

COLLEGIUM WITELONA
Uczelnia Państwowa

Wydział Nauk Technicznych i Ekonomicznych
Kierunek Informatika
Specjalność Programowanie Aplikacji Mobilnych i Internetowych

Sebastian Górska

**Nethelt – wieloplatformowy system monitorowania urządzeń sieciowych z modułem wykrywania anomalii
opartym na AI**

**Praca dyplomowa inżynierska
napisana pod kierunkiem
dr inż. Zbigniew Fryźlewicz**

Legnica 2026 rok

Spis treści

1	Wstęp	3
1.1	Wprowadzenie do problematyki	3
1.2	Cel pracy	4
1.3	Zakres pracy	4
1.4	Założenia projektowe	4
1.4.1	Ograniczenia i potencjalne trudności	5

1. Wstęp

1.1 Wprowadzenie do problematyki

Aktualnie nie sposób wyobrazić sobie firmy, przedsiębiorstwa, a nawet gospodarstwa domowego, w którym nie korzysta się z internetu. Dostęp do zewnętrznych serwisów i usług stał się naszą codziennością. Korzystamy z internetu oraz urządzeń, łączących się do niego każdego dnia. W domach internet służy nam często do rozrywki i ułatwiania codziennych czynności, do wyszukiwania informacji i rozwiązywania problemów. Dzięki niemu jesteśmy w stałym kontakcie z innymi, mamy dostęp do płatności elektronicznych, a zakupy możemy zrobić nie wychodząc z domu. W pracy korzystamy z niego w celu wyszukiwania informacji i przesyłania ich dalej, wysyłania maili, pobierania danych z Państwowych spółek, czy chociażby robienia wideokonferencji z klientami z całego świata.

Poza dostępem do zewnętrznych usług coraz większą popularność zyskują urządzenia Smart oraz IoT (Internet of Things). Wielu z nas nawet nie ma pojęcia na temat tego jak wiele takich urządzeń znajduje się w naszym otoczeniu. Według IoT Analytics pod koniec 2025 r. liczba podłączonych urządzeń IoT na świecie miała osiągnąć 21,1 miliarda To wzrost o ok. 14% w stosunku do roku poprzedniego **TODO: zweryfikować dane przed oddaniem pracy**. Według prognozy liczba ta będzie tylko rosła i w 2030 roku może sięgnąć 39 miliardów[1]. Tendencję tę można zaobserwować na rysunku 1.1 przedstawiający liczbę aktywnych urządzeń typu IoT wraz z prognozą na przyszłe lata.

Pojawienie się tak dużej ilości urządzeń, z których specyfikacji wynika, że wręcz wymagają stałego połączenia z siecią, tworzy nowe problemy i pytania. Kluczowym pytaniem, które powinniśmy sobie zadać, jest to, czy przy utracie łączności z siecią takiego urządzenia pojawią się poważne komplikacje. Chociaż w przypadku np. lodówki bądź ekspresu do kawy działającym w sieci, brak połączenia nie wyrządzi sporych szkód, tak jednak w przypadku maszyn na hali produkcyjnej lub sygnalizacji świetlnej w centrum miasta, szkody mogą być spore i mogą zagrażać nie tylko sytuacji majątkowej, ale też bezpieczeństwu ludzi. W związku z powyższym należy monitorować stan takich urządzeń, a w przypadku ich awarii jak najszybciej dążyć do naprawy usterki.

