الجمهورية العربية السورية
وزارة التعليم العالي
الجامعة السورية الافتراضية
ماجستير علوم الحاسوب

# تصميم وتطوير نظام ذكي لتحليل وإدارة إنذارات أعطال الشبكة الخلوية في بيئة عمل MTN Syria

مشروع أعد لنيل درجة ماجستير التأهيل والتخصص في علوم الحاسوب

**إعداد الطالب**

**م. عباده عز الدين**

**إشراف**

**د.م. سيرا أستور**

العام الدراسي
2025-2024

# كلمة شكر وتقدير

جزيل الشكر والعرفان، ووافر التقدير والامتنان، إلى من كان لهم الفضل الكبير في إنجاز هذا المشروع.

الشكر الخاص للدكتورة سيرا آستور، التي كانت معنا خطوة بخطوة لإنجاح هذا المشروع ووثقت بنا وبقدرتنا على المضي قدماً لتحقيق الهدف المنشود.

إلى الإدارة الكريمة في شركة MTN Syria، وأخص بالشكر منهم الدكتور محمد مازن محايري، المهندس بشير عرقسوسي، المهندس أسامة ليون، الدكتور محمد عماد العلبي، المهندس سامرعبد الرحيم ، المهندس أسامة شموط ،المهندس سداد نفوري، المهندس حسام الدين الجمال، وإلى كل من قدم لنا الدعم والمشورة. لقد كانت ثقتكم بنا ورؤيتكم الثاقبة هي الحافز الأكبر الذي دفعنا للمضي قدمًا وتجاوز كل الصعوبات. شكرًا لكم على توفير البيئة الداعمة التي مكنتنا من تحويل أفكارنا إلى واقع ملموس.

إلى دكاترة وأساتذة الجامعة الإفتراضية السورية، وأخص بالشكر منهم الدكتور باسل الخطيب، الدكتور طارق الساطي، الدكتور عصام سلمان، الدكتور عبده الخوري، الدكتور محمد مازن المصطفى .لقد كانت توجيهاتكم القيمة وعلمكم الغزير بمثابة النور الذي أضاء لنا الطريق. بفضل إرشاداتكم، تمكنا من إثراء بحثنا وتقديم عمل متكامل يجمع بين المعرفة النظرية والتطبيق العملي.

إلى أسرة عمل شركة MTN Syria ، التي كانت بمثابة العائلة الداعمة لنا. لقد كانت روح التعاون التي لمسناها في كل فرد منكم، ومساعدتكم المستمرة، هي الوقود الذي أشعل حماسنا. كل لحظة قضيناها معكم كانت إضافة حقيقية لنا على الصعيدين المهني والشخصي.ولدكاترة ومهندسي الجامعة الافتراضية السورية ونخص منهم بالشكر:

لا يسعنا إلا أن نتوجه بالعرفان لكل من ساهم في إثراء تجربتنا العلمية والعملية.

م. عبادة عزالدين

# الإهداءات

إلى من أثقلته همومنا ولم يشتكي يوماً، إلى من كان كفه سنداً لنا في السراء والضراء، وقد كان نعم السند، إلى قدوتي ومعلمي الأول، إلى من حملنا اسمه فخراً، إلى من يرجع إليه ثمرة وفضل نجاحنا.

والدي العزيز

إلى التي لا أعتقد أن الحروف قادرة على إنصافها، ويبدو أن كلماتي عاجزة على وصف حنانها وفضلها، إلى من علمتني وعانت الصعاب لأصل لما أنا فيه، إلى من كان دعاؤها سر نجاحي.

والدتي العزيزة

إلى النجمتين اللتان سطعتا في سماء حياتي فازدانها حباً وحناناً ولطفاً وفرحاً.

أخوتي "عامر، نيرمين"

لم أعتقد يومها أن هذه المجموعة الصغيرة سوف تكون لي قلباً يوماً ما، فكم تشاركنا من هموم... وكم تجاوزنا من صعاب... كم بكينا على أكتاف بعضنا... وكم شاركنا لحظات الفرح والنجاح... كنا إخوة وأحباء بما تحمله تلك الصفات من معاني.. لا أعتقد أني كنت أستطيع تجاوز الأمر دونكم.

فريق الكيمياء "سارة، طارق، أبو زيد،كردي"

إلى زملائي الأعزاء، رفاق الدرب وشركاء النجاح. لقد كنتم أكثر من مجرد زملاء، كنتم عائلة. بفضل تعاونكم وتفانيكم، تحولت الصعوبات إلى فرص، والمجهودات إلى إنجازات. كل لحظة قضيناها معًا، سواء في العمل الجاد أو في الأوقات الممتعة، ستبقى محفورة في ذاكرتي.

فريق تخطيط وتطوير الشبكة

إلى رفقاء دربي الطويل، مشيناه سوياً يداً بيد وقلباً بقلب..دمعاً بدمع... إلى من جعلتم تلك السنين ذكريات لا تنسى.

أصدقائي الأعزاء

المرحوم عدي،حسام، اياس،عبدالرحمن،

سامي،ربى، مرام ، شام،عاليا،حبيب،كرم

لو استطعنا تجسيد العطاء لكان العطاء أنتم .. إلى القلوب التي حمل دماء العلم مشبعاً بأوكسجين العطاء... فقلوبنا تنبض بالحب دوماً.

فريق الكريات الحمراءRBCs

إلى كل من لم تسعفني ذاكرتي على ذكر اسمه ويعلم مكانه في قلبي، كل الشكر لكم.

م. عبادة عزالدين

# فهرس المحتويات

ملخص.....14

**الفصل الأول-مقدمة إلى المشروع**

1.1-المقدمة.....16
2.1-شركة MTN Syris.....16
1.2.1-نظرة عامة والتاريخ.....16
2.2.1-الخدمات والمكانة في السوق.....17
3.1-مشكلة المشروع.....17
4.1-الحل المقترح.....19
5.1-أهداف المشروع.....21
1.5.1-أهداف تحسين الأداء(Performance Objectives).....21
2.5.1-الأهداف الوظيفية Functional Objectives.....22
3.5.1-الأهداف الاستراتيجيةStrategic Objectives.....22
6.1-فصول الأطروحة.....23

**الفصل الثاني-الشبكات الخلوية Cellular Network**

1.2-الشبكات الخلوية Cellular network.....26
1.1.2- تعريف الشبكة الخلوية.....26
21.2.-أجيال الشبكات المتنقلة.....27
3.1.2-معمارية الشبكة الخلوية.....28
2.2-المحطات الخلوية Base Station.....30
1.2.2-تعريف المحطات الخلوية وأهميتها.....30
2.2.2-مكونات محطة التغطية الخلوية Base Station Components.....31
3.2- الأعطال والتحديات الشائعة في شبكات الاتصالات الخلوية.....33
1.3.2-أسباب انقطاع الشبكة وتدهورها.....33
2.3.2-التحديات في تحديد الأعطال وحلها.....34
3.3.2-الكشف التلقائي عن الأعطال وتشخيصها في الشبكات الخلوية.....35
4.2-آلية الكشف عن الأعطال يدوياً وأهمية الأتمتة.....35

**الفصل الثالث-خوارزميات الذكاء الاصطناعي**


Artificial Intelligence Algorithm.....37
1.3-تعريف الذكاء الاصطناعي وخوارزمياته.....38
1.1.3-خورزميات الذكاء الاصطناعي.....38
2.1.3-أنواع خوارزميات الذكاء الاصطناعي .....38
3.1.3-تطبيقات خوارزميات الذكاء الاصطناعي في الحياة اليومية.....39
4.1.3-مزايا خوارزميات الذكاء الاصطناعي.....39
4.1.3-التحديات التي تواجه خوارزميات الذكاء الاصطناعي.....39
2.3-خوارزمية الغابة العشوائية Randon Forest.....40
1.2.3-المكونات الأساسية لـ Random Forest .....40
2.2.3-ميزات العشوائية(Randomness) .....41
3.2.3-آلية عمل Random Forest .....42
4.2.3- مميزات Random Forest .....42
5.2.3-عيوب Random Forest.....42
6.2.3- تحسين أداء Random Forest (Hyperparameter Tuning).....43
3.3-خوارزميةLogistic Regression .....43
1.3.3- الأساس الرياضي للانحدار اللوجستي .....43
2.3.3-آلية عملLogistic Regression .....45
3.3.3-مميزات وعيوبLogistic Regression .....45
4.3.3-معايير ضبط النموذج(Hyperparameter Tuning) .....46
4.3-خوارزميةXGBoost .....46
1.4.3-استخدام خوارزميةXGBoost .....46
2.4.3-الأساس الرياضي لـXGBoost .....47
3.4.3- آلية عمل XGBoost .....48
4.4.3-مميزات وعيوبXGBoost .....48
5.4.3- معايير ضبط النموذج(Hyperparameter Tuning) .....49
5.3-مقارنة بين RandomForest و LogisticRegression وXGBoost .....49
6.3-خوارزمية(Term Frequency-Inverse Document Frequency) TF-IDF .....51
1.6.3-استخدام TF-IDF .....51
2.6.3-مكوناتTF-IDF .....51
3.6.3-آلية عملTF-IDF .....52

4.6.3-مميزات وعيوبTF-IDF ..... 52
7.3-خوارزمية MultiOutputClassifier ..... 52
1.7.3-مشاكل التصنيف متعدد المخرجات..... 52
2.7.3-استخدام خوارزمية MultiOutputClassifier..... 53
3.7.3-مميزات وعيوب MultiOutputClassifier..... 53

**الفصل الرابع-تحليل المتطلبات والبنية المعمارية للنظام**

1.4-تحليل المشكلة وسير العمل الحالي Problem Analysis and Current Workflow 55.
2.4-أهم الأعطال في شركة MTN Syria..... 56
1.2.4-أعطال في الشبكة الخلوية..... 56
2.2.4-أعطال في شبكة النقل..... 56
3.2.4-أعطال في الطاقة ..... 56
4.2.4-أعطال في المحطة بشكل عام ..... 56
5.2.4-أعطال عابرة ..... 57
3.4- المتطلبات غير الوظيفية Non-Functional Requirements..... 59
1.3.4- الأداء (Performance) ..... 59
2.3.4- الأمان (Security)..... 60
3.3.4- الموثوقية والإتاحة (Reliability and Availability)..... 60
4.3.4- قابلية التوسع (Scalability)..... 60
5.3.4- سهولة الاستخدام (Usability) ..... 61
4.4-البنية المعمارية للنظام(System Architecture) ..... 61
1.4.4- الطبقة الأولى: طبقة المنطق والتطبيقات(Logic/Application Tier) ..... 61
2.4.4- الطبقة الثانية: طبقة البيانات(Data Tier) ..... 61
3.4.4- الطبقة الثالثة: طبقة العرض(Presentation Tier) ..... 62
4.4.4-مزايا هذه البنية المعمارية..... 62

**الفصل الخامس-التقنيات البرمجية المستخدمة**

1.5- لغة البرمجة ***python***..... 65
2.5 -إطار العمل فلاسك Flask framework..... 65
3.5 - لغات تطوير الويب..... 66
1.3.5- لغة ترميز النص التشعبي HTML..... 66
2.3.5-صفحات الأنماط المتتالية CSS ..... 66
3.3.5- JavaScript (JS) ..... 67

4.5- ***Visual Studio Code*** ..... 67
5.5 - قواعد البيانات و MySQL ..... 68
1.5.5 -قواعد البيانات DataBase..... 68
2.5.5 -قواعد البيانات MySQL..... 68
6.5-برنامج XAMPP ..... 69
1.6.3- وظيفة كل مكون ..... 69
2.6.5-استخدام XAMPP ..... 70
7.5-Flutter ..... 70
8.5- المكتبات المستخدمة..... 71
18..5- مكتبة Pandas ..... 71
2.8.5 -مكتبة scikit-learn (sklearn)..... 72
3.8.5 -xgboost (xgb) ..... 72
4.8.5-مكتبات الأدوات المساعدة وحفظ النماذج joblib..... 73
5.8.5 -مكتبة Numpy..... 73
6.5.5- مكتبةOS ..... 73
7.5.5-مكتبات الأمان والتشفير werkzeug.security..... 73
8.5.5- مكتبة sqlalchemy..... 74
9.5.5-Leaflet.js ..... 74

**الفصل السادس-تدريب نماذج الذكاء الاصطناعي**

1.6-تحليل البيانات ..... 76
1.1.6-استيراد البيانات والتحليل الأولي..... 76
2.1.6-تنقية البيانات وفهمها ..... 78
3.1.6-إعداد البيانات لتعلم الآلة..... 79
2.6-بناء نماذج التعلم الآلي ..... 80
3.6-تدريب نموذج لتوقع العطل التالي ..... 82
4.6-اختيار النموذج المناسب..... 82
1.4.6-اختبار النماذج..... 82
5.6-توليد إنذارات متناسبة مع البيانات التاريخيةProportional Alarm Generator ..... 84
1.5.6- الفكرة العامة لتوليد الإنذارات ..... 84
2.5.6-المكونات الرئيسية للأداة ..... 85
3.5.6- مزايا أداة توليد الإنذارات ..... 85

6.6-ربط الذكاء الاصطناعي مع توليد الأعطال.....86

**الفصل السابع-تصميم الموقع الالكتروني وتطبيق الأندرويد**

1.7-تصميم قاعدة بيانات نظام إدارة أعطال الشبكةNOC .....89
1.1.7-الهيكل التفصيلي للجداولTable Schema .....89
2.1.7-آلية عمل النظام المعتمدة على قاعدة البيانات.....91
2.7-تصميم موقع الويب.....92
1.2.7-الخادم Backend.....92
2.2.7-تصميم الواجهات frontend.....96
3.2.5-تشغيل موقع الويب .....114
3.5-تصميم تطبيق الأندرويد .....115
1.3.5-الخادم Backend.....115
2.3.5-الواجهة الأمامية لتطبيق الموبايل(Flutter) .....116
3.3.5-تشغيل تطبيق الأندرويد.....119
النتائج والخاتمة.....120
الآفاق المستقبلية.....122
المراجع.....125
الملحق.....127
Abstract.....139

# قائمة الأشكال


الشكل (1-1) شعار شركة MTN Syria.....17
الشكل (1-2) توزع المحطات الخلوية.....26
الشكل (2-2) معمارية الشبكة الخلوية.....29
الشكل(2-3) مكونات محطة التغطية الخلوية.....33
الشكل(3-1) خوارزمية Random forest.....40
الشكل(3-2) خوارزمية Logistic Regression.....44
الشكل(3-3) خوارزمية XGBoost.....47
الشكل(4-1) آلية التعامل مع الأعطال.....59
الشكل(4-2) البنية المعمارية للنظام.....63
الشكل(5-1) شعار شركة بايثون.....65
الشكل(5-2) شعار إطار العمل Flask.....66
الشكل(5-3) شعار تطبيق Visual Studio Code. .....68
الشكل(5-4) تطبيق ***XAMPP*** .....69
الشكل(5-5) شعار ***Flutter*** .....70
الشكل(6-1) لمحة عن البيانات.....77
الشكل(6-2) معالجة القيم المفقودة.....78
الشكل(6-3) عدد الأعطال حسب الشدة.....79
الشكل(6-4) عدد الأعطال حسب التكنولوجيا .....79
الشكل(6-5) عدد الأعطال حسب الفريق المختص.....79
الشكل(6-6) اختبار نماذج الذكاء الاصطناعي.....84
الشكل(6-7) الإنذارات المولدة .....86
الشكل(6-8) معالجة وإدراج الأعطال الجديدة.....86
الشكل(6-9) آلية عمل محرك الذكاء الاصطناعي AI Engine .....87
الشكل(7-1) مخطط ERD .....91
الشكل(7-2) واجهة تسجيل الدخول .....96
الشكل(7-3) القالب الأساسي.....98
الشكل(7-4) بطاقة الإحصاءات.....100
الشكل(7-5) قسم الرسائل الديناميكية.....100
الشكل(7-6) قسم المرشحات.....100
الشكل(7-7) جدول الإنذارات النشطة.....101
الشكل(7-8) تفاصيل إضافية عن الأعطال .....101
الشكل(7-9) جدول معلومات الموقع .....102
الشكل(7-10) خريطة المواقع.....102
الشكل(7-11) قسم المرشحات.....104
الشكل(7-12) قسم الرسوم البيانية.....105
الشكل(7-13) صفحة إدارة المستخدمين.....106

الشكل(7-14) إضافة مستخدم جديد .....109
الشكل(7-15) صفحة تعديل مستخدم.....109
الشكل(7-16) صفحة تذاكر الأعطال.....111
الشكل(7-17) تشغيل برنامج XAMPP .....114
الشكل(7-18) واجهة تسجيل الدخول والواجهة الرئيسية.....118
الشكل(7-19) واجهة التذاكر .....119

# قائمة الجداول

الجدول(1-1) أهم مكونات وحدة معالجة القاعدة.....31
الجدول (1-2) أهم أعطال الشبكة الخلوية.....58
الجدول(2-1) تحسين أداء Random forest.....43
الجدول(2-2) تحسين أداء logistic Regression.....46
الجدول(2-3) تحسين أداء XGBoost.....49
الجدول(2-4) مقارنة أداء الخوارزميات على أنواع البيانات المختلفة.....51
الجدول(2-5) مقارنة خوارزمية MultiOutputClassifier مع خوارزميات مشابهة.....53
الجدول(4-1) مقارنة نتائج الخوارزميات .....81

# جدول أهم المصطلحات والاختصارات الأجنبية

| الترجمة العربية | المصطلح الأجنبي | الترجمة العربية | المصطلح الأجنبي |
|---|---|---|---|
| دالة الخسارة | Loss Function | الشبكات الخلوية | Cellular network |
| دالة خسارة لوغاريتمية | Log Loss (Binary Cross-Entropy) | الأبراج الخلوية | sites |
| انحدار التدرج | Gradient Descent | خلايا | Cells |
| معدل التعلم | Learning Rate | التسليم السلس | handover |
| تطبيع البيانات | Feature Scaling | إنترنت الأشياء | Internet of Things (IoT) |
| نوع التنظيم | penalty | جهاز إرسال واستقبال | transceiver |
| القيم المتطرفة | Outliers | محطة البث القاعدية | Base Transceiver site |
| تعزيز التدرج المتطرف | XGBoost (Extreme Gradient Boosting) | شبكات النفاذ الراديوية | Radio Access Networks (RAN) |
| التعزيز التدرجي | Boosting | محطة الإرسال القاعدية | Base Transceiver Station (BTS) |
| الأشجار الضعيفة | weak learners | التطور طويل الأمد(الجيل الرابع) | LTE |
| دالة الهدف | Objective Function | نظام الاتصالات المتنقلة العالمي(الجيل الثالث) | UMTS |
| دالة التعقيد (التنظيم) | Regularization | النظام العالمي للاتصالات | GSM |

| | | | |
|---|---|---|---|
| | المتنقلة(الجيل الثاني) | | |
| Radio Frequency unit (RF) | وحدة الترددات اللاسلكية | LinearSVC | آلة المتجهات الداعمة الخطية |
| Universal Baseband Processing Board | لوحة معالجة النطاق الأساسي الشاملة | MultiOutputClassifier | مصنف متعدد المخرجات |
| Universal External Interface Unit / Power Entry Unit | وحدة الواجهة الخارجية الشاملة / وحدة إدخال الطاقة | TF-IDF (Term Frequency-Inverse Document Frequency) | تكرار المصطلح - التردد العكسي للمستند |
| Universal Main Processing and Transmission Board | لوحة المعالجة والنقل الرئيسية الشاملة | Term Frequency (TF) | تكرار المصطلح |
| Universal External Interface Unit | وحدة الواجهة الخارجية الشاملة | Inverse Document Frequency (IDF) | التردد العكسي للمستند |
| Universal Environment Interface Unit | وحدة واجهة البيئة الشاملة | Count Vectorizer | عداد المتجهات |
| Multi-band RRUs | وحدات تردد لاسلكي متعددة النطاقات | Object-Oriented Programming (OOP) | البرمجة غرضية التوجه |
| Active Antenna Unit (AAU) | وحدة الهوائي الفعال | Interpreted language | لغة مُفسَّرة |
| Rectifiers | المقومات | Graphical User Interfaces (GUI) | الواجهات الرسومية |
| Network congestion/overload | ازدحام الشبكة/الحمل الزائد | Scripting language | لغة برمجة نصية |
| Hardware faults | أعطال الأجهزة | Computer Algebra Systems (CAS) | أنظمة الجبر الحاسوبي |
| Short-Term Cell Outages (STCOs) | انقطاعات الخلايا قصيرة المدى | Flask Framework | إطار العمل فلاسك |
| Root cause analysis | تحديد السبب الجذري | Web development | تطوير الويب |
| Quality of Service (QoS) | جودة الخدمة | HTML (HyperText Markup Language) | لغة ترميز النص التشعبي |
| Remote diagnosis | التشخيص عن بعد | CSS (Cascading Style Sheets) | صفحات الأنماط المتتالية |
| Automatic Fault Detection and Diagnosis | الكشف التلقائي عن الأعطال وتشخيصها | JavaScript (JS) | جافاسكربت |
| Performance Support System (PSS) data | بيانات نظام دعم الأداء | Data Validation | التحقق من صحة البيانات |
| Artificial Intelligence (AI) | الذكاء الاصطناعي | API (Application Programming Interface) | واجهة برمجة التطبيقات |
| Machine Learning (ML) | التعلم الآلي | DataBase (DB) | قاعدة البيانات |
| Unsupervised learning | التعلم غير الخاضع للإشراف | Database Management System (DBMS) | نظام إدارة قواعد البيانات |
| Classification algorithms | خوارزميات التصنيف | Relational Database | قاعدة بيانات ترابطية |
| NOC (Network Operation Center) | مركز التشغيل (مركز عمليات الشبكة) | eCommerce companies | شركات التجارة الإلكترونية |
| MAE (Maintenance and Alarm Explorer) | برنامج وسيط من ) (شركة هواوي | Generative Artificial Intelligence | الذكاء الاصطناعي التوليدي |
| Fault Management | إدارة الأعطال | Machine learning | التعلم الآلي |
| Statistical Alarm | إنذار إحصائي | Natural Language Processing (NLP) | المعالجة اللغوية الطبيعية |
| Neural Networks | الشبكات العصبية الاصطناعية | RandomForestRegressor | منحدِر الغابة العشوائية |
| Supervised Learning | التعلم الخاضع للإشراف | Object-Relational Mapper (ORM) | ربط الكائنات بالعلاقات |

| | | | |
|---|---|---|---|
| Linear Regression | الانحدار الخطي | Initial analysis | التحليل الأولي |
| Decision Trees | أشجار القرار | RRU Name (Remote Radio Unit) | اسم وحدة الراديو البعيدة |
| Clustering | التجميع/التجزئة | BBU Name (Baseband Unit) | اسم وحدة الشريط الأساسي |
| Reinforcement Learning | التعلم المعزز | eNodeB ID (evolved Node B) | معرف العقدة الإلكترونية |
| Automation | الأتمتة | Log Serial Number | الرقم التسلسلي للسجل |
| Self-learning | التعلم الذاتي | Maintenance Status | حالة الصيانة |
| User experience | تجربة المستخدم | low-frequency faults | الأعطال منخفضة التكرار |
| Big data | البيانات الضخمة | Human Intervention | تدخل بشري |
| Data bias | التحيز في البيانات | DataBase | قاعدة البيانات |
| Transparency | الشفافية | Data preparation | إعداد البيانات |
| Responsible and Ethical | مسؤول وأخلاقي | TfidfVectorizer | محوّل المتجهات TF-IDF |
| Random Forest | الغابة العشوائية | Unbalanced distribution | التوزيع غير المتوازن |
| Classification | التصنيف | SMOTE (Synthetic Minority Over-sampling Technique) | تقنية SMOTE (للموازنة) |
| Regression | الانحدار | Gradient Boosting | التعزيز المتدرج |
| Ensemble Learning | تعلم المجموعات | Accuracy | الدقة |
| Variance | التباين | Vectorizer | محوّل المتجهات |
| CART (Classification and Regression Trees) | أشجار التصنيف والانحدار | Encoder | محوّل (أو مرمّز) |
| Gini Impurity | معيار جيني (نجاسة جيني) | Proportional Alarm Generator | مولّد الإنذارات المتناسبة |
| Mean Squared Error (MSE) | متوسط الخطأ التربيعي | NOC (Network Operations Center) | مركز عمليات الشبكة |
| Majority Voting | التصويت بالأغلبية | Downtime | زمن انقطاع الخدمة |
| Randomness | العشوائية | longitude/latitude | خط الطول/خط العرض |
| Bootstrap Aggregating (Bagging) | تجميع العينات مع الإحلال | department | القسم |
| Random Feature Selection | اختيار الميزات العشوائي | predicted_alarms | الإنذارات المتوقعة |
| Fully Grown Trees | أشجار غير مقيدة النمو | Predictive Maintenance | الصيانة التنبؤية (الاستباقية) |
| Prediction | التوقع/التنبؤ | ER (Entity-Relationship) diagram | مخطط الكينونات والعلاقات |
| Overfitting | التجهيز الزائد (فرط التجهيز) | Backend | الخادم (الجانب الخلفي) |
| Feature Importance | أهمية الميزات | Session | الجلسة |
| Missing Values | القيم المفقودة | frontend | الواجهات الأمامية |
| Model Interpretability | قابلية تفسير النموذج | Don't Repeat Yourself (DRY) | لا تكرر نفسك (مبدأ برمجي) |
| Logistic Regression | الانحدار اللوجستي | responsive | متجاوب |
| Binary Classification | التصنيف الثنائي | NOC Dashboard | لوحة تحكم مركز عمليات الشبكة |
| Multiclass Classification | التصنيف متعدد الفئات | Pagination | التصفح (تقسيم البيانات على صفحات) |

| | | | |
|---|---|---|---|
| الرسوم البيانية والتحليلات | Charts & Analytics | واحد مقابل البقية | One-vs-Rest (OvR) |

# الملخص

تعد شبكات الاتصالات الخلوية حجر الزاوية في عالمنا الحديث المترابط، حيث أحدثت ثورة في طريقة تواصل الأفراد والشركات على حد سواء.

فمنذ ظهور الجيل الأول (1G) في أواخر السبعينيات، تطورت هذه الشبكات بشكل هائل لتنتقل من مجرد توفير المكالمات الصوتية التناظرية إلى تقديم خدمات البيانات عالية السرعة التي تدعم تطبيقات معقدة مثل الفيديو عالي الدقة، والواقع المعزز، وإنترنت الأشياء(IoT) .

لقد أصبحت الهواتف المحمولة والأجهزة اللاسلكية الأخرى جزءاً لا يتجزأ من حياتنا اليومية، مما يجعل الاعتماد على هذه الشبكات أمراً بالغ الأهمية .ومع تزايد هذا الاعتماد، تتزايد أيضاً أهمية فهم الأعطال والتحديات التي تواجه هذه الأنظمة المعقدة.

فالأعطال في شبكات الاتصالات الخلوية يمكن أن تؤدي إلى تعطيل الخدمات، وخسائر اقتصادية كبيرة، وتأثيرات سلبية على جودة تجربة المستخدم.

تواجه شركات الاتصالات مثل شركة MTN Syria تحديات كبيرة في إدارة أعطال الشبكة بسبب تزايد تعقيد البنية التحتية وارتفاع حجم الإنذارات الصادرة عن مركز عمليات الشبكة NOC .

يهدف هذا المشروع إلى تطوير نظام ذكي متكامل يستخدم تقنيات الذكاء الاصطناعي لتحليل وإدارة إنذارات الأعطال الصادرة عن مركز عمليات الشبكة (NOC) في شركةMTN Syria ، وربط هذه الإنذارات مباشرة مع نظام التذاكر (Ticketing System) الحي لتسريع عملية الاستجابة والمعالجة .كما يسعى المشروع إلى تمثيل الأعطال على خريطة تفاعلية باستخدام نظم المعلومات الجغرافية (GIS) لتوفير صورة مرئية دقيقة وفورية عن مواقع الأعطال، بالإضافة إلى إرسال إشعارات آنية لموظفي الدعم الفني عبر تطبيق أندرويد يمكنهم من الاطلاع على التفاصيل واتخاذ الإجراءات المناسبة .تبرز أهمية المشروع في كونه خطوة نحو أتمتة إدارة الأعطال وتقليل الاعتماد على الإجراءات اليدوية، مما يرفع من جودة الخدمة المقدمة ويقلل من وقت التعافي(MTTR) .

# الفصل الأول
# مقدمة إلى المشروع

## 1.1-المقدمة

تعد شبكات الاتصالات الخلوية حجر الزاوية في عالمنا الحديث المترابط، حيث أحدثت ثورة في طريقة تواصل الأفراد والشركات على حد سواء .فمنذ ظهور الجيل الأول (1G) في أواخر السبعينيات، الذي اقتصر على توفير المكالمات الصوتية التناظرية، شهدت هذه التقنية تطوراً هائلاً ومتسارعاً .ومع كل جيل جديد، كانت الشبكات الخلوية تقدم قدرات فائقة: بدءاً من الجيل الثاني (2G)الذي أتاح إرسال الرسائل النصية(SMS) ، مروراً بالجيل الثالث (3G) الذي فتح الباب أمام تصفح الإنترنت ومكالمات الفيديو، وصولاً إلى الجيل الرابع (4G) الذي رسخ خدمات البيانات عالية السرعة، وانتهاءً بالجيل الخامس (5G) الذي يَعِد بإحداث ثورة في مجالات متقدمة مثل المدن الذكية وإنترنت الأشياء(IoT) .

لقد أصبحت الهواتف المحمولة والأجهزة اللاسلكية الأخرى جزءاً لا يتجزأ من حياتنا اليومية، مما يجعل الاعتماد على استقرار وموثوقية هذه الشبكات أمراً بالغ الأهمية .ومع هذا التطور وزيادة تعقيد البنية التحتية للشبكات، تتزايد أيضاً أهمية فهم الأعطال والتحديات التي تواجه هذه الأنظمة المعقدة . فالأعطال في شبكات الاتصالات الخلوية، مهما كانت صغيرة، يمكن أن تؤدي إلى تعطيل الخدمات الحيوية، وخسائر اقتصادية كبيرة للشركات المشغلة، وتأثيرات سلبية مباشرة على جودة تجربة المستخدم النهائي.

في هذا السياق، تواجه شركات الاتصالات مثل شركة **MTN Syria** تحديات كبيرة في إدارة أعطال الشبكة بفعالية .ويعود ذلك بشكل أساسي إلى ارتفاع حجم الإنذارات اليومية الصادرة عن مركز عمليات الشبكة (NOC) ، والتي تتطلب تحليلاً وتصنيفاً يدوياً من قبل المهندسين، مما يجعل العملية بطيئة وعرضة للأخطاء البشرية .إن هذا النهج التقليدي في التعامل مع الأعطال لم يعد كافياً لمواكبة حجم وتعقيد الشبكات الحديثة، مما يستدعي الحاجة الماسة إلى حلول مبتكرة وأكثر ذكاءً.

## 2.1-شركة MTN Syris

### 1.2.1-نظرة عامة والتاريخ

MTN Syria، التي كانت تُعرف سابقًا باسم "أريبا"، هي إحدى شركتي الاتصالات الخلوية الرئيسيتين في سوريا، إلى جانب شركة سيريتل. تأسست الشركة في عام 2001 ودخلت السوق السورية كجزء من موجة تحرير قطاع الاتصالات في البلاد. في عام 2007، استحوذت عليها مجموعة MTN العالمية، وهي شركة اتصالات ضخمة متعددة الجنسيات مقرها في جنوب إفريقيا، وتم تغيير علامتها التجارية إلىMTN Syria .

الشكل (1-1) شعار شركة **MTN Syria**

### 2.2.1-الخدمات والمكانة في السوق

تقدم MTN Syria مجموعة شاملة من خدمات الاتصالات التي تشمل:

- **خدمات الجوال** :تغطي الشركة تقنيات الجيل الثاني(2G) ، والجيل الثالث(3G) ، والجيل الرابع.(4G/LTE)
- **خدمات الإنترنت** :توفر الشركة خدمات الإنترنت عبر الهاتف المحمول وخدمات الإنترنت المنزليADSL .
- **الباقات والخطط** :تقدم الشركة باقات متنوعة للدفع المسبق والدفع الآجل للمكالمات والبيانات والرسائل.

تاريخياً، لعبت MTN Syria دوراً حيوياً في ربط المجتمع السوري وتوفير الوصول إلى الإنترنت، مما ساهم في تمكين الأفراد ودعم قطاع الأعمال. وقد بلغ عدد مشتركيها حوالي 6 ملايين مشترك في عام 2012.

## 3.1-مشكلة المشروع

يتمحور هذا المشروع حول البيئة التشغيلية لشركة **MTN Syria**، التي تعد إحدى الشركات الرائدة في قطاع الاتصالات في سوريا. كواحدة من المشغلين الرئيسيين، تدير الشركة بنية تحتية خلوية ضخمة ومعقدة تخدم ملايين المشتركين في جميع أنحاء البلاد، وتلعب دوراً حيوياً في نسيج الاتصالات اليومي للأفراد وقطاع الأعمال على حد سواء. تتطور هذه الشبكة باستمرار لدعم الأجيال الأحدث من التكنولوجيا ، مما يزيد من قدراتها وتعقيدها في آن واحد. إن موثوقية هذه الشبكة واستقرارها أمران بالغا الأهمية، حيث يؤثر أي انقطاع بشكل مباشر على حياة الناس وأعمالهم.

تنشأ المشكلة الأساسية من الأعطال الحتمية التي تحدث داخل شبكة بهذا الحجم. فبينما الأعطال متوقعة، يكمن التحدي الحقيقي في إدارتها بكفاءة وسرعة. إن عواقب توقف الخدمة وتدهورها وخيمة ومتعددة الأوجه:

- **الأثر الاقتصادي** :تؤدي انقطاعات الخدمة بشكل مباشر إلى **خسارة في إيرادات** الشركة، كما تسبب **خسائر اقتصادية كبيرة** للشركات والمؤسسات التي تعتمد على شبكة MTN في عملياتها اليومية.
- **تجربة المستخدم** :تترك الأعطال **تأثيراً سلبياً عميقاً على جودة تجربة المستخدم**، مما يؤدي إلى إحباط العملاء واحتمالية فقدهم لصالح المنافسين، وهو ما يُعرف بمعدل التغيير (Churn Rate).
- **التكاليف التشغيلية** :تتطلب عمليات التشخيص والإصلاح اليدوية **موارد بشرية ومادية كبيرة**، من ساعات عمل المهندسين إلى إرسال فرق الصيانة الميدانية، مما يرفع من تكاليف التشغيل بشكل ملحوظ.
- **سمعة العلامة التجارية** :في سوق الاتصالات التنافسي، تُعتبر الموثوقية عاملاً أساسياً. تؤدي الأعطال المتكررة أو طويلة الأمد إلى **الإضرار بسمعة الشركة** وتقويض ثقة العملاء بها.

تكمن جذور هذه المشكلات داخل **مركز عمليات الشبكة(NOC)** ، وهو المحور المركزي لمراقبة صحة الشبكة. تعاني عملية إدارة الأعطال الحالية في MTN Syria من كونها يدوية وتواجه عدة تحديات حرجة:

- **فيضان الإنذارات(Alarm Flood)** : يستقبل مركز عمليات الشبكة يومياً آلاف الإنذارات من مختلف مكونات الشبكة. الكثير من هذه الإنذارات تكون متكررة أو مجرد أعراض لمشكلة أكبر، مما يسبب "ضوضاء" معلوماتية تغمر المهندسين وتجعل من الصعب تمييز الإنذارات الحرجة التي تتطلب تدخلاً فورياً.
- **تعقيد البنية التحتية وصعوبة تحليل السبب الجذري :(Root Cause Analysis)** الشبكات الحديثة هي نظام بيئي معقد يتكون من أجهزة (أبراج، محطات قاعدية، خوادم) وبرمجيات من موردين متعددين. يمكن لعطل واحد في قطعة جهاز أن يطلق سلسلة من الإنذارات المتتالية عبر أنظمة مختلفة، مما يجعل مهمة تحديد **السبب الجذري الحقيقي** للمشكلة يدوياً أشبه بالبحث عن إبرة في كومة قش.
- **النهج التفاعلي :(Reactive Approach)** إن العملية اليدوية هي بطبيعتها **تفاعلية**؛ أي أنها تعتمد على انتظار وقوع المشكلة ثم البدء في تحليلها وحلها. هذا التأخير بين لحظة وقوع

العطل ولحظة فهمه بالكامل يؤدي إلى إطالة **متوسط زمن الإصلاح(MTTR)** ، وهو أحد أهم مؤشرات الأداء (KPIs) لأي مشغل اتصالات.

إن هذا الاعتماد على الإجراءات اليدوية التفاعلية لم يعد مستداماً؛ فهو غير فعال، ومكلف، وبطيء للغاية لمواكبة متطلبات إدارة شبكة حديثة ومعقدة. هذا الوضع يخلق حاجة ماسة وملحة لنظام ذكي، مؤتمت، وقادر على تحليل الإنذارات بشكل استباقي لتسريع عملية الإصلاح وضمان تقديم خدمة موثوقة ومستقرة لمشتركي.MTN Syria

## 4.1-الحل المقترح

لمواجهة التحديات التشغيلية المعقدة في إدارة شبكةMTN Syria ، يقدم هذا المشروع حلاً تكنولوجياً متكاملاً يتمثل في **تطوير نظام ذكي(Intelligent System)** .

هذا النظام مصمم لإعادة هندسة العمليات الحالية بشكل جذري، ناقلاً إياها من نموذج العمل اليدوي والتفاعلي (Reactive) إلى نهج استباقي، مؤتمت، وذكي (Proactive and Intelligent) يعتمد بشكل أساسي على البيانات لاتخاذ القرارات.

