórkowa, dekompozycja sympleksyjna), metody podziału przestrzeni (drzewa: BSP, Kd, czwórkowe, ósemkowe).

Metody próbkowania krzywych i powierzchni niejawnych: root isolation, interval arithmetic, root finding. Rekonstrukcja i generowanie siatek dla krzywych i powierzchni niejawnych: metody kontynuacji, metody podziału przestrzennego.

Dopasowywanie krzywych i powierzchni niejawnych - powierzchnie: blob, LS, RBF, MPU. Niejawne modelowanie zjawisk złożonych: techniki niejawnego modelowania szkieletowego, niejawne modelowanie statycznych obiektów przyrodniczych (przykład muszli Murex Cabritii).

Wprowadzenie do analizy elementów skończonych.

Analiza izogeometryczna: struktury topologiczne dla powierzchni i obiektów wolumetrycznych, metody reprezentacji.

Metody numeryczne dla analizy izogeometrycznej.

Metody reprezentacji i rekonstrukcji terenu.

Bazy hierarchiczne dla reprezentacji obiektów geometrycznych.

Analiza falkowa w modelowaniu geometrycznym.

Nazwa zajęć: **Protokoły kryptograficzne**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna współczesne protokoły kryptograficzne oraz zasady działania bezpiecznej komunikacji środowiska pracy.

**w zakresie umiejętności:**

1. potrafi bezpiecznie skonfigurować środowisko kryptograficzne.

**Treści programowe dla zajęć:**

Infrastruktura Klucza Publicznego PKI.

Certyfikaty X.509.

Protokół TLS 1.3.

Protokół WireGuard.

Nieinteraktywny protokół zk-SNARK.

Nazwa zajęć: **Warsztaty tłumaczenia automatycznego**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. Rozumie pojęcie „jednostka nazwana” i potrafi wskazać jej wystąpienia w tekście.

**w zakresie umiejętności:**

1. Potrafi przygotować korpus danych niezbędny do stworzenia modelu rozpoznawania jednostek nazwanych.
2. Potrafi wytrenować model rozpoznawania jednostek nazwanych.
3. Umie stworzyć system rozpoznający jednostki nazwane.
4. Umie ocenić skuteczność systemu rozpoznawania jednostek nazwanych.

5. Rozumie pojęcie korpusu równoległego i potrafi zastosować w praktyce metody jego pozyskania.
6. Potrafi wydzielić zbiory danych konieczne do wytrenowania i ewaluacji modelu neuronowego tłumaczenia automatycznego.
7. Rozumie pojęcie jednostek podwyrazowych i potrafi zastosować w praktyce algorytm Byte Pair Encoding w celu przetworzenia zbiorów danych.
8. Potrafi wytrenować model neuronowego tłumaczenia automatycznego oraz zastosować go do tłumaczenia tekstu.
9. Potrafi zastosować rozpoznawanie jednostek nazwanych w celu poprawienia jakości tłumaczenia automatycznego.
10. Umie ocenić jakość modelu tłumaczenia automatycznego.

**Treści programowe dla zajęć:**

Pojęcie i charakterystyka jednostek nazwanych. Różnica pomiędzy rozpoznawaniem jednostek nazwanych a ekstrakcją informacji. Wizualizacja jednostek nazwanych w tekście za pomocą biblioteki spaCy.

Opracowanie cech dobrego korpusu danych dla zadania rozpoznawania jednostek nazwanych oraz metod jego pozyskiwania. Pozyskanie korpusu danych i przygotowanie go do przetworzenia za pomocą biblioteki flair.

Implementacja modułu trenującego model rozpoznawania jednostek nazwanych. Wykorzystanie biblioteki flair.

Implementacja systemu rozpoznającego jednostki nazwane w języku programowania Python 3.

Charakterystyka metod ewaluacji systemu rozpoznawania jednostek nazwanych. Ewaluacja stworzonego systemu.

Pojęcie korpusu równoległego. Pozyskanie korpusów równoległych ze zbiorów ogólnodostępnych. Zastosowanie narzędzi ekstrakcji tekstu z dokumentów. Zastosowanie narzędzia hunalign w celu stworzenia korpusu równoległego.

Wstępne przetwarzanie korpusu. Wydzielenie zbiorów danych uczących i testowych z korpusu przy zachowaniu odpowiednich proporcji.

