



Ayça Begüm Taşcıoğlu
PhD student at University of Bologna,
in Computer Science and Engineering

- aycabegum.tascioglu2@unibo.com
- <https://www.linkedin.com/in/aycabegumtascioglu/>
- <https://www.github.com/aeyc>

Competenze

Java
C++
Python
C
PHP
Assembly Language
SQL
Javascript
CSS
HTML
MATLAB

Lingue

Turca (Madrelingua)
Inglese (Full professional proficiency)
Tedesco (Limited working proficiency)
Russa (Limited working proficiency)
Italiano (Limited working proficiency)

Certificati

CCNA - Issued by Cisco, Novembre 2021
Process Mining Data Science in Action -
Issued by Eindhoven University of
Technology, Aprile 2022
Introduction to Genomic Technologies - Issued
by Johns Hopkins University, Settembre
2022

Esperienza Professionale

Luglio 2022 - Novembre 2022 **DXC Luxoft Germany**
Automotive Embedded Software Engineer
Ha lavorato a un progetto di guida autonoma con un team situato a Monaco di
Baviera/Germania.

Marzo 2022 - Luglio 2022 **Mobisec Italia Srl**
Cyber Security Research Intern
Ha lavorato a un motore di analisi del comportamento autonomo per profilare il
comportamento dei dispositivi e il traffico di rete e rilevare le anomalie.

Settembre 2021 - Luglio 2022 **Cisco**
Cisco Incubator 9.0 Program Trainee
Ha lavorato come ingegnere di rete tirocinante nel settore della sicurezza informatica.

Febbraio 2021 - Luglio 2021 **Unicredit**
Software Engineering Intern
Ha lavorato all'applicazione dell'intelligenza artificiale agli eventi di sicurezza fisica per
sviluppare un modello predittivo.

Giugno 2020 - Luglio 2020 **Innova**
Data Science Intern
Ha lavorato sulla previsione delle serie temporali con Vanilla, Stacked, CNN e Conv LSTM.

Marzo 2020 - Maggio 2020 **Konneka Bilgi İletişim Teknolojileri**
Software Engineer
Ha collaborato con i team di Cybersecurity e Machine Learning.

Gennaio 2020 - Agosto 2020 **Ratio Technologies**
Data Scientist
Ha lavorato al progetto di stoccaggio e ottimizzazione dell'energia. Preparare set di dati
sul consumo energetico e analizzarli con diversi modelli di apprendimento automatico:
regressione lineare, SVM e reti neurali.

Settembre 2019 - Luglio 2020 **Bilkent University**
Undergraduate Laboratory Assistant
Ha assistito gli studenti sui compiti di laboratorio relativamente al linguaggio di
programmazione Python.

Agosto 2019 - Settembre 2019 **JotForm**
Data Science Intern
Ha lavorato con il team di Data Science per sviluppare un Phishing Detector in grado di
rilevare se un modulo è spam o sicuro. Ha pulito i dati in base ai punteggi delle parole e li
ha analizzati per implementare l'apprendimento automatico con SVM. Dopo il primo
progetto, ha lavorato con il team SecOps su problemi di vulnerabilità, in particolare sul
Cross-Site scripting (XSS).

Agosto 2018 - Settembre 2018 **Turkey IsBank**
IT Intern
Ha lavorato sulla virtualizzazione e sulle macchine virtuali (creazione, aggiunta al dominio
e gestione delle preferenze di rete) con VMware vSphere Client.

Formazione

Alma Mater Studiorum - Università di Bologna PhD, Dipartimento di Informatica - Scienza e Ingegneria	2022-ad oggi
Università degli Studi di Padova Master of Science, ICT for Internet and Multimedia, Telecommunications Engineering	2020-2022
Bilkent University Bachelor's degree, Informatica	2016 - 2020
Ankara Atatürk Anadolu Lisesi High School Diploma, Matematica e Scienze	2012 - 2016

Progetti

Mobile Behavioral Analysis Engine

Un motore di analisi comportamentale per profilare il comportamento dei dispositivi mobili e rilevare le anomalie. Per l'analisi dinamica sono stati sviluppati e testati modelli con LSTM, Decision Tree, SARIMAX, ARIMA, K-Means Clustering, Isolation Forest e Interquartile Range Algorithm. Per lo sviluppo degli algoritmi sono state utilizzate le librerie RapidMiner, Disco e Python. Per il motore sono stati scelti gli algoritmi K-Means Clustering, Isolation Forest e Interquartile Range, sviluppati con Python; i dati sono stati estratti da MongoDB.