الفكرة الجوهرية للحل هي إنشاء منظومة مترابطة تعالج دورة حياة العطل بالكامل،بدءاً من لحظة اكتشافه وحتى إغلاقه بأقل قدر من التدخل البشري. يتكون هذا الحل من أربعة مكونات أساسية ومتكاملة:

### 1. محرك الذكاء الاصطناعي (AI Engine)

هذا المكون هو العقل المدبر للنظام. بدلاً من أن يقوم المهندسون بتحليل آلاف الإنذارات يدويًا، يقوم هذا المحرك باستلام سيل البيانات الخام وتحويلها إلى معلومات قابلة للتنفيذ. يتم ذلك عبر:

- **التحليل والتصنيف الفوري** :باستخدام خوارزميات تعلم الآلة المتقدمة مثل **XGBoost**، يقوم النظام بتصنيف كل إنذار بشكل فوري، محدداً :
  - **نوع العطل**
  - **مستوى خطورته**
  - **التكنولوجيا المتأثرة (2G/3G/4G)**
  - **الفريق الفني المسؤول** عن الإصلاح.
- **التنبؤ الاستباقي** :لا يقتصر دور المحرك على تحليل الأعطال الحالية فقط، بل يتجاوزه إلى التنبؤ بالمستقبل .فهو يتوقع **العطل التالي المحتمل حدوثه** في نفس الموقع بناءً على الأنماط التاريخية ، ويقدر**الوقت المتوقع للإصلاح** ، مما يمنح فرق الصيانة رؤية مستقبلية قيمة.

### 2. نظام التذاكر المؤتمت (Automated Ticketing System)

يعمل هذا المكون كجسر يربط بين مرحلة التحليل ومرحلة التنفيذ. فبمجرد أن ينتهي محرك الذكاء الاصطناعي من تحليل الإنذار، يقوم النظام تلقائيًا بـ:

- **إنشاء تذكرة عطل** :يتم إنشاء تذكرة جديدة في النظام بشكل فوري ، مما يلغي الحاجة إلى إدخال البيانات يدويًا.
- **تعبئة البيانات تلقائياً** :تُملأ التذكرة بجميع المعلومات التي استنتجها الذكاء الاصطناعي: موقع العطل، الأولوية، الحل المبدئي المقترح، والفريق المسؤول.
- **الإسناد المباشر** :يتم إسناد التذكرة مباشرة إلى الفريق الفني المعني ، مما يقلل بشكل كبير من الوقت الضائع في التواصل والتنسيق اليدوي.

### 3. الخريطة التفاعلية (GIS Visualization)

توفر هذه الواجهة لمركز عمليات الشبكة (NOC) لوحة تحكم بصرية شاملة. بدلاً من متابعة الأعطال كقوائم نصية، يمكن للمشغلين الآن رؤية حالة الشبكة بالكامل على خريطة جغرافية حية:

- **عرض جغرافي للأعطال** :يتم عرض جميع الأعطال النشطة كنقاط على خريطة تفاعلية باستخدام نظم المعلومات الجغرافية.(GIS)
- **ترميز لوني ذكي** :يتم تلوين كل نقطة بناءً على مستوى خطورة العطل (أحمر للحرجة، أصفر للمتوسطة، إلخ) ، مما يسمح للمشغلين بتحديد المناطق الأكثر تضرراً واستيعاب الوضع العام للشبكة بنظرة واحدة.

### 4. تطبيق الموبايل لفرق الدعم الفني (Mobile App for Technicians)

يمثل هذا التطبيق امتداداً للنظام ليصل إلى الفرق الميدانية، ويحول هواتفهم الذكية إلى أدوات عمل فعالة:

- **إشعارات فورية** :يتلقى الفنيون إشعاراً فورياً على هواتفهم **ضمن التطبيق** عند إسناد تذكرة جديدة إليهم ، مع كافة التفاصيل اللازمة.
- **الوصول للمعلومات من الميدان** :يمكن للفني عرض جميع تفاصيل العطل، بما في ذلك الحل المقترح من قبل الذكاء الاصطناعي، مباشرة من موقع العطل.

- **تحديثات آنية** :يسمح التطبيق للفني بتحديث حالة التذكرة (مثل "قيد التنفيذ" أو "تم الحل") مباشرة من الميدان ، وتنعكس هذه التحديثات فورًا في النظام المركزي.

من خلال تكامل هذه المكونات الأربعة، يخلق الحل المقترح منظومة عمل سلسة ومؤتمتة بالكامل، تعالج بشكل مباشر المشاكل الأساسية المتمثلة في بطء الاستجابة، صعوبة التشخيص، والتكاليف التشغيلية المرتفعة.

# 5.1-أهداف المشروع

لترجمة الحل المقترح إلى نتائج ملموسة، تم تحديد مجموعة من الأهداف الدقيقة التي تشكل معايير نجاح المشروع. تنقسم هذه الأهداف إلى ثلاث فئات رئيسية: أهداف الأداء التي تركز على تحسينات قابلة للقياس، والأهداف الوظيفية التي تصف قدرات النظام، والأهداف الاستراتيجية التي توضح القيمة طويلة الأمد للمشروع.

## 1.5.1-أهداف تحسين الأداء(Performance Objectives)

هذه هي المؤشرات الكمية المباشرة التي تعكس مدى فعالية النظام في تحسين العمليات اليومية. تُعتبر هذه الأهداف مؤشرات أداء رئيسية (KPIs) لمركز عمليات الشبكة.

- **خفض زمن الاستجابة (MTTR)** : يُعد **متوسط زمن الإصلاح Mean Time To (Repair**أحد أهم المقاييس في عالم الاتصالات، فهو يقيس المدة الزمنية من لحظة اكتشاف العطل حتى لحظة إصلاحه بالكامل. الهدف هنا ليس فقط تسريع عملية الإصلاح، بل تقليل فترة انقطاع الخدمة عن المشترك، مما ينعكس مباشرة على رضاه واستمراريته.

- **زيادة دقة التصنيف** :دقة التصنيف الآلي هي حجر الزاوية الذي يعتمد عليه النظام بأكمله. إذا فشل الذكاء الاصطناعي في تحديد الفريق المسؤول الصحيح أو الأولوية الحقيقية للعطل، فإن عملية الأتمتة بأكملها تفقد قيمتها. يضمن تحقيق هذه الدقة العالية توجيه كل مشكلة إلى الشخص المناسب من المرة الأولى، مما يمنع إهدار الوقت والموارد.

- **تقليل الإنذارات الكاذبة بنسبة** :يعاني مهندسو الشبكات من "ضوضاء الإنذارات"، حيث يكون الكثير منها غير مؤثر أو مجرد أعراض لمشكلة أخرى. يهدف النظام إلى تصفية هذه الإنذارات الكاذبة بذكاء، مما يحرر وقت المهندسين للتركيز على المشاكل الحقيقية التي تؤثر على جودة الخدمة بدلاً من مطاردة أعطال وهمية.

### 2.5.1-الأهداف الوظيفية Functional Objectives

تصف هذه الأهداف القدرات والمهام المحددة التي يجب أن يكون كل جزء من النظام قادراً على تنفيذها لتحقيق أهداف الأداء المذكورة أعلاه.

- **لنظام التحليل الذكي** :يجب أن يكون النظام قادراً على **التصنيف الآلي متعدد الأبعاد** لكل إنذار، وتحديد ليس فقط القسم المسؤول، بل أيضاً التكنولوجيا المتأثرة والحاجة للتدخل البشري. كما يجب أن يمتلك القدرة على **استخلاص الأنماط الخفية** من البيانات التاريخية لتقديم رؤى استباقية، مثل التنبؤ بالعطل التالي.

- **لنظام التذاكر المتكامل** :الهدف هنا هو تحقيق **تدفق عمل سلس وغير متقطع** .يجب أن يقوم النظام بإنشاء التذاكر وتعبئتها وتعيينها تلقائياً، مما يلغي تماماً التأخير الناتج عن الإدخال اليدوي. كما يجب أن يوفر **تحديثاً لحظياً لحالة التذاكر** لضمان أن جميع الأطراف على اطلاع دائم بتقدم عملية الإصلاح.

- **لنظام الخرائط التفاعلية** :يجب أن يقدم النظام **وعياً مكانياً Situational Awareness** لمشغلي مركز عمليات الشبكة. الهدف هو تحويل البيانات النصية المجردة إلى خريطة بصرية حية، تمكن المشغلين من فهم التأثير الجغرافي للأعطال وتحديد المناطق التي تحتاج إلى اهتمام فوري.

- **لتطبيق الجوال** :الهدف هو **تمكين الفرق الميدانية** وتحويلهم من مجرد منفذين للمهام إلى جزء متفاعل من الحل. يجب أن تزودهم الإشعارات الفورية بالمعلومات في الوقت المناسب، وأن تسمح لهم الوظائف التفاعلية بإغلاق حلقة التواصل عبر تحديث حالة الإصلاح مباشرة من الموقع.

### 3.5.1-الأهداف الاستراتيجيةStrategic Objectives

تمثل هذه الأهداف القيمة التجارية طويلة الأمد التي يضيفها المشروع إلى شركةMTN Syria ، وتتجاوز مجرد تحسين العمليات اليومية.

- **على مستوى الأعمال** :يهدف المشروع إلى تحقيق **ميزة تنافسية مستدامة** من خلال بناء سمعة قوية في موثوقية الشبكة. ففي سوق تنافسي، يعتبر تحسين تجربة العملاء وتقليل معدل تغييرهم (Churn) هدفاً استراتيجياً رئيسياً. كما يساهم تحسين كفاءة الموارد في خفض التكاليف التشغيلية على المدى الطويل.

- **على مستوى البيانات** :الهدف هو تحويل بيانات الأعطال من مجرد سجلات تاريخية إلى **أصل استراتيجي (Strategic Asset)** من خلال إنشاء قاعدة معرفية منظمة، يمكن

للشركة استخدامها في المستقبل للتحليل المتقدم، وتحسين التخطيط للشبكة، واتخاذ قرارات استراتيجية مبنية على أدلة بيانات قوية.

- **على مستوى التطوير المستقبلي** :تم تصميم النظام ليكون **منصة مرنة وقابلة للتوسع،** وليس مجرد أداة ثابتة. يضمن هذا الهدف أن الاستثمار في هذا المشروع يمكن أن ينمو ويتطور مع الشركة، من خلال إمكانية دمجه مستقبلاً مع أنظمة أخرى مثل إدارة علاقات العملاء(CRM) ، مما يفتح آفاقاً جديدة لتقديم خدمات أفضل.

## 6.1-فصول الأطروحة

تم تنظيم هذه الأطروحة في فصول متسلسلة منطقياً، حيث يبني كل فصل على الذي يسبقه، وذلك لتقديم عرض شامل ومتكامل للمشروع بدءاً من الفكرة وانتهاءً بالنتائج والتوصيات المستقبلية. يتكون الهيكل من الفصول التالية:

- **الفصل الأول: مقدمة المشروع** يقدم هذا الفصل نظرة شاملة ومتكاملة للمشروع. يبدأ بمقدمة عامة عن أهمية الشبكات الخلوية، ثم يعرف بشركة MTN Syria كدراسة حالة للمشروع، ويحدد بدقة المشكلة التشغيلية التي تهدف الأطروحة لحلها، ويعرض الحل التقني المقترح ومكوناته الرئيسية، ويضع الأهداف المحددة والقابلة للقياس، وأخيراً يقدم هذا الهيكل العام للأطروحة كخريطة طريق للقارئ.

- **الفصل الثاني: الخلفية النظرية - الشبكات الخلوية** يستعرض هذا الفصل المفاهيم الأساسية والتقنية المتعلقة بالشبكات الخلوية، لتزويد القارئ بالأساس النظري اللازم لفهم سياق المشكلة. يشمل ذلك شرحاً لتاريخ وتطور أجيال الشبكات) من 1 Gإلى 5(G، بنيتها المعمارية، المكونات التفصيلية للمحطات القاعدية، بالإضافة إلى استعراض لأشهر أنواع الأعطال والتحديات التي تواجهها.

- **الفصل الثالث: خوارزميات الذكاء الاصطناعي** يركز هذا الفصل على الجانب النظري للتقنيات الذكية المستخدمة في المشروع. يقدم شرحاً مفصلاً لآليات عمل الخوارزميات الرئيسية التي تم اختيارها، مثل Random Forest وXGBoost ، ويوضح المفاهيم الرياضية وراءها. كما يتناول تقنيات معالجة اللغات الطبيعية المستخدمة لتحويل البيانات النصية إلى بيانات رقمية، مثلTF–IDF .

- **الفصل الرابع: تحليل المتطلبات والبنية المعمارية للنظام** هذا الفصل مخصص لوضع المخطط الهندسي للنظام. يبدأ بتحليل معمق للمشكلة لتحديد المتطلبات الوظيفية وغير الوظيفية للنظام. بناءً على ذلك، يقدم التصميم المعماري عالي المستوى (High–level Architecture)، الذي يوضح كيفية ترابط المكونات الرئيسية (الواجهة الخلفية، قاعدة البيانات، محرك الذكاء الاصطناعي، الواجهات الأمامية).

- **الفصل الخامس: الأدوات والتقنيات المستخدمة** يقدم هذا الفصل نظرة عامة على حزمة الأدوات والتقنيات البرمجية التي تم اختيارها لتنفيذ المشروع. يشمل ذلك تبريراً لاختيار لغات البرمجة) مثل(Python ، أطر العمل) مثل Flask و(Flutter ، أنظمة قواعد البيانات (MySQL)، والمكتبات البرمجية الرئيسية.

- **الفصل السادس: بناء وتدريب نموذج الذكاء الاصطناعي** يعرض هذا الفصل الجانب العملي من بناء "عقل" النظام. يشرح بالتفصيل دورة حياة تطوير النموذج بالكامل: بدءاً من جمع البيانات وتنظيفها، مروراً بعملية هندسة الميزات(Feature Engineering) ، ثم تدريب الخوارزميات المختلفة وتقييم أدائها بدقة لاختيار النموذج النهائي والأكثر فعالية.

- **الفصل السابع: تطوير تطبيق الويب والموبايل** هذا الفصل مخصص لبناء التطبيقات التي يتفاعل معها المستخدم النهائي. يغطي التنفيذ العملي للواجهة الخلفية(Backend API) ، ثم ينتقل إلى تصميم وتطوير واجهة الويب الأمامية (Frontend) بكل مكوناتها التفاعلية من خرائط ورسوم بيانية، وأخيراً يستعرض تصميم وتطوير تطبيق الأندرويد المخصص للفرق الميدانية.

- **الخاتمة والآفاق المستقبلية** يلخص هذا الجزء الأخير النتائج الرئيسية التي تم التوصل إليها في المشروع، ويقيم مدى تحقيق الأهداف المحددة. كما يناقش القيود التي واجهت العمل ويقدم مجموعة من التوصيات والأفكار لأعمال تطوير مستقبلية يمكن أن تبنى على هذا المشروع.

# الفصل الثاني

# الشبكات الخلوية

# Cellular Network

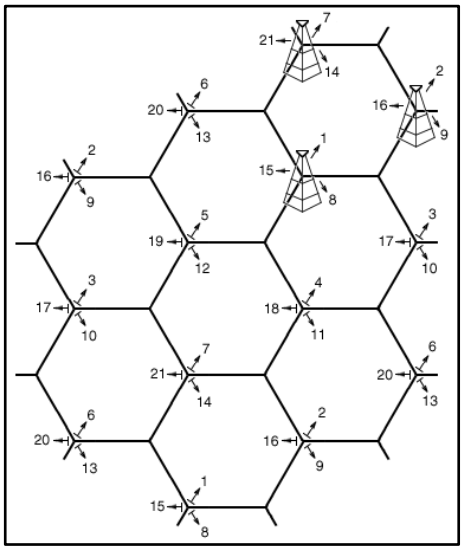
# 1.2-الشبكات الخلوية Cellular network

## 1.1.2- تعريف الشبكة الخلوية

هي نظام اتصالات لاسلكي يُستخدم للإرسال والاستقبال اللاسلكي للمعلومات بين الأجهزة المحمولة، والأبراج الخلوية sites التي تغطي المنطقة المحيطة بها وتتكون هذه الشبكة من مجموعة من الأبراج الخلوية المنتشرة في جميع أنحاء المنطقة المراد تغطيتها، وتتفاعل مع الأجهزة المحمولة لنقل البيانات.

تعتمد الشبكة الخلوية على تقنية الاتصالات الرقمية، حيث تحول المعلومات إلى إشارات رقمية تستطيع الأجهزة المحمولة والأبراج الخلوية فهمها ونقلها، تقسم المنطقة التي تغطيها الشبكة إلى مناطق صغيرة تسمى خلايا Cells، وتوجه الإشارات اللاسلكية بين الأجهزة المحمولة والأبراج الخلوية عبر هذه الخلايا.

تتحدد منطقة تغطية كل خلية بعوامل مثل قوة جهاز الإرسال والاستقبال، والتضاريس، ونطاق التردد المستخدم. تستخدم الخلية عادةً مجموعة مختلفة من الترددات عن الخلايا المجاورة لتجنب التداخل وتوفير جودة خدمة مضمونة داخل كل خلية. عندما تتحد هذه الخلايا معًا، فإنها توفر تغطية لاسلكية على مساحة جغرافية واسعة. يتيح ذلك للعديد من الأجهزة، بما في ذلك الهواتف المحمولة، والأجهزة اللوحية، وأجهزة الكمبيوتر المحمولة المزودة بمودم النطاق العريض المتنقل، والأجهزة القابلة للارتداء مثل الساعات الذكية، التواصل مع بعضها البعض ومع أجهزة الإرسال والاستقبال الثابتة والهواتف في أي مكان في الشبكة، عبر المحطات القاعدية، حتى لو كانت بعض الأجهزة تتحرك عبر أكثر من خلية واحدة أثناء الإرسال. يسمح تصميم الشبكات الخلوية بالتسليم السلس (handover)، مما يتيح اتصالاً غير منقطع عندما ينتقل الجهاز من خلية إلى أخرى.



الشكل (2-1) توزع المحطات الخلوية

تتميز الشبكة الخلوية بالقدرة على نقل المعلومات بسرعة وبكفاءة عاليتين، ويمكنها تغطية مساحات واسعة بشكل فعال وبما أن الشبكة الخلوية تعتمد على الاتصالات اللاسلكية، فإن الأجهزة المحمولة يمكنها الاتصال بالشبكة من أي موقع داخل منطقة التغطية، مما يوفر مستوى عالٍ من الراحة والمرونة للمستخدمين.

تستخدم الشبكة الخلوية في العديد من التطبيقات، بما في ذلك الاتصالات الصوتية والمرئية، والتراسل النصي، والإنترنت المتنقل، وخدمات الموقع الجغرافي، والتحكم عن بعد، والمزيد وتستخدم العديد من الشركات الكبيرة في مجال الاتصالات الشبكة الخلوية كنواة أساسية لخدماتها.

وقد شهدت الشبكة الخلوية تطورًا كبيرًا على مدار السنوات، حيث أُطلقت أجيال متعددة من التقنيات اللاسلكية، مثل 2G,3G,4G,5G، التي توفر سرعات وقدرات متزايدة للاتصالات الخلوية ومن المتوقع أن يستمر هذا التطور في المستقبل، مع إطلاق جيل جديد من التقنيات اللاسلكية والتي ستوفر ميزات وقدرات جديدة للاتصالات الخلوية.

توفر الشبكات الخلوية عددًا من الميزات المرغوبة:

- **سعة أكبر:** مقارنة بجهاز إرسال كبير واحد، حيث يمكن استخدام نفس التردد لروابط متعددة طالما أنها في خلايا مختلفة.
- **استهلاك طاقة أقل:** تستخدم الأجهزة المحمولة طاقة أقل من جهاز إرسال واحد أو قمر صناعي نظرًا لأن أبراج الخلايا أقرب.
- **منطقة تغطية أكبر:** من جهاز إرسال أرضي واحد، حيث يمكن إضافة أبراج خلايا إضافية إلى أجل غير مسمى ولا يحدها الأفق.
- **القدرة على استخدام إشارات بتردد أعلى:** (وبالتالي نطاق ترددي متاح أكبر / معدلات بيانات أسرع) التي لا تستطيع الانتشار لمسافات طويلة.
- ضغط البيانات وتعدد الإرسال: يمكن لعدة قنوات فيديو (بما في ذلك الفيديو الرقمي) وقنوات صوتية أن تنتقل عبر إشارة تردد أعلى على حامل نطاق عريض واحد.

لقد نشر مقدمو خدمات الاتصالات الرئيسيون شبكات خلوية للصوت والبيانات على معظم المناطق المأهولة بالسكان على الأرض. يسمح هذا للهواتف المحمولة والأجهزة الأخرى بالاتصال بالشبكة الهاتفية العمومية المحولة والوصول إلى الإنترنت العام. بالإضافة إلى خدمات الصوت والبيانات التقليدية، تدعم الشبكات الخلوية الآن تطبيقات إنترنت الأشياء (IoT)، وربط الأجهزة مثل العدادات الذكية والمركبات وأجهزة الاستشعار الصناعية.

## 21.2.-أجيال الشبكات المتنقلة

تطورت الشبكات المتنقلة بشكل كبير على مر السنين، حيث جلب كل جيل قدرات وتحسينات جديدة في السرعة والسعة والكفاءة. فيما يلي لمحة عامة عن الأجيال الخمسة للشبكات المتنقلة:

### الجيل الأول (1G)

اعتمدت شبكات الجيل الأول (1G)، التي تم تقديمها في عام 1979، على أنظمة الراديو التناظرية. وقد دعمت بشكل أساسي المكالمات الصوتية وكانت لها قيود من حيث الموثوقية والأمان (نقص التشفير) وتغطية الإشارة. كان التداخل من إشارات الراديو الأخرى شائعاً.

### الجيل الثاني (2G-GSM)

حلت شبكات الجيل الثاني (2G) محل الجيل الأول في عام 1991، واستخدمت إشارات رقمية، مما أدى إلى تحسين كبير في الأمان والسعة. أتاح الجيل الثاني إرسال رسائل SMS و MMS. وقد أتاح إدخال GPRS في عام 1997 للمستخدمين إرسال واستقبال رسائل البريد الإلكتروني أثناء التنقل.

**الجيل الثالث (3G-UMTS)**

أحدثت شبكات الجيل الثالث (3G) ثورة في الاتصال المتنقل من خلال توفير سرعات أعلى بكثير وقدرات نقل بيانات أكبر مقارنة بالجيل الثاني. وقد أتاح ذلك ميزات مثل مكالمات الفيديو، ومشاركة الملفات، وتصفح الإنترنت، ومشاهدة التلفزيون عبر الإنترنت، والألعاب المتنقلة. حول الجيل الثالث الهواتف المحمولة إلى مراكز للاتصال الاجتماعي .

**الجيل الرابع (4G-LTE)**

تُعد شبكات الجيل الرابع (4G) تقدماً كبيراً على الجيل الثالث، وهي أسرع بخمس مرات من شبكة الجيل الثالث، وتوفر نظريًا سرعات تصل إلى 100 ميجابت في الثانية. يوفر الجيل الرابع زمن انتقال أفضل، وجودة صوت أعلى، وسهولة الوصول إلى خدمات الرسائل الفورية ووسائل التواصل الاجتماعي، وبث عالي الجودة، وتنزيلات أسرع. تدعم معظم الأجهزة المحمولة التي تم إصدارها منذ عام 2013 فصاعدًا الجيل الرابع.

**الجيل الخامس (5G-NR)**

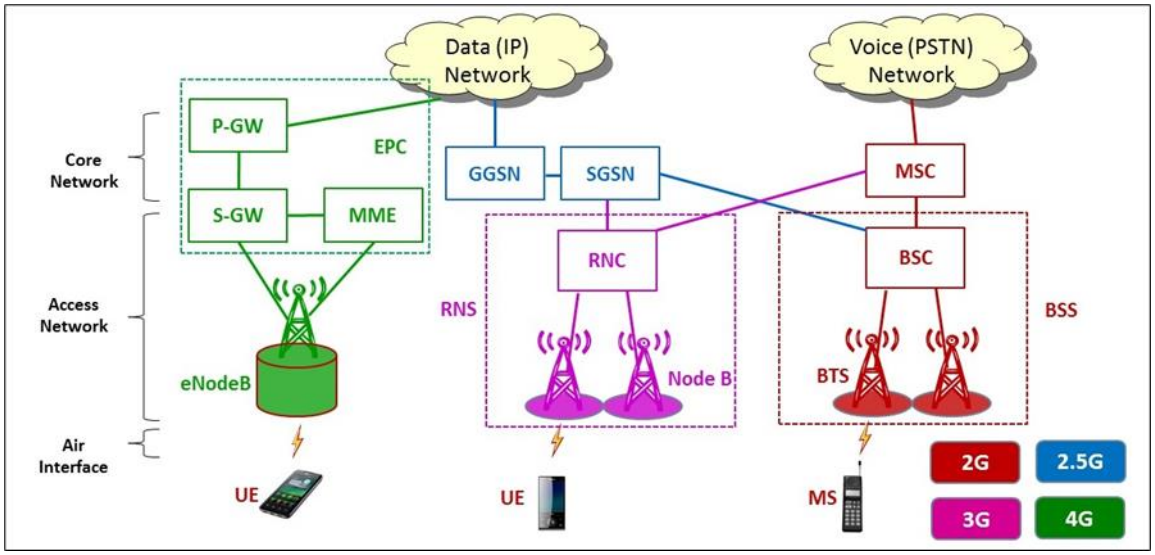
لقد أحدث الجيل الخامس (5G)، وهو أحدث جيل، تغييرات كبيرة في كيفية اتصال الأجهزة بالإنترنت. إنه يوفر نظرياً سرعات تنزيل أسرع 20 مرة من الجيل الرابع ويتميز بزمن انتقال منخفض جداً. وقد أتاح الجيل الخامس اتجاهات جديدة في إنترنت الأشياء (IoT)، بما في ذلك المدن الذكية، والرعاية الصحية المتصلة، والسيارات المتصلة. ولا يزال يتم استكشاف إمكاناته الكاملة، مع توقع المزيد من التطورات في السنوات القادمة.

**ما بعد الجيل الخامس: فجر الجيل السادس (6G)**

بينما يستمر الجيل الخامس في التطور، فإن الجيل السادس (6G) قيد المناقشة بالفعل، مع إطلاق متوقع حوالي عام 2030. يهدف الجيل السادس إلى توفير سرعات أسرع 100 مرة من الجيل الخامس، مما يتيح تطبيقات متقدمة مثل الاتصالات ثلاثية الأبعاد عالية الدقة، والشبكات التي تعتمد على الذكاء الاصطناعي، والاستشعار فائق الدقة للموقع. من المتوقع أن يحدث الجيل السادس تقدمًا كبيرًا في التكنولوجيا، مما يزيد من ترابط العالم بطرق لا يمكننا تخيلها بعد.

## 3.1.2-معمارية الشبكة الخلوية

كل شبكة خلوية تحتوي على الأقل على جهاز للإرسال والاستقبال (transceiver) في موقع ثابت، المعروف باسم محطة البث القاعدية Base Transceiver site، توفر تغطية الشبكة والتي يمكن استخدامها لنقل الصوت والبيانات وغيرها. كل خلية تستخدم مجموعة مختلفة من الترددات التي يجب أن تختلف عن ترددات الخلايا المجاورة لتفادي أي تشويش وتوفير الخدمة بجودة داخل نطاق تغطية كل خلية .توفر مجموعة من هذه الخلايا تغطية لاسلكية لمساحة جغرافية واسعة. وهذا يتيح لعدد كبير من أجهزة الإرسال والاستقبال المحمولة للاتصال مع بعضها البعض ومع الهواتف الثابتة في أي مكان في الشبكة عبر محطات القاعدة حتى لو كان بعض من الأجهزة تتحرك بين أكثر من خلية واحدة أثناء الإرسال.




الشكل (2-2) معمارية الشبكة الخلوية

- **شبكات النفاذ الراديوية(RAN) :**

**الجيل الثاني (2G - GSM)** :

يتم توصيل محطات الإرسال (BTS) عبر خطوط E1 إلى شبكة PDH ثم إلىSDH ، قبل الربط بالبنية الأساسية IP/MPLS باستخدامSTM-1 .

**الجيل الثالث (3G - UMTS)** :

ترتبط وحدات NodeB عبر Fast Ethernet و Packet Microwave إلى EoSDH، ثم تنتقل عبر Gigabit Ethernet إلى شبكة IP/MPLS .

**الجيل الرابع (4G - LTE)** :

تتصل محطات eNodeB بوحدة ONU ضمن بنيةGPON ، ثم إلىOLT ، ومنها إلى شبكة Metro Ethernet التي تتكامل مع.IP/MPLS

- **شبكات النقل والتجميع:**

يتم تجميع حركة البيانات من مختلف تقنيات RAN عبر شبكات مثل EoSDH وMetro Ethernet، ثم تمريرها إلى الشبكة الأساسية IP/MPLS التي تشكّل طبقة النقل المركزية باستخدام أجهزة توجيه متطورة.(PE Routers)

- **الشبكة الأساسية(Core Network) :**

**القسم التحويلي CS Core :**

يدعم خدمات الاتصال الصوتي ويتكامل مع شبكات PSTN عبر بوابات MGW ومراكزMSS/VLR ، باستخدام بروتوكول SIGTRAN للإشارات.

**القسم التبديل بالحزم PS Core:**

يدير جلسات البيانات وحركة المستخدمين في G/4G3، من خلال عناصر مثلMME ، S-GW، P-GW، SGSN، وGGSN.

- **وظائف أساسية إضافية:**
تشمل مكونات مثلHLR ، HSS، AuCوPCRF، المسؤولة عن التحقق من هوية المشترك وتطبيق السياسات وتحديد الرسوم. كما يتم دعم خدمات الصوت والفيديو عبر بروتوكول الإنترنت باستخدامIMS مثل VoLTE وVoWiFi.
- **الشبكات الخارجية (الإنترنت):**
يتم الوصول إلى الإنترنت من خلال وصلات عالية السرعة (10GbE) عبر بوابة P-GW، مع توجيه وتأمين البيانات القادمة من وإلى العالم الخارجي.

# 2.2-المحطات الخلوية Base Station

## 1.2.2-تعريف المحطات الخلوية وأهميتها

المحطة الخلوية Base Stationهي العنصر الأساسي في أي شبكة اتصالات خلوية، وهي المسؤولة عن ربط أجهزة المستخدم (مثل الهواتف المحمولة) بالشبكة اللاسلكية، وتحويل الإشارات بين الجهاز والشبكة الأساسية Core Network وهي تعمل ضمن خلية تغطية محددة، ومن هنا جاء اسمها.

وهي نقطة الاتصال الحاسمة بين المستخدم وشبكة الاتصالات. تطورت عبر الأجيال (من 2Gحتى 5G) لتوفر سرعات أعلى، تغطية أوسع، وخدمات متقدمة مثل الإنترنت، الفيديو، والمكالمات عالية الجودة(VoLTE) .

**وظائف المحطة الخلوية:**

- الاتصال اللاسلكي مع الأجهزة عبر(LTE/UMTS/GSM ).
- نقل البيانات الصوتية والإنترنت إلى الشبكة الأساسية.
- تبديل المكالمات والجلسات عند الانتقال بين الخلايا(Handover) .
- التحكم في مستوى الإشارة والجودة باستخدام خوارزميات ذكية.
- تشفير وحماية البيانات المرسلة بين الجهاز والشبكة.

**آلية عمل المحطة الخلوية:**

- يقوم الهاتف بإرسال طلب الاتصال إلى أقرب محطة.
- تستقبله وحدة الراديو(RRU) ، وتنقله إلى المعالج(BBU) .
- تعالجه الـ BBU وتحدد الوجهة (مكالمة، إنترنت...).
- تنقله إلى الشبكة الأساسية عبر نظام النقل.
- يتم تنفيذ الخدمة والرد بنفس المسار في الاتجاه المعاكس.

## 2.2.2-مكونات محطة التغطية الخلوية Base Station Components

### 1. وحدة المعالجة القاعدية BaseBand Unit

- تعتبر عقل المحطة، وهي المسؤولة عن جميع العمليات الحسابية والمعالجة الرقمية للإشارات.
- تُعالج الإشارات القادمة من الهوائيات وتنفذ الوظائف الأساسية مثل:
  - التحكم في الاتصال بالراديو.
  - تشفير وفك تشفير البيانات.
  - التفاعل مع الشبكة الأساسية(Core Network) .

تتكون وحدة معالجة القاعدة مما يلي:

| اسم الكرت | الوظيفة | ملاحظات |
|---|---|---|
| Universal Baseband Processing Board | يعالج الإشارات القاعدية لجميع التقنيات (2G/3G/4G) | يدعم الترددات المتعددة والتقنيات المتكاملة |
| Universal External Interface Unit / Power Entry Unit | وحدة تزويد الطاقة للـBBU | تدعم توصيل الطاقة DC 48V |
| Universal Main Processing and Transmission Board | المعالج الرئيسي + نقل البيانات | يشمل وظائفCPU ، إدارة الإشارات، والتحكم |
| Universal External Interface Unit | يتعامل مع إشارات النقل (Transport) مثل E1/GbE | - |
| Baseband Resource Interface Board | يربط بين BBU والـ RRU عبرCPRI | - |
| Universal Environment Interface Unit | يعالج إشارات المراقبة والإنذارات (alarms) من النظام إلى نظام O&M الخارجي. | يعد وحدة مركزية لجمع الإشارات البيئية (الطاقة، درجة الحرارة...) |

الجدول(1-1) أهم مكونات وحدة معالجة القاعدة

### 2. وحدة الترددات اللاسلكية Radio Frequency unit

- تُثبت بالقرب من الهوائيات (غالباً أعلى الأبراج).
- تقوم بتحويل الإشارة الرقمية إلى إشارة تردد لاسلكي (RF) والعكس.
- تدعم تعدد الحوامل وتعمل على أكثر من تردد(Multi-band RRUs) .
- تقلل من فقد الإشارة لأنها قريبة من الهوائي.

### 3. كابل الألياف البصرية Optical Fiber Cable

- يربط بين الـ BBU و الـRRU
- يستخدم بروتوكول CPRI أو eCPRI لنقل البيانات بسرعة عالية بين المكونات

**4. نظام الهوائيات Antenna System**

- تبث وتستقبل الإشارات اللاسلكية بين المستخدم والمحطة.
- أنواع الهوائيات:
  - هوائيات شاملة الاتجاه Omni Antennas
  - هوائيات قطاعية Sector Antennas غالبًا 3 قطاعات لكل برج.
  - متعددة الهوائيات MIMO Antennas تعدد الهوائيات لتحسين السعة والتغطية.
  - الهوائيات الفعال Active Antenna Unit - AAU تكون مدمجة مع الـ RRU.

**5. نظام الطاقة Power System**

يضمن استمرارية عمل المحطة بشكل مستقر ويتكون من:

- المقومات Rectifiers :لتحويل التيار المتردد AC إلى DC.
- البطاريات Batteries :لتوفير الطاقة أثناء انقطاع الكهرباء.
- وحدة توزيع الطاقة Power Distribution Unit (PDU) :لتنظيم توزيع الطاقة.
- المولدة Generator.
- نظام الطاقة الشمسية Solar system.

**6. وحدة النقل Transmission Unit**

تربط المحطة بالشبكة الأساسية Core Network.

تستخدم تقنيات مختلفة حسب الموقع:

- Microwave Links
- Fiber Optics
- E1/T1 Lines
- IP/MPLS Routers

**7. الكابينة الخارجية أو الداخلية Cabinet / Outdoor Enclosure**

تحتوي على BBU والأنظمة المساعدة.

تأتي بعدة أنواع:

- Indoor Cabinets للأماكن المغلقة.
- Outdoor Cabinets مقاومة للعوامل الجوية.
- Pole-mounted Units تُركب على الأعمدة.

**8. نظام المراقبة والتنبيه Monitoring and Alarm System**

لمراقبة أداء المعدات وحالة الطاقة ودرجات الحرارة.

يرسل تنبيهات لمراكز التشغيل والصيانة (O&M) في حال حدوث أعطال.

الشكل(2-3) مكونات محطة التغطية الخلوية

## 3.2- الأعطال والتحديات الشائعة في شبكات الاتصالات الخلوية

على الرغم من التطورات الكبيرة التي شهدتها الشبكات الخلوية، إلا أنها تواجه تحديات مختلفة ومعرضة لأعطال يمكن أن تؤثر على جودة الخدمة والموثوقية. يُعد فهم هذه المشكلات أمراً بالغ الأهمية للحفاظ على أنظمة اتصالات قوية.

### 1.3.2-أسباب انقطاع الشبكة وتدهورها

وفقًا لاستطلاعات رأي مشغلي الشبكات المتنقلة، تُعد أعطال الشبكة، وأعطال الروابط المادية، وازدحام الشبكة(الحمل الزائد) من أهم أسباب الانقطاعات. يمكن أن تشمل المشكلات المحددة ما يلي:

- **أعطال الشبكة:** يمكن أن تنشأ هذه الأعطال من مصادر مختلفة، بما في ذلك الأخطاء البرمجية، أو أخطاء التكوين، أو الأعطال في الأجهزة ضمن البنية التحتية للشبكة .
- **أعطال الروابط المادية:** يمكن أن يؤدي تلف البنية التحتية المادية، مثل كابلات الألياف الضوئية أو خطوط النقل، بسبب الحوادث أو الكوارث الطبيعية أو حتى سرقة معدات الموقع، إلى انقطاعات كبيرة في الخدمة.
- **ازدحام الشبكة/الحمل الزائد:** خلال أوقات الذروة أو في المناطق ذات الكثافة السكانية العالية، يمكن أن تصبح الشبكة محملة بشكل زائد، مما يؤدي إلى تباطؤ السرعات، وانقطاع المكالمات، وتدهور جودة الخدمة. تُعد هذه مشكلة شائعة، خاصة مع تزايد الطلب على التطبيقات التي تستهلك كميات كبيرة من البيانات .