Pojęcie jednostek podwyrazowych i algorytm Byte Pair Encoding. Porównanie narzędzi Subword-NMT, fastBPE i SentencePiece. Wytrenowanie modelu SentencePiece i zastosowanie go w celu przetworzenia zbiorów danych.

Zaznajomienie z narzędziem fairseq. Binarzacja danych oraz wytrenowanie modelu tłumaczenia automatycznego. Zastosowanie wytrenowanego modelu do tłumaczenia tekstu.

Pojęcie ograniczeń leksykalnych. Zastosowanie rozpoznawanych jednostek nazwanych jako ograniczeń leksykalnych podczas tłumaczenia automatycznego w celu polepszenia jakości tłumaczenia.

Ewaluacja ludzka oraz ewaluacja automatyczna z wykorzystaniem metryki BLEU. Ograniczenia metryki BLEU. Wykorzystanie narzędzia GEval do automatycznej ewaluacji stworzonego systemu tłumaczenia.

**Nazwa zajęć: Programowanie w języku C**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna i rozumie koncepcje języka ANSI C oraz jego ograniczenia.
2. zna i rozumie rolę, możliwości i ograniczenia biblioteki standardowej.
3. zna i rozumie zasady zarządzania zasobami w systemach uniksopodobnych.
4. zna i rozumie problemy współbieżności i synchronizacji.

**w zakresie umiejętności:**

1. potrafi posługiwać się językiem ANSI C.
2. potrafi korzystać z biblioteki standardowej oraz funkcji systemowych.
3. potrafi posługiwać się kompilatorem oraz wybranymi narzędziami wspomagającymi proces kompilacji.
4. potrafi implementować proste programy wielowątkowe.
5. potrafi utworzyć bibliotekę statyczną i dynamiczną.
6. potrafi pogłębiać swoją wiedzę w oparciu o dokumentację języka i wykorzystywane narzędzia.
7. potrafi samodzielnie zgłębiać wiedzę z obszarów programowania i systemów operacyjnych.

**Treści programowe dla zajęć:**

Podstawowe koncepcje języka ANSI C. Zmienne i funkcje, struktury i unie, wskaźniki i tablice, operatory, instrukcje warunkowe, pętle, podstawowe operacje wejścia/wyjścia. Elementy procesu kompilacji. Narzędzia gcc i clang.

Standardowe wejście i wyjście. Argumenty przekazywane podczas wywoływania programu.

Biblioteka standardowa i matematyczna. Samodzielna implementacja wybranych funkcji z biblioteki standardowej.

Funkcje systemowe. Operacje na plikach. Samodzielna implementacja wybranych narzędzi (np. cat i cp) z wykorzystaniem funkcji systemowych.

Struktura procesu. Stos i sarta. Zarządzanie pamięcią i zaawansowane operacje na wskaźnikach. Samodzielna implementacja podstawowych struktur danych (stos, lista z dowiązaniem, drzewo wyszukiwań binarnych).

Funkcje systemowe. Tworzenie procesów i zarządzanie współdzielonymi zasobami.

Współbieżność i synchronizacja. Obsługa współbieżności przez jądro. Wątki POSIX i synchronizacja wątków.

Tworzenie bibliotek statycznych i dynamicznych.

Integracja kodu w języku C z Pythonem.

Nazwa zajęć: **Zaawansowane algorytmy rozproszone**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. rozumie, w jaki sposób modeluje się obliczenia w sieciach różnego typu.
2. zna i rozumie zaawansowane techniki projektowania algorytmów rozproszonych, złożone z wielu faz, często wykorzystujące pomocnicze struktury danych (spannery, klastry).
3. zna metody ułatwiające tworzenie algorytmów asynchronicznych.

**w zakresie umiejętności:**

1. potrafi twórczo rozwinąć znane algorytmy i/lub znaleźć dla nich nowe zastosowania.
2. potrafi się posługiwać narzędziem do automatycznej analizy algorytmów rozproszonych/współbieżnych.
3. potrafi się posługiwać symulatorem algorytmów rozproszonych w celu eksperymentów, analizy i testowania własnych algorytmów.

**Treści programowe dla zajęć:**

Modelowanie obliczeń rozproszonych w sieciach (model z przesyłaniem komunikatów); wersja synchroniczna i asynchroniczna; pojęcia: konfiguracja, zdarzenie, egzekucja; złożoność komunikatowa i czasowa algorytmów; implementowanie najprostszyc algorytmów rozproszonych w symulatorze.