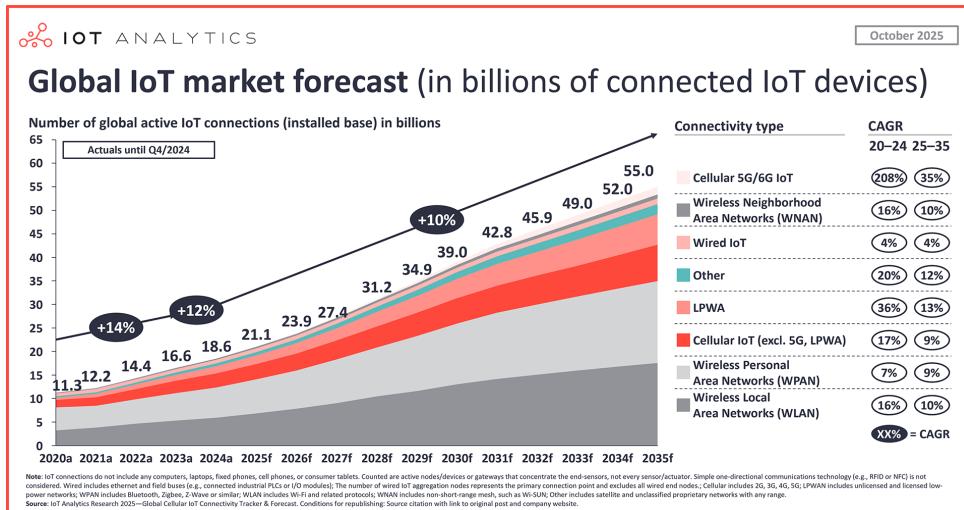


Figure 1.1: Liczba urządzeń typu IoT w latach 2020–2035
 Źródło: <https://iot-analytics.com/number-connected-iot-devices/>

1.2 Cel pracy

1.3 Zakres pracy

1.4 Założenia projektowe

Projekt ma charakter wielomodułowy i obejmuje trzy główne komponenty:

- aplikację webową,
- aplikację desktopową,
- moduł detekcji anomalii oparty na algorytmie AI typu *isolation forest*.

System zostanie zaprojektowany w taki sposób, aby umożliwić użytkownikowi rejestrację, logowanie, konfigurację połączenia ze swoją siecią LAN, dodawanie lokalnych urządzeń sieciowych, monitorowanie ich stanu oraz zmianę ustawień powiadomień. Aplikacja desktopowa będzie pełniła rolę klienta komunikującego się z serwerem, co jest niezbędne do prawidłowego funkcjonowania systemu w środowisku wielomodułowym. Dodatkowym wyzwaniem projektu będzie integracja wytrenowanego modelu AI, który będzie wspierał wczesne wykrywanie anomalii w sieci.

System przewiduje co najmniej dwa typy użytkowników: administratora oraz użytkownika zwykłego. Zalogowanie do systemu będzie obowiązkowe, a proces logowania zostanie udostępniony zarówno poprzez protokół OAuth2 (Google), jak i w formie lokalnej autoryzacji. Administrator będzie posiadał rozszerzone uprawnienia, w tym możliwość zarządzania użytkownikami oraz przeglądania podstawowych statystyk dotyczących korzystania z systemu.

Aplikacja webowa będzie realizowana w oparciu o nowoczesną wersję języka Java (wersja 25) oraz framework Spring Boot (w wersji 4.x.x), który ułatwia implementację i utrzymanie systemu. Interfejs użytkownika zostanie zbudowany przy użyciu TypeScript oraz frameworka Angular (wersja 21). Moduł AI zostanie opracowany w języku Python z wykorzystaniem odpowiednich

bibliotek do uczenia maszynowego, co przyspieszy proces modelowania i analizy danych. Aplikacja desktopowa zostanie stworzona w tej samej wersji Javy, z wykorzystaniem biblioteki Swing, co pozwoli na jej uruchamianie na różnych systemach operacyjnych.

Dane pomiędzy modułami systemu będą współdzielone poprzez centralną bazę danych, której silnik wykorzysta PostgreSQL w wersji 18.2.

Projekt przewiduje wdrożenie aplikacji w środowisku chmurowym w celu zapewnienia dostępności dla użytkowników końcowych z dowolnej lokalizacji. Wstępnie planowane jest wykorzystanie platformy Google Cloud Console do tego celu.

1.4.1 Ograniczenia i potencjalne trudności

W ramach projektu należy uwzględnić następujące ograniczenia i możliwe trudności:

- Moduł AI będzie wykrywał jedynie wybrane typowe anomalie w sieci i może nie uwzględniać wszystkich możliwych scenariuszy awarii.
- Desktopowy klient nie będzie posiadał zaawansowanego GUI i jego funkcjonalność ograniczy się do komunikacji z serwerem oraz wyświetlania podstawowych powiadomień.
- Wersja mobilna aplikacji nie jest przewidziana w początkowej fazie wdrożenia.
- Wydajność i dostępność systemu mogą być ograniczone przez wybraną infrastrukturę chmurową oraz przez ograniczenia przepustowości sieci użytkowników.

Spis rysunków

1.1 Liczba urządzeń typu IoT w latach 2020–2035 Źródło: <https://iot-analytics.com/number-connected-iot-devices/> 4

Spis tabel

Spis listingów

Bibliography

- [1] S. Sinha, *State of IoT 2025: Number of connected IoT devices growing 14% to 21.1 billion globally*, 2025, <https://iot-analytics.com/number-connected-iot-devices/> [dostęp: 07.12.2025].