- **أعطال الأجهزة:** يمكن أن تتعرض المكونات مثل المحطات القاعدية، والخوادم، والموجهات، والمودمات، ووحدات التبديل لأعطال في اللوحات أو رقائق المعالجة. يمكن أن تساهم

لعوامل البيئية، مثل نظام التبريد المعطل الذي يؤدي إلى ارتفاع درجة الحرارة، في أعطال الأجهزة.

- **انقطاع التيار الكهربائي:** يمكن أن يتسبب نقص إمدادات الطاقة المستمرة لمواقع الخلايا ومعدات الشبكة في اضطرابات واسعة النطاق في الخدمة، خاصة إذا كانت أنظمة الطاقة الاحتياطية غير كافية أو تفشل.
- **التداخل:** يمكن أن تتداخل إشارات الراديو الأخرى مع الإشارات الخلوية، مما يؤدي إلى ضعف جودة المكالمات أو مشكلات في نقل البيانات. يمكن أن يكون هذا تحدياً خاصاً في البيئات الحضرية التي تحتوي على العديد من الإشارات المتداخلة [3].
- **مخاوف الأمن والخصوصية:** مع نقل أنظمة الاتصالات اللاسلكية لكميات متزايدة من المعلومات السرية، تصبح أهدافًا للهجمات السيبرانية، مما يتطلب تدابير أمنية قوية لحماية البيانات ومنع الوصول غير المصرح به .
- **مناطق التغطية المحدودة:** على الرغم من الانتشار الواسع، لا يزال من الممكن وجود مناطق ذات تغطية خلوية ضعيفة أو معدومة، غالباً بسبب الحواجز الجغرافية، أو مواد البناء، أو المواقع النائية.
- **انقطاعات الخلايا قصيرة المدى (STCOs):** هذه انقطاعات قصيرة (تصل إلى 30 دقيقة) تؤثر على جميع أو بعض خلايا المحطة القاعدية وقد لا يتم اكتشافها مباشرة بواسطة أنظمة مراقبة الشبكة. غالباً ما يدركها المستخدمون على أنها انقطاعات خفية ويمكن أن تؤدي إلى شكاوى المشتركين وانخفاض حركة المرور للخدمة. تُعد انقطاعات الخلايا قصيرة المدى أكثر شيوعًا في المناطق الحضرية ولكن لها تأثير أقوى على المستخدمين في المناطق الريفية، ويرتبط حدوثها بمستويات حركة المرور والظروف الجوية السيئة.

### 2.3.2-التحديات في تحديد الأعطال وحلها

يمكن أن يكون تحديد السبب الجذري لأعطال الشبكة أمراً صعباً بسبب الترابط المعقد بين عناصر الشبكة المختلفة. قد لا يتم تحديد العلاقة بين مكونات الأجهزة المختلفة وعناصر التحكم البيئية (مثل أنظمة التبريد) بشكل صريح دائماً، مما يجعل من الصعب ربط الأعطال تلقائياً بالسبب الجذري. يمكن أن يؤدي هذا التعقيد إلى فترات حل طويلة وزيادة تكاليف الصيانة.

**1 تأثير مشكلات الشبكة**

تترتب على انقطاعات الشبكة وتدهور الخدمة عواقب وخيمة، بما في ذلك:

- **فقدان الإيرادات:** تؤدي انقطاعات الخدمة مباشرة إلى خسارة الإيرادات لمشغلي الشبكات المتنقلة.
- **زيادة تكاليف الصيانة:** يتطلب تشخيص وإصلاح المشكلات المعقدة موارد كبيرة ويمكن أن يؤدي إلى ارتفاع نفقات التشغيل.
- **تأثير جودة الخدمة (QoS):** يعاني المستخدمون من تدهور الخدمة، مما يؤدي إلى الإحباط واحتمال فقدان العملاء.
- **عدم رضا المستخدمين:** يمكن أن يؤدي ضعف أداء الشبكة إلى تجربة مستخدم سلبية، مما يؤثر على ولاء العملاء وسمعة العلامة التجارية.

للتخفيف من هذه التحديات، يستثمر مشغلو الشبكات المتنقلة بشكل متزايد في الحلول التي توفر رؤية أفضل لأداء الشبكة وتتيح التشخيص عن بعد وحل المشكلات، بهدف بناء شبكات أكثر مرونة وكفاءة.

### 3.3.2-الكشف التلقائي عن الأعطال وتشخيصها في الشبكات الخلوية

يُعد الكشف عن الأعطال وتشخيصها أمراً بالغ الأهمية للحفاظ على أداء وموثوقية الشبكات الخلوية. وبينما حظي الكشف عن الأعطال بدراسة مستفيضة، فإن مهمة تشخيص الأعطال الأكثر تعقيداً، والتي تتضمن تحديد السبب الجذري للعطل، لا تزال مجالاً بحثياً مستمراً .

#### 1.3.3.2-أساليب الكشف عن الأعطال وتشخيصها

يستكشف الباحثون طرقاً مختلفة للكشف التلقائي عن الأعطال وتشخيصها، وغالباً ما يستفيدون من البيانات التي يتم جمعها من الشبكة. تشمل هذه الأساليب ما يلي:

- **بيانات نظام دعم الأداء:** يتم جمع هذه البيانات تلقائياً بواسطة الشبكة ويمكن تحليلها باستخدام خوارزميات التعلم غير الخاضع للإشراف والتصنيف للكشف عن الأعطال وتشخيصها مثل الازدحام والأعطال في تخصيص قنوات المرور.
- **بيانات اختبار القيادة Drive test data:** يتم جمع هذه البيانات يدوياً من خلال اختبارات القيادة في سيناريوهات مكالمات مختلفة (المكالمات القصيرة والطويلة والخاملة). تساعد المكالمات القصيرة في تحديد الأعطال في إعداد المكالمات، وتم تصميم المكالمات الطويلة لتحديد أعطال التسليم وانقطاع المكالمات، وتساعد اختبارات الوضع الخامل في فهم خصائص الإشارات القياسية في الشبكة .
- **الذكاء الاصطناعي (AI) والتعلم الآلي (ML):** يتم تطبيق تقنيات الذكاء الاصطناعي والتعلم الآلي، بما في ذلك الشبكات العصبية، بشكل متزايد لتحليل بيانات الشبكة، وتحديد الأنماط، والتنبؤ بالأعطال. يمكن أن تساعد هذه الأساليب في أتمتة عملية التشخيص وتحسين دقة تحديد الأعطال.
- **خوارزميات قائمة على دمج المعرفة والبيانات:** بالنسبة لشبكات الجيل الخامس، يتم تطوير خوارزميات تجمع بين الأساليب القائمة على المعرفة وتحليل البيانات الضخمة لتعزيز قدرات تشخيص الأعطال .
- **الشبكات العصبية التلافيفية الرسومية (GCN):** تُستخدم الشبكات العصبية التلافيفية الرسومية لتطوير خوارزميات تشخيص أعطال الشبكات اللاسلكية غير المتجانسة، خاصة لشبكات الجيل الرابع والجيل الخامس، من خلال الاستفادة من العلاقات بين عناصر الشبكة المختلفة .

## 4.2-آلية الكشف عن الأعطال يدوياً وأهمية الأتمتة

في شبكة MTN Syria يتم الكشف عن الأعطال بالطريقة التالية:

- يحدث العطل على محطة ما، فرضاً DAM001.
- يتحسس الكرت المسؤول عن الأعطال الموجود في الـ BBU عن وجود عطل، ويكون الكرت المسؤول هو كرت UPEU اذا كان عطل رئيسي(معرف من قبل الشركة المصنعة)، أو كرت UEIU اذا كان عطل ثانوي(معرف من قبل المشغل وفي حالتنا MTN Syria)

- بعد التعرف على العطل ترسل وحدة المعالجة العطل باستخدام وحدة النقل.
- يصل العطل إلى مركز التشغيل NOC عبر برنامج وسيط مقدم من الشركة المصنعة، وفي حالتنا هو MAE المقدم من شركة Huawei.
- مركز التشغيل مكون من مهندسين مسؤولين عن مراقبة الأعطال ومتابعتها، حيث بعد ورود العطل يقومون بتحديد التالي:
  - نوع العطل.
  - مدى تأثير العطل على الخدمة.
  - آلية حل العطل.
  - حل هو بحاجة تدخل بشري.
  - الجيل المتأثر بهذا العطل.
  - القسم المختص بالعطل(Radio,Maintenance,Power,Tranissmisom)
- يقوم مركز الأعطال بتحليل العطل وفق السابق وتحديد القسم المختص وارسال العطل إليه.
- يقوم القسم المختص بحل المشكلة أو بتوجيه ورشة إلى الموقع واصلاح العطل.
- يصل إشعار من النظام بحل المشكلة عبر واجهة MAE.

تُعد الشبكات ذاتية الإصلاح مهمة بشكل خاص لإدارة الأعطال وفشل النظام من خلال الكشف التلقائي عن أسبابها وتحديدها، وبالتالي تقليل الوقت الذي يؤثر فيه العطل على الشبكة وتسريع عملية الاسترداد.

يهدف التشخيص التلقائي للأعطال إلى تقليل التدخل البشري، الذي يمكن أن يؤدي إلى أخطاء وتناقضات. من خلال الاستفادة من البيانات الدقيقة وتقنيات التحليل المتقدمة، يتمثل الهدف في إنشاء شبكات خلوية أكثر قوة ومرونة يمكنها الكشف عن الأعطال وتشخيصها وحتى إصلاحها تلقائياً، مما يضمن خدمة مستمرة وعالية الجودة للمستخدمين.

.

# الفصل الثالث

# خوارزميات الذكاء الاصطناعي

# Artificial Intelligence Algorithm

## 1.3-تعريف الذكاء الاصطناعي وخوارزمياته

الذكاء الاصطناعي والذي يُعرف اختصاراً بـ AI هو تقنية ذات قدرات حل تشبه قدرات الإنسان في حل المشكلات. يبدو أن الذكاء الاصطناعي في العمل يحاكي الذكاء البشري - يمكنه التعرف على الصور وكتابة القصائد وإجراء تنبؤات قائمة على البيانات .

تجمع المؤسسات الحديثة كمياتٍ كبيرةً من البيانات من مصادر متنوعة مثل أجهزة الاستشعار الذكية والمحتوى الذي ينشئه الإنسان وأدوات المراقبة وسجلات النظام. تقوم تقنيات الذكاء الاصطناعي بتحليل البيانات واستخدامها لمساعدة العمليات التجارية بفعالية. على سبيل المثال، يمكن لتقنية الذكاء الاصطناعي الاستجابة للمحادثات البشرية في دعم العملاء، وإنشاء صور ونصوص أصلية للتسويق، وتقديم اقتراحات ذكية للتحليلات.

في النهاية، يتعلق الذكاء الاصطناعي بجعل البرامج أكثر ذكاءً لتفاعلات المستخدم المخصصة وحل المشكلات المعقدة.

### 1.1.3-خورزميات الذكاء الاصطناعي

تعتمد خوارزميات الذكاء الاصطناعي على تقنيات رياضية ومنطقية لتحليل البيانات واستخراج الأنماط، مما يتيح لها اتخاذ قرارات ذكية تستند إلى تلك التحليلات. ما يميزها هو قدرتها على التطور المستمر باستخدام أدوات مثل التعلم الآلي (Machine Learning) والشبكات العصبية الاصطناعية (Neural Networks)، حيث تزداد دقتها وكفاءتها بمرور الوقت وتكرار العمليات.

### 2.1.3-أنواع خوارزميات الذكاء الاصطناعي

**1. التعلم الخاضع للإشراف Supervised Learning**

يستخدم هذا النوع بيانات تحتوي على مدخلات ومخرجات معروفة، مما يُمكن الخوارزميات من التعرف على الأنماط والعلاقات بينهما،مثال:

- في القطاع المالي، تُستخدم هذه الخوارزميات في الكشف عن محاولات الاحتيال عن طريق تحليل الأنماط المالية المشبوهة باستخدام تقنيات مثل الانحدار الخطي وأشجار القرار.

**2. التعلم غير الخاضع للإشراف Unsupervised Learning**

على عكس النوع السابق، لا تعتمد خوارزميات هذا النوع على بيانات مُعلمة، بل تحلل البيانات الخام لاستكشاف الأنماط والعلاقات الخفية، مثال:

- في التسويق الرقمي، تُستخدم خوارزميات التجميع Clustring لتقسيم العملاء إلى شرائح بناءً على سلوكياتهم الشرائية، مما يساعد الشركات على فهم العملاء بشكل أفضل واستهدافهم بدقة.

**3. التعلم المعزز (Reinforcement Learning)**

يركز هذا النوع على مبدأ التجربة والخطأ، حيث يتعلم النظام من التفاعل مع البيئة المحيطة والحصول على تغذية راجعة لتحسين أدائه، مثال:

- في السيارات ذاتية القيادة، تعتمد الخوارزميات على التعلم المستمر لاختيار المسارات المثلى وتجنب الحوادث من خلال التفاعل مع البيئة في الوقت الفعلي.

## 3.1.3-تطبيقات خوارزميات الذكاء الاصطناعي في الحياة اليومية

1. **الرعاية الصحية**
   - تُستخدم الخوارزميات في تحليل صور الأشعة وتشخيص الأمراض بدقة وسرعة.
   - توفر أدوات مثل التحليل الطبي بالذكاء الاصطناعي دعماً للأطباء من خلال تحسين كفاءة التشخيص وتقليل الأخطاء.
2. **المساعدات الرقمية**
   - تعتمد المساعدات الشخصية مثل "أليكسا" و"سيري" على خوارزميات متقدمة لتحليل الأوامر الصوتية وتنفيذها، مما يجعل حياة المستخدمين أكثر سهولة وتنظيمًا.
3. **الأمن المالي**
   - تُساعد خوارزميات الذكاء الاصطناعي في حماية الحسابات المصرفية من الاحتيال، من خلال تحليل الأنماط المالية وكشف أي أنشطة غير عادية.
4. **السيارات ذاتية القيادة**
   - تعتمد السيارات على معالجة البيانات المحيطة بها لاتخاذ قرارات فورية وآمنة، مما يجعل القيادة الذاتية أكثر أماناً وفعالية يوماً بعد يوم.

## 4.1.3-مزايا خوارزميات الذكاء الاصطناعي

1. **تحليل الأنماط والتنبؤ:**

تتيح الخوارزميات التنبؤ بالأحداث المستقبلية بناءً على البيانات الحالية، مما يساعد الشركات في اتخاذ قرارات استراتيجية وتحديد احتياجات السوق.

2. **الأتمتة والتعلم الذاتي:**

تقلل الخوارزميات من التدخل البشري في المهام الروتينية، مما يزيد من الكفاءة ويضمن دقة متزايدة بمرور الوقت.

3. **التخصيص وتجربة المستخدم:**

تقدم تجربة مخصصة للمستخدمين من خلال تحليل تفضيلاتهم، مثل توصيات الأفلام على نتفليكس أو قوائم التشغيل على سبوتيفاي.

## 4.1.3-التحديات التي تواجه خوارزميات الذكاء الاصطناعي

1. **الاعتماد على البيانات الضخمة:**

تحتاج هذه الخوارزميات إلى كميات كبيرة من البيانات ، مما قد يكون عائقًا أمام المؤسسات التي لا تمتلك الموارد الكافية.

2. **التحيز في البيانات:**

يؤدي التحيز في البيانات إلى قرارات غير عادلة، خصوصًا في مجالات مثل العدالة الجنائية والتوظيف.

**3. الشفافية والتفسير:**

يصعب تفسير قرارات بعض الخوارزميات، لا سيما تلك القائمة على الشبكات العصبية، مما يحد من استخدامها في القطاعات التي تتطلب وضوحًا عاليًا.

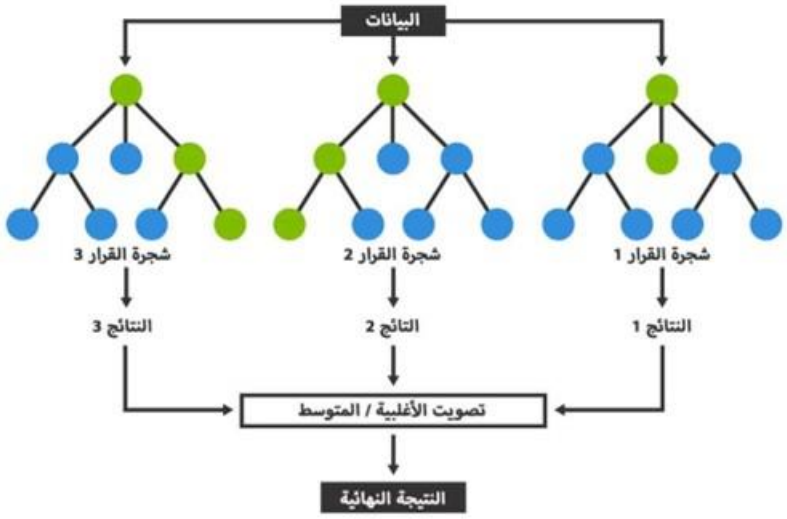
تُمثل خوارزميات الذكاء الاصطناعي ركيزة أساسية في عالمنا الحديث، حيث تقدم حلولاً مبتكرة لتحسين حياتنا اليومية وزيادة كفاءة القطاعات المختلفة. ومع ذلك، يبقى التحدي الأكبر هو ضمان استخدامها بشكل مسؤول وأخلاقي يعزز رفاهية المجتمع دون التضحية بالشفافية أو العدالة.

## 2.3-خوارزمية الغابة العشوائية Randon Forest

هي إحدى خوارزميات التعلم الآلي الخاضع للإشراف (Supervised Learning) التي تُستخدم في مهام التصنيف (Classification) والانحدار (Regression) .

تعتمد هذه الخوارزمية على مفهوم تعلم المجموعات(Ensemble Learning) ، حيث يتم دمج عدة نماذج (أشجار قرار) لتحسين الأداء وتقليل التباين(Variance) .

.




الشكل(3-1) خوارزمية Random forest

### 1.2.3-المكونات الأساسية لـ Random Forest

1. أشجار القرار(Decision Trees)

كل شجرة قرار في الغابة العشوائية (Random Forest) تُبنى باستخدام خوارزمية CART . (Classification and Regression Trees)، التي تقوم بتقسيم البيانات بناءً على معايير مثل:

- التقسيم في التصنيف Gini Impurity
- التقسيم في الانحدار Mean Squared Error (MSE)
- معيار جيني(Gini Impurity)

2. معادلة الغابة العشوائية

ليست معادلة واحدة بسيطة مثل المعادلات التقليدية، بل هي خوارزمية مبنية على تجميع عدة أشجار قرار (Decision Trees) لتكوين نموذج أقوى وأكثر دقة. ومع ذلك، يمكننا شرح الفكرة الأساسية لها بطريقة رياضية مبسطة.

تعريف مبسط:

| الغابة العشوائية = مجموعة من أشجار القرار + التصويت أو المتوسط |
|---|

**إذا كانت المهمة تصنيف (Classification)**

يتم استخدام التصويت بالأغلبية:

$$y = mode(h_1(x), h_2(x), \ldots, h_T(x))$$

حيث:

$y$: التنبؤ النهائي.

$h_T(x)$: التنبؤ من الشجرة رقم T.

$T$: عدد الأشجار في الغابة.

Mode: الدالة التي تعيد الأكثر تكرارًا (تصويت بالأغلبية).

**إذا كانت المهمة تنبؤ عددي(Regression) :**

يتم استخدام المتوسط الحسابي:

$$h_t(x) \sum_{t=1}^{T} \frac{1}{T} = y$$

## 2.2.3-ميزات العشوائية(Randomness)

تتميز Random Forest بـ ثلاثة أنواع من العشوائية لتحسين الأداء:

- Bootstrap Aggregating (Bagging)
يتم اختيار عينات التدريب بشكل عشوائي مع الإحلال (مع تكرار بعض العينات).
كل شجرة تُدرَّب على مجموعة مختلفة من البيانات.
- Random Feature Selection
عند بناء كل عقدة في شجرة القرار، يتم اختيار مجموعة فرعية عشوائية من الميزات (Features) بدلاً من استخدام جميع الميزات.
- بناء أشجار غير مقيدة(Fully Grown Trees)
تُبنى الأشجار حتى الوصول إلى العقد النهائية (الأوراق) دون تقليم(Pruning) .

### 3.2.3-آلية عمل Random Forest

**1. خطوات التدريب(Training)**

- إنشاء مجموعة بيانات Bootstrap
يتم سحب n عينات من مجموعة البيانات الأصلية مع الإحلال (أي بعض العينات قد تتكرر).
- بناء أشجار القرار:
لكل شجرة يتم اختيار مجموعة فرعية عشوائية من الميزات في كل تقسيم.
يتم بناء الشجرة حتى الوصول إلى الحد الأقصى للعمق أو حتى تصبح جميع العينات في العقدة من فئة واحدة.
- التكرار حتى اكتمال الغابة:
يتم بناء Tأشجار (حيث Tهو عدد الأشجار المحدد، مثل 50 أو 100).

**2. خطوات التوقع(Prediction)**

- للتصنيف(Classification)
تمرير العينة عبر كل شجرة.
كل شجرة تصوت على الفئة (Class) التي تعتقد أن العينة تنتمي إليها.
يتم اختيار الفئة الأكثر تصويتًا (Majority Voting).
- للانحدار (Regression)
يتم أخذ متوسط توقعات جميع الأشجار.

### 4.2.3- مميزات Random Forest

- مقاومة للتجهيز الزائد (Overfitting) بسبب استخدام Bagging و Feature Randomness.
- تعمل مع البيانات الكبيرة والمتنوعة (Numerical & Categorical).
- لا تحتاج إلى تطبيع (Normalization/Scaling) البيانات.
- تقدم قياساً لأهمية الميزات(Feature Importance) .
- تتعامل مع القيم المفقودة (Missing Values) بشكل جيد.

### 5.2.3-عيوب Random Forest

- بطيئة نسبياً عند وجود عدد كبير جداً من الأشجار.
- تستهلك ذاكرة أكبر من بعض الخوارزميات الأخرى مثلLogistic Regression .
- صعوبة تفسير النموذج (Model Interpretability) مقارنة بشجرة قرار واحدة.

### 6.2.3- تحسين أداء Random Forest (Hyperparameter Tuning)

| المعامل(Parameter) | الوصف | القيم المقترحة |
|---|---|---|
| n_estimators | عدد الأشجار في الغابة | 50-500 |
| max_depth | أقصى عمق لكل شجرة | 3-20 |
| min_samples_split | أقل عدد عينات مطلوب لتقسيم عقدة | 2-10 |
| min_samples_leaf | أقل عدد عينات في العقدة الورقية | 1-5 |
| max_features | عدد الميزات المدخلة في كل تقسيم | 'auto', 'sqrt', 'log2' |
| bootstrap | استخدام أخذ العينات مع الإحلال (Bagging) | True/False |

الجدول(3-1) تحسين أداء Random forest

## 3.3-خوارزمية Logistic Regression

هي إحدى خوارزميات التعلم الآلي الخاضع للإشراف (Supervised Learning) التي تُستخدم بشكل أساسي في مشاكل التصنيف الثنائي(Binary Classification) ، لكن يمكن توسيعها للتعامل مع التصنيف متعدد الفئات (Multiclass Classification) باستخدام إستراتيجيات مثل One-vs-Rest (OvR).

استخدام خوارزمية الانحدار اللوجستي

- إذا كنت تريد نموذجاً سريعاً وقابلاً للتفسير.
- إذا كانت العلاقة بين الميزات والهدف شبه خطية.
- إذا كانت البيانات متوازنة (ليس هناك تباين كبير في توزيع الفئات).
- إذا كنت بحاجة إلى احتمالات التصنيف (ليس مجرد توقع الفئة).

### 1.3.3- الأساس الرياضي للانحدار اللوجستي

#### 1. دالة السيجمويد(Sigmoid Function)

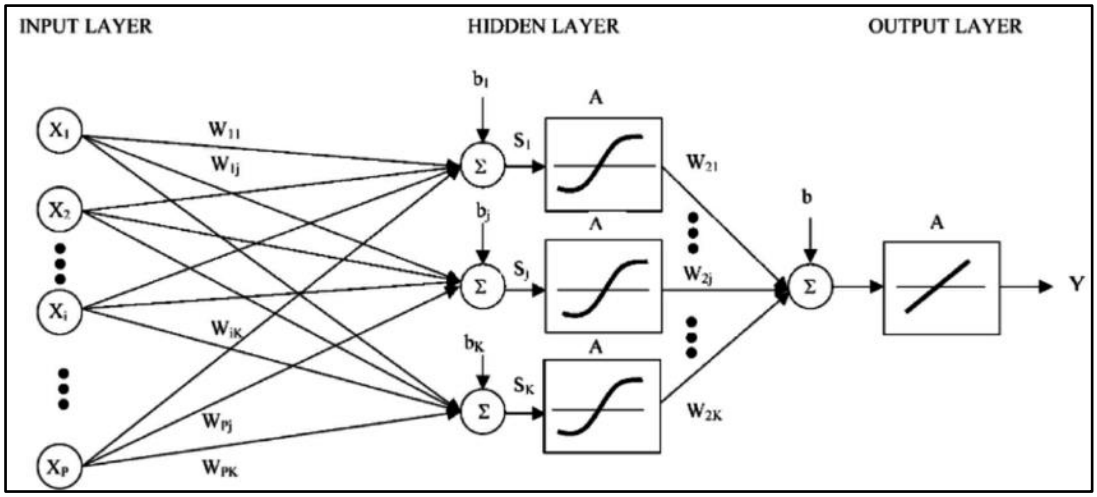
الانحدار اللوجستي يحول المخرجات إلى نطاق [0، 1] باستخدام دالة السيجمويد:

$$\sigma(x) = \frac{1}{1-e^{-z}}$$

حيث:

$$z = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n$$

σ(z): يمثل احتمالية أن تنتمي العينة للفئة 1.
z: أي عدد حقيقي (المدخل).
e: العدد النيبيري (تقريبًا 2.718).




الشكل(3-2) خوارزمية **Logistic Regression**

**2. اتخاذ القرار(Decision Boundary)**

- إذا كان σ (z)≥0.5 الفئة 1
- إذا كان σ(z)<0. 5 الفئة 0

**3. دالة الخسارة(Loss Function)**

يستخدم Log Loss (Binary Cross-Entropy)لقياس الخطأ:

$$J(w) = \sum_{i=1}^{N} \frac{1}{N} - [y_i \log(\sigma(z_i)(1 - yi)log(1 - \sigma(z_i))]$$

حيث:

N:عدد العينات

$y_i$:القيمة الحقيقية (0 أو 1)

$\sigma(zi)$ :الاحتمال المتوقع

### 2.3.3-آلية عمل Logistic Regression

1. مرحلة التدريب(Training)
   - تهيئة الأوزان wعشوائيًا.
   - حساب $z$ لكل عينة.
   - تحويل z إلى احتمالية باستخدام السيجمويد.
   - حساب الخطأ باستخدام.Log Loss
   - تحديث الأوزان باستخدام:Gradient Descent

$$W_j = w_j - \alpha \frac{\partial w_j}{\partial J(w)}$$

حيث:

$w_j$:الوزن الحالي رقم j.

α:معدل التعلم (Learning Rate).

$\frac{\partial w_j}{\partial J(w)}$:المشتق الجزئي لدالة الخطأ بالنسبة للوزن $w_j$، أي مقدار التغير في الخطأ عند تغيير الوزن.

J(w):دالة الخطأ (Log Loss) .

2. مرحلة التوقع(Prediction)
   - حساب z للبيانات الجديدة.
   - تطبيق السيجمويد للحصول على الاحتمال σ(z) .
   - إذا σ (z)≥0.5الفئة 1، وإلا الفئة 0.
3. توسيع الانحدار اللوجستي للتعامل مع عدة فئات One-vs-Rest (OvR)
   - يتم تدريب مصنف منفصل لكل فئة ضد جميع الفئات الأخرى.
     مثال: إذا كان لدينا 3 فئات(A, B, C) ، يتم تدريب 3 نماذج:
     النموذج1:A vs (B + C)
     النموذج 2:B vs (A + C)
     النموذج 3:C vs (A + B)

     يتم اختيار الفئة ذات الأعلى احتمالية.

### 3.3.3-مميزات وعيوب Logistic Regression

**المميزات:**

- سريع في التدريب والتنبؤ (مناسب للبيانات الكبيرة).
- نتائجه قابلة للتفسير (يمكن معرفة تأثير كل ميزة).
- يعمل جيداً عندما تكون العلاقة بين المدخلات والمخرجات شبه خطية.
- يمكن استخدامه لاستخراج أهم الميزات.(Feature Importance)

**العيوب:**

- يفترض علاقة خطية بين المدخلات واللوغاريتم(Logit) .
- غير مناسب للمشاكل غير الخطية المعقدة.
- حساس للقيم المتطرفة(Outliers) .
- يحتاج إلى تطبيع البيانات (Feature Scaling) ليعمل بكفاءة.

### 4.3.3-معايير ضبط النموذج(Hyperparameter Tuning)

| المعامل (Parameter) | الوصف | القيم المقترحة |
|---|---|---|
| penalty | نوع التنظيم(L1, L2, None) | 'l2', 'l1', 'elasticnet' |
| C | عكس قوة التنظيم (كلما زاد C قل التنظيم) | 0.001 - 100 |
| solver | خوارزمية التحسين ('liblinear', 'lbfgs', 'sag', 'saga') | يعتمد على حجم البيانات |
| max_iter | الحد الأقصى لعدد التكرارات | 100-1000 |

الجدول (3-2) تحسين أداء logistic Regression

## 4.3-خوارزميةXGBoost

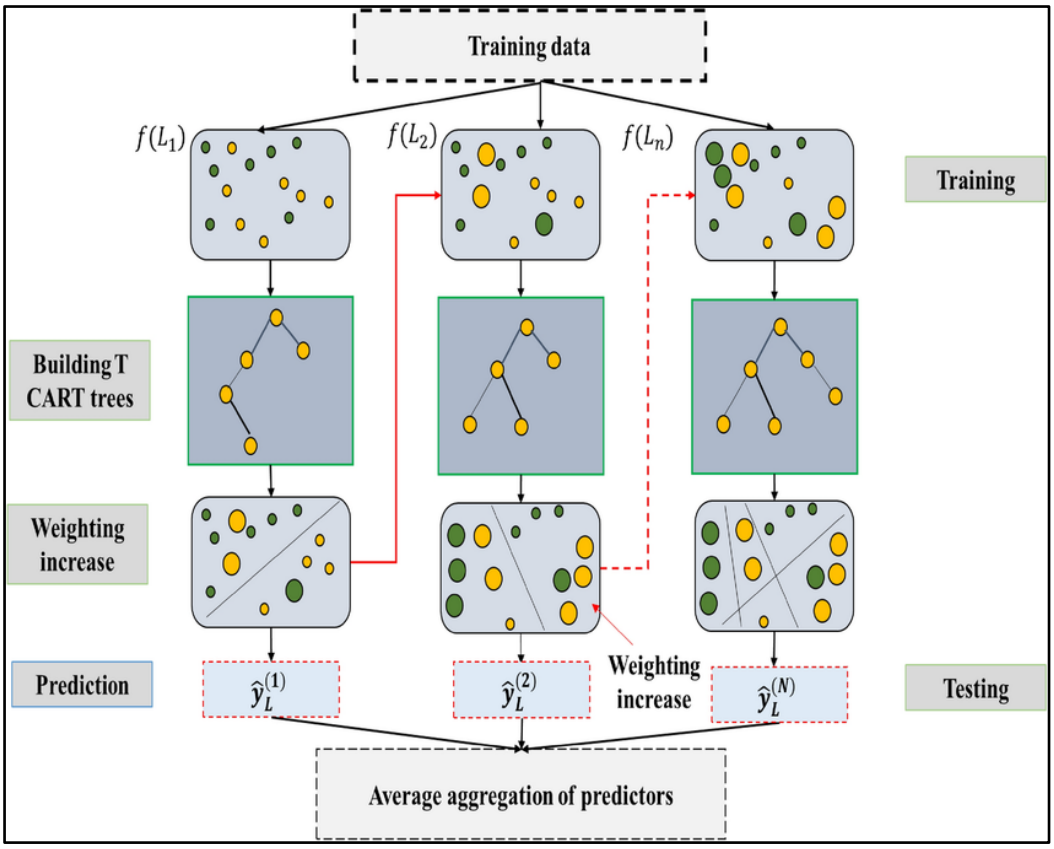
هي اختصار لـ **Extreme Gradient Boosting**، وتُعد من أقوى خوارزميات التعلم الآلي الخاضع للإشراف(Supervised Learning) ، وتُستخدم بشكل فعال في كل من مشاكل التصنيف (Classification)والانحدار (Regression) ،تتميز بدقتها العالية وسرعتها في التدريب.

### 1.4.3-استخدام خوارزميةXGBoost

- إذا كنت تريد نموذجًا قويًا وفعّالاً في اكتشاف الأنماط المعقدة.
- إذا كانت البيانات تحتوي على علاقات غير خطية أو ميزات مترابطة.
- إذا كان لديك عدد كبير من الميزات وتريد اكتشاف الأهم بينها.
- إذا كنت تريد نموذجًا قابلاً للتعديل الدقيق (Fine-tuning) من خلال معايير كثيرة.

### 2.4.3-الأساس الرياضي لـXGBoost

يعتمد XGBoost على تقنية **Boosting**، حيث يتم تدريب مجموعة من الأشجار الضعيفة(weak learners) بشكل تسلسلي، بحيث تُحسّن كل شجرة من أخطاء السابقة.




الشكل(3-3) خوارزمية **XGBoost**

**دالة الهدف(Objective Function) :**

XGBoostيحاول تقليل دالة الهدف التالية:

$$l(t) = \sum_{k=1}^{K} \Omega(f_k) + \sum_{i=1}^{N} l(y_i, \widehat{\boldsymbol{y}_i^t})$$

حيث:

- $l(y_i, \widehat{\boldsymbol{y}_i^t})$: دالة الخسارة(Log Loss)
- $\Omega(f_k)$: دالة التعقيد (Regularization) للشجرة $f_k$، وتُعرّف بـ:

$$\Omega(f_k) = \gamma T + \frac{1}{2} w \sum_{j=1}^{T} \lambda_j^2$$

T:عدد العقد (الأوراق) في الشجرة.

$w_j$:وزن كل ورقة.

### 3.4.3- آلية عمل XGBoost

- **مرحلة التدريب(Training)**

1. يبدأ النموذج بتوقع ثابت (عادة متوسط القيم).
2. يُحسب الخطأ بين التوقعات والقيم الحقيقية.
3. تُبنى شجرة جديدة لتوقع **الخطأ.(residuals)**
4. يتم تحديث التوقعات بإضافة مخرجات الشجرة الجديدة.
5. تُكرر الخطوات حتى يتم الوصول إلى عدد معين من الأشجار أو يتوقف التحسين.

- **التحديث الرياضي لكل تكرار:**

$$\widehat{y_i^t} = \widehat{y_i}^{\,t-1} + f_k(x_i)$$

حيث:

$\widehat{y_i}^{\,t-1}$:التنبؤ من التجربة السابقة

$f_k(x_i)$:تنبؤ الشجرة الجديدة.

- **مرحلة التوقع:(Prediction)**

يتم جمع مخرجات كل الأشجار وتطبيقها على العينة الجديدة:

$$\widehat{y_i^t} = \sum_{k=1}^{K} f_k(x_i)$$

ثم يُطبّق Softmax أو Sigmoid للحصول على احتمالات في حالة التصنيف.

### 4.4.3-مميزات وعيوبXGBoost

**<u>المميزات:</u>**

- دقة عالية جدًا في أغلب التحديات الواقعية.
- سرعة عالية في التدريب باستخدام موازاة متعددة.(Parallelism)
- يحتوي على تنظيم داخلي يقلل من فرط التعلّم.
- يدعم البيانات الناقصة تلقائيًا.
- يوفر إمكانية استخراج أهمية الميزات.

**<u>العيوب:</u>**

- يمكن أن يكون بطيئًا في التوقع مع عدد كبير من الأشجار.
- عدد كبير من المعاملات يجب ضبطها(Hyperparameters) .
- أقل قابلية للتفسير مقارنة بنماذج بسيطة كـLogistic Regression .