Problem wyboru lidera (LE) w cyklu, wersje o złożoności komunikatowej  $O(n^2)$  i  $O(n \log n)$  w modelu asynchronicznym; dowód optymalności algorytmu o złożoności  $O(n \log n)$ ; algorytmy LE w modelu synchronicznym o złożoności  $O(n)$ ; implementacja w symulatorze.

Obliczanie 3-kolorowania wierzchołków drzewa metodą Cole&Vishkin oraz różne uogólnienia (w szczególności (Delta+1)-kolorowanie grafu ogólnego) oraz zastosowania tego algorytmu; implementacja w symulatorze; dolne oszacowanie na 2 i 3-kolorowanie wierzchołkowe cyklu.

Obliczanie skojarzeń w grafie metodą skojarzenia frakcyjnego i deterministycznego zaokrąglania (algorytm M. Fischer z 2017); ten sam problem z użyciem tzw. "spannerów"; inne zastosowania spannerów; uogólnienia skojarzeń i ich zastosowania.

Zastosowania struktury danych zwanej "klastrami" (z małym brzegiem) do aproksymowania problemów grafowych: MIS, Maximum Matching, MDS; procedura szybkiego budowania klastów; implementacja elementów tej procedury w symulatorze.

Narzędzia do budowania algorytmów asynchronicznych: synchronizatory alfa, beta, gamma, ni1, eta; uogólnienie synchronizatora alfa; implementacja w symulatorze synchronizatorów alfa i beta.

Algorytmy odporne na błędy sieci: problem Consensus; problem "2 generałów"; dolne oszacowania związane z problemem Consensus; implementacja w symulatorze algorytmu Consensus.

Narzędzie do automatycznej weryfikacji algorytmów rozproszonych: sieci Petriego, spin i język Promela, CSP czyli Communicating Sequential Processes; zastosowanie tych narzędzi do analizy algorytmów asynchronicznych.

Nazwa zajęć: **Zabezpieczenia protokołów sieciowych**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. Rozumie pojęcia związane z transmisją danych w sieciach komputerowych opartych o model TCP/IP.
2. Rozumie niebezpieczeństwo pętli w sieci komputerowej.
3. Zna pojęcie routingu w sieciach komputerowych.
4. Zna metody filtrowania niepożądanego ruchu sieciowego przy użyciu list dostępu.

**w zakresie umiejętności:**

1. Potrafi używać wiersz poleceń systemu Cisco IOS.

2. Potrafi użyć protokoły Telnet i SSH do zdalnego zarządzania sprzętem sieciowym oraz zna ich wady i zalety.
3. Rozumie problemy związane z użyciem dużej liczby komputerów w jednej domenie rozgłoszeniowej i potrafi je rozwiązać stosując VLANy.
4. Potrafi zabezpieczyć urządzenia przed atakiem zalewania adresami MAC.
5. Zna problemy związane z użyciem Dynamic Trunking Protocol i potrafi im zapobiegać.
6. Umie konfigurować Spanning Tree Protocol.
7. Potrafi wdrożyć redundancję łącza.
8. Potrafi użyć protokołów OSPF i zabezpieczyć go.
9. Potrafi użyć protokołów EIGRP i zabezpieczyć go.
10. Potrafi wdrożyć redundancję bramy sieciowej przy użyciu protokołu HSRP.
11. Potrafi przeciwdziałać atakom związanym z protokołem DHCP.
12. Potrafi uruchomić szyfrowany tunel pomiędzy urządzeniami obsługującymi ruch sieciowy.

**Treści programowe dla zajęć:**

Omówienie pojęć związanych z transmisją danych w sieciach komputerowych.  
Zapoznanie ze sprzętem sieciowym. Podstawy obsługi systemu operacyjnego Cisco IOS.  
Konfiguracja protokołów zdalnego zarządzania: Telnet i SSH.  
Zagrożenia związane z dużą domeną rozgłoszeniową. Konfiguracja wirtualnej sieci lokalnej.  
Zabezpieczanie portów przy użyciu Port Security.  
Omówienie Dynamic Trunking Protocol i zagrożeń z nim związanych.  
Spanning Tree Protocol jako rozwiązanie problemu przypadkowych pętli.  
Zabezpieczanie sieci lokalnej przed awarią pojedynczego łącza.  
Podstawy routingu.  
Routing dynamiczny na przykładzie OSPF. Zabezpieczanie protokołu OSPF.  
Routing dynamiczny na przykładzie EIGRP. Zabezpieczanie protokołu EIGRP.  
Redundancja bramy sieciowej.  
Przeciwdziałanie atakom związanym z protokołem DHCP.  
Filtrowanie niepożądanego ruchu sieciowego przy użyciu list dostępu.  
Szyfrowanie ruchu w tunelu sieciowym.