<https://github.com/aevc/BehaviorAnalysisEngine>

Csion

Csion mira a ottimizzare le decisioni delle persone su qualsiasi argomento in base ai loro comportamenti caratteristici in un breve lasso di tempo. Gli utenti devono sottoporsi a un test della personalità fornito al momento della registrazione, quindi porre una domanda su qualsiasi argomento. Grazie a un'API IBM Watson integrata, il programma sarà in grado di comprendere la domanda e, in base ai dati sulla personalità dell'utente, fornirà una risposta alla domanda data. Il progetto è sviluppato con Python e JavaScript, come applicazione mobile multiplatforma per Android e iOS. Per i database viene utilizzato MongoDB.

<https://github.com/aevc/csion>

BloomFilter

Filtro Bloom per eseguire un test di appartenenza approssimativo per le sequenze di DNA, per gli algoritmi di corrispondenza delle stringhe in bioinformatica. Il progetto è sviluppato in Python, con sequenze di lettura fornite in formato FASTA.

<https://github.com/aevc/bloomFilter>

Company Review System

La piattaforma per le recensioni di colloqui aziendali e di lavoro è una proposta che consente agli utenti di fornire recensioni su aziende, posti di lavoro e processi di colloquio. Il sistema di database è stato implementato in Java e i dati sono stati gestiti con comandi SQL. Per la progettazione del sito web sono stati utilizzati PHP, Javascript e CSS.

<https://github.com/aevc/Company-Interview-and-Employment-Review-Platform-Database-System>

The Wall

Basato su un gioco da tavolo - Walls & Warriors - è stato creato un nuovo gioco da tavolo in Java con nuove modalità come la modalità sfida e la modalità sviluppatore e nuovi livelli; implementato in Java, con IntelliJ IDEA. Viene utilizzata la programmazione orientata agli oggetti.

<https://github.com/aevc/CS319-03-Group3E>

TDD-GoogleMaps

Una pagina reattiva sviluppata con Test Driven Development (TDD). Prende le coordinate dall'utente o dal GPS del dispositivo, quindi mostra la posizione su Google Maps. In alternativa, prende le coordinate del GPS e calcola la distanza tra il centro città più vicino e la posizione attuale, oppure prende le coordinate dall'utente o dal GPS del dispositivo, quindi mostra la distanza tra la posizione attuale e il centro della Terra. Implementato con le API di Google Maps in Java; per la progettazione di siti web multiplatforma, vengono utilizzati PHP, JavaScript, CSS, HTML. Creato sulla base del Test Driven Development, con JUnit e Selenium. <https://github.com/aevc/TDD-GoogleMaps>

Phishing Detector

È stato progettato un programma per rilevare se un modulo è spam o meno con l'apprendimento supervisionato. Con i moduli forniti, etichettati come "SPAM" o "NORMALE". Per prima cosa, ho dovuto ripulire i miei dati attraverso la tokenizzazione, la rimozione delle stopwords, la conservazione dei dati in diversi dizionari e l'inizializzazione degli z-score per la parte di analisi. In secondo luogo, alla luce dei punteggi di ogni parola e delle loro etichette, ho analizzato l'importanza delle parole per la classificazione. Infine, ho utilizzato modelli di regressione logistica, SVM e regressione lineare. Nel mio caso, ho preferito scegliere SVM; ho separato i miei dati in parti di addestramento e di test prima di mescolarli. Ho scritto una funzione di previsione per ottenere un risultato sul funzionamento o meno del mio progetto e ho stampato una matrice di confusione, quindi ho terminato il mio progetto.

<https://github.com/aevc/PhishingDetector>

Ceramic Tiles for Hybrid Armor

Un programma per scoprire un modo efficace di posizionare tessere esagonali e quadrate su armature ibride, con apprendimento per rinforzo da zero. L'algoritmo è stato utilizzato come agente stesso. L'area rimanente nell'armatura ibrida influisce sulla ricompensa. Sono state utilizzate due matrici Q: la matrice Q completa, una matrice con i numeri dei modelli; in questo caso, le azioni sono state intraprese in base ai modelli migliori con la ricompensa più alta; e la matrice Q locale, una matrice che include la selezione casuale della forma, le coordinate nella tela e il numero di passi; le azioni sono state intraprese in base alla selezione della forma e delle coordinate del modello migliore. L'algoritmo Greedy è stato utilizzato per intraprendere le azioni e il progetto è stato implementato in Python.

<https://github.com/aevc/CeramicTilesForHybridArmor>

Panther

Un analizzatore lessicale e un parser. Sviluppato in linguaggio C. <https://github.com/aevc/Panther>