### 5.4.3- معايير ضبط النموذج(Hyperparameter Tuning)

| المعامل | الوصف | القيم المقترحة |
|---|---|---|
| n_estimators | عدد الأشجار | 100 – 1000 |
| learning_rate | معدل التعلم(η) | 0.01 – 0.3 |
| max_depth | العمق الأقصى لكل شجرة | 3 – 10 |
| subsample | نسبة العينات المستخدمة لكل شجرة | 0.5 – 1 |
| colsample_bytree | نسبة الميزات المستخدمة في كل شجرة | 0.5 – 1 |
| gamma | الحد الأدنى للربح المطلوب للتقسيم | 0 – 5 |
| lambda | تنظيم L2 على الأوزان | 0 – 10 |
| alpha | تنظيم L1 على الأوزان | 0 – 10 |

الجدول (3-3) تحسين أداء XGBoost

## 5.3-مقارنة بين RandomForest و LogisticRegression وXGBoost

| المعيار | RandomForest | LogisticRegression | XGBoost |
|---|---|---|---|
| نوع الخوارزمية | تعلم مجموعة (Ensemble) | نموذج خطي | تعلم مجموعة (Boosting) |
| آلية العمل | أشجار قرار متعددة مع عشوائية | دالة لوجستية + تحسين احتمالي | تعزيز تدرجي مع أشجار قرار |
| أفضل ل | البيانات المعقدة غير الخطية | البيانات الخطية أو شبه الخطية | البيانات المنظمة والمعقدة |

| المعيار | RandomForest | LogisticRegression | XGBoost |
|---|---|---|---|
| معالجة البيانات | لا يحتاج تطبيع | يحتاج تطبيع | لا يحتاج تطبيع |
| حساسية للقيم المتطرفة | منخفضة | عالية | منخفضة |
| سرعة التدريب | متوسط (يعتمد على عدد الأشجار) | سريع | بطيء (خاصة مع التكرارات الكثيرة) |
| سرعة التوقع | متوسط | سريع جداً | سريع (بعد التدريب) |
| التفسيرية | متوسطة (أهمية الميزات) | عالية (معاملات واضحة) | متوسطة |
| التجهيز الزائد (Overfitting) | مقاوم جيد (مع ضبط العمق) | متوسط (بحاجة لتنظيم) | مقاوم جيد (مع معاملات التنظيم) |
| معالجة البيانات غير المتوازنة | جيد (يمكن وزن الفئات) | متوسط (بحاجة لمعالجة مسبقة) | ممتاز scale_pos_weight |
| دعم متعدد المخرجات | ممتاز) مع MultiOutput) | جيد) مع MultiOutput) | ممتاز) مدمج في XGBoost) |
| المعاملات الرئيسية | n_estimators, max_depth | C, penalty | n_estimators, max_depth, learning_rate |

| نوع البيانات | RandomForest | LogisticRegression | LinearSVC | XGBoost |
|---|---|---|---|---|
| البيانات النصية | جيد جدًا | جيد | ممتاز (الأفضل عادةً) | جيد جدًا |

| نوع البيانات | RandomForest | LogisticRegression | LinearSVC | XGBoost |
|---|---|---|---|---|
| البيانات الرقمية | ممتاز | جيد | جيد | ممتاز |
| البيانات الفئوية | ممتاز (بدون ترميز) | يحتاج ترميز | يحتاج ترميز | ممتاز (بدون ترميز) |
| البيانات عالية الأبعاد | جيد (مع تقليل الميزات) | متوسط (بحاجة لتنظيم قوي) | ممتاز | جيد (مع ضبط المعاملات) |
| البيانات الصغيرة | متوسط (قد يتجهز) | جيد | جيد | متوسط (قد لا يتقارب) |

الجدول (3-4) مقارنة أداء الخوارزميات على أنواع البيانات المختلفة

## 6.3-خوارزمية TF-IDF (Term Frequency-Inverse Document Frequency)

هي إحدى أهم طرق تمثيل النصوص رقمياً في معالجة اللغات الطبيعية (NLP) وتعلم الآلة. تستخدم لقياس أهمية الكلمة في مستند بالنسبة لمجموعة مستندات .

### 1.6.3-استخدام TF-IDF

- يحول النصوص إلى أرقام يمكن للنماذج الرياضية فهمها
- يعطي وزناً أعلى للكلمات المهمة والمميزة في المستند
- يقلل من تأثير الكلمات الشائعة غير المهمة .

### 2.6.3-مكونات TF-IDF

1. تكرار المصطلح Term Frequency (TF)

تقيس كم مرة تظهر الكلمة في المستند مقارنة بجميع كلماته.

TF(t,d) :عدد تكرار الكلمة t في المستند

d:إجمالي عدد الكلمات في المستند

2. التردد العكسي للمستند Inverse Document Frequency (IDF)

تقيس ندرة الكلمة في جميع المستندات. الكلمات النادرة تكون أكثر أهمية.

3. TF-IDFالنهائي
هو حاصل ضرب القيمتين السابقتين:

### 3.6.3-آلية عمل TF-IDF

- إنشاء مصفوفة التكرارات
- حساب عدد مرات ظهور كل كلمة في كل مستند.
- حساب قيم TF لكل كلمة في كل مستند.
- حساب قيم IDF لكل كلمة .
- حساب TF-IDF بضربTF × IDF .
- تطبيع النتائج.

### 4.6.3-مميزات وعيوبTF-IDF

**المميزات:**

- بسيط وسهل التطبيق
- فعال في تمييز الكلمات المهمة
- أفضل من مجرد حساب التكرارات(Count Vectorizer)

**العيوب:**

- لا يأخذ في الاعتبار ترتيب الكلمات
- لا يفهم معاني الكلمات (مثل المرادفات)

## 7.3-خوارزمية MultiOutputClassifier

هو غلاف (meta-estimator) في مكتبة scikit-learn يسمح بتطبيق خوارزميات التصنيف العادية على مشاكل التصنيف متعدد المخرجات(multi-output classification) .

### 1.7.3-مشاكل التصنيف متعدد المخرجات

عندما يكون لديك عدة متغيرات هدف (Y) تحتاج التنبؤ بها في نفس ، يكون له فئات مختلفة

- **الإستراتيجية الأساسية ( One-vs-Restلكل هدف)**
  - ينشئ مصنف منفصل لكل متغير هدف
  - كل مصنف يتدرب بشكل مستقل على بيانات التدريب
  - عند التنبؤ، يجمع توقعات جميع المصنفات

| الطريقة | MultiOutputClassifier | ClassifierChain | MultiTask Learning |
|---|---|---|---|

| الطريقة | MultiOutputClassifier | ClassifierChain | MultiTask Learning |
|---|---|---|---|
| **التنفيذ** | مستقل لكل هدف | متسلسل (يستخدم توقعات الأهداف السابقة) | مشترك لجميع الأهداف |
| **الدقة** | جيدة | أفضل (يستفيد من علاقات الأهداف) | الأفضل إذا كانت الأهداف مرتبطة |
| **التنفيذ** | أسهل | أكثر تعقيداً | معقد |

الجدول (3-5) مقارنة خوارزمية MultiOutputClassifier مع خوارزميات مشابهة

## 2.7.3-استخدام خوارزمية MultiOutputClassifier

- عندما لا تكون هناك علاقة قوية بين المتغيرات الهدف
- عندما تريد حلاً بسيطاً وسريع التطبيق
- عند استخدام خوارزميات تصنيف عادية مثل RandomForest

## 3.7.3-مميزات وعيوب MultiOutputClassifier

**المميزات**

- سهل الفهم والتطبيق
- يعمل مع أي مصنف عادي
- لا يحتاج تعديلات على الخوارزمية الأساسية

**العيوب:**

- يتجاهل العلاقات بين المتغيرات الهدف
- يزيد حجم النموذج (يحفظ نسخة منفصلة لكل هدف)
- ليس الأمثل عندما تكون الأهداف مترابطة بقوة

# الفصل الرابع
# تحليل المتطلبات والبنية المعمارية للنظام

## 1.4-تحليل المشكلة وسير العمل الحالي Problem Analysis and Current Workflow

لفهم حجم الفرصة التي يقدمها هذا المشروع، يجب علينا التعمق في تحليل سير العمل اليدوي الحالي لإدارة الأعطال في مركز عمليات الشبكة (NOC) بشركة MTN Syria. هذا التحليل لا يقتصر على تحديد الخطوات، بل يكشف عن التحديات الكامنة والتكاليف الخفية المرتبطة بهذه العملية التقليدية.

دورة حياة العطل في ظل النظام الحالي هي عملية خطية ومتعددة المراحل، كل مرحلة منها تمثل نقطة تأخير محتملة:

1. **حدوث العطل ونقله (Steps 1 & 2)**: عندما يقع عطل في محطة تغطية، يتم إرسال إنذار إلى مركز العمليات. هذه المرحلة الأولية مؤتمتة بطبيعتها.
2. **استقبال الإنذار والتحليل اليدوي (Steps 3 & 4)**: هنا تبدأ المشكلة الحقيقية. يصل الإنذار كسطر نصي على شاشة المراقبة أمام مهندس الـ NOC. في هذه اللحظة الحرجة، يواجه المهندس **حملاً معرفياً زائداً (Cognitive Overload)**، حيث يجب عليه:
   - **فك شيفرة الإنذار**: فهم معنى الإنذار التقني، الذي قد يكون غامضاً أحياناً.
   - **تقييم الأثر**: ربط الإنذار بموقعه الجغرافي وتحديد عدد المشتركين أو الخدمات المتأثرة.
   - **البحث عن السياق**: قد يحتاج المهندس للرجوع إلى سجلات الأعطال السابقة أو مخططات الشبكة لفهم ما إذا كان هذا العطل متكرراً أو مرتبطاً بمشاكل أخرى.
   - **اتخاذ القرار**: بناءً على خبرته الشخصية (التي تختلف من مهندس لآخر)، يجب عليه أن يقرر القسم المسؤول (طاقة، راديو، نقل، إلخ).
3. **الإسناد والتوجيه (Step 5)**: بعد اتخاذ قرار التحليل، تبدأ مرحلة جديدة من العمل اليدوي. يقوم المهندس بالتواصل مع القسم المعني عبر الهاتف أو البريد الإلكتروني، وشرح المشكلة، وتمرير التفاصيل. هذه العملية بحد ذاتها تستهلك وقتاً ثميناً وقد يحدث فيها سوء فهم.
4. **الإصلاح والإغلاق (Steps 6 & 7)**: يقوم الفريق المختص بالعمل على الإصلاح، وبعد حل المشكلة، يتم تحديث الحالة في النظام يدوياً من قبل مهندس الـ NOC.
5. **نقاط الضعف الجوهرية وتأثيراتها**

هذا التحليل العميق يوضح أن العملية الحالية ليست مجرد "بطيئة"، بل هي منظومة تعاني من نقاط ضعف جوهرية تؤثر على الشركة بأكملها:

- **التأخير المركب (Compounded Delays)**: كل خطوة يدوية في العملية تضيف دقائق ثمينة إلى **متوسط زمن الإصلاح (MTTR)**. التأخير في التحليل يضاف إلى التأخير في التواصل، والذي يضاف إلى التأخير في توجيه الفريق الصحيح. هذه الدقائق تتراكم لتصبح ساعات من انقطاع الخدمة المحتمل، مما يعني خسارة مباشرة في الإيرادات وتزايد استياء العملاء.

- **مخاطر الخطأ البشري العالية**: الاعتماد على الخبرة الفردية يجعل النظام غير متسق. **مثال عملي**: قد يقوم مهندس بتشخيص عطل "انقطاع إشارة" على أنه مشكلة في "الراديو"، بينما السبب الحقيقي هو عطل في "الطاقة". سيتم إرسال فريق الراديو إلى الموقع ليكتشف الخطأ، ثم يتم استدعاء فريق الطاقة، مما يضخم زمن الإصلاح والتكاليف بشكل كبير.
- **النهج التفاعلي وانعدام الرؤية المستقبلية**: النظام الحالي مصمم للتعامل مع المشاكل **بعد وقوعها**. إنه يفتقر تماماً للقدرة على النظر إلى البيانات بشكل شامل لاكتشاف الأنماط التي تسبق الأعطال الكبيرة. **على سبيل المثال**: قد يكون هناك نمط متكرر لارتفاع درجة حرارة جهاز معين قبل أن يتعطل تماماً بأسبوع. النظام اليدوي عاجز عن اكتشاف مثل هذه الأنماط، وبالتالي تضيع فرصة إجراء صيانة وقائية كانت ستمنع العطل من الحدوث أصلاً.
- **تأثير الصوامع (Silo Effect)**: لأن كل قسم (طاقة، نقل، راديو) يرى فقط الأعطال المتعلقة به، لا توجد رؤية شاملة وموحدة للمشاكل. هذا يخلق "صوامع" معلوماتية، حيث قد لا يدرك فريق النقل أن مشكلة الطاقة التي يعاني منها فريق آخر هي السبب الجذري لمشاكلهم، مما يؤدي إلى إهدار الوقت في التشخيص الخاطئ.

إن تحليل سير العمل الحالي يثبت أن المشكلة ليست مجرد "عدم كفاءة"، بل هي مشكلة استراتيجية تحد من قدرة الشركة على ضمان جودة الخدمة، وتزيد من تكاليفها التشغيلية، وتمنعها من الاستفادة من بياناتها كأصل استراتيجي لاتخاذ قرارات أفضل.

## 2.4-أهم الأعطال في شركة MTN Syria

تعدد الأعطال الممكنة الحدوث على الشبكة وتختلف أسبابها، ولكن يمكن تحديد أهم الأعطال وفق القسم المختص بالشكل التالي:

### 1.2.4-أعطال في الشبكة الخلوية

القسم المختص في هذه الحالة هو قسم التغطية الراديوية Radio Team وتشمل هذه الأعطال عطل على مستوى الخلية أو المحطة، ويكون غالباً خطأ في التهيئة أو عطل في العتاد المادي Hardware fault .

### 2.2.4-أعطال في شبكة النقل

القسم المختص في هذه الحالة هو قسم النقل Transmission Team(TX) وتشمل هذه الأعطال عطل على مستوى المحطة أو الربط بين المحطات.

### 3.2.4-أعطال في الطاقة

القسم المختص هو قسم الطاقة Power team، وتحدث هذه الأعطال نتيجة وجود كهرباء في المحطة مثل انقطاع التغذية الكهربائية أو مشاكل بالألواح الشمسية...

### 4.2.4-أعطال في المحطة بشكل عام

القسم المختص في هذه الحالة هو قسم الصيانة Maintenance team وتشمل هذه الأعطال جميع أنواع الأعطال الممكنة الحدوث عن المحطة بشكل عام مثل عطل في التكييف أو ارتفاع درجة الحرارة أو...

### 5.2.4-أعطال عابرة

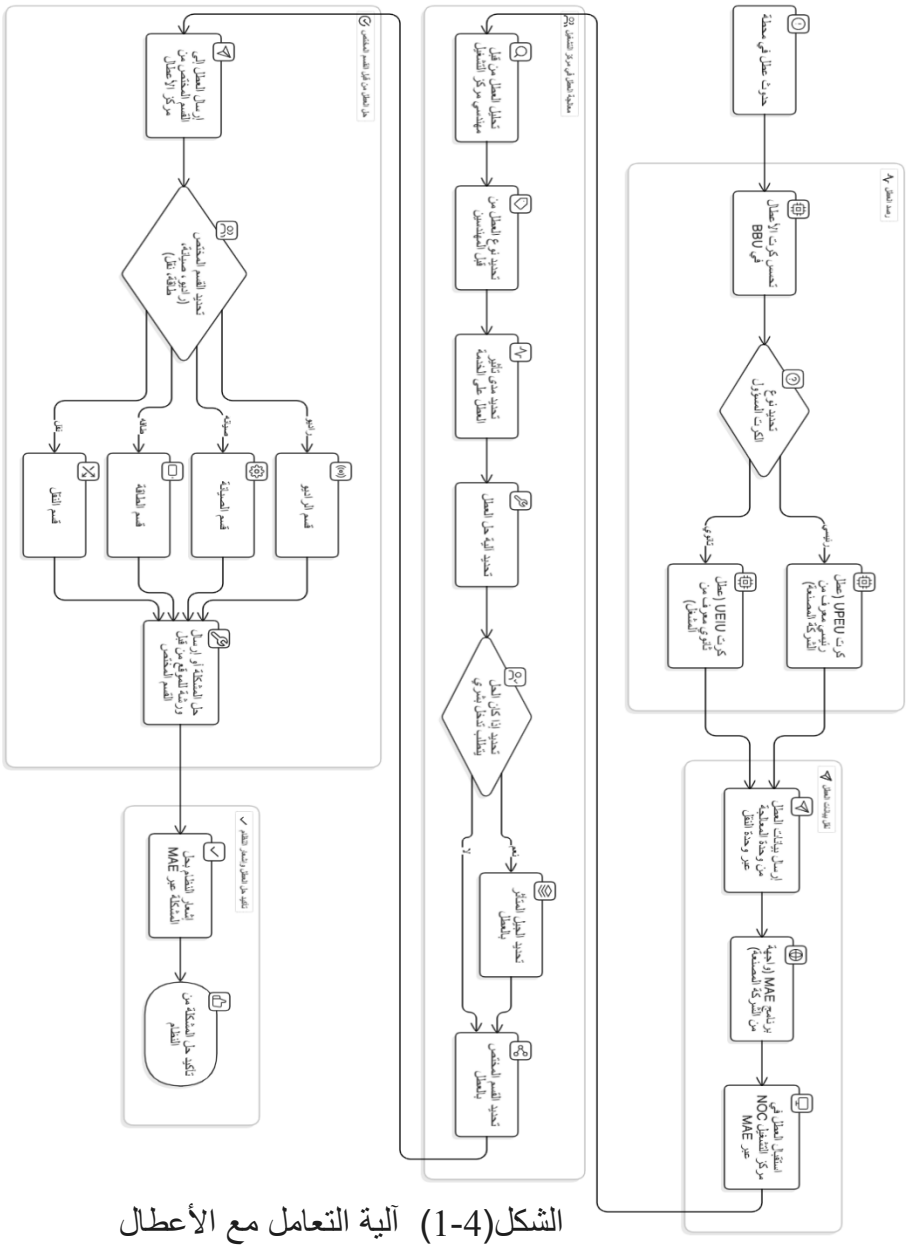
القسم المختص في هذه الحالة هو قسم Fault Management وتكون هذه الأعطال ناتجة عن أعطال اخرى أو أعطال ذات أولوية منخفضة ولا تحتاج سوى إلى مراقبة.

يوضح الجدول التالي أهم الأعطال مع القسم المختص

| القسم | نوع العطل | نوع العطل |
|---|---|---|
| FM | UMTS Cell Unavailable | GSM Local Cell Unusable |
| | NE Is Disconnected | Licensed Feature Entering Keep-Alive Period |
| | Monitoring Device Maintenance Link Failure | NodeB Audit No Response |
| | UMTS Cell MC-HSDPA Function Fault | Licensed Feature Unusable |
| | ALD Maintenance Link Failure | Task execution failure alarm |
| | Statistical Alarm | Configuration Data Inconsistency |
| MA | NodeB Unavailable | Solar Cabinet Door Open |
| | High Temperature | AC Battery Room Fault |
| | Fan Faulty | Battery Subrack First Cabinet High Temperature |
| | Battery & Rectifier First Cabinet Door Open | Solar Cabinet High Temperature |
| | Switch AC2 Fault | Cabinet High Temperature Outdoor |
| | Fan Stalled | Burglar Alarm |
| | AC Equipment Room Fault | Sensor Failure |
| Power | Solar System Mains Failure | Generator On |
| | Solar System Module Failure | Solar System DC Low Voltage |
| | Mains Input Out of Range | Generator On (Outdoor) |
| | Base Station DC Power Supply Abnormal | Solar System DC Low Voltage (Outdoor) |
| | Mains Failure | Power Module Abnormal |
| | RF Unit DC Input Power Failure | System Mains Failure Outdoor |
| | RF Unit External Power Supply Insufficient | AC Surge Protector Fault |
| | Battery Power Unavailable | Power Supply DC Output Out of Range |
| | Solar System Module Failure (Outdoor) | Rectifier Subrack First Cabinet High Temperature |

| | | |
|---|---|---|
| | Battery Current Out of Range | Power Module and Monitoring Module Communication Failure |
| | Generator Fault | Solar System Mains Failure (Outdoor) |
| Radio | GSM Cell out of Service | UMTS Cell Common Channel Reconfiguration Failed |
| | Cell Unavailable | RF Unit ALD Current Out of Range |
| | Local Cell Unusable | Radio Signaling Link Disconnected |
| | BBU CPRI Interface Error | UMTS Cell Max DL Power Mismatch |
| | RF Unit TX Channel Gain Out of Range | RF Unit Maintenance Link Failure |
| | RF Unit CPRI Interface Error | UMTS Cell Common Channel Setup Failed |
| | RF Unit RX Channel RTWP/RSSI Unbalanced | RF Unit VSWR Threshold Crossed |
| | Cell RX Channel Interference Noise Power Unbalanced | Cell Output Power too Low |
| | RF Unit Overload | Radio Link Failure |
| TX | Adjacent Node IP Address Ping Failure | SCTP Link Congestion |
| | IP Clock Link Failure | Time Synchronization Failure |
| | ESL Link Fault | LAPD Link Congestion |
| | OML Fault | M3UA link transmission quality Faulty |
| | CSL Fault | Inter-System Communication Failure |
| | External Clock Reference Problem | SCTP Link Fault |
| | User Plane Fault | Path Fault |
| | eNodeB S1 Control Plane Transmission Interruption | Ethernet Link Fault |
| | Remote Maintenance Link Failure | |

الجدول (4-1) أهم أعطال الشبكة الخلوية



الشكل(4-1) آلية التعامل مع الأعطال

## 3.4- المتطلبات غير الوظيفية Non-Functional Requirements

إذا كانت المتطلبات الوظيفية هي "ماذا" يفعله النظام، فإن المتطلبات غير الوظيفية هي التي تحدد "كيف" يجب أن يؤدي تلك الوظائف بجودة وكفاءة. إنها تمثل العمود الفقري الذي يضمن أن النظام ليس مجرد فكرة جيدة، بل أداة قوية ومستقرة يمكن الاعتماد عليها في بيئة عمل حرجة.

### 1.3.4- الأداء (Performance)

الأداء لا يعني فقط السرعة، بل يعني تمكين المستخدمين من أداء مهامهم بفعالية تحت الضغط. في مركز عمليات الشبكة (NOC)، كل ثانية لها ثمن.

- **زمن استجابة الواجهة :** مهندس الـ NOC يراقب عشرات المؤشرات في وقت واحد. إذا استغرقت لوحة التحكم وقتاً طويلاً للتحميل أو التحديث، فإنه يفقد تركيزه وقدرته على

الاستجابة السريعة. زمن الاستجابة السريع يضمن أن تظل الواجهة أداة مساعدة وليست عائقاً، مما يسمح للمشغل بالانتقال بسلاسة بين الخرائط والجداول والرسوم البيانية دون انقطاع.

- **سرعة التحليل الذكي :**تكمن قيمة الذكاء الاصطناعي في قدرته على تقديم رؤى فورية. إذا تأخر النظام في تحليل الإنذار، فإنه يفقد ميزته التنافسية على التحليل البشري. يجب أن تتم عملية التصنيف والتنبؤ بسرعة البرق لضمان وصول المعلومة الصحيحة إلى الفريق المناسب في أقصر وقت ممكن، مما يساهم بشكل مباشر في تقليل متوسط زمن الإصلاح (MTTR).

### 2.3.4- الأمان (Security)

يتعامل النظام مع بيانات حساسة تتعلق بالبنية التحتية للشبكة وحسابات الموظفين، مما يجعل الأمان متطلباً غير قابل للتفاوض.

- **حماية كلمات المرور (التشفير)**: تخزين كلمات المرور كنص عادي هو بمثابة ترك باب الخزنة مفتوحاً. استخدام دوال التجزئة (Hashing) القوية يعني أنه حتى لو تمكن شخص من اختراق قاعدة البيانات، فإنه لن يتمكن من قراءة كلمات المرور، مما يحمي حسابات المستخدمين من السرقة والاستغلال.
- **صلاحيات الوصول (Role-Based Access Control)**: لا ينبغي أن يمتلك كل مستخدم نفس الصلاحيات. الفني الميداني يحتاج فقط إلى رؤية التذاكر المسندة إليه وتحديث حالتها، بينما يحتاج مدير مركز العمليات إلى رؤية شاملة للشبكة وتقارير الأداء. يضمن هذا المبدأ حماية النظام من التغييرات غير المقصودة أو المتعمدة ويطبق مبدأ "الحاجة للمعرفة" (Need-to-Know Basis).

### 3.3.4- الموثوقية والإتاحة (Reliability and Availability)

مركز عمليات الشبكة هو مركز عصبي يعمل على مدار الساعة طوال أيام الأسبوع (7/24). أي توقف في نظام المراقبة يعني أن الشركة أصبحت "عمياء" وغير قادرة على رؤية حالة شبكتها.

- **الإتاحة العالية (Uptime %99.5)**: نسبة إتاحة 99.5% تعني أن النظام قد يكون غير متاح لمدة لا تزيد عن 44 ساعة في السنة بأكملها. هذا المستوى من الموثوقية يضمن أن النظام سيكون جاهزاً للعمل في أي لحظة، ليلاً أو نهاراً، لدعم عمليات المراقبة المستمرة واتخاذ القرارات الحرجة دون انقطاع.

### 4.3.4- قابلية التوسع (Scalability)

شبكة الاتصالات هي كيان حي ينمو ويتوسع باستمرار. النظام الذي يعمل بكفاءة اليوم مع 10,000 إنذار يومياً قد ينهار غداً إذا زاد العدد إلى 20,000.

- **التعامل مع البيانات المتزايدة**: يجب تصميم قاعدة البيانات والبنية البرمجية للنظام بحيث يمكنها استيعاب النمو المستقبلي في حجم البيانات وعدد المستخدمين. هذا يعني أن إضافة

مواقع شبكة جديدة أو زيادة عدد الإنذارات لن يؤدي إلى تباطؤ النظام أو انهياره، مما يضمن أن يكون الاستثمار في هذا المشروع طويل الأمد.

### 5.3.4- سهولة الاستخدام (Usability)

النظام القوي تقنياً ولكنه معقد للاستخدام هو نظام فاشل. يجب تصميم الواجهات لتناسب طبيعة عمل كل مستخدم.

- **بساطة تطبيق الموبايل**: الفني في الميدان يعمل في ظروف صعبة وقد يكون تحت ضغط الوقت. يجب أن يكون التطبيق مصمماً ليسمح له بالحصول على المعلومة التي يحتاجها وتحديث حالة التذكرة بأقل عدد ممكن من النقرات، مع واجهة واضحة تعمل بشكل جيد حتى على شاشات الهواتف الصغيرة.
- **وضوح لوحة التحكم**: على عكس الفني، يحتاج مشغل الـ NOC إلى رؤية كمية كبيرة من المعلومات في وقت واحد. يجب تصميم لوحة التحكم لعرض البيانات بكثافة ولكن بطريقة منظمة، باستخدام الألوان والرسوم البيانية والخرائط لتسليط الضوء على المعلومات الأكثر أهمية وتمكينه من فهم "الصورة الكبيرة" بسرعة.

## 4.4-البنية المعمارية للنظام (System Architecture)

تم تصميم النظام وفقاً **للبنية المعمارية ثلاثية الطبقات .(Three-Tier Architecture)** هذا النموذج هو معيار صناعي لتطوير التطبيقات القوية والقابلة للتطوير، حيث يتم فصل النظام إلى ثلاث طبقات منطقية رئيسية ومستقلة: طبقة العرض، طبقة المنطق، وطبقة البيانات. هذا الفصل الدقيق بين المسؤوليات يضمن أن يكون النظام مرناً، قابلاً للتوسع، وسهل الصيانة.

### 1.4.4- الطبقة الأولى: طبقة المنطق والتطبيقات(Logic/Application Tier)

هذه هي الطبقة المركزية التي تحتوي على "عقل" النظام، حيث تتم معالجة جميع البيانات وتنفيذ منطق العمل .(Business Logic) تتكون هذه الطبقة من:

- **محرك الذكاء الاصطناعي(AI Engine)** : هو مكون متخصص ضمن هذه الطبقة .يتكون من نماذج تعلم الآلة المدربة مسبقاً .(XGBoost, Random Forest) عندما تتطلب عملية ما تحليلاً ذكياً، تقوم الواجهة الخلفية باستدعاء هذا المحرك لتنفيذ مهام التصنيف والتنبؤ.
- **الواجهة الخلفية(Backend Server - Flask API)** : هو الخادم المركزي الذي تم بناؤه باستخدام إطار العمل **Flask**فيPython .

  يعمل كوسيط ومنسق لجميع العمليات. هو المسؤول عن إدارة المستخدمين، التحقق من الصلاحيات، معالجة الطلبات القادمة من طبقة العرض، والتواصل مع قاعدة البيانات.

### 2.4.4- الطبقة الثانية: طبقة البيانات(Data Tier)

هذه هي طبقة تخزين البيانات الدائمة للنظام.(Persistence Layer)

- **قاعدة البيانات (Database - MySQL)** هي المسؤولة عن تخزين واسترجاع وإدارة جميع بيانات النظام بشكل آمن وفعال.

**وظيفتها** :يتم الوصول إلى هذه الطبقة حصراً من خلال طبقة المنطق .(Backend) هذا العزل يضمن حماية البيانات وتناسقها، حيث لا يمكن للواجهات الأمامية الوصول مباشرة إلى قاعدة البيانات، مما يمثل مبدأ أمان أساسياً.

### 3.4.4- الطبقة الثالثة: طبقة العرض(Presentation Tier)

هذه هي الطبقة التي يتفاعل معها المستخدم مباشرة، وهي مسؤولة عن عرض واجهة المستخدم (UI) وجمع المدخلات. في هذا النظام، تتكون هذه الطبقة من مكونين:

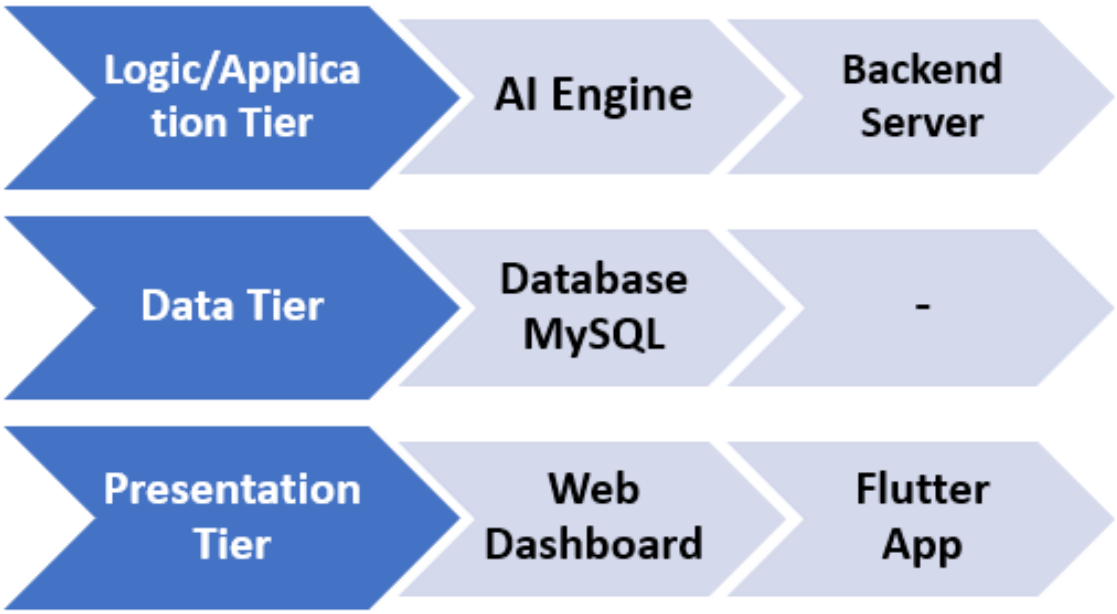
- **لوحة تحكم الويب(Web Dashboard)** : تطبيق ويب تفاعلي تم بناؤه باستخدام (HTML, CSS, JavaScript)وهو مخصص لمشغلي مركز عمليات الشبكة.(NOC)
- **تطبيق الموبايل(Flutter App)** : تطبيق يعمل على نظام أندرويد تم بناؤه باستخدام إطار العمل **Flutter**، وهو مخصص للفرق الفنية الميدانية.

**وظيفتها** :هذه الطبقة لا تحتوي على أي منطق برمجي معقد .وظيفتها الأساسية هي إرسال طلبات (API Requests)إلى طبقة المنطق بناءً على تفاعل المستخدم، واستقبال الردود (JSON data) لعرضها بشكل مرئي ومفهوم من خلال الجداول والرسوم البيانية والخرائط.

### 4.4.4-مزايا هذه البنية المعمارية

هذا الفصل الدقيق بين الطبقات يوفر مزايا استراتيجية مهمة:

- **الفصل بين الاهتمامات(Separation of Concerns)** : كل طبقة مسؤولة عن مهمة محددة، مما يجعل تطوير كل جزء واختباره وصيانته أسهل بكثير.
- **قابلية التوسع المستقل(Independent Scalability)** : يمكن زيادة موارد الخوادم في طبقة المنطق للتعامل مع عدد أكبر من المستخدمين، دون التأثير على طبقة البيانات أو طبقة العرض.
- **المرونة التكنولوجية** :يمكن إعادة بناء واجهة الويب الأمامية بالكامل باستخدام تقنية مختلفة في المستقبل، طالما أنها لا تزال تتواصل مع نفس الـ API الذي توفره الواجهة الخلفية، دون الحاجة لتغيير أي شيء في منطق النظام.
- **الأمان المعزز** :يوفر الهيكل حدود أمان طبيعية بين الطبقات، وأهمها حماية قاعدة البيانات من أي وصول مباشر من الإنترنت.




الشكل(4-2) البنية المعمارية للنظام

# الفصل الخامس

# التقنيات البرمجية المستخدمة

## 1.5- لغة البرمجة $python$

هي لغة برمجة، عالية المستوى سهلة التعلم مفتوحة المصدر قابلة للتوسيع، تعتمد أسلوب البرمجة غرضية التوجه $(oop)$ .

هي لغة مُفسَّرة، ومُتعدِدة الاستخدامات، وتستخدم بشكل واسع في العديد من المجالات، كبناء البرامج المستقلة باستخدام الواجهات الرسومية وفي تطبيقات الويب، ويمكن استخدامها كلغة برمجة نصية للتحكم في أداء العديد من البرمجيات مثل بلندر.

بشكل عام، يمكن استخدام بايثون لعمل البرامج البسيطة للمبتدئين، ولإنجاز المشاريع الضخمة في الوقت نفسه، غالباً ما يُنصح المبتدؤون في ميدان البرمجة بتعلم هذه اللغة لأنها من بين أسرع اللغات البرمجية تعلماً.

طُوِّرت بايثون في معهد الرياضيات والمعلوماتية الهولندي $(CWI)$ في مدينة أمستردام على يد جايدو فان روسم في أواخر ثمانينات القرن العشرين، وكان أول إعلان عنها في عام 1991 م، كُتبت نواة اللغة باستعمال لغة سي، أطلق روسم الاسم "بايثون" على لغته تعبيرًا عن إعجابه بفِرقَة مسرحية هزلية شهيرة من بريطانيا، كانت تطلق على نفسها اسم مونتي بايثون.

تتميز بايثون بمجتمعها النشط، كما أن لها الكثير من المكتبات البرمجية ذات الأغراض الخاصة.

الشكل (5-1) شعار شركة بايثون .

## 2.5 -إطار العمل فلاسك Flask framework

إطار العمل Flask هو عبارة عن إطار عمل مصغر أو ما يسمى بالـ Micro Framework يتم استخدامه في لغة البرمجة بايثون من أجل بناء مواقع وتطبيقات الويب البسيطة. إذ يتوفر على حزمة من الأدوات والـ Functions التي تساعدك على تحقيق مجموعة من الـ Tasks في عملية إنتاج موقع وتطبيق ويب. يعتبر Flask إطار سهل للغاية مقارنة مع إطارات أخرى نظيرة مثل Django .

إطار Flask وسهولته يجعله مثالياً لمن ولج للتو للتطوير بلغة البرمجة، ولديه آفاق وآمال لصناعة مواقع الويب وتطبيقات الويب بهذه اللغة البرمجية. إن تهيئة تطبيقات ومواقع الويب باستخدامه سهل

للغاية، ويكفي فقط أن تكون لديك خبرة بسيطة في لغة بايثون، وبعض الخبرة في تقنيات الويب المختلفة ( HTML, CSS, JS ... ).

يدعم Flask بدوره كل المكتبات والإطارات المستخدمة في تقنيات الويب، وهذا يجعل إمكانية إدراجها في برمجية متكاملة أمر سهل للغاية.

يتميز Flask بحزمة إيجابيات منها:

- ✓ أحد أسهل الإطارات للمبتدئين: مقارنة مع إطارات أخرى في البايثون نفسها مثلDjango ، او إطارات أخرى غير إطارات البايثون، فإن تعلم Flask سهل للغاية. اذ لا يأتي بتلك الصعوبة التي تتخيلها، وسهولته تجعلك تود استكشاف إطارات أخرى واستخدامها في المستقبل .
- ✓ دعم كامل لكل مكتبات البايثون المختلفة: إن البايثون مثل الجافاسكربت، تعتمد بشكل كبير على مكتبات (libs) في مجالات مختلفة. يمنحك إطار Flask إمكانية دمج أي مكتبة عبر مدير الحزم الخاص بها pip واستخدامها في الويب .
- ✓ قادر على التطور لاستخدامه في البرمجيات المعقدة

هذه تقريبا أهم الإيجابيات و لا يمتلك Flask الكثير من السلبيات غير كونه محدود نوعاً ما مقارنة مع أقرانه، وهذا ما يجعله أساسا سهل للمبتدئين .

الشكل (5-2) شعار إطار العمل **Flask**.

## 3.5 - لغات تطوير الويب

### 1.3.5 - لغة ترميز النص التشعبي HTML

تُعد **HTML** العمود الفقري لأي موقع ويب. وظيفتها الأساسية هي تحديد **هيكل ومحتوى** الصفحة. تخيل صفحة الويب كجسم الإنسان، فإن HTML هي الهيكل العظمي الذي يحدد أجزاءه.

- **المحتوى :**العناوين، الفقرات، الصور، الروابط، القوائم، والجداول.
- **الوظيفة :**تنظيم المحتوى بشكل منطقي ووضع كل عنصر في مكانه الصحيح.
- **مثال :**وسم <p>يُستخدم لإنشاء فقرة، ووسم <img>لإضافة صورة.

### 2.3.5-صفحات الأنماط المتتالية CSS

تُستخدم **CSS**لتحديد **شكل ومظهر** الموقع. إذا كانت HTML هي الهيكل، فإن CSS هي الملابس والمكياج التي تجعل الهيكل يبدو جذابًا.

- **المظهر :**الألوان، الخطوط، الخلفيات، المسافات، وحجم العناصر.
- **الوظيفة :**تنسيق العناصر التي تم إنشاؤها بواسطة HTML لجعلها تبدو جميلة ومنظمة.
- **مثال :**يمكن لـ CSS تغيير لون نص معين، أو حجم الخط، أو وضع عنصر في منتصف الصفحة.

### JavaScript (JS) -3.3.5

تُعد **JavaScript**اللغة المسؤولة عن إضافة **التفاعل والديناميكية** إلى الموقع. إذا كانت HTML هي الهيكل و CSSهي المظهر، فإن JavaScript هي التي تمنح الموقع "الحياة" وتجعله يتفاعل مع المستخدم.