Nazwa zajęć: **Systemy rozmyte**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. zna pojęcie zbioru rozmytego, umie z jego pomocą modelować pojęcia nieprecyzyjne
2. rozumie istotę i przeznaczenie sterowania rozmytego w porównaniu z konwencjonalnymi metodami sterowania, zna zasady działania sterownika rozmytego, potrafi zaprojektować i zaimplementować prosty sterownik rozmyty

**w zakresie umiejętności:**

1. umie operować na zbiorach rozmytych, zna praktyczny aspekt tych operacji
2. zna dostępne biblioteki oprogramowania w standardzie FCL i potrafi je zastosować do implementacji własnych sterowników

**Treści programowe dla zajęć:**

Pojęcie zbioru rozmytego, jego modelowanie i przykłady  
Pojęcie zmiennej lingwistycznej i jej modelowanie  
Sterowanie rozmyte wraz z przykładami  
Język FCL i jego implementacje  
Algorytmy rozmyte w systemach podejmowania decyzji  
Projekt systemu rozmytego

Nazwa zajęć: **Testy penetracyjne**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:**

1. Rozumie etyczne oraz prawne obostrzenia dotyczące testów penetracyjnych.
2. Zna metodykę prowadzenia testu penetracyjnego aplikacji webowej.
3. Zna techniki postexploitacyjne w systemach Linux i Windows.
4. Zna podstawowe techniki omijania mechanizmów bezpieczeństwa.

**w zakresie umiejętności:**

1. Potrafi zainicjować test penetracyjny oraz przygotować niezbędne środowisko pracy.
2. Potrafi przygotować prawidłowy raport z testu penetracyjnego.

3. Potrafi przeprowadzić rekonesans pasywny na temat sieci/domeny/firmy będącej celem testu penetracyjnego.
4. Potrafi przeprowadzić rekonesans aktywny na temat sieci/systemu będącego celem testu penetracyjnego.
5. Potrafi zaprogramować proste narzędzia automatyzujące pracę podczas testu penetracyjnego.
6. Potrafi przeprowadzić enumerację systemu Linux.
7. Potrafi przeprowadzić enumerację systemu Windows.
8. Potrafi wykorzystać w praktyce zagadnienia związane z podnoszeniem uprawnień.
9. Potrafi korzystać ze skanera podatności oraz interpretować wyniki skanów.
10. Potrafi wyszukiwać podatności na podstawie zdobytych informacji.
11. Zna aspekty pracy z exploitami oraz potrafi z nich skorzystać.
12. Potrafi przeprowadzić zdalny atak na systemy uwierzytelniające różnych usług.
13. Potrafi przeprowadzić atak offline na różnego rodzaju skróty haseł.
14. Potrafi mapować aplikację webową oraz wykrywać ukryte zasoby.
15. Potrafi namierzyć i wykorzystać podstawowe błędy obsługi danych od użytkownika.
16. Potrafi namierzyć i wykorzystać podatności związane z obsługą plików
17. Potrafi namierzyć i wykorzystać inne podatności w aplikacjach webowych.
18. Potrafi wykorzystać oprogramowanie podczas testu penetracyjnego.
19. Zna zasadę działania ataków typu client-side.
20. Potrafi przeprowadzić symulację ataku phishingowego.
21. Potrafi przełamać zabezpieczenia sieci-wifi opartej o WPA2.

**w zakresie kompetencji społecznych:**

1. Zna metodykę ataków socjotechnicznych.

**Treści programowe dla zajęć:**

Wprowadzenie do testów penetracyjnych: Metodyki prowadzenia testów penetracyjnych. Zapoznanie z etycznymi oraz prawnymi aspektami testów penetracyjnych. Wskazanie kierunków rozwoju, źródeł i sposobów nabywania umiejętności oraz omówienie przydatnej literatury. Przygotowanie środowiska laboratoryjnego.