- **التفاعل :**التحقق من صحة البيانات في النماذج، إنشاء تأثيرات حركية، وعرض وإخفاء العناصر.
- **الوظيفة :**الاستجابة لأفعال المستخدم (مثل النقر على زر)، تحديث محتوى الصفحة دون الحاجة لإعادة تحميلها، والتواصل مع الخادم.
- **مثال :**عندما تنقر على زر في صفحة ويب لتغيير الصورة المعروضة، فإن JavaScript هي التي تنفذ هذا الإجراء.

## *Visual Studio Code* -4.5

هو محرر شيفرات مجاني و مفتوح المصدر "MIT License" مطور من قبل Microsoft موجه للعديد من الانظمة ."$Windows, Linux\ and\ macOS$" يدعم هذا المحرر العديد من اللغات ويحتوي على عدد كبير من الاضافات التي ستسهل عليك العمل من كتابة وتحليل الكود.

في عام 2018 حصل على المركز الأول في استبيان $stackoverflow$ كأكثر بيئة تطوير مستخدمة بين المطورين.

يدعم الكثير من اللغات مثل $JavaScript, TypeScript, CSS, HTML, Python$ولغات اخرى عديدة ويمكنك دعم لغات اكثر عن طريق VS Code Marketplace وكما يوفر توثيق للتعامل مع بعض اللغات يمكن ان تستكشفها في التوثيق الرسمي للمحرر، هذا المحرر يجمع بعض بيانات الاستخدام ويرسلها الى Microsoft لتحسين المنتج والخدمات المقدمة، يمكن تعطيل جمع البيانات في صفحة الأسئلة الشائعة.

الشكل (5-3) شعار تطبيق **Visual Studio Code**.

# 5.5 - قواعد البيانات و MySQL

## 1.5.5 -قواعد البيانات DataBase

قاعدة البيانات هي مجموعة منظّمة من المعلومات المهيكلة أو البيانات المخزّنة عادةً بصيغة إلكترونيّة أو في نظام كمبيوتر. عادةً ما تكون قاعدة البيانات تحت تحكم نظام إدارة قاعدة بيانات (DBMS). ومعاً تتم الإشارة إلى البيانات ونظام إدارة قواعد البيانات جنباً إلى جنب مع التطبيقات المرتبطة بهما باعتبارها نظام قواعد بيانات وغالباً ما يتم اختصاره إلى قاعدة بيانات فقط.
وعادةً ما تتم صياغة البيانات ضمن الأنواع الأكثر شيوعاً من قواعد البيانات المستعملة اليوم على هيئة صفوف وأعمدة في سلسلة من الجداول لإضفاء الفاعلية على المعالجة والاستعلام عن البيانات. ويمكن حينئذٍ الوصول إلى البيانات وإدارتها وتعديلها وتحديثها والتحكم فيها وتنظيمها بسهولة تستخدم غالبية قواعد البيانات لغة الاستعلام المركّبة (SQL) لكتابة البيانات والاستعلام عنها.
تتطلب قاعدة البيانات عادةً برنامج قواعد بيانات شاملاً يُعرف باسم نظام إدارة قواعد البيانات يعمل نظام DBMS كواجهة بين قواعد البيانات ومستخدميها النهائيين أو البرامج، ما يسمح للمستخدمين باسترداد المعلومات وتحديثها وإدارة طريقة تنظيم المعلومات وتحسينها. يسهّل نظام DBMS كذلك مراقبة قواعد البيانات والتحكم فيها، مما يمكّن مجموعة متنوعة من العمليات الإدارية، مثل مراقبة الأداء والضبط والنسخ الاحتياطي والاسترداد.

بعض نماذج برامج قواعد البيانات المعروفة أو أنظمة DBMS تشمل MySQL و Microsoft Access وMicrosoft SQL Server وFileMaker Pro وOracle Database.

## 2.5.5 -قواعد البيانات MySQL

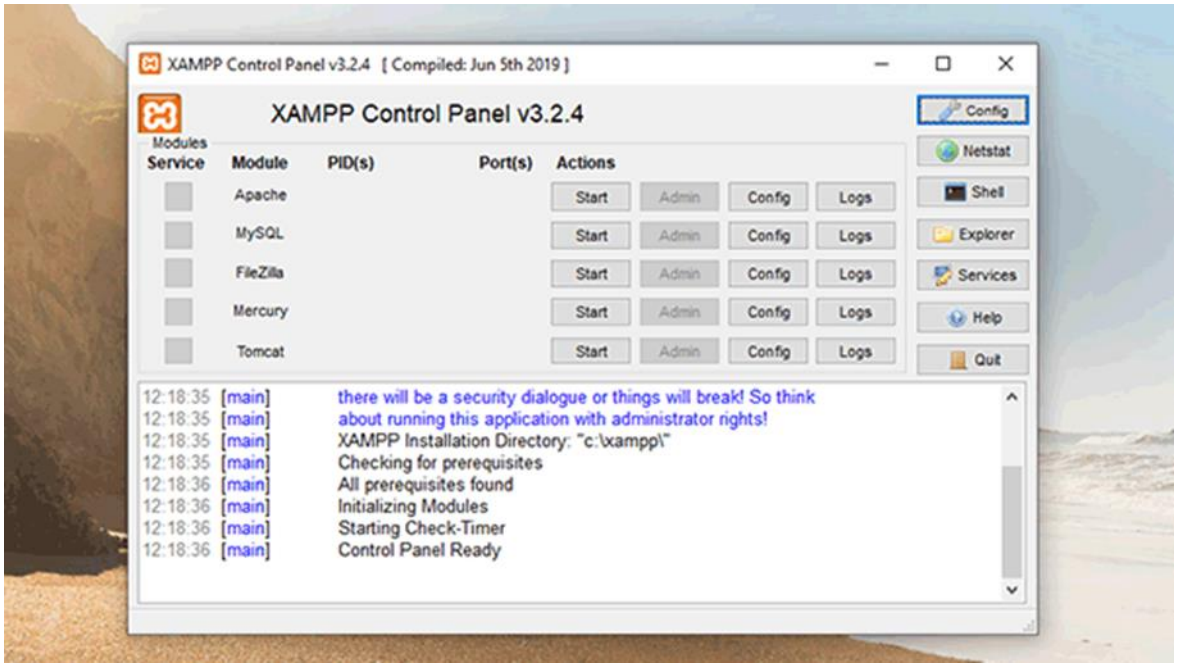
MySQL هو نظام مفتوح المصدر لإدارة قواعد البيانات الترابطية ويستند إلىSQL ، وقد تم تصميمه وتحسينه لتطبيقات الويب ويمكنه العمل على أي نظام أساسي. وعندما نشأت متطلبات جديدة ومختلفة مع ظهور الإنترنت، أصبح MySQL النظام الأساسي المفضل لمطوري الويب والتطبيقات المستندة إلى الويب. ولأنه مصمم لمعالجة ملايين الاستعلامات وآلاف المعاملات، يمثل MySQL خيارًا شائعاً لشركات التجارة الإلكترونية التي تحتاج إلى إدارة العديد من التحويلات المالية. المرونة حسب الطلب هي الميزة الرئيسة لـMySQL .

و MySQL هو نظام DBMS الذي يمثل الأساس لأشهر مواقع الويب والتطبيقات المستندة إلى الويب في العالم، بما في ذلك Airbnb و Uberو LinkedInو Facebookو Twitter.

يتم التعامل مع قواعد بيانات MySQL باستخدام برنامج MySQL work Branch وتتم كتابة الأوامر البرمجية فيه باستخدام لغة SQL.

## 6.5 -برنامج XAMPP

XAMPPهو حل خادم ويب مجاني ومفتوح المصدر، يتيح لك إعداد خادم محلي على جهاز الكمبيوتر الخاص بك بسهولة لتطوير واختبار مواقع الويب. اسم XAMPP هو اختصار لمكوناته الأساسية **X** :منصة متعددة **A**pache، **M**ariaDB، **P**HP، و**P**erl.




الشكل (5-4) تطبيق ***XAMPP***

### 1.6.3- وظيفة كل مكون

- **Xمنصة متعددة :**يعني هذا أن XAMPP يمكن استخدامه على أنظمة تشغيل مختلفة، بما في ذلك ويندوز، ماك أو إس، ولينكس.
- **Apache:**هذا هو برنامج خادم الويب. يتولى معالجة الطلبات من متصفحات الويب ويرسل لها صفحات الويب والملفات التي تحتاجها. فكر فيه كمحرك أساسي يخدم ملفات موقعك.
- **MariaDB:**هذا هو نظام إدارة قواعد البيانات. إنه نسخة مطورة من MySQL ويُستخدم لتخزين وإدارة بيانات موقعك، مثل معلومات المستخدمين، تفاصيل المنتجات، أو منشورات المدونات.

- **PHP:** هذه لغة برمجة نصية من جانب الخادم. تُستخدم لإنشاء صفحات وتطبيقات ويب ديناميكية. على سبيل المثال، يقوم PHP بمعالجة النماذج، والتفاعل مع قاعدة البيانات، وتوليد المحتوى بشكل فوري.
- **Perl:** هذه لغة برمجة أخرى تُستخدم غالبًا في البرمجة النصية ومهام إدارة الأنظمة. يتم تضمينها في XAMPP لتوفير وظائف إضافية للمطورين.

### 2.6.5-استخدام XAMPP

يستخدم المطورون XAMPP لإنشاء بيئة اختبار محلية، تُعرف أيضًا باسم خادم **localhost**.

يسمح لهم هذا ببناء واختبار مواقع الويب على أجهزتهم الخاصة دون الحاجة إلى رفع الملفات إلى خادم ويب حقيقي. هذا مفيد جدًا من أجل:

1. **التطوير دون اتصال:** يمكنك العمل على موقع الويب حتى بدون اتصال بالإنترنت.
2. **الاختبار السريع:** يمكن اختبار التغييرات فورًا دون تأخيرات رفع الملفات.
3. **بيئة آمنة:** يمكنك تجربة ميزات جديدة، إضافات، أو قوالب دون المخاطرة بإتلاف موقع ويب مباشر.

توفر لوحة تحكم XAMPP واجهة بسيطة لتشغيل وإيقاف الخدمات المختلفة Apache, MariaDB, إلخ والوصول إلى إعداداتها.

## Flutter-7.5

Flutterهو إطار عمل (Framework) و مجموعة أدوات لتطوير واجهات المستخدم UI Toolkit من إنتاج شركة جوجل. يسمح للمطورين ببناء تطبيقات جميلة وعالية الأداء للهواتف المحمولة iOS وAndroid ، الويب، وأجهزة سطح المكتب، كل ذلك باستخدام قاعدة كود واحدة (Single Codebase)ولغة برمجة واحدة هي Dart.

ببساطة، بدلاً من كتابة تطبيق منفصل لنظام أندرويد وآخر لنظام آيفون، يمكنك مع Flutter كتابته مرة واحدة فقط، وسيعمل على كلا النظامين بكفاءة تامة.

الشكل (5-5) شعار ***Flutter***

# 8.5- المكتبات المستخدمة

المكتبة $Library$ في لغة البرمجة؛ هي عبارة عن مجموعة من الدوال الجاهزة التي يمكن استيرادها في الكود، وتقدر الإحصائيات عدد مكتبات لغة البايثون بنحو 137000 مكتبة في مختلف المجالات.

## 18..5- مكتبة Pandas

تحليل البيانات هو عملية جمع وتنظيم وتحليل البيانات لاكتشاف الأنماط واتخاذ القرارات. أصبحت تحليل البيانات مهمة بشكل متزايد في مجموعة متنوعة من المجالات، بما في ذلك الأعمال التجارية والحكومة والعلوم. تتوفر العديد من المكتبات لتحليل البيانات، بما في ذلك Pandas و NumPy و SciPy.

تُعد Pandas واحدة من المكتبات الأكثر شيوعًا لتحليل البيانات، لأنها توفر أدوات قوية لقراءة وكتابة ومعالجة البيانات. مكتبة Pandas AI هي مكتبة Python تضيف إمكانات الذكاء الاصطناعي التوليدي generative artificial intelligence إلىPandas ، مما يجعل إطارات البيانات data framesحواريةconversational

مكتبة Pandas AI هي مكتبة Python تدمج إمكانات الذكاء الاصطناعي التوليدي فيPandas ، مما يجعل إطارات البيانات data frames حوارية .conversational تسمح مكتبة Pandas AI للمستخدمين بطرح أسئلة على مجموعة البيانات الخاصة بهم بلغة طبيعية، وستجيب المكتبة على هذه الأسئلة باستخدام الذكاء الاصطناعي. على سبيل المثال، يمكنك طرح الأسئلة التالية

تستخدم مكتبة Pandas AI مجموعة متنوعة من تقنيات الذكاء الاصطناعي، بما في ذلك:

- التعلم الآليmachine learning
- المعالجة اللغوية الطبيعيةnatural language processing
- التعلم العميقdeep learning

تجعل هذه التقنيات من الممكن للمكتبة فهم أسئلة المستخدمين بشكل صحيح والإجابة عليها بدقة.

تقدم مكتبة Pandas AI العديد من نقاط القوة، بما في ذلك:

- سهولة الاستخدام: تُعد مكتبة Pandas AI سهلة الاستخدام نسبيًا، حتى بالنسبة للمبتدئين في مجال تحليل البيانات.
- القوة: توفر مكتبة Pandas AI مجموعة متنوعة من الأدوات القوية لتحليل البيانات، بما في ذلك التعلم الآلي والمعالجة اللغوية الطبيعية والتعلم العميق.
- الحداثة: تُعد مكتبة Pandas AI مكتبة جديدة نسبيًا، ولكنها تتطور باستمرار لإضافة ميزات ووظائف جديدة. كيفية استخدام مكتبةPandas AI

تُعد مكتبة Pandas AI أداة قوية لتحليل البيانات باستخدام الذكاء الاصطناعي. مع استمرار تطور المكتبة، من المرجح أن تصبح أداة أكثر أهمية لتحليل البيانات في مجموعة متنوعة من المجالات.

### 2.8.5 -مكتبة scikit-learn (sklearn)

- **الوصف :**هي المكتبة الأساسية والأكثر شمولاً لتعلم الآلة في .Python توفر أدوات لكل مراحل بناء النموذج، من تجهيز البيانات إلى تقييم الأداء.
- **أهم توابع المكتبة:**

  - **model_selection.train_test_split**: تقوم بتقسيم البيانات إلى مجموعة للتدريب وأخرى للاختبار، وهي خطوة ضرورية للتأكد من أن النموذج يعمل بشكل جيد على بيانات لم يرها من قبل.
  - **preprocessing.LabelEncoder**: تحوّل البيانات النصية إلى أرقام، لأن النماذج الرياضية لا تفهم النصوص.
  - **feature_extraction.text.TfidfVectorizer:**أداة ذكية لتحويل النصوص (مثل اسم العطل) إلى متجهات رقمية. تقوم بحساب أهمية كل كلمة، مما يساعد النموذج على فهم معنى النص.
  - **ensemble.RandomForestClassifier / RandomForestRegressor** نماذج قوية تُستخدم في المشروع لمهمتين:
    - **RandomForestClassifier:** للتصنيف، مثل توقع العطل التالي (Next Fault).
    - **RandomForestRegressor**: للتنبؤ بقيمة رقمية، مثل توقع زمن حل العطل .(Resolution Time)
  - **linear_model.LogisticRegression:** نموذج تصنيف كلاسيكي، يُستخدم هنا كأحد النماذج المرشحة للمقارنة واختيار الأفضل.
  - **multioutput.MultiOutputClassifier**: أداة رائعة تسمح باستخدام نماذج التصنيف العادية (التي تتوقع شيئًا واحدًا) للتنبؤ بعدة أشياء في نفس الوقت (مثل توقع التقنية، المسؤول، والأولوية معًا).
  - **metrics.classification_report**: تُستخدم لتقييم أداء نماذج التصنيف وإعطاء مقاييس دقيقة مثل (F1-score) لمعرفة مدى جودة النموذج.

### 3.8.5- xgboost (xgb)

- **الوصف :**مكتبة متقدمة وعالية الأداء، تشتهر بسرعتها ودقتها في حل مسائل التصنيف والانحدار. غالبًا ما تحقق أفضل النتائج في المسابقات والمشاريع المعقدة.
- **دورها في المشروع :**تُستخدم كأحد النماذج القوية المرشحة (XGBClassifier) ضمن حلقة المقارنة لاختيار النموذج الأفضل لمهمة التصنيف متعددة المخرجات.

### 4.8.5-مكتبات الأدوات المساعدة وحفظ النماذج joblib

- **الوصف :**مكتبة بسيطة وفعالة، استخدامها الرئيسي هنا هو حفظ الكائنات البرمجية (مثل نماذج تعلم الآلة المدربة) في ملفات على جهازك.
- **دورها في المشروع :**بعد تدريب النماذج، تقوم بحفظها (best_multi_model.pkl, resolution_time_predictor2.pkl, etc.)حتى يمكن استخدامها لاحقًا في تطبيق الويب للتنبؤ دون الحاجة لإعادة التدريب في كل مرة.

### 5.8.5 -مكتبة Numpy

سُميّت بهذا الاسم اختصاراً للمصطلح *Numerical Python*، وهي أساسية في علوم البيانات وهناك العديد من مكتبات علم البيانات وتعلم الآلة تعتمد اعتماداً قوياً عليها، وتتيح قدرة التعامل مع المصفوفات بطريقة أفضل من Lists الموجودة تلقائياً كشكل من أشكال تراكيب البيانات في بايثون.

تتمثل هذه الأفضلية التي تُقدمها *Numpy* على Lists في أن المصفوفات في مكتبة *Numpy* تكون متجانسة، كما أنها سريعة في عمليات القراءة والكتابة وتُعتبر أكثر كفاءة.
أحد أهم الأدوات في المكتبة هي المصفوفة ذات الأبعاد المتعددة والكفاءة الفائقة *High Performance Multidimensional Array*، والتي نستطيع أن نُنفذ من خلالها الكثير من الدوال الرياضية والعمليات الحسابية والعمل على حل العديد من المشاكل.

تتيح هذه المكتبة أيضاً عمليات ال Index, Slice, Subset.

### 6.5.5- مكتبة OS

تساعد على التعامل مع نظام التشغيل ومعرفة معلومات عن نظام التشغيل، كالتعامل مع المجلدات من حيث الانشاء والنسخ والحذف والتعامل مع الملفات في نظام التشغيل والتعديل عليها.

### 7.5.5 -مكتبات الأمان والتشفير werkzeug.security

- **الوصف :**هي جزء من مكتبة Werkzeug التي يعتمد عليهاFlask ، وتوفر أدوات مهمة للأمان، خصوصًا للتعامل مع كلمات المرور.
- **دورها في المشروع:**
  - **generate_password_hash**: تُستخدم عند إضافة مستخدم جديد لتشفير (Hashing)كلمة المرور الخاصة به قبل حفظها في قاعدة البيانات. هذا يضمن عدم تخزين كلمات المرور كنص عادي، مما يزيد من أمانها.
  - **check_password_hash**: تُستخدم عند محاولة تسجيل الدخول للتحقق من أن كلمة المرور التي أدخلها المستخدم تتطابق مع النسخة المشفرة المحفوظة في قاعدة البيانات، دون الحاجة إلى فك تشفيرها

## 8.5.5- مكتبة sqlalchemy

- **الوصف العام :**هي مجموعة أدوات SQL قوية جداً ومكتبة لربط الكائنات بالعلاقات (ORM)في .Python إنها توفر طريقة برمجية مرنة وفعالة للتفاعل مع قواعد البيانات بدلاً من كتابة استعلامات SQL كنصوص عادية، مما يزيد من الأمان ويحسن تنظيم الكود.
- **دورها في المشروع:**

- **create_engine:** هي نقطة البداية لأي تطبيق يستخدم .SQLAlchemy تُستخدم في دالة get_db_engineلإنشاء "محرك اتصال" بقاعدة بيانات .MySQL. هذا المحرك يدير الاتصالات بقاعدة البيانات بكفاءة.
- **text وbindparam:**تُستخدمان معاً لإنشاء استعلامات SQL آمنة. في دالة run_prediction، يتم استخدامهما لبناء استعلام يجلب معرفات (IDs) الإنذارات الجديدة من قاعدة البيانات. هذه الطريقة تحمي التطبيق من هجمات حقن SQL (SQL Injection)عن طريق فصل الاستعلام عن البيانات التي يتم إدخالها فيه.

## Leaflet.js-9.5.5

Leaflet.jsهي مكتبة JavaScript مفتوحة المصدر، تُعتبر من أشهر وأسهل الأدوات المستخدمة لبناء خرائط تفاعلية على الويب. تتميز بأنها خفيفة الوزن، سريعة، وسهلة الاستخدام، وتعمل بكفاءة على كل من أجهزة الكمبيوتر والهواتف المحمولة.

مكتبة Leaflet.jsهي الأداة المثالية التي يمكنك استخدامها في واجهتك الأمامية لعرض هذه المواقع كنقاط على خريطة تفاعلية، مما يسهل على المستخدمين تصور أماكن الأعطال والمواقع المختلفة.

- **أهم مميزات Leaflet.js**

البساطة وخفة الوزن :حجم المكتبة صغير جداً (حوالي 42 كيلوبايت)، مما يجعلها سريعة التحميل ولا تبطئ من أداء صفحة الويب.
سهولة الاستخدام :واجهتها البرمجية (API) بسيطة وواضحة جداً. يمكنك إنشاء خريطة وإضافة علامات (Markers) عليها ببضعة أسطر من الكود.
مرونة عالية :تدعم العديد من الإضافات (Plugins) التي توسع من قدراتها، مثل:
تجميع العلامات :(Marker Clustering) لتجميع النقاط المتقاربة في نقطة واحدة عند التصغير.
الخرائط الحرارية :(Heatmaps) لإظهار كثافة البيانات في مناطق معينة.
أدوات الرسم :للسماح للمستخدم برسم أشكال على الخريطة.
دعم مصادر خرائط متعددة :يمكنك استخدامها مع مختلف مزودي الخرائط .
مفتوحة المصدر :مجانية بالكامل ولها مجتمع ضخم من المطورين الذين يساهمون في تطويرها وتقديم الدعم.

# الفصل السادس
# تدريب نماذج الذكاء الاصطناعي
# Train AI mdodels

## 1.6-تحليل البيانات

### 1.1.6-استيراد البيانات والتحليل الأولي

تم تحميل بيانات تاريخية حقيقية من ملف CSV ، حيث يحتوي الملف على الأعطال لمدة 5 أيام وتقدر بحوالي 340 ألف عطل مختلفة الأسباب والأنواع ويتحوي على الأعمدة التالية:

- Root alarm - نوع الإنذار (Root alarm أو -)
- Severity - مستوى الخطورة (Major, Minor, Critical)
- Alarm ID - معرف الإنذار (أرقام مثل 21807، 25600)
- Name - اسم الإنذار (مثل " OML Fault", "Monitoring Device Maintenance Link Failure")
- NE Type - نوع عنصر الشبكة (GBTS, BSC6900 GU, BTS3900)
- SiteCode - رمز الموقع (مثل SWD072, DAM005)
- Alarm Source - مصدر الإنذار
- MO Name - اسم الكائن المُدار
- Location Information - معلومات الموقع التفصيلية
- Additional Information - معلومات إضافية
- Occurred On (NT) - وقت حدوث الإنذار
- Cleared On (NT) - وقت إزالة الإنذار
- Acknowledged On (ST) - وقت الإقرار بالإنذار
- Acknowledged By - من أقر بالإنذار
- RRU Name - اسم وحدة الراديو البعيدة
- BBU Name - اسم وحدة الشريط الأساسي
- eNodeB ID - معرف العقدة الإلكترونية
- Log Serial Number - الرقم التسلسلي للسجل
- User Label - تسمية المستخدم
- Equipment Alarm Serial Number - الرقم التسلسلي لإنذار المعدات
- Maintenance Status - حالة الصيانة (NORMAL)
- Subnet - الشبكة الفرعية (ROOT/BSS, ROOT/BSS/BTS3900)

ثم قمنا بعرض أولى الصفوف وتحليل البيانات باستخدام أوامر head( ) وinfo( ) وdescribe( ) للحصول على تصور عام حول البنية، الأنواع، والقيم الفارغة.

| | Severity | Alarm ID | Name | NE Type | SiteCode | Alarm Source | NO Name | Location Information | Additional Information | ... | Acknowledged On (ST) | Acknowledged By | MMI Name | eNodeB ID | Log Serial Number | User Label | Equipment Alarm Serial Number | Maintenance Status | Subnet |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Root alarm | Major | 21807 | CNAL Fault | GBTS | SNDO72 | HESC07_SwitchWheel_SNDO72 | Wheel_SNDO72 | Site Holder=92 BSC Subrack No.=1 BSC Shl No.... | ... | 9202026 10:05 | <System operator> | - | - | - | 79400108 | - | 16921080 | NORMAL | ROOTBSS |
| 1 | - | Major | 25800 | Monitoring Device Maintenance Link Failure | GBTS | DAM005 | HESC10_YourCity_DanielAbdallahRoad_DAM005 | DanielAbdallahRoad_DAM005 | Cabinet No.=0, Subrack No.=7, Slot No.=0, Boar... | RAT_INFO=GSM, AFFECTED_RAT=GSM, DID=1006 | 9202026 8:59 | <System operator> | - | - | - | 79400107 | - | 20180837 | NORMAL | ROOTBSS |
| 2 | - | Major | 25800 | Monitoring Device Maintenance Link Failure | GBTS | DAM005 | HESC10_YourCity_DanielAbdallahRoad_DAM005 | DanielAbdallahRoad_DAM005 | Cabinet No.=0, Subrack No.=7, Slot No.=0, Boar... | RAT_INFO=GSM, AFFECTED_RAT=GSM, DID=1006 | 9202026 8:59 | <System operator> | - | - | - | 79400213 | - | 20180840 | NORMAL | ROOTBSS |
| 3 | - | Major | 25800 | Monitoring Device Maintenance Link Failure | GBTS | DAM005 | HESC10_YourCity_DanielAbdallahRoad_DAM005 | DanielAbdallahRoad_DAM005 | Cabinet No.=0, Subrack No.=7, Slot No.=0, Boar... | RAT_INFO=GSM, AFFECTED_RAT=GSM, DID=1006 | 9202026 8:59 | <System operator> | - | - | - | 79400254 | - | 20180841 | NORMAL | ROOTBSS |
| 4 | - | Major | 25800 | Monitoring Device Maintenance Link Failure | GBTS | DAM075 | HESC02_ThamarDAM_AbdelHkim_DAM075 | DAM_AbdelHkim_DAM075 | Cabinet No.=0, Subrack No.=7, Slot No.=0, Boar... | RAT_INFO=GSM, AFFECTED_RAT=GSM, DID=1075 | 9202026 8:59 | <System operator> | - | - | - | 79400275 | - | 11797742 | NORMAL | ROOTBSS |
| 5 | Root alarm | Minor | 21641 | SCTP Link Fault | BSC6900 GU | SNDO01 | HESC07_Swiella | SRVHAVA, SHAVA, SCTPLINKHAVA, SCTP Link ID=1650 | Subrack No.=1, Slot No.=4, Subsystem No.=2, SCTP... | ... | 9202026 10:00 | <System operator> | - | - | - | 79400283 | - | 16921078 | NORMAL | ROOTBSS |
| 6 | Root alarm | Minor | 21641 | SCTP Link Fault | BSC6900 GU | SNDO01 | HESC07_Swiella | SRVHAVA, SHAVA, SCTPLINKHAVA, SCTP Link ID=1651 | Subrack No.=1, Slot No.=4, Subsystem No.=2, SCTP... | ... | 9202026 10:00 | <System operator> | - | - | - | 79400287 | - | 16921078 | NORMAL | ROOTBSS |
| 7 | - | Major | 21802 | Adjacent Node PAth Address Proj Failure | BSC6910 GU | DAM169 | HESC10_YourCity | Subrack No.=0, Slot No.=9, Adjacent Node PAL... | Adjacent Node ID=NULL, Name of Adjacent Node 3... | ... | 9202026 10:03 | <System operator> | - | - | - | 79400283 | - | 30180846 | NORMAL | ROOTBSS |
| 8 | - | Critical | 65073 | High Temperature | BTS3900 | DAM088 | NBTS_Tehreen_DAM088 | Cabinet No.=0, Subrack No.=0, Slot No.=0, Fra... | AFFECTED_RAT=GSM, DID=NULL, alAd... | RAT_INFO=GSM, ... | 9202026 10:00 | <System operator> | - NBTS_Tehreen_DAM088 | 411088 | 79400289 | Tehreen_DAM088 | 88681 | NORMAL | ROOTBSS:BTS3900 |

الشكل(6-1) لمحة عن البيانات

تمت معالجة القيم المفقودة إما بالتعبئة بقيم افتراضية أو بحذف السجلات غير الضرورية، مع تحويل أعمدة التواريخ إلى النوع datetime لحساب مدة العطل بدقة.

```
Severity                          0
Alarm ID                          0
Name                              0
NE Type                           0
SiteCode                       2246
Alarm Source                      0
MO Name                           0
Location Information              0
Additional Information            0
Occurred On (NT)                  0
Cleared On (NT)                   0
Acknowledged On (ST)              0
Acknowledged By                   0
RRU Name                         16
BBU Name                          0
eNodeB ID                         0
Log Serial Number                 0
User Label                        0
Equipment Alarm Serial Number     0
Maintenance Status                0
Subnet                            1
dtype: int64
```

```
Severity                          0
Alarm ID                          0
Name                              0
NE Type                           0
SiteCode                          0
Alarm Source                      0
MO Name                           0
Location Information              0
Additional Information            0
Occurred On (NT)                  0
Cleared On (NT)                   0
Acknowledged On (ST)              0
Acknowledged By                   0
RRU Name                          0
BBU Name                          0
eNodeB ID                         0
Log Serial Number                 0
User Label                        0
Equipment Alarm Serial Number     0
Maintenance Status                0
Subnet                            0
dtype: int64
```

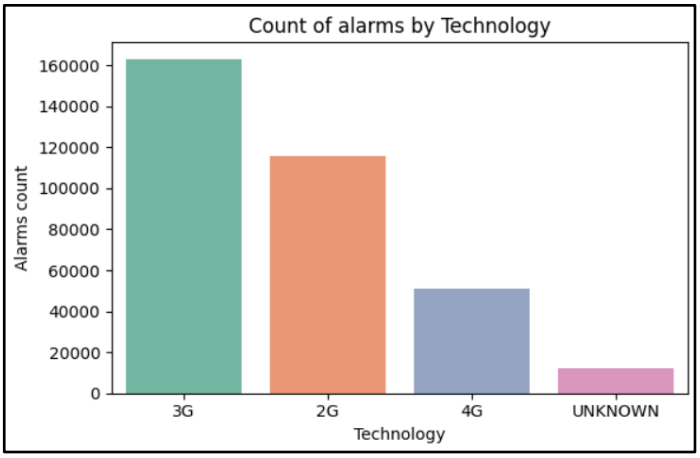
الشكل(6-2) معالجة القيم المفقودة

### 2.1.6-تنقية البيانات وفهمها

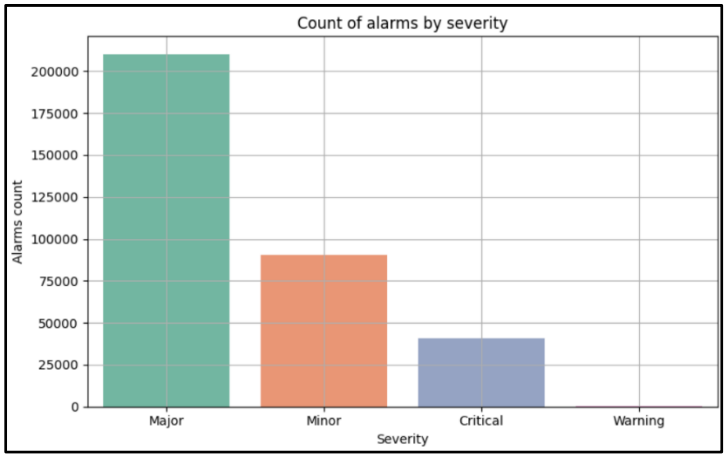
تم تصفية الأعطال ذات التكرار المنخفض لأغراض تعزيز جودة التدريب. ثم أُضيفت أعمدة جديدة مثل:

- Repair Duration :لحساب مدة العطل بالدقائق.
- Technology : تصنيف التكنولوجيا (2G/3G/4G) من خلال تحليل نصي لعدة أعمدة.
- Responsible, Human Intervention, Proposed Solution, priority:من خلال مطابقة الأعطال مع قاعدة معرّفات خارجية تم ادراجها في قاعدة البيانات DataBase باسم alarmstr.

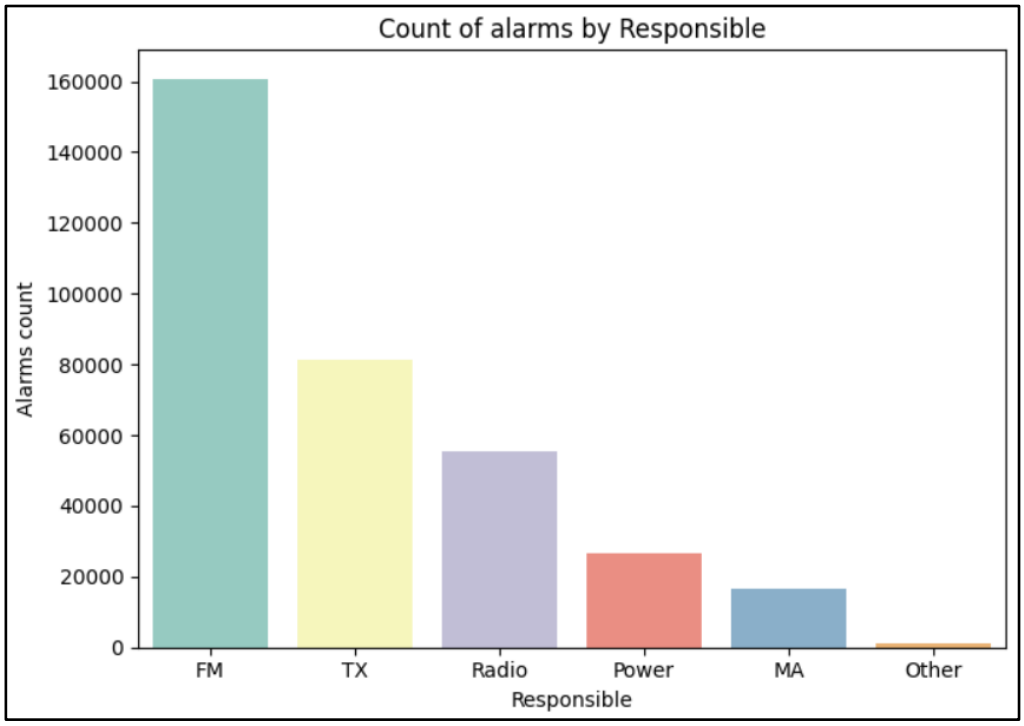
كما تم استخدام الرسوم البيانية (باستخدام مكتبات Matplotlib و Seaborn) لتحليل الأعطال الأكثر تكرارًا حسب الشدة والموقع والتكنولوجيا والقسم المختص.




الشكل(6-4) عدد الأعطال حسب التكنولوجيا




الشكل(6-3) عدد الأعطال حسب الشدة




الشكل(6-5) عدد الأعطال حسب الفريق المختص

### 3.1.6-إعداد البيانات لتعلم الآلة

لتحضير البيانات للنماذج، قمنا بدمج النصوص من عدة أعمدة وصفية
(Name, NE Type, SiteCode, ...)
في عمود جديد موحد باسم text لتكوين وصف شامل لكل عطل. ثم قمنا بالخطوات التالية:

- استخدمنا خوارزمية **TfidfVectorizer** لتحويل البيانات النصية إلى ميزات رقمية، مع تحديد استخلاص أهم **1000خاصية** (max_features=1000)لتمثيل كل عطل.

- تحديد الأعمدة التي سيتم اعتمادها لتدريب النموذج:

كثرة الأعمدة ستؤثر بالسلب على تدريب النموذج لأنها ستزيد من الوقت اللازم للتدريب وستؤدي لأخطاء في التواقع لذلك تم اختيار الأعمدة التالية كونها الأعمدة الأكثر أهمية من الناحية التقنية والعملية :

$$'Name','NE\ Type','SiteCode','MO\ Name','Alarm\ Source',$$

$$'Severity','Location\ Information','Additional\ Information'$$

- حددنا الأعمدة المستهدفة 5 أعمدة مستهدفة لمشكلة التصنيف Classification تشمل :
  - نوع التكنولوجيا Technology
  - المسؤول Responsible
  - مدى الحاجة لتدخل بشري Human Intervention
  - الحل المقترح Proposed Solution
  - الأولوية Priority
- تم ترميز القيم النصية في الأعمدة المستهدفة إلى أرقام باستخدام **LabelEncoder**.

```
df['text'] = df[['Name', 'NE Type', 'SiteCode',
...]].fillna('-').agg(' '.join, axis=1)

vectorizer = TfidfVectorizer(max_features=500)

X = vectorizer.fit_transform(df['text'])
```

- أما عمود **مدة الإصلاح Repair Duration** ، فقد تم التعامل معه كمشكلة **انحدار** (Regression)منفصلة، نظراً لطبيعته الرقمية المستمرة.

## 2.6-بناء نماذج التعلم الآلي

تم تقسيم البيانات إلى بيانات للتدريب وبيانات للاختبار بنسبة 20/80، ومن ثم تم تدريب ثلاثة نماذج رئيسية:

- **Random Forest**
- **Logistic Regression**
- **XGBoost**(بشكل فردي لكل مُخرج)

واستخدمنا MultiOutputClassifier لدعم التصنيف متعدد الأهداف في النماذج التي تتطلب ذلك. تم تقييم كل نموذج بناءً على متوسط مقياس F1 لكل هدف على حدة، وتم اختيار النموذج الأفضل حسب الجدول التالي:

| | Random Forest | LogisticRegression | XGBoost |
|---|---|---|---|
| Technology | 0.9993 | 0.9984 | 0.998 |
| Responsible | 1.00 | 0.998 | 0.99 |
| Human Intervention | 1.00 | 0.99 | 1 |
| Proposed Solution | 1.00 | 0.998 | 1 |
| priority | 1.00 | 0.9987 | 1 |
| Repair Duration | 0.36 | 0.3406 | 0.38 |
| **Avg.** | **0.8932** | **0.89113** | **0.8947** |

الجدول (6-1) مقارنة نتائج الخوارزميات

من الجدول السابق نستنتج:

- جميع النماذج قدمت أداءً ممتازاً ومتقارباً جداً في التنبؤ بالأهداف الخمسة الأولى، مما يُظهر أن البيانات جيدة التصنيف، وخاصة بعد معالجة النصوص واستخدامTF-IDF .
- أداء **Random Forest**و **XGBoost**كان شبه مثالي ($F1 \approx 1.0$) في معظم الأهداف.
- التحدي الرئيسي كان في توقع **مدة الإصلاح Repair Duration** ، بسبب التوزيع غير المتوازن للبيانات.
- يمكن تحسين دقة "مدة الإصلاح" باستخدام تقنيات مثل تحويلها إلى مشكلة انحدار (Regression)أو استخدام تقنيات موازنة الفئات مثلSMOTE .
- تم اعتماد القيم الحالية حيث أن مدة العطل تعتمد على عوامل خارجية كثيرة يصعب قياسها، مثل سرعة استجابة فريق الصيانة.

بناءً على الأداء القوي وقابليته للتطوير، **تم اختيار نموذج XGBoost** ليكون النموذج الأساسي للمشروع. يعود هذا الاختيار لعدة أسباب فنية وعلمية تجعله الأنسب لبيانات أعطال الشبكات الخلوية:

- **الدقة العالية والقدرة على التعلم من الأخطاء :**يعتمد XGBoost على آلية **التعزيز المتدرج (Gradient Boosting)**، حيث يقوم ببناء النماذج بشكل تسلسلي، وكل نموذج جديد يحاول تصحيح أخطاء النموذج الذي سبقه. هذه المنهجية تؤدي إلى دقة عالية يصعب تحقيقها بالنماذج الأخرى.
- **الكفاءة وسرعة الأداء :**صُمم XGBoost ليكون سريعاً وفعالاً في استغلال موارد النظام (المعالج والذاكرة)، مما يجعله مثالياً للتعامل مع مجموعات البيانات الكبيرة والمعقدة التي تنتج عن بيانات الشبكات.
- **المرونة في ضبط النموذج لكل هدف :**على الرغم من أننا احتجنا لتدريب نموذج XGBoostمنفصل لكل مُخرج (لكل هدف من الأهداف الستة)، إلا أن هذه الميزة أتاحت لنا **ضبط المعلمات (Hyperparameters)** بشكل دقيق ومخصص لكل مهمة على حدة، مما يزيد من قوة وموثوقية كل تنبؤ.

- **مقاومة فطرية للمُبالغة في التعميم(Overfitting) :** يحتوي XGBoost على أدوات مدمجة وقوية لتجنب الإفراط في ملاءمة النموذج لبيانات التدريب، مثل **تنظيم L1 و L2 (Regularization)**، مما يضمن قدرة النموذج على التعميم على بيانات جديدة لم يرها من قبل.
- **القدرة على تفسير النتائج :**يوفر النموذج إمكانية استخراج **أهمية الميزات Feature) (Importance**، مما يساعد على فهم العوامل الأكثر تأثيراً في تحديد نوع العطل وأولويته، وهو أمر حيوي لتحسين العمليات التشغيلية.

## 3.6-تدريب نموذج لتوقع العطل التالي

تم تدريب نموذج لتنبؤ العطل التالي **(Next Fault Prediction)**، حيث تم بناء نموذج تنبؤي جديد باستخدام RandomForestClassifierلتوقع نوع العطل التالي اعتمادًا على التسلسل الزمني للأعطال في نفس الموقع. تم ترميز الأعطال بشكل تسلسلي، ثم تدريب النموذج وتقييمه باستخدام classification_reportوحصلنا على **دقة (Accuracy) بقيمة 70 .%**تعتبر هذه القيمة جيدة نظراً لأن توزع الأعطال غير متناسق، وبهدف تجنب حصول overfitting للنموذج المولّد .

```
sequence_df = pd.DataFrame(sequence_data,
columns=['CurrentFault', 'NextFault'])

model.fit(X_train, y_train)
```

ويُستخدم النموذج في توقع العطل التالي المحتمل بعد عطل معين:

```
predict_next_fault("Power Failure") → "Battery Door Open"
```

## 4.6-اختيار النموذج المناسب

نموذج XGBoost تم اختياره لأنه يجمع بين الدقة، والمرونة، والقدرة على التعامل مع بيانات معقدة ومتعددة الأبعاد، وهو مناسب جدًا لتصنيف أنواع الأعطال وتوقع الحلول في بيئة شبكات الاتصالات.

وفي توقع العطل التالي تم اختيار Random Forest.

تم حفظ النماذج بصيغة joblib لاستخدامه لاحقاً، وتم عمل API يجهز العطل المولد(سيتم التحدث عنه لاحقاً) ليتم ادخاله لنموذج الذكاء الاصطناعي وتوقع الأعمدة الهدف target columns.

### 1.4.6-اختبار النماذج

1. **تحميل النماذج والأدوات الجاهزة**

في البداية، يتم تحميل كل المكونات الضرورية التي تم إنشاؤها وحفظها خلال مرحلة التدريب. هذه المكونات هي:

- xgboost_models_per_column.pkl:مجموعة نماذج **XGBoost**، حيث يوجد نموذج مخصص لكل خاصية مطلوب التنبؤ بها (مثل القسم المسؤول، الحل المقترح، الأولوية).
- tfidf_vectorizer.pkl **(Vectorizer)**:لتحويل النصوص الوصفية للأعطال إلى أرقام (متجهات) يمكن للنماذج فهمها.
- label_encoders.pkl:مجموعة أدوات **(Encoders)**لترجمة التنبؤات الرقمية التي تنتجها النماذج مرة أخرى إلى نصوص مفهومة (مثلاً، تحويل الرقم 2إلى كلمة Highفي خانة الأولوية).
- next_fault_predictor_model.pkl: النموذج الخاص بالتنبؤ **بالعطل التالي** في السلسلة.
- resolution_time_predictor.pkl:النموذج الجديد المخصص لتوقع **زمن حل العطل** بالدقائق نموذج انحدارRegression .

2. **تحميل بيانات الاختبار وتحضيرها**

- يقرأ الكود ملف alarmstr.csv للحصول على بيانات حقيقية يمكن اختبار النماذج عليها.
- ينظف البيانات بملء أي قيم فارغة بعلامة .

3. **أخذ عينة عشوائية للتجربة**

- بدلاً من تشغيل التنبؤ على كل البيانات، يختار الكود **عدداً محدداً من الأعطال بشكل عشوائي** (sample(n=1))من البيانات المحضّرة ليكون حالة الاختبار.

**4. تنفيذ عملية التنبؤ المتكاملة**

لكل عطل في العينة ، يقوم الكود بالآتي:

1. **تحضير المُدخلات** :يأخذ النص الوصفي للعطل ويستخدم loaded_vectorizerلتحويله إلى صيغة رقمية.
2. **التنبؤ بالخصائص المتعددة**:
   - يمر على الخصائص المستهدفة (Technology, Responsible, Priority,..)
   - يستخدم نموذج XGBoost المناسب لكل خاصية للتنبؤ بقيمتها.
   - يستخدم loaded_encoders لترجمة النتيجة الرقمية إلى نص واضح مثال Predicted Priority: High
3. **التنبؤ بزمن حل العطل** :يستخدم نموذج resolution_time_model لتوقع المدة التي سيستغرقها حل العطل بالدقائق.
4. **التنبؤ بالعطل التالي** :يستدعي دالة predict_next_fault التي تأخذ اسم العطل الحالي وتستخدم نموذج next_fault_modelلتوقع العطل التالي المحتمل حدوثه في نفس الموقع.

**5. عرض النتائج**

في النهاية، يطبع الكود جميع التنبؤات التي توصل إليها بشكل منظم وواضح في الشاشة، ليرينا كيف استطاعت النماذج المختلفة تحليل العطل من جميع جوانبه.

```
Predicting on 1 random alarms:

--- Alarm: Local Cell Unusable (BTS3900) ---
  Predicted Technology: 2G
  Predicted Responsible: Radio
  Predicted Human Intervention: Yes
  Predicted Proposed Solution: Reboot affected unit and update software.
  Predicted Priority: high
  Predicted Resolution Time: 99.93 minutes
  Predicted Next Fault: Local Cell Unusable
------------------------
```

الشكل(6-6) اختبار نماذج الذكاء الاصطناعي

# 5.6-توليد إنذارات متناسبة مع البيانات التاريخية Proportional Alarm Generator

في سياق تطوير أنظمة ذكية قادرة على فهم وتحليل أعطال الشبكات الخلوية، كان من الضروري وجود بيانات اختبارية قابلة للاستخدام في تدريب النماذج، مع الحفاظ على نفس الخصائص الإحصائية للبيانات الحقيقية. ولتلبية هذا الغرض، قمنا بتطوير أداة برمجية بلغة Python تقوم بتوليد إنذارات مصطنعة مستندة إلى النسب الواقعية في البيانات التاريخية، بهدف ضمان الواقعية والتوازن في البيانات المُدخلة للنماذج.

## 1.5.6- الفكرة العامة لتوليد الإنذارات

الأداة عبارة عن مولد ذكي للإنذارات يقوم بقراءة البيانات التاريخية الحقيقية من ملف CSV وتحليل الأنماط الإحصائية المختلفة مثل:

- توزيع مستويات الخطورة(Severity)
- توزيع أسماء الإنذارات(Alarm Name)
- توزيع أنواع عناصر الشبكة(NE Type)
- نسبة الإنذارات الجذرية(Root Alarm)

ثم يتم استخدام هذه النسب لتوليد عينة جديدة من الإنذارات الحقيقية تتوافق مع نفس التوزيع الإحصائي، بحيث تكون أقرب ما يمكن للواقع.

## 2.5.6-المكونات الرئيسية للأداة

1. **تحميل البيانات وتحليل الأنماط**

- يتم تحميل البيانات من ملف البيانات التاريخية .
- يتم التأكد من صحة الصفوف وعدد الأعمدة.
- يتم تحليل الأعمدة الأساسية (Severity, Name, NE Type) واستخراج نسب التكرار باستخدام مكتبة collections.Counter.
- يتم حساب نسبة الإنذارات من نوع "Root alarm" بشكل خاص.

2. **توليد عينة من الإنذارات**

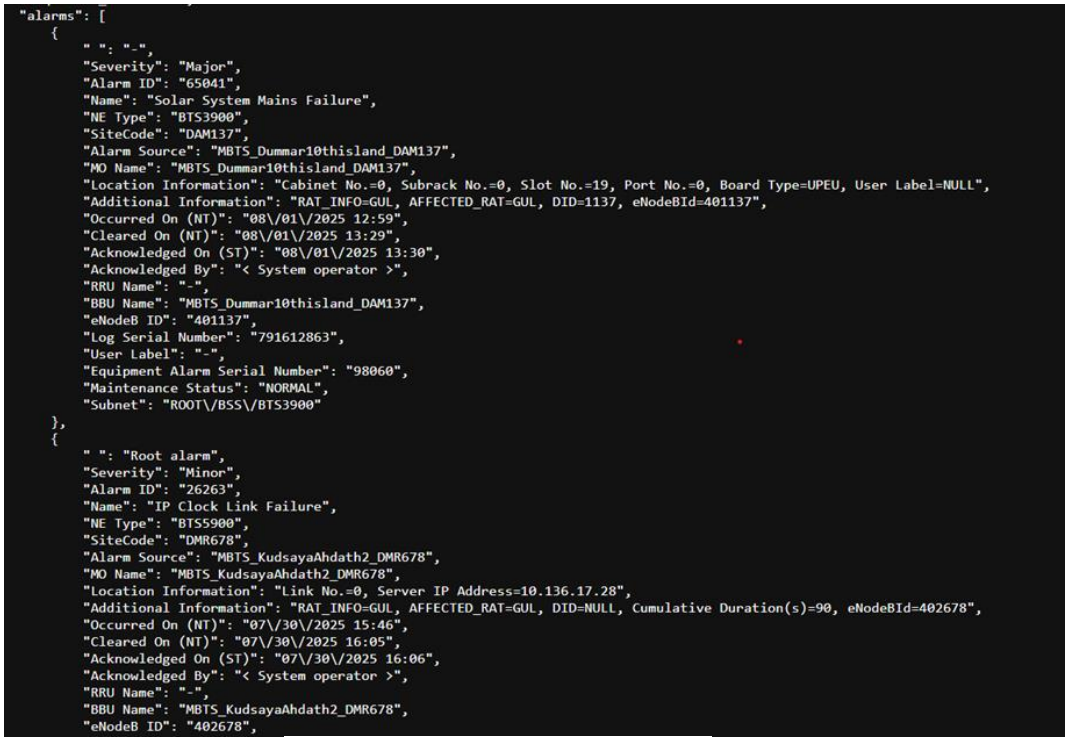
- يتم تحديد عدد الإنذارات المطلوب توليدها (مثل 10 أو 100).
- يتم حساب عدد الإنذارات المطلوبة لكل مستوى خطورة بناءً على النسب التاريخية.
- يتم اختيار إنذارات حقيقية من نفس المستوى بشكل عشوائي من البيانات المحملة.
- يتم تحديث الطوابع الزمنية لكل إنذار ليبدو حديثًا (وقته الحالي).
- يتم توليد معلومات إضافية تشمل:
  - الطابع الزمني للظهور، الإزالة، والإقرار.
  - معلومات عن الموقع(Site Info)

- **3.3حفظ النتائج**

- يتم حفظ الإنذارات المولدة في قاعدة البيانات في جدول باسم Alarms لتتم معالجتها لاحقاً.
- يتم عرض إحصائيات عن الإنذارات الجديدة، مثل توزيع الخطورة وأسماء الإنذارات.
- تم تطوير API باستخدام Flask لربط أداة توليد الإنذارات مع النظام لاحقاً

## 3.5.6- مزايا أداة توليد الإنذارات

- **الواقعية** :لا تعتمد الأداة على توليد بيانات عشوائية بالكامل، بل تعتمد على بيانات حقيقية.
- **التحكم بالتوزيع** :تحافظ الأداة على نفس نسب مستويات الخطورة والتكرار الموجودة في الواقع.
- **المرونة** :يمكن التحكم بحجم العينة، نوع الفلترة، وعدد الإنذارات المطلوب.
- **سهولة التصدير والتحليل** :تُصدر الأداة النتائج ضمن قاعدة البيانات لتكون جاهز للاستخدام في النماذج أو التحليلات.



```
"alarms": [
    {
        " ": "-",
        "Severity": "Major",
        "Alarm ID": "65041",
        "Name": "Solar System Mains Failure",
        "NE Type": "BTS3900",
        "SiteCode": "DAM137",
        "Alarm Source": "MBTS_Dummar10thisland_DAM137",
        "MO Name": "MBTS_Dummar10thisland_DAM137",
        "Location Information": "Cabinet No.=0, Subrack No.=0, Slot No.=19, Port No.=0, Board Type=UPEU, User Label=NULL",
        "Additional Information": "RAT_INFO=GUL, AFFECTED_RAT=GUL, DID=1137, eNodeBId=401137",
        "Occurred On (NT)": "08\/01\/2025 12:59",
        "Cleared On (NT)": "08\/01\/2025 13:29",
        "Acknowledged On (ST)": "08\/01\/2025 13:30",
        "Acknowledged By": "< System operator >",
        "RRU Name": "-",
        "BBU Name": "MBTS_Dummar10thisland_DAM137",
        "eNodeB ID": "401137",
        "Log Serial Number": "791612863",
        "User Label": "-",
        "Equipment Alarm Serial Number": "98060",
        "Maintenance Status": "NORMAL",
        "Subnet": "ROOT\/BSS\/BTS3900"
    },
    {
        " ": "Root alarm",
        "Severity": "Minor",
        "Alarm ID": "26263",
        "Name": "IP Clock Link Failure",
        "NE Type": "BTS5900",
        "SiteCode": "DMR678",
        "Alarm Source": "MBTS_KudsayaAhdath2_DMR678",
        "MO Name": "MBTS_KudsayaAhdath2_DMR678",
        "Location Information": "Link No.=0, Server IP Address=10.136.17.28",
        "Additional Information": "RAT_INFO=GUL, AFFECTED_RAT=GUL, DID=NULL, Cumulative Duration(s)=90, eNodeBId=402678",
        "Occurred On (NT)": "07\/30\/2025 15:46",
        "Cleared On (NT)": "07\/30\/2025 16:05",
        "Acknowledged On (ST)": "07\/30\/2025 16:06",
        "Acknowledged By": "< System operator >",
        "RRU Name": "-",
        "BBU Name": "MBTS_KudsayaAhdath2_DMR678",
        "eNodeB ID": "402678",
```


الشكل(6-7) الإنذارات المولدة

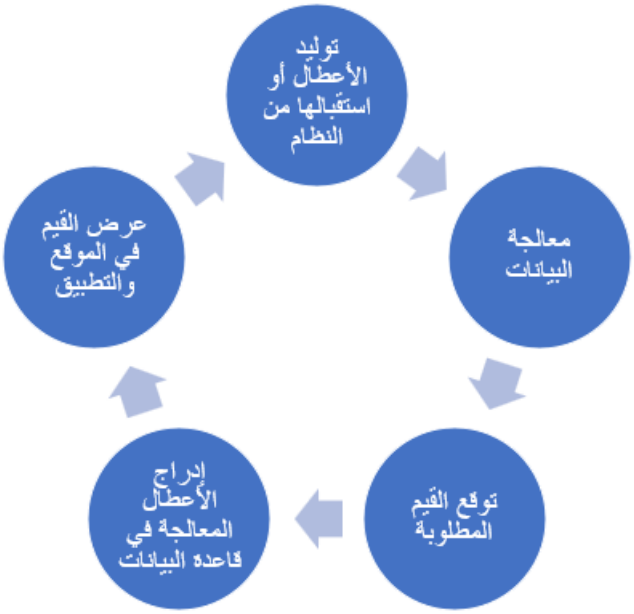
## 6.6-ربط الذكاء الاصطناعي مع توليد الأعطال

بعد توليد الأعطال يتم معالجتها بنفس الطريقة التي تحدثنا عنها سابقاً (اختبار النماذج)،وبعدها يتم إضافتها إلى قاعدة البيانات في جدول alarms_ai ليتم استدعاؤها في موقع الويب.

```
--- Predicting for Alarm: NE Is Disconnected (ID: 770) ---
  Predicted Technology: UNKNOWN
  Found value from alarmstr for Responsible: FM
  Predicted Responsible: FM
  Found value from alarmstr for HumanIntervention: Yes
  Predicted HumanIntervention: Yes
  Found value from alarmstr for ProposedSolution: Check configuration and reset unit
  Predicted ProposedSolution: Check configuration and reset unit
  Found value from alarmstr for Priority: high
  Predicted Priority: high
  Predicted Resolution Time: 49.65 minutes
  Predicted Next Fault: SCTP Link Fault

--- Saving new predictions and creating tickets ---
Successfully created database engine.
Successfully saved 10 new predictions to 'alarms_ai'.
```

الشكل(6-8) معالجة وإدراج الأعطال الجديدة




الشكل(6-9) آلية عمل محرك الذكاء الاصطناعي AI Engine

# الفصل السابع

# تصميم الموقع الالكتروني

# وتطبيق الأندرويد

## 1.7-تصميم قاعدة بيانات نظام إدارة أعطال الشبكةNOC

تم تصميم قاعدة البيانات mtn_noc_system لتكون المحور الأساسي لنظام ذكي متكامل لإدارة ومراقبة أعطال شبكات الاتصالات في مركز عمليات الشبكةNOC .

يهدف التصميم إلى أتمتة عملية تحليل الإنذارات، وتوجيه فرق الصيانة بشكل فعال، والتنبؤ بالأعطال المستقبلية، مما يساهم في تقليل زمن انقطاع الخدمة Downtime ورفع كفاءة العمليات.

تتكون قاعدة البيانات من سبعة جداول رئيسية مترابطة، لكل منها دور محدد في دورة حياة العطل، وهي كالتالي:

### 1.1.7-الهيكل التفصيلي للجداولTable Schema

**1. جدول الإنذارات الخام alarms**

هذا الجدول هو نقطة البداية، حيث يستقبل ويخزن جميع الإنذارات الواردة مباشرة من الشبكة مثل BTS, NodeB, eNodeB بصورتها الأولية أو من نظام توليد الأعطال الذكي.

- **أهم الحقول**

alarm_id **:**معرف فريد، nameاسم الإنذار، severity:خطورة الإنذار Critical, Major, Minor، site_code رمز الموقع، و occurred_on تاريخ ووقت الحدوث.

- **دوره :**يعمل كمستودع للبيانات الخام والتاريخية التي ستخضع للمعالجة والتحليل لاحقاً.

**2.جدول القاعدة المعرفية alarmstr**

يعمل هذا الجدول كقاعدة معرفية ثابتة Knowledge Base تحتوي على الإجراءات القياسية والحلول المقترحة لكل نوع من الإنذارات المعروفة مسبقًا لتساعد في عملية تدريب النموذج.

- **أهم الحقول :**

name اسم الإنذار، Responsibleالفريق المسؤولFM, TX, Radio :، ProposedSolutionالحل المقترح، و Priorityالأولوية.

- **دوره :**يوفر دليلاً مرجعياً للنظام لمساعدة الذكاء الاصطناعي أثناء التدريب.

**3. جدول الإنذارات المعالَجة alarms_ai**

هذا هو الجدول المحوري الذي يتم فيه تطبيق الذكاء الاصطناعي. يتم فيه تخزين النتائج بعد معالجتها من نموذج الذكاء الاصطناعي.

- **أهم الحقول:**

alarm_id يرتبط بالإنذار الأصلي، Technology تكنولوجيا الشبكة: 2G, 3G, 4G، Responsibleالفريق المسؤول المقترح، ProposedSolution الحل المقترح، RepairDuration الوقت المتوقع للإصلاح، و NextFault العطل التالي المتوقع.

- **دوره :**تحويل الإنذارات الخام إلى بيانات ذكية قابلة للاستخدام، جاهزة لإنشاء بطاقة عمل وتوجيهها للفريق المختص.

**4. جدول المواقع sites**

يحتوي هذا الجدول على معلومات جغرافية وتقنية ثابتة حول جميع مواقع الشبكة.

- **أهم الحقول:**

site_id رمز الموقع، site_name اسم الموقع، city المدينة، technology التكنولوجيا، و longitude/latitude الإحداثيات.

- **دوره :**إثراء بيانات الإنذار بمعلومات الموقع الجغرافي، مما يساعد في تحديد مكان العطل بدقة وتوجيه فرق الصيانة.

**5.جدول بطاقات الأعطال tickets**

يمثل هذا الجدول نظام التذاكر Ticketing System الذي يتتبع دورة حياة العطل من لحظة إنشائه وحتى إغلاقه.

- **أهم الحقول:**

id رقم البطاقة، alarm_ai_id يرتبط بالإنذار المعالج، status حالة البطاقة Open, In Progress, Closed، و assigned_to_user_id الموظف المسؤول.

- **دوره :**تنظيم ومتابعة عمليات الصيانة وضمان حل جميع الأعطال المسجلة.

**6. جدول المستخدمين users**

يُستخدم لإدارة حسابات وصلاحيات مستخدمي النظام، سواء كانوا مشغلين في مركز المراقبة أو فنيين ميدانيين.

- **أهم الحقول:**

idمعرف المستخدم، username اسم المستخدم، roleالدور : admin, operator, technician، و departmentالقسم.

- **دوره :**تنظيم الوصول إلى النظام وتحديد صلاحيات كل مستخدم، وربط بطاقات الأعطال بالفنيين المسؤولين.

**7. دول الإنذارات المتوقعةpredicted_alarms**

هذا الجدول مخصص لتخزين نتائج نماذج التنبؤ المستقبليةPredictive Maintenance ، حيث يقوم النظام بتحليل الأنماط التاريخية وتوقع الأعطال قبل حدوثها.

- **أهم الحقول :**

Predicted_Next_Fault العطل المتوقع، Predicted_Responsible الفريق المسؤول المقترح، و Predicted_Resolution_Time الوقت المقترح للإصلاح.

- **دوره :**تحقيق الصيانة الاستباقية وتقليل الأعطال المفاجئة.

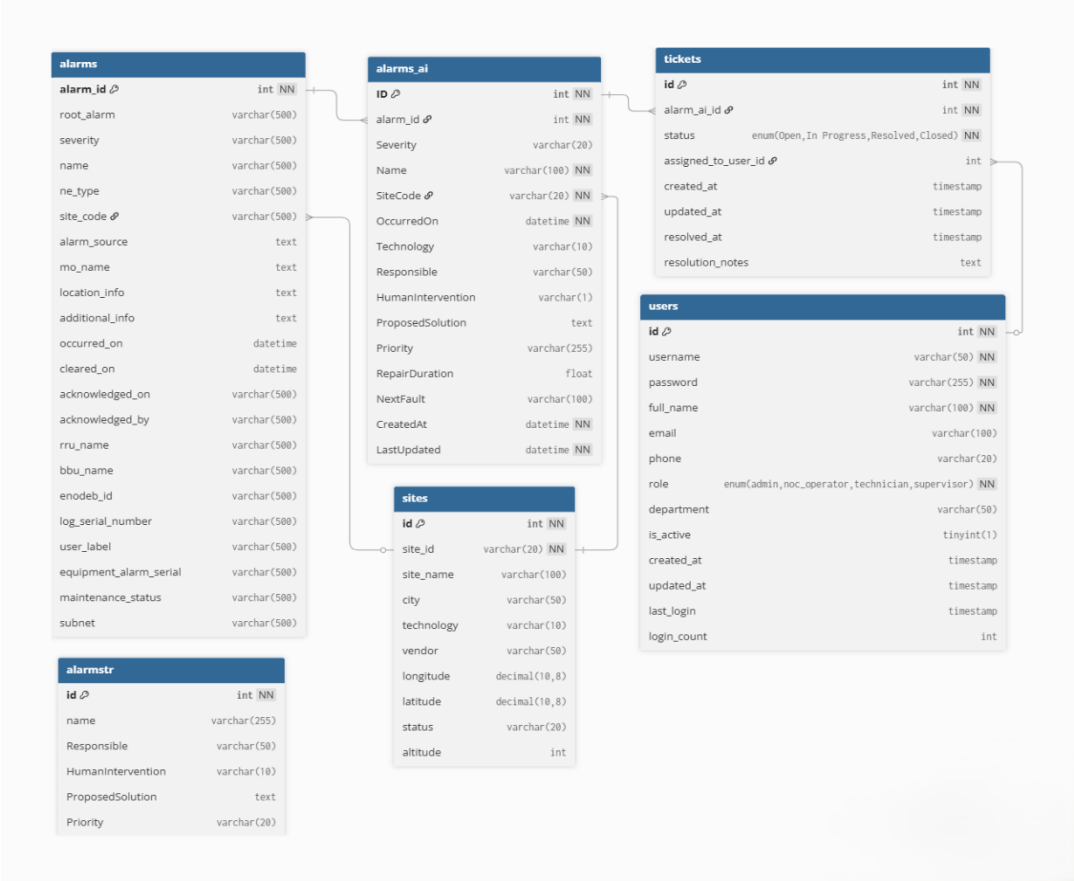
### 2.1.7-آلية عمل النظام المعتمدة على قاعدة البيانات

1. **استقبال الإنذار :**يتم إدراج إنذار جديد في جدول alarms من النظام بشكل مباشر أو من مولد الإنذارات.
2. **المعالجة الذكية :**يقوم النظام بمعالجة الإنذار الجديد، وتحديد التكنولوجيا والفريق المسؤول والحل المقترح بناءً على نماذج الذكاء الاصطناعي، ثم يضيف السجل المعالج إلى alarms_ai.
3. **إنشاء بطاقة عطل :**يتم إنشاء سجل جديد تلقائيًا في جدول tickets مرتبط بالإنذار المعالج في حال كان العطل بحاجة تدخل بشريي.
4. **إسناد ومتابعة :**يتم إسناد البطاقة إلى فني من جدول users، الذي يقوم بتحديث حالتها حتى يتم حل المشكلة.

بهذا التصميم، توفر قاعدة البيانات بنية قوية ومرنة تدعم نظامًا فعالاً لإدارة الأعطال، مع تكامل مميز للذكاء الاصطناعي لتحسين الأداء وتسريع عملية اتخاذ القرار.

### 3.1.7-مخطط ER والعلاقات بين الجداول

يوضح المخطط كيف تترابط أجزاء النظام المختلفة لتتبع الأعطال من لحظة ظهورها حتى حلها:




الشكل(7-1) مخطط ERD

**العلاقات الأساسية:**

1. **المواقع والإنذارات sites و alarms**
   - **العلاقة :**واحد إلى متعدد.One-to-Many
   - **الشرح :**كل موقع sites يمكن أن يصدر عنه العديد من الإنذاراتalarms ، ولكن كل إنذار يتبع لموقع واحد فقط. يتم الربط من خلال حقل site_code
2. **الإنذارات الخام والمعالجة alarms و alarms_ai**
   - **العلاقة :**واحد لواحد.One-to-One
   - **الشرح :**كل إنذار خام alarms يتم تحليله ومعالجته بواسطة النظام الذكي ليتحول إلى سجل واحد فقط في جدول الإنذارات المعالجة .alarms_ai يتم الربط عبر alarm_id.
4. **الإنذارات المعالجة وبطاقات الأعطال alarms_ai و tickets**
   - **العلاقة :**واحد إلى متعدد.One-to-Many
   - **الشرح :**كل إنذار معالج في alarms_aiيؤدي إلى إنشاء بطاقة عطل ticket واحدة على الأقل لتتبعه.
5. **المستخدمون وبطاقات الأعطال users و tickets**
   - **العلاقة :**واحد إلى متعدد.One-to-Many
   - **الشرح :**كل مستخدم فني أو مهندس يمكن أن يُسند إليه العديد من بطاقات الأعطال tickets، بينما كل بطاقة تُسند إلى مستخدم واحد في كل مرة.

يمثل هذا المخطط بنية متكاملة تضمن تدفق البيانات بسلاسة من مرحلة رصد العطل الخام إلى تحليله الذكي ثم تحويله إلى مهمة عملية يتم إسنادها ومتابعتها حتى الحل.

# 2.7-تصميم موقع الويب

## 1.2.7-الخادم Backend

تم تطوير ال backend باستخدام flask حيث يوضح التالي أهم الدوال والتوابع التي تم تعريفها:

- **نظام إدارة المستخدمين**

يعتمد هذا المشروع على نظام إدارة مستخدمين قوي وآمن يتيح للمسؤولين التحكم الكامل في صلاحيات المستخدمين وإدارة حساباتهم. يتجاوز النظام مجرد تسجيل الدخول ليغطي دورة حياة المستخدم الكاملة من الإنشاء إلى الحذف، مع التركيز على الأمان والوظائف الأساسية.

1. **تسجيل الدخول وإدارة الجلسات**

- **تسجيل الدخول(/login) :**

  - عندما يرسل المستخدم بياناته، يقوم النظام بالاتصال بقاعدة البيانات للبحث عن اسم المستخدم.
  - لا يتم تخزين كلمة المرور كنص عادي، بل يتم تشفيرها باستخدام دالة generate_password_hashعند إنشائها. عند تسجيل الدخول، تُستخدم دالة check_password_hashلمقارنة كلمة المرور المدخلة بالتشفير المخزن في قاعدة البيانات، مما يوفر أمانًا عاليًا.
  - بعد التحقق من كلمة المرور وتأكيد أن الحساب **نشط** is_activeيساوي 1، يتم تحديث سجل المستخدم في قاعدة البيانات بآخر وقت تسجيل دخول وزيادة عداد مرات تسجيل الدخول.
  - يتم تخزين معلومات المستخدم الأساسية مثل id, username, roleفي **الجلسة (Session)**. الجلسة هي الطريقة التي يتعرف بها الخادم على المستخدم عبر الطلبات المختلفة، وتظل صالحة حتى يقوم المستخدم بتسجيل الخروج.

2. **إدارة الصلاحيات والأدوار**

- **الديكوراتور @admin_required:**

  - هذه هي إحدى أهم ميزات الأمان في المشروع .الديكوراتور هي دالة خاصة في بايثون تُستخدم لتعديل سلوك دالة أخرى.
  - في هذا السياق، يقوم @admin_requiredبالتحقق قبل تنفيذ أي دالة مرتبطة به (مثل usersأو add_user)من أن المستخدم الحالي لديه صلاحية **admin**.
  - إذا لم يكن لدى المستخدم هذه الصلاحية، يتم إيقاف العملية وإعادة توجيه المستخدم إلى لوحة التحكم (dashboard) مع رسالة تحذير، مما يمنع المستخدمين العاديين من الوصول إلى الصفحات الحساسة.

3. **العمليات على المستخدمين**

تُدار جميع العمليات المتعلقة بالمستخدمين من قبل المسؤول (admin) عبر واجهة مخصصة:

- **عرض المستخدمين:(/users)**

  - هذه الصفحة محمية بديكوراتور .@admin_required
  - تقوم بجلب جميع المستخدمين من قاعدة البيانات وعرضهم في جدول، بما في ذلك معلومات مهمة مثل الاسم الكامل، الدور، البريد الإلكتروني، وتاريخ آخر تسجيل دخول.
  - هذه الميزة تمنح المسؤول نظرة عامة شاملة على جميع الحسابات في النظام.

- **إضافة مستخدم جديد(/users/add) :**
  - تُستخدم هذه الوظيفة لإنشاء حسابات جديدة للمستخدمين.
  - يتم التحقق من أن اسم المستخدم غير موجود بالفعل لتجنب التكرار.
  - يتم تشفير كلمة المرور الجديدة تلقائيًا باستخدام generate_password_hashقبل تخزينها في قاعدة البيانات لضمان الأمان.
- **تعديل مستخدم موجود(/users/edit/<int:id>) :**
  - تسمح هذه الوظيفة للمسؤول بتعديل معلومات المستخدمين الآخرين.
  - يمكنه تغيير الاسم، البريد الإلكتروني، الدور، أو حالة التفعيل.(is_active)
  - إذا تم إدخال كلمة مرور جديدة، يتم تشفيرها وتحديثها في قاعدة البيانات.
- **حذف مستخدم:(/users/delete/<int:id>)**
  - توفر هذه الوظيفة للمسؤول القدرة على حذف الحسابات.
  - تتضمن ميزة أمان إضافية تمنع المسؤول من حذف حسابه الخاص، مما يضمن وجود حساب مسؤول دائمًا في النظام.

**4. الجانب الأمني**

- **تشفير كلمات المرور :**استخدام Werkzeug.securityيمنع هجمات سرقة كلمات المرور حتى في حالة اختراق قاعدة البيانات.
- **إدارة الجلسات :**استخدام os.urandom(24)لإنشاء مفتاح سري عشوائي يمنع هجمات التلاعب بالجلسات.
- **فصل الصلاحيات :**يضمن ديكوراتور admin_required@أن المستخدمين العاديين لا يمكنهم الوصول إلى الوظائف الحساسة، مما يقلل من مخاطر سوء الاستخدام.

**5. نظام البيانات والتحليلات**

لا يقتصر هذا المشروع على إدارة المستخدمين فحسب، بل يمثل نظامًا شاملاً **لجمع وعرض بيانات الشبكة** بطريقة منظمة. الهدف من هذا الجزء من المشروع هو تحويل البيانات الخام (مثل معلومات المواقع والإنذارات) إلى معلومات مفيدة وقابلة للتحليل، لمساعدة فرق التشغيل على اتخاذ قرارات سريعة ومبنية على حقائق.

**1. مصدر البيانات: قاعدة البيانات**

البيانات الأساسية للمشروع تأتي من عدة جداول في قاعدة بياناتMySQL :

- **جدول sites:** يحتوي على معلومات عن كل موقع (برج اتصالات)، مثل رمزه أو اسمه.
- **جدول alarms_ai:** هذا الجدول هو الأكثر حيوية. يخزن بيانات الإنذارات التي تم توليدها.

- SiteCode:يربط الإنذار بموقع معين.
- Name:يصف نوع الإنذار (مثل انقطاع التيار الكهربائي).
- OccurredOn: يحدد متى حدث الإنذار.
- Priority: يحدد أهمية الإنذار (مثل حرجة، عالية، متوسطة).

- **جدول tickets:** يربط كل تذكرة مشكلة بإنذار معين، مما يتيح تتبع حالة حل المشكلة.

**2. نقاط الوصول (API) لجلب البيانات**

البيانات لا تُعرض بشكل مباشر في صفحاتHTML ، بل يتم توفيرها من خلال واجهات برمجة تطبيقات (API) منفصلة. هذا النهج مفيد لأنه:

- **يفصل المنطق عن العرض :**يمكن للواجهة الأمامية باستخدام JavaScript جلب البيانات بشكل ديناميكي دون إعادة تحميل الصفحة بالكامل.
- **يدعم التطبيقات الأخرى :**يمكن لتطبيق جوال أو أي خدمة أخرى استخدام نفس نقاط النهاية لجلب البيانات.