Raportowanie testu penetracyjnego: Przygotowanie szablonu raportu z testu penetracyjnego. Omówienie istotnych elementów raportu oraz kryteriów świadczących o jego jakości. Otwarcie raportu z opisem przeprowadzonych działań w trakcie trwania kursu, który będzie głównym elementem zaliczenia kursu.

Rekonesans pasywny: Omówienie metod przeprowadzania białego wywiadu, informacji jakie mogą być przydatne do dalszych działań oraz przedstawienie publicznie dostępnych źródeł.

Rekonesans aktywny: Wprowadzenie do aktywnej enumeracji sieci wraz z przedstawieniem niezbędnika narzędziowego każdego pentestera.

Programowanie: Automatyzacja prac wykonywanych przez pentestera z wykorzystaniem języków programowania i poleceń powłoki (Python, Bash, Powershell).

Enumeracja w systemach Linux: Przeprowadzenie rekonesansu w systemie Linux. Wykorzystanie błędów konfiguracji systemu i działającym na nich usług do podniesienia uprawnień oraz pozyskania informacji użytecznych podczas testu penetracyjnego.

Enumeracja w systemach Windows: Przeprowadzenie rekonesansu w systemie Windows. Wykorzystanie błędów konfiguracji systemu i działającym na nich usług do podniesienia uprawnień oraz pozyskania informacji użytecznych podczas testu penetracyjnego.

Ocena podatności: Analiza zdobytych informacji, metody wyszukiwania podatności. Wykorzystanie skanerów podatności. Praca z exploitami.

Ataki na hasła: Przeprowadzenie ataków na zdalne mechanizmy uwierzytelniające. Przeprowadzenie ataków offline na różnego rodzaju skróty haseł.

Testy penetracyjne aplikacji webowych: Metodyka prowadzenia testów penetracyjnych aplikacji webowych. Modelowanie zagrożeń. Przydatne narzędzia. Zdobywanie informacji o celu. Mapowanie aplikacji. Wykrywanie ukrytych zasobów.

Podstawowe błędy obsługi danych od użytkownika: Wykrywanie i wykorzystywanie podatności typu SQL Injection, Cross-Site Scripting, Code Injection.

Podatności związane z obsługą plików: Wykrywanie podatności typu Local/Remote File Inclusion. Wykrywanie podatności związanych z uploadem plików oraz obsługą różnych rozszerzeń (XML, SVG).

Pozostałe podatności w aplikacjach webowych: Ataki na mechanizmy zarządzania sesją, wykorzystanie podatności Cross-Site Request Forgery, analiza podatności w logice biznesowej aplikacji, wykorzystanie podatności typu Server Side Request Forgery. Wykorzystanie podatności związanych z deserializacją danych.

Techniki postexploitacyjne w systemach Linux i Windows: Alternatywne sposoby transferu plików. Podnoszenie uprawnień w systemie. Lateral movements. Przekierowania portów i tunelowanie ruchu. Metasploit framework: Praca z narzędziem msfconsole. Wykorzystanie narzędzia meterpreter. Omówienie dodatkowych narzędzi z pakietu msf.

Ataki typu client-side: Opracowanie ataku typu client-side. Wprowadzenie do ataków socjotechnicznych i przeprowadzenie ataku phishingowego.

Techniki omijania: Omówienie zasad funkcjonowania mechanizmów bezpieczeństwa typu anty wirus, web application firewall czy filtr antyspamowy. Przeprowadzenie działań mających na celu ich ominięcie.

Sieci Wi-Fi: Omówienie sposobów ataków na sieci wi-fi oraz ich klientów. Przeprowadzenie ataku na standard WPA2.

Nazwa zajęć: **Kryptografia post-kwantowa**

**Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka**

**w zakresie wiedzy:**

1. zna i rozumie metody współczesnych algorytmów odpornych na ataki z wykorzystaniem komputera kwantowego.

**w zakresie umiejętności:**

1. potrafi implementować algorytmy demonstrujące wybrane zagadnienia kryptografii post-kwantowej.

**w zakresie kompetencji społecznych:**

1. jest świadomy zagrożenia bezpieczeństwa komputerowego wynikającego z rozwoju nowych technologii.

**Treści programowe dla zajęć:**

Podstawowe własności krat

Problemy obliczeniowe na kratach

Kryptosystem z kluczem publicznym NTRU

Schemat podpisu cyfrowego NTRU

Podstawy teorii kodów korygujących

Kody Turbo i kody Golaya

Protokół McEliece

KEM wraz z przykładami