هناك نقطتا وصول رئيسيتان للبيانات:

- **/api/data:** تُستخدم هذه النقطة لتقديم البيانات الأساسية للوحة التحكم. تقوم بجلب معلومات المواقع والإنذارات وعدد التذاكر الإجمالي.
- **/api/tickets:** تُستخدم خصيصًا لعرض بيانات تذاكر المشاكل. تقوم بتنفيذ استعلام معقد (JOIN)لربط تذكرة المشكلة ببيانات الإنذار والمستخدم المسؤول، مما يوفر رؤية شاملة لحالة التذكرة.

**3. التحليل والتصور**

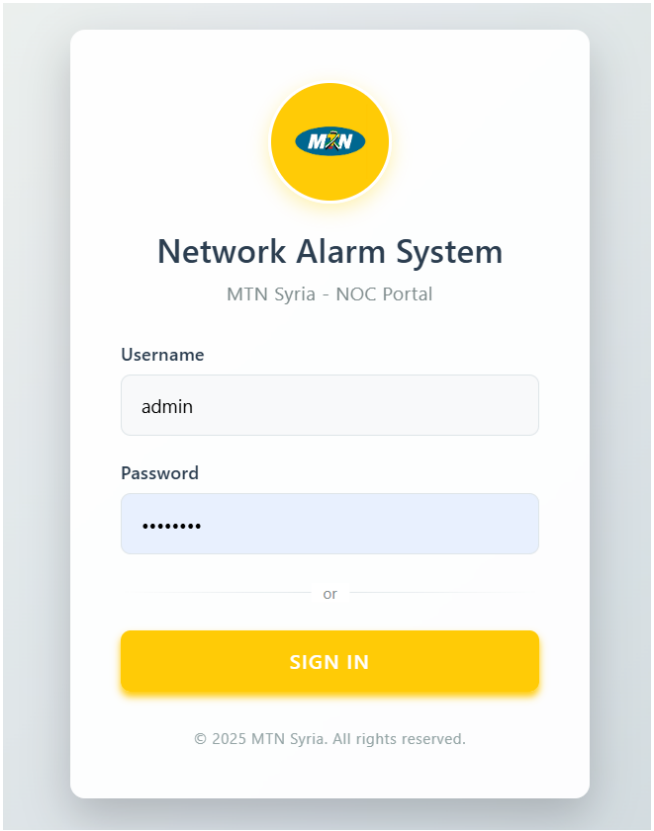
في حين أن الكود يركز على الجانب الخلفي(Backend) ، فإن البيانات التي يجمعها ويقدمها عبر الـ APIهي أساس التحليلات. على سبيل المثال، يمكن استخدام هذه البيانات لبناء:

- **رسوم بيانية :(/charts)** يمكن استخدام مكتبات JavaScript مثل Chart.js أو D3.js لإنشاء رسوم بيانية توضح عدد الإنذارات حسب الأولوية، أو عدد الإنذارات لكل موقع، أو متوسط وقت حل التذكرة.
- **لوحة تحكم تفاعلية :**يمكن لصفحة admin.htmlاستخدام البيانات من /api/dataلتحديث الجداول والعدادات في الوقت الفعلي، مما يمنح المستخدم رؤية حية لحالة الشبكة.

## 2.2.7-تصميم الواجهات frontend

### 1. الصفحة الرئيسية(Login Page)

تُعد هذه الصفحة واجهة المستخدم الأولى للنظام، وهي مصممة لتكون بوابة الدخول الآمنة إلى لوحة التحكم. تعتمد الصفحة على مجموعة من تقنيات الويب الأساسية لتقديم تصميم وظيفي وتفاعلي.




الشكل(7-2) واجهة تسجيل الدخول

❖ **الهيكل والمحتوى(HTML)**

يُشكل كود HTML البنية الأساسية للصفحة، حيث يُحدد العناصر المرئية وترتيبها. . يتضمن ذلك:

- **العنوان :(<title>)** يُظهر اسم النظام في متصفح المستخدمNetwork Alarm System
- **عناصر التعريف :**تظهر نصوص ترحيبية وشعارات مثل شعار MTN لتعريف المستخدم بهوية البوابة.
- **النموذج :(<form>)** يُستخدم لجمع بيانات المستخدم. يحتوي على حقلين رئيسيين:
  - **اسم المستخدم :(<input type="text">)** لجمع اسم المستخدم.
  - **كلمة المرور :(<input type="password">)** لجمع كلمة المرور بشكل آمن (يخفي الأحرف).

- **الرسائل التفاعلية :**تُستخدم قوالب Jinja2 ({% ... %}) لعرض رسائل ديناميكية مثل flash messagesمن الخادم، لتنبيه المستخدم بحالة تسجيل الدخول (نجاح، خطأ، إلخ).

❖ **المظهر والتصميم(CSS)**

يُعنى كود CSS بالجانب الجمالي وتجربة المستخدم، ويُطبق من خلال:

- **النمط المخصص :**يتم استدعاء ملف style.cssلتحديد أنماط العناصر المختلفة، مثل الألوان، والخطوط، والمسافات، وتنسيق الأزرار.
- **الخطوط الخارجية :**يُستخدم رابط Font Awesomeلاستيراد أيقونات احترافية (مثل أيقونة العين) تُضيف لمسة بصرية ووظيفية للعناصر.

❖ **التفاعل والسلوك(JavaScript)**

يُضيف كود JavaScript التفاعلية إلى الصفحة، مما يعزز تجربة المستخدم. يُلاحظ استخدام كود JavaScriptقصير لتحقيق ميزة واحدة رئيسية:

- **إظهار/إخفاء كلمة المرور :**عند النقر على أيقونة العين بجانب حقل كلمة المرور، يقوم الكود بتغيير نوع الحقل من passwordإلى textوالعكس، مما يسمح للمستخدم برؤية كلمة المرور أثناء كتابتها. هذا السلوك التفاعلي يُحسن من قابلية الاستخدام ويقلل من الأخطاء في إدخال كلمة المرور.

❖ **الارتباط مع بيئةFlask**

تُظهر الصفحة التكامل العميق مع إطار عمل Flask من خلال:

- **قوالبJinja2 :** تُستخدم لإنشاء محتوى ديناميكي، مثل مسار الملفات الثابتة (url_for('static', ...))أو عرض رسائل من الخادم.(get_flashed_messages)
- **التوجيه :(<form method="POST" action="/">)** يوجه النموذج البيانات المُدخلة إلى المسار الجذر (/) في الخادم، حيث تقوم دالة loginفي Flask بمعالجتها.

**2. القالب الأساسي(Base Template)**

يُشكل هذا الملف، المسمى base.html، **القالب الأساسي** لجميع صفحات النظام. بدلًا من تكرار نفس الهيكل (الرأس، الشريط الجانبي، الذيل) في كل صفحة، يسمح هذا القالب بتعريفها مرة واحدة ثم إعادة استخدامها في جميع الصفحات الأخرى. يعكس هذا النهج مبدأ **لا تكرر نفسك Don't Repeat Yourself - DRY**، مما يجعل الكود أكثر كفاءة وقابلية للصيانة.

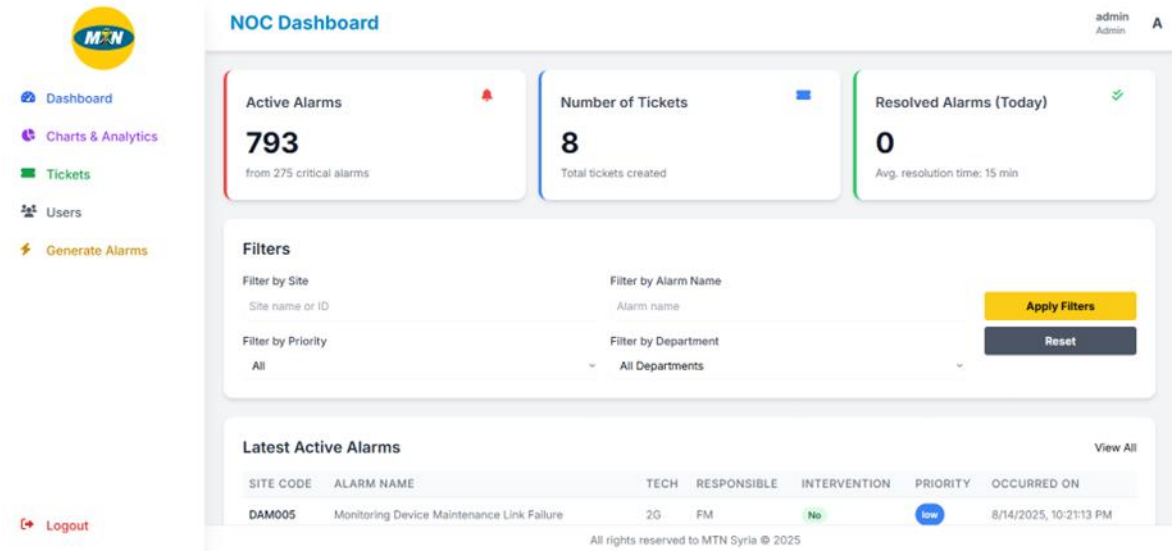
❖ **هيكل الصفحة والبنية(HTML)**

يُعرّف القالب الأجزاء الرئيسية للمنصة:

- **الرأس :(<head>)** يحتوي على البيانات الوصفية للصفحة(meta tags) ، وعنوان الصفحة، والروابط الخارجية للموارد.
- **الشريط الجانبي :(<aside>)** يمثل قائمة التنقل الرئيسية.

- يحتوي على روابط لصفحات النظام المختلفة مثل لوحة التحكم(Dashboard) ، والرسوم البيانية(Charts) ، وتذاكر المشاكل(Tickets) .
- يُظهر روابط إضافية (مثل Users و Generate Alarms)بشكل شرطي بناءً على دور المستخدم .(session.role) هذا يعكس مبدأ **التحكم في الوصول** على مستوى الواجهة الأمامية.

- **الرأس الرئيسي(<header>) :** يحتوي على عنوان الصفحة الحالي، ومعلومات المستخدم المسجل دخوله (الاسم الكامل والدور)، وزرًا لتبديل الشريط الجانبي في الشاشات الصغيرة.
- **المحتوى الرئيسي(<main>) :** هو الجزء الديناميكي من الصفحة حيث سيتم إدخال محتوى كل صفحة فرعية (مثل محتوى صفحة لوحة التحكم أو صفحة المستخدمين).
- **الذيل(<footer>) :** يحتوي على معلومات حقوق الطبع والنشر.




الشكل(7-3) القالب الأساسي

❖ **تقنيات التصميم(CSS & Tailwind CSS) :**

- **Tailwind CSS:** يبرز استخدام tailwindcssفي هذا القالب. بدلاً من كتابة قواعد CSS مخصصة، يُستخدم Tailwind لإضافة فئات CSS جاهزة ومسبقة التحديد مباشرةً في كود HTML (w-64, bg-white, shadow-xl, flex).هذا يُسرّع عملية التطوير ويحافظ على نمط تصميم ثابت ومتناسق في جميع أنحاء التطبيق.
- **الأنماط المخصصة :**يتم تعريف بعض الأنماط المخصصة (مثل mtn-blue, mtn-yellow)مباشرة في وسم .<style>هذا يضمن استخدام الألوان الخاصة بالعلامة التجارية MTNفي جميع أنحاء الموقع بسهولة.
- **الخطوط الخارجية :**يتم استخدام خط Interمن Google Fonts لضمان مظهر عصري واحترافي.

❖ **التفاعل والديناميكية(JavaScript & Jinja2)**

يُظهر القالب استخدامًا متقدمًا لتقنيات الويب:

- **قوالبJinja2 :**
  - **الكتل({% block %}) :** تُستخدم لتعريف مناطق قابلة للتعديل. على سبيل المثال، يمكن للصفحة الفرعية أن تملأ {% block content %}بمحتواها الخاص.
  - **الوراثة(Inheritance) :** تسمح الصفحات الفرعية بوراثة هذا القالب الأساسي وتجاوز أو إضافة محتوى إلى الكتل المحددة.
  - **المتغيرات({{ session.role }}) :** تُستخدم لعرض بيانات الجلسة (مثل دور المستخدم واسمه)، مما يجعل المحتوى شخصيًا وديناميكيًا.
- **JavaScript:**
  - **إخفاء/إظهار الشريط الجانبي :**يُستخدم كود JavaScript للتحكم في وسم aside، مما يجعله قابلًا للطي والفتح في الشاشات الصغيرة (مثل الهواتف المحمولة)، مما يوفر تجربة مستخدم متجاوبة(responsive) .
- **تنبيهاتToast :** يُجهز القالب حاوية (#toast-container) لعرض رسائل التنبيه (toasts)بطريقة جذابة ومؤقتة، مما يُستخدم لإظهار إشعارات مثل "تم تسجيل الخروج بنجاح."

### 3. لوحة التحكم(NOC Dashboard)

تُعد لوحة التحكم هي الواجهة الرئيسية للمستخدمين في النظام، حيث تُقدم رؤية شاملة وحية لحالة الشبكة. هذه الصفحة لا تعرض محتوى ثابتًا، بل تعمل كواجهة تفاعلية تُحدث نفسها باستمرار لجلب البيانات من الخادم وعرضها بطريقة منظمة.

❖ **الوراثة وتضمين الموارد(Jinja2 & Assets)**

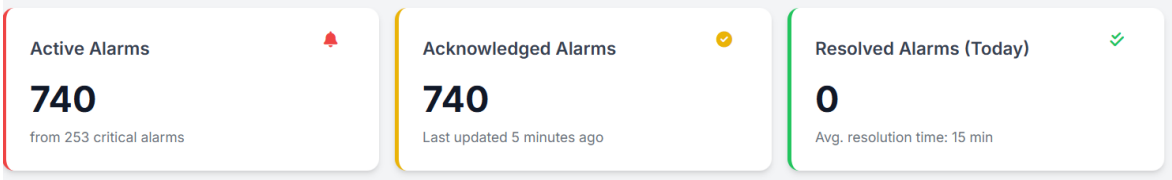
تستخدم هذه الصفحة ميزة **وراثة القوالب** منJinja2 ، حيث تبدأ بالتعليمة {% extends "base.html" %}. هذا يعني أنها ترث كل هيكل وتصميم القالب الأساسي (base.html) وتُعيد تعريف الكتل المحددة:

- **{% block title %}:** يغير عنوان الصفحة ليصبح."NOC Dashboard"
- **{% block styles %}:**يضيف روابط لمكتبات CSS و JavaScriptخارجية ومحلية خاصة بهذه الصفحة، مثل:
  - **Leaflet.js:** مكتبة JavaScript مفتوحة المصدر لإنشاء خرائط تفاعلية.
  - dashboard.css: ملف CSS مخصص لتنسيق عناصر لوحة التحكم.
- **{% block content %}:** يُدرج كل المحتوى الخاص بلوحة التحكم داخل هذا الجزء، بما في ذلك البطاقات الإحصائية والجداول والخريطة.

❖ **المكونات الرئيسية للوحة التحكم(HTML)**

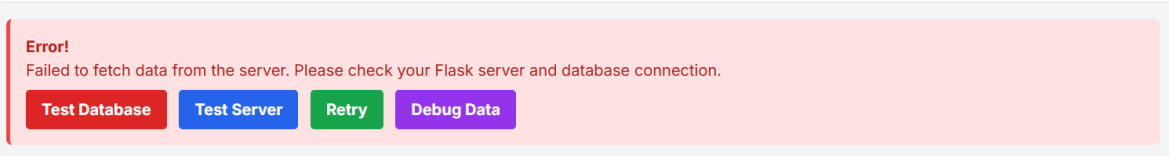
تُقسم الصفحة إلى عدة أقسام وظيفية لتقديم المعلومات بوضوح:

- **بطاقات الإحصائيات :**مجموعة من البطاقات المنسقة بشكل أنيق تعرض مقاييس رئيسية للنظام مثل **عدد الإنذارات النشطة، الإنذارات الحرجة، عدد تذاكر المشاكل، والإنذارات التي تم حلها اليوم.**



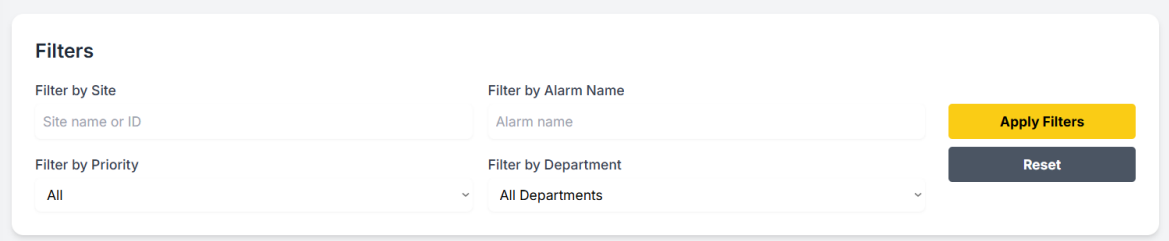

الشكل(7-4) بطاقة الإحصاءات

- **قسم الرسائل الديناميكية :**يتم حجز مساحة لعرض رسائل خطأ، ومؤشرات تحميل، ورسائل نجاح. هذا يعزز تجربة المستخدم من خلال تقديم ملاحظات فورية حول حالة النظام (مثلاً، إذا فشل الاتصال بالخادم).



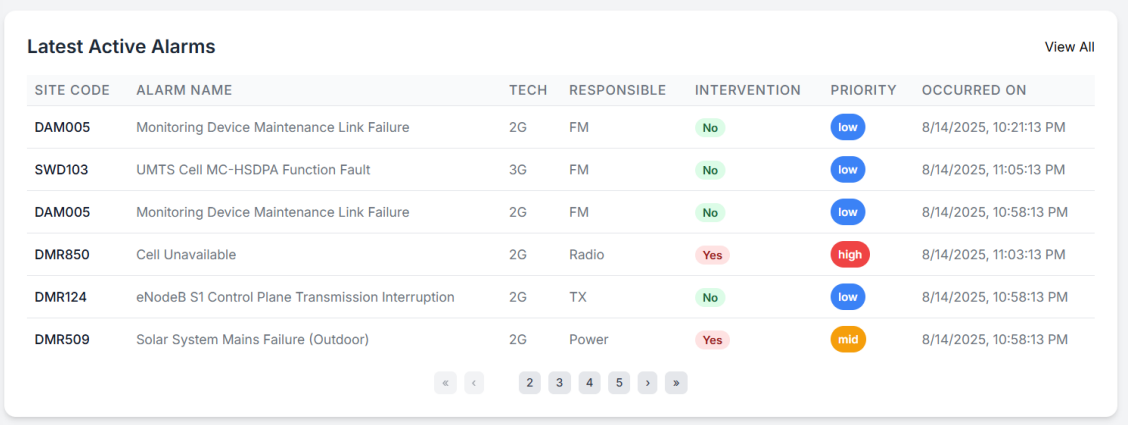

الشكل(7-5) قسم الرسائل الديناميكية

- **قسم المرشحات(Filters) :** يسمح للمستخدم بتصفية البيانات المعروضة في الجداول بناءً على معايير مختلفة مثل **الموقع، اسم الإنذار، الأولوية، والقسم المسؤول** .هذا يُمكن المستخدمين من العثور على المعلومات التي يحتاجونها بسرعة.

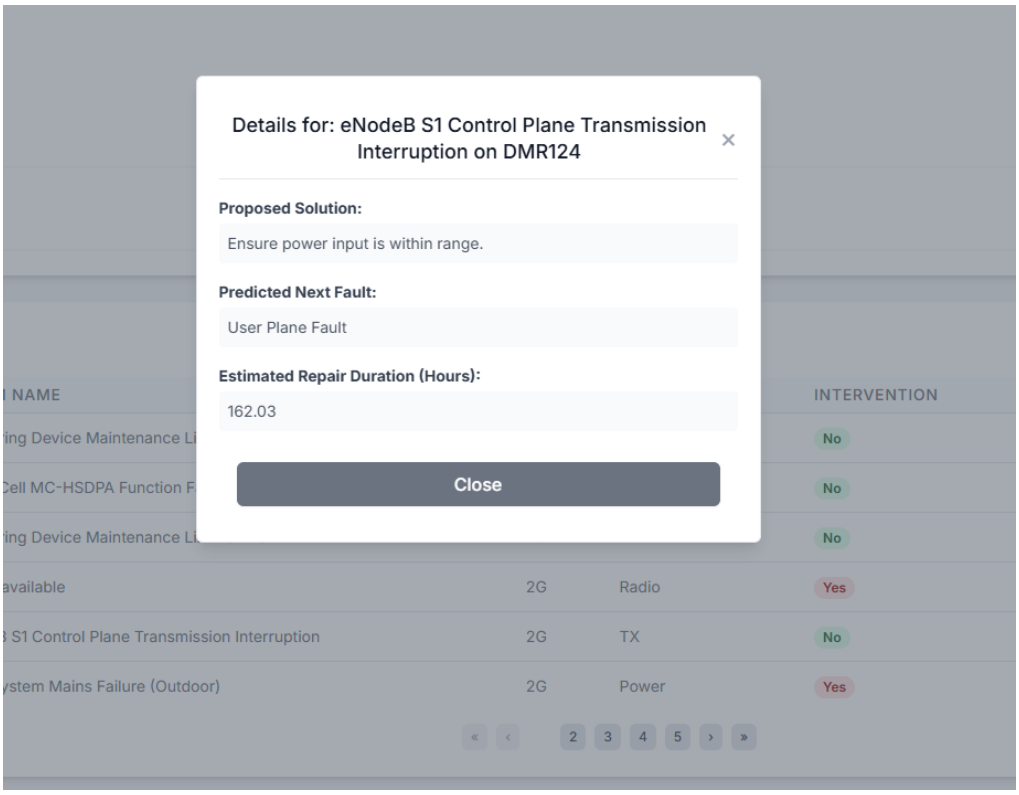



الشكل(7-6) قسم المرشحات

- **الجداول التفاعلية :**يتم عرض البيانات في جداول منظمة وقابلة للتصفية والتصفح. تحتوي الصفحة على جدولين رئيسيين:
  - **جدول الإنذارات النشطة :**يعرض تفاصيل الإنذارات الأخيرة، مثل كود الموقع، واسم الإنذار، والأولوية.

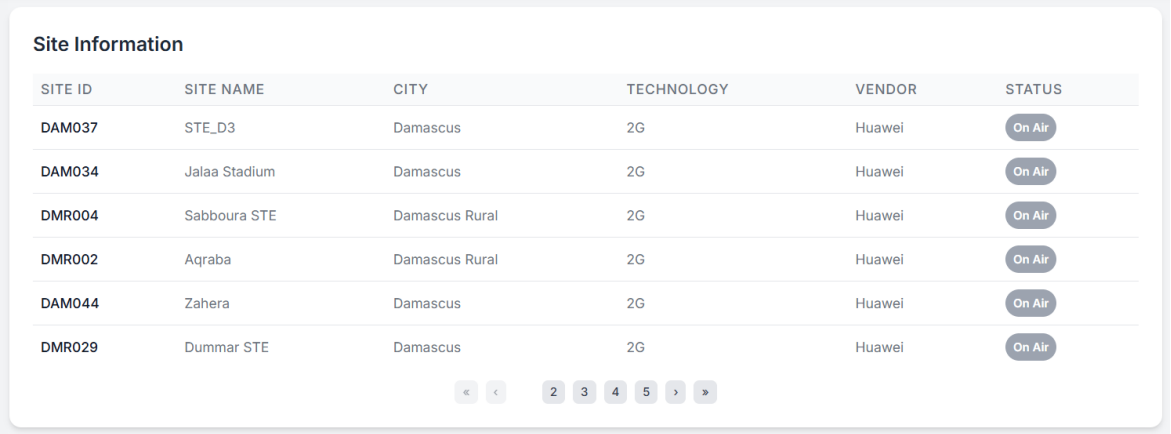



الشكل(7-7) جدول الإنذارات النشطة



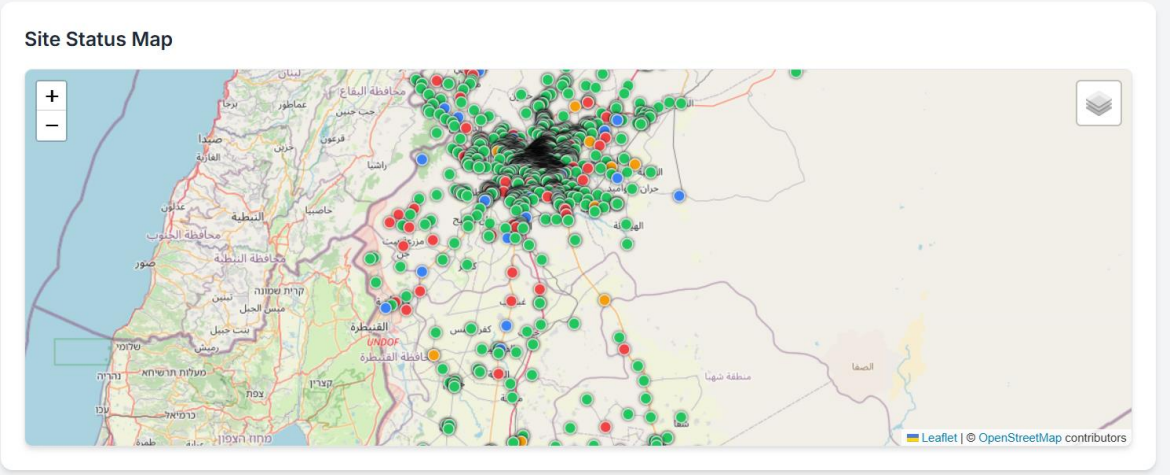

الشكل(7-8) تفاصيل إضافية عن الأعطال

- **جدول معلومات المواقع :**يُظهر تفاصيل كل موقع في الشبكة، مثل المدينة، التقنية، وحالة الموقع.




الشكل(7-9) جدول معلومات الموقع

- **خريطة المواقع(Site Status Map) :** تُستخدم مكتبة Leaflet لعرض خريطة تفاعلية تظهر مواقع الأبراج. يتم وضع علامات (markers) على الخريطة بألوان مختلفة تعكس حالة الموقع (مثلاً، أحمر للإنذارات الحرجة).




الشكل(7-10) خريطة المواقع

### ❖ التفاعل ومعالجة البيانات(JavaScript)

يُشكل كود JavaScript الجزء الأكثر أهمية في هذه الصفحة، حيث يوفر الديناميكية والوظائف الرئيسية:

- **جلب البيانات(fetch) :** يستخدم JavaScript لجلب البيانات من واجهة برمجة تطبيقات (API)الخادم بشكل غير متزامن، مما يمنع تجمد الصفحة.
- **تحديث الواجهة :**بعد جلب البيانات، يقوم الكود بتحديث جميع عناصر الصفحة ديناميكيًا، بما في ذلك:
  - تحديث أرقام البطاقات الإحصائية.
  - تعبئة الجداول بالبيانات الجديدة.
  - إضافة علامات ملونة للمواقع على الخريطة بناءً على حالة كل موقع.
- **المرشحات والتصفح(Pagination) :** يُمكن كود JavaScript المستخدم من تصفية البيانات في الجداول وعرضها على صفحات متعددة، مما يُحسن الأداء عند التعامل مع كميات كبيرة من البيانات.
- **الخريطة التفاعلية :**يتم تهيئة الخريطة وتحديد أنواع الخرائط المختلفة (مثل الشارع، القمر الصناعي) وتخصيص الأيقونات لتمثيل حالة كل موقع.
- **النافذة المنبثقة(Modal) :** تُعرض نافذة منبثقة تفاعلية عند النقر على إنذار في الجدول، لتوفير معلومات إضافية مثل الحل المقترح أو العطل المتوقع.

### 4. صفحة الرسوم البيانية والتحليلات(Charts & Analytics)

تُركز هذه الصفحة بشكل أساسي على **تحليل البيانات** وتقديمها بطريقة مرئية وسهلة الفهم. على عكس لوحة التحكم الرئيسية التي تُقدم رؤية لحظية، تهدف هذه الصفحة إلى عرض الاتجاهات والأنماط في بيانات الإنذارات باستخدام رسوم بيانية تفاعلية.

❖ **الهيكل الأساسي وتضمين المكتبات(Jinja2 & Assets)**

تتبع هذه الصفحة نفس هيكل صفحة لوحة التحكم، حيث ترث من القالب الأساسي .base.html الفرق الرئيسي هنا هو في الموارد المُضافة:

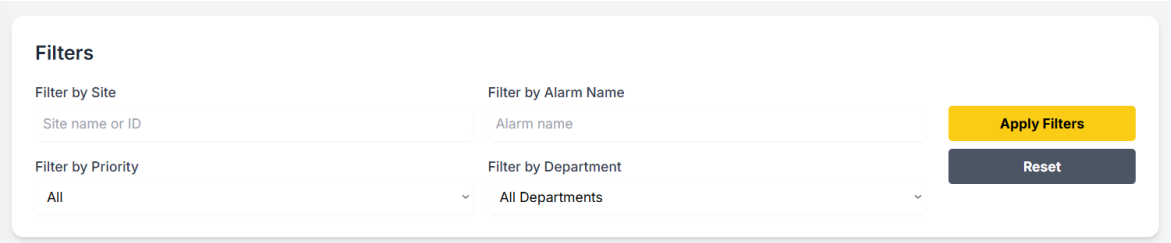
- **{% block title %}:** يحدد عنوان الصفحة."Charts & Analytics"
- **{% block styles %}:** يضيف رابطًا لمكتبة **Chart.js**، وهي مكتبة JavaScript مفتوحة المصدر وممتازة لإنشاء رسوم بيانية متنوعة وجذابة.
- **{% block content %}:** يحتوي على الأقسام الرئيسية للصفحة، وهي مرشحات البيانات والرسوم البيانية نفسها.

❖ **المكونات الرئيسية للصفحة(HTML)**

تُقسم الصفحة إلى قسمين رئيسيين يُكمل كل منهما الآخر:

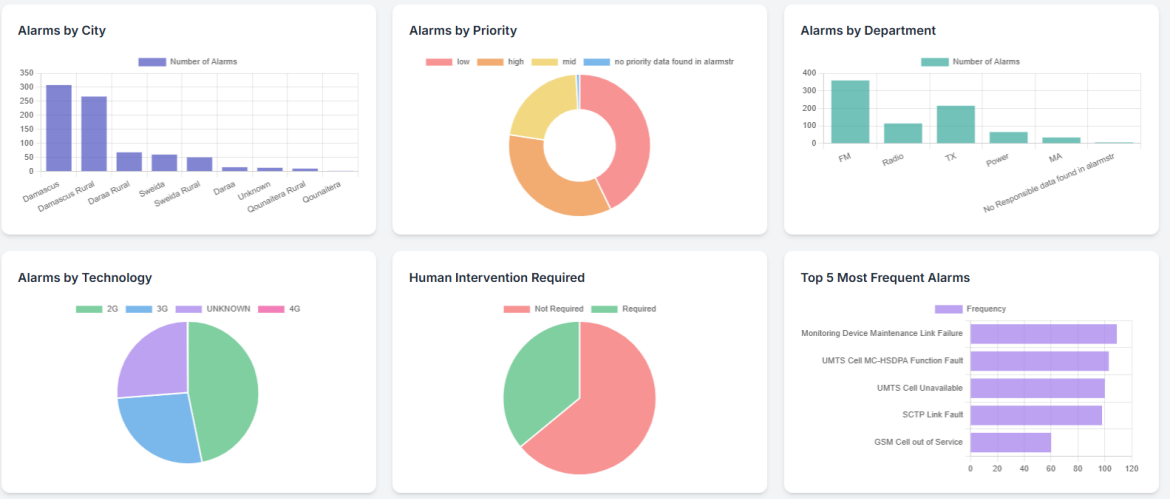
- **قسم المرشحات(Filters) :**
  - هذا القسم مطابق لقسم المرشحات في لوحة التحكم الرئيسية.

- يسمح للمستخدم بتضييق نطاق البيانات المعروضة في الرسوم البيانية.
- يمكن التصفية حسب **الموقع، اسم الإنذار، الأولوية، والقسم المسؤول.**
- الأزرار (Apply Filters و Reset) تتيح للمستخدم التحكم في تطبيق هذه المرشحات على الرسوم البيانية.




الشكل(7-11) قسم المرشحات

- **قسم الرسوم البيانية(Charts Grid) :**
  - هو شبكة من الرسوم البيانية(grid) ، كل رسم بياني داخل بطاقة (div) أنيقة ومنسقة.
  - كل رسم بياني يُعرض داخل عنصر <canvas>، وهو العنصر الذي تستخدمه مكتبة Chart.js لرسم المخططات.
  - تتضمن الصفحة مجموعة متنوعة من المخططات لتغطية جوانب مختلفة من البيانات:
    - **Alarms by City مخطط شريطي:** يُظهر توزيع الإنذارات حسب المدينة، مما يُساعد على تحديد المناطق التي بها أكبر عدد من المشاكل.
    - **Alarms by Priority مخطط حلقي:** يُبين نسبة الإنذارات حسب الأولوية (عالية، متوسطة، منخفضة).
    - **Alarms by Department مخطط شريطي:** يُوضح عدد الإنذارات لكل قسم مسؤول.
    - **Alarms by Technology مخطط دائري:** يُظهر توزيع الإنذارات على مختلف التقنيات) مثل G3، G4، إلخ.(
    - **Human Intervention Required مخطط دائري:** يُقارن بين الإنذارات التي تتطلب تدخلاً بشريًا وتلك التي لا تتطلب ذلك.
    - **Top 5 Most Frequent Alarms مخطط شريطي أفقي:** يُسلط الضوء على أكثر 5 إنذارات تكرارًا في النظام.




الشكل(7-12) قسم الرسوم البيانية

### ❖ الوظائف الديناميكية(JavaScript)

يُعتبر كود JavaScript في هذه الصفحة هو العقل المُدبر، حيث يقوم بمهام أساسية لتحويل البيانات الخام إلى رؤى:

- **جلب البيانات :**يستخدم الدالة fetchData()لجلب بيانات الإنذارات والمواقع من الخادم.
- **إدارة المخططات:**
  - يتم استخدام كائن chartInstancesلتخزين ومراقبة جميع المخططات المُنشأة.
  - الدالة renderChart()تتحقق مما إذا كان المخطط موجودًا بالفعل، وتُدمره قبل إنشاء نسخة جديدة، مما يضمن تحديثًا سلسًا للبيانات.
- **تحديث المخططات:**
  - لكل رسم بياني، توجد دالة مخصصة(updateCityChart ، updatePriorityChart، إلخ) تقوم بمعالجة البيانات الخام.
  - هذه الدوال تقوم بتجميع البيانات (مثلاً، عدّ الإنذارات لكل مدينة أو قسم) وتجهيزها لتغذية مكتبة.Chart.js
- **التصفية الديناميكية(applyFilters) :**
  - هذه الدالة هي جوهر الصفحة. عندما يقوم المستخدم بتغيير المرشحات، تقوم الدالة بتصفية قائمة الإنذارات بالكامل بناءً على المدخلات.
  - بعد التصفية، تقوم الدالة باستدعاء updateAllCharts()لتحديث جميع الرسوم البيانية دفعة واحدة بالبيانات الجديدة والمُصفاة.

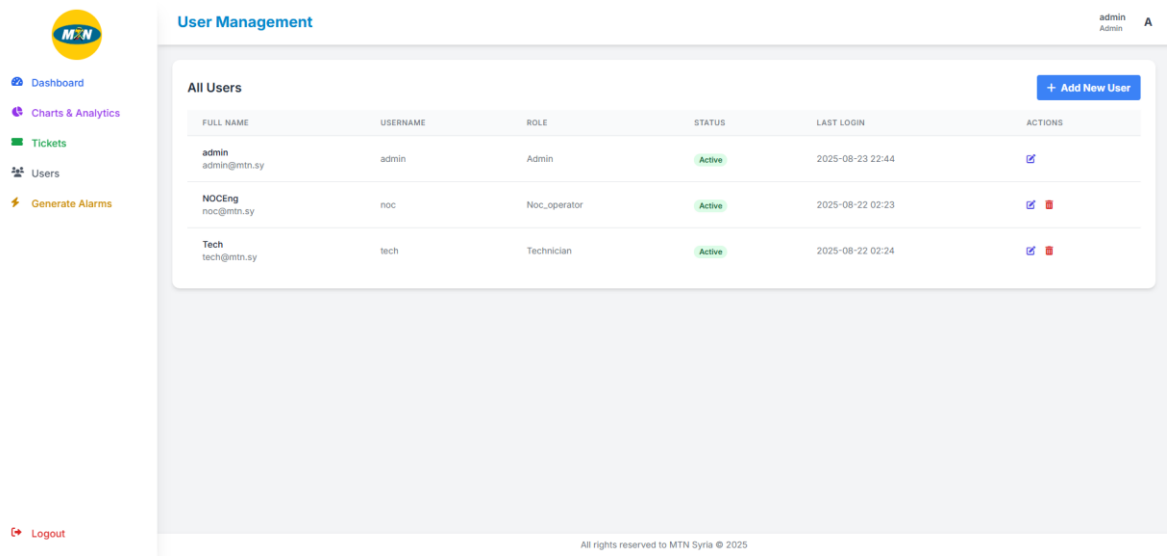
- **الاستماع للأحداث:**
  - عند تحميل الصفحة(DOMContentLoaded) ، يتم جلب البيانات لأول مرة.
  - يتم ربط أزرار المرشحات apply-filters و reset-filtersبوظائف JavaScript المقابلة، مما يجعل الصفحة تفاعلية بالكامل.

### 5. صفحة إدارة المستخدمين(User Management)

تُعد صفحة إدارة المستخدمين جزءًا حيويًا من أي نظام، فهي تُمكّن المسؤولين من التحكم في الوصول والصلاحيات. يهدف هذا القالب إلى تقديم واجهة بسيطة وفعالة لعرض، إضافة، تعديل، وحذف حسابات المستخدمين.

❖ **الهيكل الأساسي للصفحة**

- **{% extends "base.html" %}:**كما هو الحال في الصفحات الأخرى، يرث هذا القالب الهيكل الأساسي من base.htmlلضمان التناسق في التصميم، مما يوفر شريط التنقل العلوي وتذييل الصفحة.
- **{% block page_title %}: و{% block title %}**يُحددان عنوان الصفحة في متصفح الويب وفي واجهة المستخدم على التوالي باسم."User Management"
- **{% block content %}:**هو الحاوية الرئيسية لكل المحتوى الخاص بالصفحة.




الشكل(7-13) صفحة إدارة المستخدمين

❖ **المكونات الرئيسية لواجهة المستخدم(HTML)**

- **العنوان وزر الإضافة:**
  - يحتوي رأس الصفحة على عنوان رئيسي **"All Users"**وزر **"Add New User".**
  - يستخدم الزر <**a**>كوصلة لإعادة توجيه المستخدم إلى مسار **url_for('add_user')**، وهو مسار Flask يُشير إلى صفحة إضافة مستخدم جديد.
- **رسائل"Flash" :**
  - الجزء المُعلق في الكود (داخل ``) يُمثل آلية **"Flash Messages"**في **Flask.**
  - تُستخدم هذه الآلية لعرض رسائل للمستخدم بعد إتمام عملية معينة (مثل إضافة مستخدم بنجاح أو فشل عملية الحذف).
  - يتم عرض رسالة بسيطة (message) مع تنسيق مُناسب-(bg-green-100, bg-red-100)بناءً على الفئة(category) ، مما يمنحها مظهرًا مختلفًا (أخضر للنجاح، أحمر للفشل.(
- **جدول المستخدمين:**
  - يُعد هذا الجزء هو المحور الرئيسي للصفحة. يُستخدم عنصر <table>لعرض قائمة بجميع المستخدمين في النظام.
  - **رأس الجدول :(<thead>)** يحتوي على عناوين الأعمدة:
    - **Full Name**:اسم المستخدم الكامل وبريده الإلكتروني.
    - **Username**:اسم المستخدم الخاص به لتسجيل الدخول.
    - **Role:**صلاحيات المستخدم) مثل.(Admin, User, etc.
    - **Status**:حالة الحساب (نشط Activeأو غير نشط .(Inactive
    - **Last Login**:تاريخ ووقت آخر تسجيل دخول.
    - **Actions**:الإجراءات المتاحة للمستخدم (تعديل، حذف).
- **جسم الجدول(<tbody>) :**
  - يستخدم حلقة **{% for u in users %}**من محرك القوالب **Jinja2**لتكرار وعرض كل مستخدم من قائمة **users**التي تم تمريرها من الخادم.
  - **{% else %}:** في حالة عدم وجود مستخدمين في القائمة، يتم عرض صف واحد برسالة **"No users found"**لتقديم ملاحظة واضحة للمستخدم.

- **التحقق الشرطي ({% if u.is_active %})** يتم استخدامه لتغيير مظهر حالة المستخدم ديناميكيًا. إذا كان المستخدم نشطًا، يتم عرض علامة خضراء، وإلا يتم عرض علامة حمراء.
- **أعمدة الإجراءات(Actions) :**
  - **زر التعديل(Edit) :** هو رابط يُعيد توجيه المستخدم إلى صفحة تعديل بيانات المستخدم، مع تمرير **id** المستخدم كجزء من الرابط (url_for('edit_user', id=u.id)).
  - **زر الحذف(Delete) :** هو نموذج form منفصل يستخدم طريقة POST لإرسال طلب حذف إلى الخادم.
  - يُستخدم "return confirm(...);" **onsubmit=** لعرض رسالة تأكيد للمستخدم قبل تنفيذ عملية الحذف لمنع الحذف غير المقصود.
  - **{% if u.id != session.user_id %}:** هذا الشرط يمنع المستخدم من حذف حسابه الخاص، مما يضيف طبقة أمان أساسية.

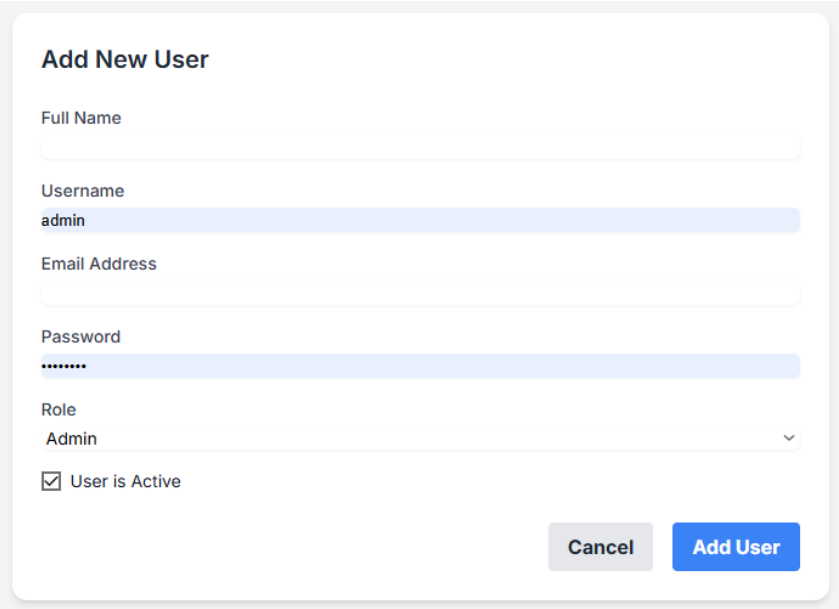
### 6. صفحة إضافة/تعديل مستخدم(Add/Edit User)

تُظهر هذه الصفحة مثالًا ممتازًا على كيفية استخدام **نفس القالب (Template)** للقيام بوظيفتين مختلفتين بناءً على سياق مُعين، وهي **إضافة مستخدم جديد** أو **تعديل مستخدم موجود** .هذا الأسلوب يقلل من تكرار الكود ويسهل عملية الصيانة.

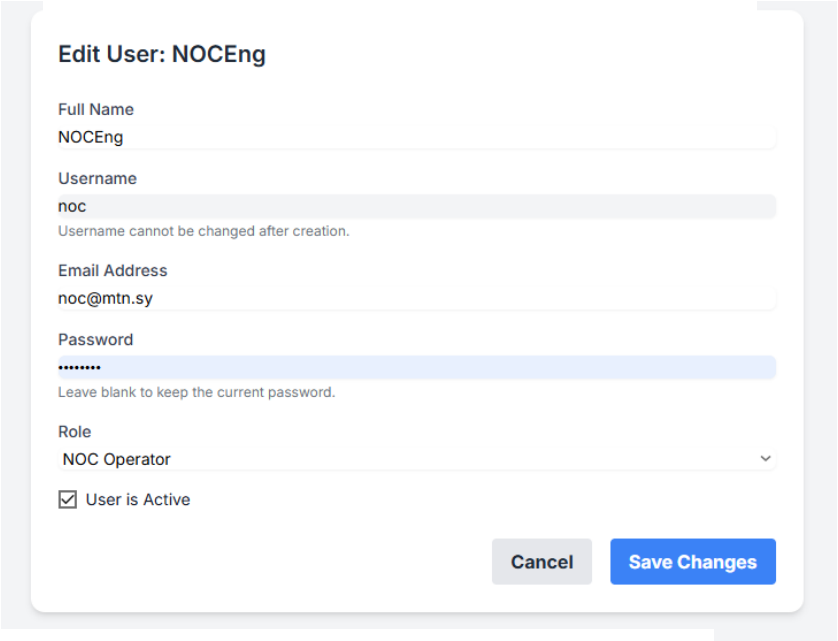
❖ **التحكم في المحتوى الديناميكي(Jinja2)**

تعتمد هذه الصفحة بشكل كبير على المتغير action الذي يتم تمريره من الخادم (Flask) لتحديد ما إذا كانت العملية هي "edit" أو ."add" يتم استخدام هذا المتغير في عدة أماكن:

- **العنوان title و page_title:**
  - {% block title %}{{ 'Edit User' if action == 'edit' else 'Add New User' }}{% endblock %}
  - هذا الشرط يغير عنوان الصفحة ديناميكيًا. إذا كان action يساوي edit، يصبح العنوان"Edit User" ، وإلا فإنه يصبح."Add New User"
  - يتم تطبيق نفس المنطق على عنوان الصفحة الظاهر للمستخدم.




الشكل(7-14) إضافة مستخدم جديد




الشكل(7-15) صفحة تعديل مستخدم

- **التحكم في حقول النموذج:**
  - **اسم المستخدم(username) :**
    - في وضع التعديل، يُصبح حقل اسم المستخدم للقراءة فقط readonly لمنع تغييره بعد إنشائه.{% if action=='edit' %}readonly{% endif %}:

- **كلمة المرور(password) :**
  - يُصبح هذا الحقل {% if action=='add' %}required{% endif %} إلزاميًا فقط عند إضافة مستخدم جديد.
  - في وضع التعديل، تُترك كلمة المرور اختيارية، ويتم توفير رسالة توضيحية للمستخدم بأنه يمكنه ترك الحقل فارغًا للحفاظ على كلمة المرور الحالية.
- **حالة التنشيط(is_active) :**
  - {% if data.is_active or (action=='add' and not data) %}checked{% endif %}: هذا الشرط الذكي يضمن أن يكون مربع الاختيار محددًا بشكل افتراضي في حالة الإضافة.

- **نص زر الإرسال:**
  - {{ 'Save Changes' if action == 'edit' else 'Add User' }}: يتغير نص الزر الرئيسي في النموذج ليعكس طبيعة العملية.

❖ **نموذج الإدخال(Form)**

- يحتوي النموذج على حقول إدخال أساسية لإدارة بيانات المستخدمين:
  - **Full Name, Username, Email:** حقول نصية لإدخال المعلومات الأساسية.
  - **Password:** حقل خاص بكلمة المرور.
  - **Role:** قائمة منسدلة (<select>) للسماح للمسؤول باختيار صلاحية المستخدم من بين الخيارات المحددة (Admin, NOC Operator, Technician, Supervisor).
  - **Is Active:** مربع اختيار (checkbox) لتحديد ما إذا كان الحساب نشطًا أم لا.
- **البيانات الأولية للحقول:("{{ data.full_name or '' }}"value=)**
  - يتم استخدام المتغير data لتعبئة الحقول بالبيانات الحالية للمستخدم.
  - {% set data = user_data or form_data or {} %}: هذا السطر يُحدد مصدر البيانات. إذا كان هناك بيانات مستخدم (user_data) قادمة من الخادم (في حالة التعديل)، يتم استخدامها. وإذا كان هناك بيانات مُرسلة من النموذج نفسه (form_data)بعد فشل التحقق (Validation), يتم استخدامها للحفاظ على المدخلات، وإلا يتم استخدام كائن فارغ({}) .

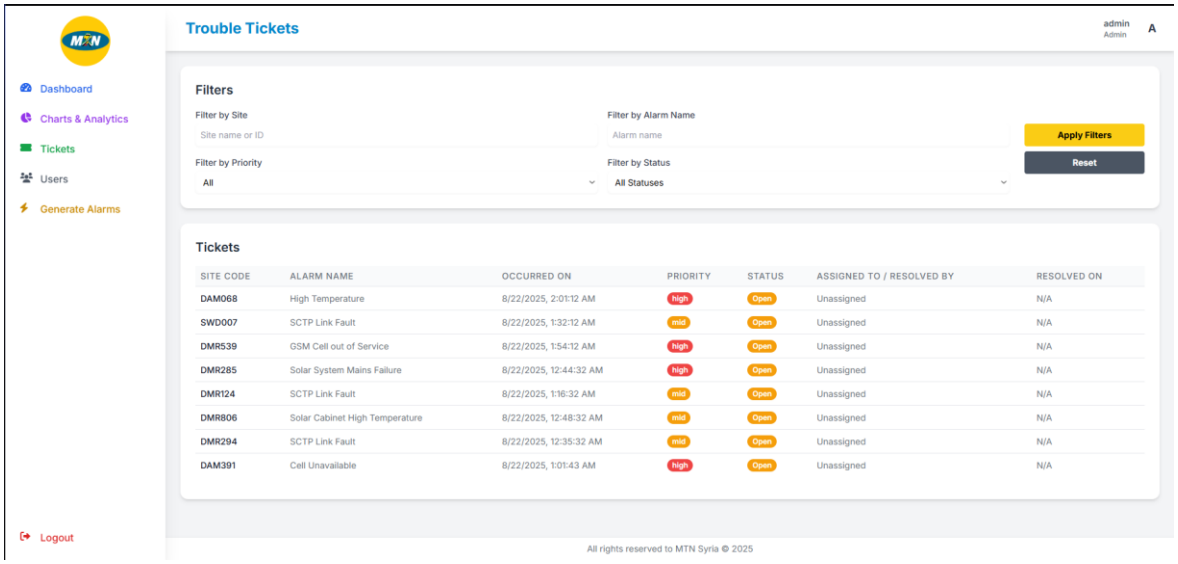
❖ **الإجراءات والأزرار**

- **زر الإلغاء(Cancel) :**
  - هو رابط <a>بسيط يُعيد المستخدم إلى صفحة إدارة المستخدمين (url_for('users')).

- **زر الإرسال(submit) :**
  - هو زر type="submit" الذي يُرسل بيانات النموذج إلى نفس الصفحة (action="").
  - يتم التعامل مع عملية الإضافة أو التعديل على مستوى الخادم (Python) بناءً على ما إذا كانت هناك بيانات مستخدم موجودة أم لا.

### 7. صفحة تذاكر الأعطال (Trouble Tickets)

تُركز هذه الصفحة على عرض وإدارة تذاكر المشاكل (Trouble Tickets) التي يتم إنشاؤها للإنذارات. الهدف هو توفير واجهة مركزية ومفصلة لفرق التشغيل والصيانة لمتابعة حالة المشاكل وحلولها.




الشكل(7-16) صفحة تذاكر الأعطال

❖ **الهيكل الأساسي والمكونات(HTML)**

- **{% extends "base.html" %}** كما هو الحال في جميع الصفحات، يعتمد هذا القالب على base.html لضمان التناسق في التصميم العام.
- **{% block title %}** و **{% block page_title %}** يُحددان عنوان الصفحة في كل من علامة تبويب المتصفح وفي الواجهة الرئيسية كـ"Trouble Tickets".
- **{% block styles %}** يتم تضمين ملف CSS خاص بلوحة التحكم لتطبيق التنسيقات المُحددة.
- **{% block content %}** يحتوي على الأقسام الرئيسية للصفحة:

- **قسم المرشحات(Filters Section) :**
  - يسمح للمستخدم بتصفية التذاكر حسب:
    - **Site:** كود الموقع.
    - **Alarm Name:** اسم الإنذار.
    - **Priority**الأولوية(High, Medium, Low) .
    - **Status:** حالة التذكرة (Open, In Progress, Resolved, Closed)
  - الأزرار Apply Filtersو Resetتتحكم في تطبيق التصفية.
- **جدول التذاكر(Tickets Table)**
  - يعرض قائمة بالتذاكر في جدول أنيق.
  - يتميز بوجود رؤوس أعمدة تُحدد البيانات المعروضة مثل: كود الموقع، اسم الإنذار، تاريخ الحدوث، الأولوية، الحالة، المسؤول عن التذكرة، وتاريخ الحل.
  - ملاحظة مهمة: جسم الجدول (<tbody id="tickets-table-body">) فارغ في البداية، وهذا يعني أن **البيانات يتم جلبها وعرضها ديناميكيًا باستخدام JavaScript** وليس من خلال محرك القوالب.Jinja2
- **صفحة التذاكر(Pagination)**
  - يتيح هذا القسم التنقل بين صفحات التذاكر في حالة وجود عدد كبير منها.
  - يتم إنشاؤه بالكامل ديناميكيًا بواسطة.JavaScript
- **النافذة المنبثقة للتفاصيل(Modal)**
  - هذه النافذة (div id="ticket-details-modal") تكون مخفية في البداية (hidden).
  - تُستخدم لعرض تفاصيل إضافية عن التذكرة، مثل **ملاحظات الحل (resolution_notes)**، عند النقر على صف معين في الجدول.
  - تحتوي على زر إغلاق لتوفير تجربة مستخدم سلسة.

❖ **الوظائف الديناميكية(JavaScript)**

يُعد كود JavaScript في هذه الصفحة معقدًا ومهمًا للغاية، حيث يتولى جميع العمليات الديناميكية:

- **المتغيرات العامة:**
  - **allTickets:** تُخزن جميع بيانات التذاكر التي يتم جلبها من الخادم.

- **currentFilteredTickets:** تُخزن قائمة التذاكر بعد تطبيق المرشحات عليها.
- **ticketsPerPage و currentTicketsPage:** تُستخدم لإدارة عملية تقسيم البيانات على الصفحات.(Pagination)

- **وظائف المساعدة(Utility Functions) :**
  - **getStatusColor(status)**: تُرجع لونًا محددًا بناءً على حالة الأولوية (High, Mid, Low).
  - **getTicketStatusColor(status)**: تُرجع لونًا مختلفًا بناءً على حالة التذكرة نفسها .(Open, In Progress, etc.) تُستخدم لإضافة دلالات بصرية للحالة.

- **عرض البيانات(renderTicketsTable)**
  - هذه هي الوظيفة الأساسية التي تقوم بعرض التذاكر في الجدول.
  - تأخذ قائمة التذاكر المُصفاة، وتُحسب أي صفوف يجب عرضها بناءً على رقم الصفحة الحالي.(currentTicketsPage)
  - تُنشئ صفوف الجدول (<tr>) وعناصر البيانات (<td>) لكل تذكرة بشكل ديناميكي.
  - تُضيف **مستمع حدث (Event Listener)** لكل صف، بحيث عند النقر عليه، يتم استدعاء وظيفة .showTicketDetailsModal()
  - تُعالج أيضًا حالة عدم وجود تذاكر مطابقة.

- **وظائف التنقل(Pagination) :**
  - **renderTicketsPagination(totalItems):** تُنشئ وتُعرض أزرار التنقل بين الصفحات (« ,‹ , ,› ,») أرقام الصفحات.
  - تتحكم في حالة الأزرار (تفعيل/تعطيل) بناءً على الصفحة الحالية.

- **جلب وتصفية البيانات:**
  - **fetchData()**:وظيفة غير متزامنة (async) لجلب بيانات التذاكر من المسار api/tickets/على الخادم.
  - **applyFilters()**: هي وظيفة رئيسية تُستدعى عند تغيير أي من المرشحات. تقوم بتصفية البيانات الموجودة في allTickets بناءً على قيم حقول الإدخال، ثم تُحدث currentFilteredTickets، وتُعيد عرض الجدول من الصفحة الأولى.

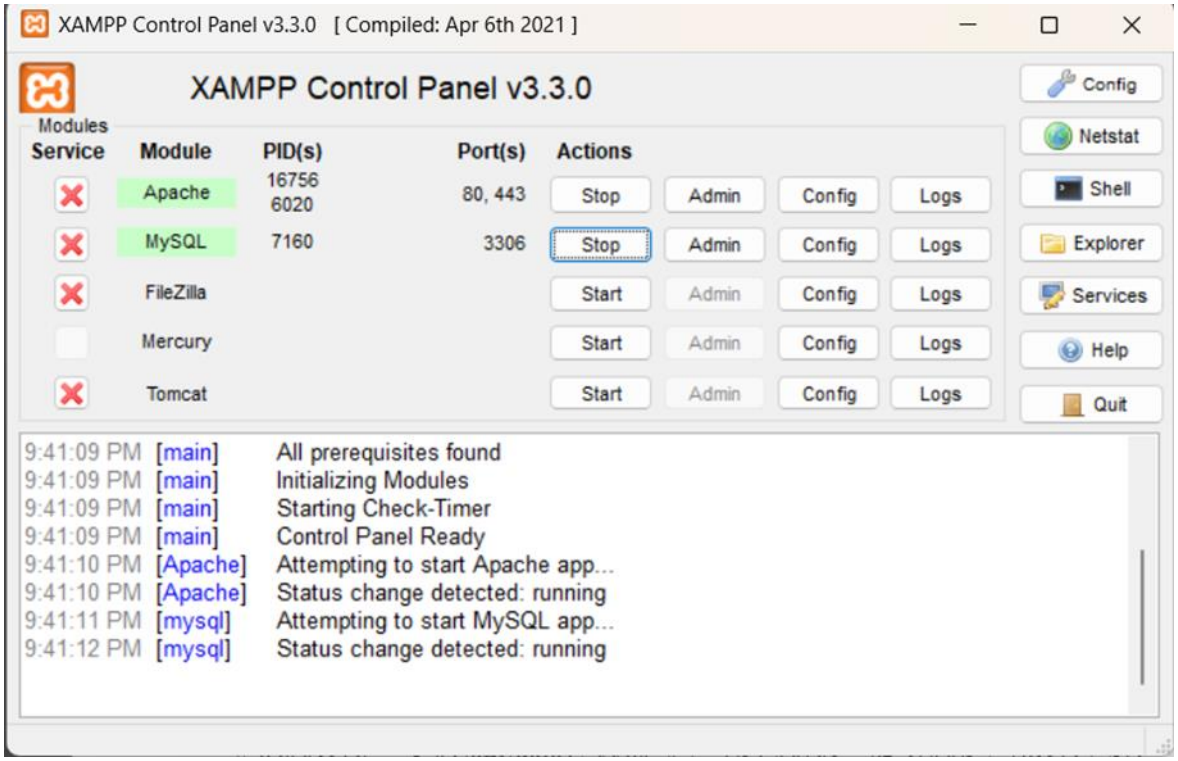
- **النافذة المنبثقة(Modal) :**
  - **showTicketDetailsModal(ticket):** تُظهر النافذة المنبثقة وتُملأ بياناتها بملاحظات الحل (resolution_notes) الخاصة بالتذكرة التي تم النقر عليها.

- **hideTicketDetailsModal()**: تُخفي النافذة.

- **مستمعي الأحداث(Event Listeners) :**
  - **DOMContentLoaded**: يتم استدعاؤه عند تحميل الصفحة، يقوم بجلب البيانات لأول مرة.
  - يتم ربط أحداث clickو inputو changeبأزرار المرشحات وحقول الإدخال، مما يتيح التصفية الفورية والديناميكية للبيانات.
  - يتم ربط أحداث النقر بأزرار إغلاق النافذة المنبثقة.

### 3.2.5-تشغيل موقع الويب

1-تشغيل برنامج XAMPP وتشغيل خدمة Apache و MySQL فيه.




الشكل(7-17) تشغيل برنامج **XAMPP**

2-الانتقال إلى php admin بالضغط على زر admin

3-إنشاء قاعدة البيانات باسم Mtn_noc_systemواسيراد ملف قاعدة البيانات.

4-تشغيل كود app.py

5-فتح موقع الويب باستخدام أحد المتصفحات والانتقال للرابط التالي:

http://127.0.0.1:5000

# 3.5-تصميم تطبيق الأندرويد

## 1.3.5-الخادم Backend

مع تطور النظام، ظهرت الحاجة لإنشاء واجهة خاصة بتطبيق الموبايل لعرض البيانات الهامة. ولكن، واجهتنا مشكلة أداء عند محاولة استخدام مسار البيانات (/api/data) المصمم لواجهة الويب، حيث كان هذا المسار يرسل كمية هائلة من البيانات (جميع المواقع وجميع الإنذارات)، مما أدى إلى بطء شديد في استجابة الخادم وانقطاع الاتصال مع التطبيق.

لحل هذه المشكلة، قمنا بتطوير مسار API جديد ومُحسَّن (/api/mobile/dashboard_data) مصمم خصيصاً لتلبية متطلبات لوحة التحكم في تطبيق الموبايل بكفاءة عالية.

1. **الغرض من المسار الجديد**

الهدف الأساسي من هذا المسار هو تزويد تطبيق الموبايل بالبيانات الضرورية فقط، مما يضمن:

- **سرعة الاستجابة** :تقليل حجم البيانات المرسلة بشكل كبير يؤدي إلى تحميل أسرع للوحة التحكم.
- **استقرار الخادم** :تجنب إرهاق الخادم بطلبات تستهلك الكثير من الذاكرة وقوة المعالجة.
- **كفاءة الشبكة** :تقليل استهلاك بيانات الإنترنت على أجهزة المستخدمين.

2. **شرح الأجزاء الرئيسية:**

1. **@app.route('/api/mobile/dashboard_data', methods=['GET'])**:
   - يُعرّف هذا السطر مساراً جديداً في الخادم يمكن الوصول إليه عبر الرابط api/mobile/dashboard_data/، ويستجيب فقط لطلبات من نوع GET.
2. **جلب الإحصائيات:**
   - يتم تنفيذ استعلامات COUNT(*) بسيطة وسريعة جداً لجلب عدد التذاكر الكلي وعدد التذاكر التي تم حلها. هذه الطريقة أكثر كفاءة من جلب جميع السجلات ثم عدّها في الخادم.
3. **الاستعلام المُحسَّن لجلب الإنذارات:**
   - **SELECT ai.Name, ai.SiteCode, ...**: بدلاً من جلب جميع الأعمدة باستخدام *، قمنا بتحديد الأعمدة التي يحتاجها تطبيق الموبايل فقط، مما يقلل من حجم البيانات.
   - **WHERE a.cleared_on IS NULL**: هذا الشرط يقوم بفلترة النتائج لعرض الإنذارات النشطة فقط، وهو ما يهم المستخدم في لوحة التحكم.
   - **ORDER BY ai.OccurredOn DESC**: يقوم بترتيب الإنذارات من الأحدث إلى الأقدم.

- **LIMIT 10**: هذا هو التحسين الأهم، حيث يقتصر عدد النتائج على أحدث 10 إنذارات فقط. هذا يمنع الخادم من إرسال آلاف السجلات ويضمن استجابة سريعة جداً.

4. **معالجة البيانات وإرسالها:**

- قبل إرسال البيانات، يتم تحويل أي كائنات من نوع datetime إلى نصوص (strings)باستخدام ()isoformat، وذلك لأن صيغة JSON القياسية لا تدعم كائنات التاريخ والوقت مباشرة.
- **return jsonify(...)**: أخيراً، يتم تجميع كل البيانات المطلوبة (الإحصائيات وقائمة الإنذارات المحدودة) وإرسالها كاستجابة بصيغةJSON ، وهي الصيغة المثالية للتواصل بين الخوادم وتطبيقات الموبايل.

بهذه الطريقة، تمكنا من بناء واجهة خلفية قوية ومستقرة قادرة على خدمة كل من واجهة الويب وتطبيق الموبايل بكفاءة عالية.

## 2.3.5-الواجهة الأمامية لتطبيق الموبايل(Flutter)

لتقديم تجربة مستخدم حديثة وسريعة الاستجابة على الأجهزة المحمولة، تم تطوير الواجهة الأمامية للتطبيق باستخدام إطار العمل **Flutter**منGoogle .

يتيح لنا Flutter بناء واجهات مستخدم جميلة وتفاعلية تعمل على كل من نظامي Android و iOS من قاعدة شيفرة برمجية واحدة، مما يضمن تجربة متناسقة ويوفر وقتاً وجهداً في عملية التطوير.

### 1. الهيكلية العامة للتطبيق (Application Architecture)

تم تصميم التطبيق باتباع مبادئ فصل الاهتمامات (Separation of Concerns) لضمان سهولة الصيانة والتطوير المستقبلي. تتكون الهيكلية من الأجزاء الرئيسية التالية:

- **طبقة الواجهة(UI Layer)** : تحتوي على جميع الشاشات والعناصر المرئية التي يتفاعل معها المستخدم. تم بناؤها باستخدام نظام الويدجت (Widgets) الخاص بـ.Flutter
- **طبقة الخدمات(Services Layer)** : مسؤولة عن التواصل مع الواجهة الخلفية (Backend)تم إنشاء ملف api_service.dart ليكون مسؤولاً عن جميع طلبات الشبكة (Network Requests)وإدارة البيانات.
- **طبقة النماذج(Models Layer)** : تحتوي على تعريفات الأصناف (Classes) التي تمثل البيانات القادمة من الخادم، مثل Ticket, Alarm,و .DashboardDataهذا يضمن التعامل مع البيانات بشكل آمن ومنظم.

**2. الميزات الرئيسية وتحليل الشيفرة البرمجية**

**أ. المصادقة وتسجيل الدخول(login_page.dart)**

هي نقطة البداية للمستخدم.

- **الواجهة** :تتكون من حقول لإدخال اسم المستخدم وكلمة المرور، بالإضافة إلى زر لتسجيل الدخول.
- **الآلية** :عند الضغط على زر تسجيل الدخول، يتم استدعاء دالة loginمن api_service.dart وإرسال بيانات المستخدم إلى الخادم.
- **إدارة الجلسة** :في حال نجاح عملية تسجيل الدخول، يتم حفظ بيانات المستخدم (مثل الاسم الكامل والدور) في ذاكرة الهاتف باستخدام مكتبة shared_preferences.
هذا يسمح للتطبيق بتذكر المستخدم عند إعادة تشغيله والانتقال مباشرة إلى لوحة التحكم.

**ب. لوحة التحكم الرئيسية(dashboard_page.dart)**

هذه الشاشة هي المركز الرئيسي للتطبيق بعد تسجيل الدخول، وهي مصممة لتكون مشابهة لواجهة الويب.

- **التنقل** :تحتوي على شريط تنقل سفلي (BottomNavigationBar) يتيح للمستخدم التبديل بسهولة بين "لوحة التحكم" و "التذاكر."
- **جلب البيانات الديناميكي** :تستخدم FutureBuilderلجلب البيانات بشكل غير متزامن من مسار API المُحسَّن .(/api/mobile/dashboard_data) هذا يضمن عدم تجميد واجهة المستخدم أثناء انتظار استجابة الخادم.
- **عرض الإحصائيات** :يتم عرض البيانات التي تم جلبها في بطاقات إحصائيات (Stat Cards)جذابة تعرض عدد الإنذارات النشطة، إجمالي التذاكر، والتذاكر التي تم حلها.
- **قائمة الإنذارات** :يتم عرض أحدث الإنذارات النشطة في قائمة (ListView) لتوفير نظرة سريعة للمستخدم.
- **التحديث بالسحب** :تم تضمين ويدجت RefreshIndicatorالتي تسمح للمستخدم بسحب الشاشة لأسفل لتحديث البيانات فوراً.

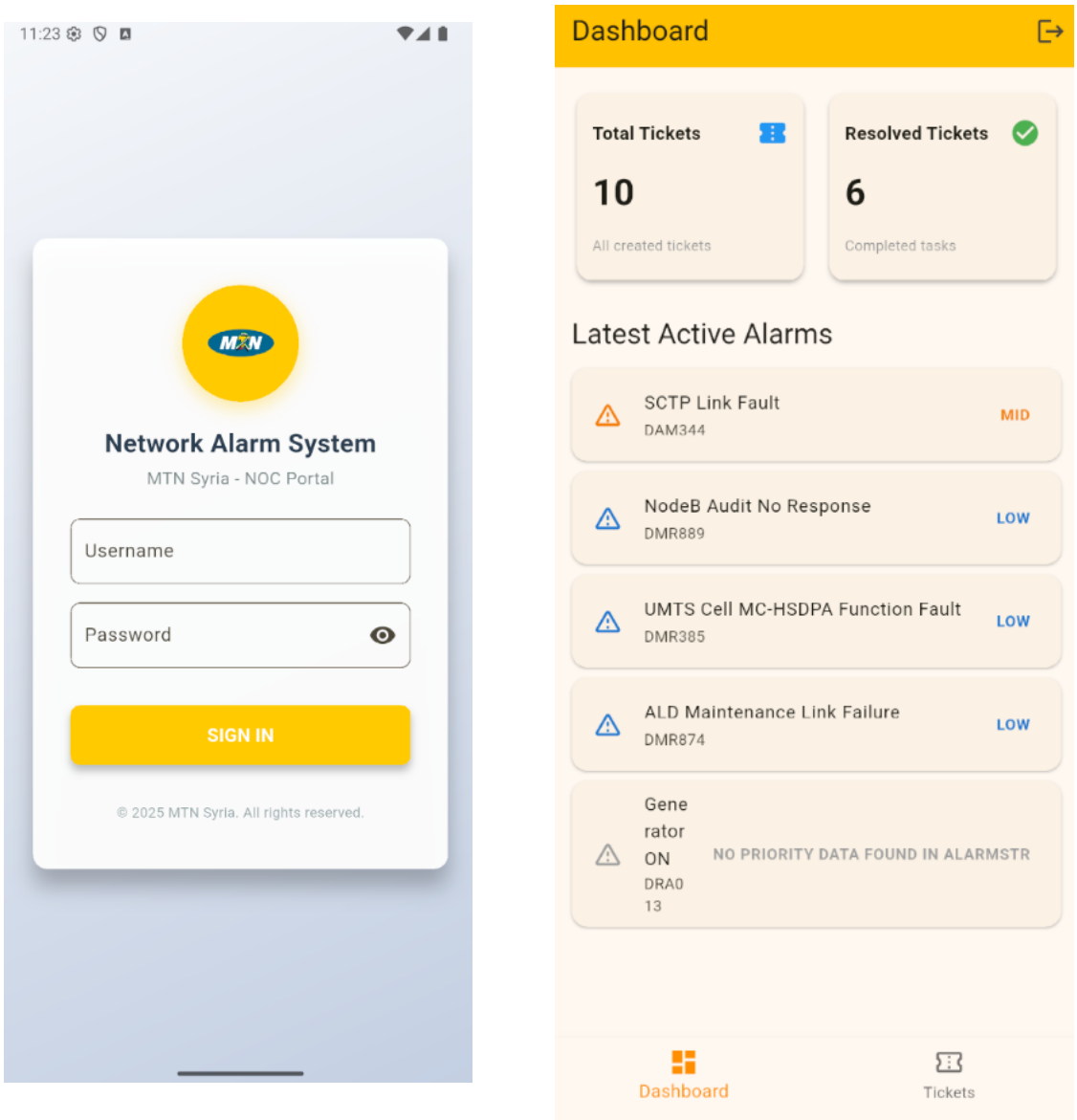
**ج. واجهة التذاكر(tickets_page.dart)**

هذه الواجهة توفر للمستخدمين، وخاصة الفنيين، القدرة على عرض وإدارة التذاكر.

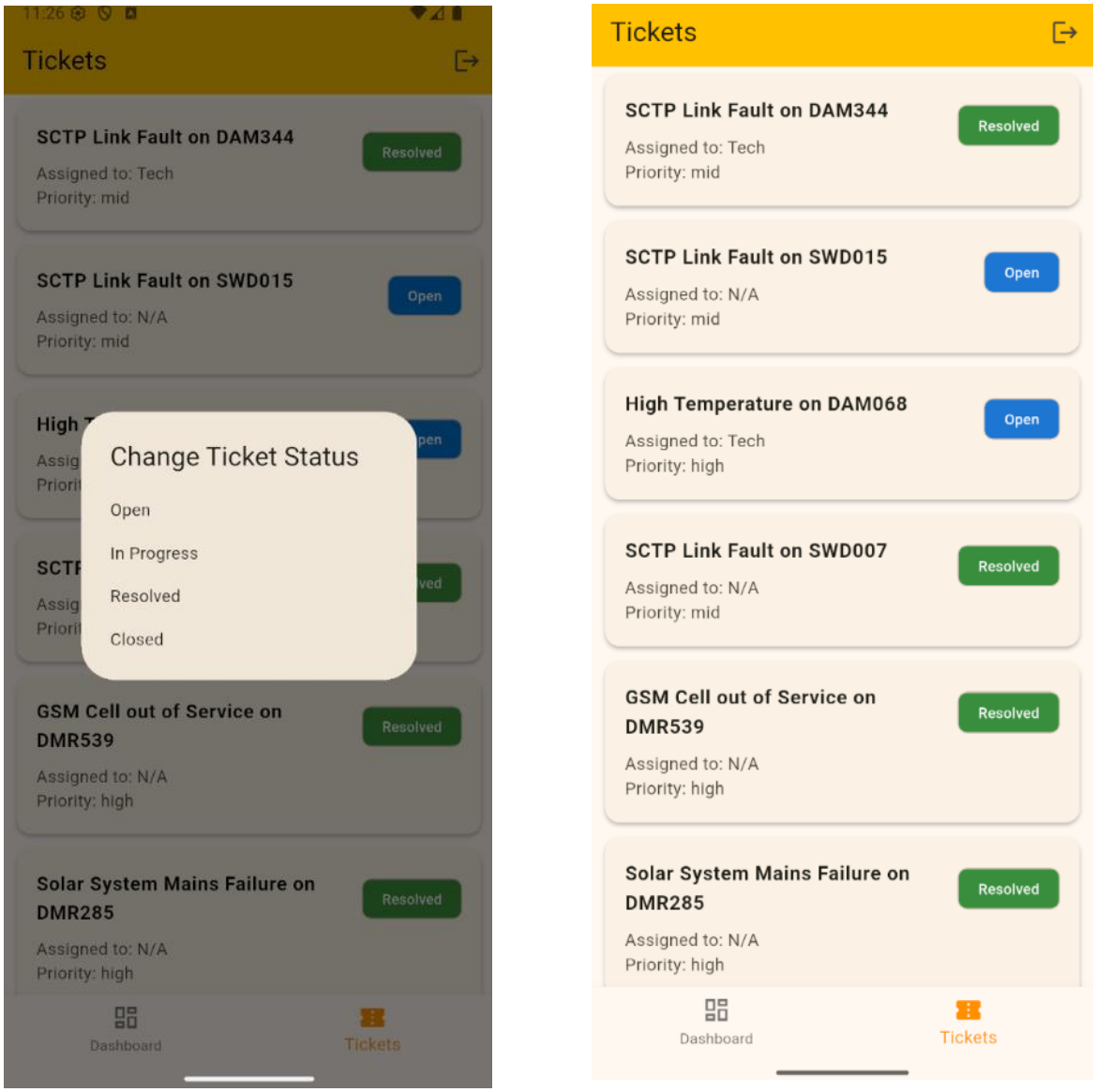
- **عرض التذاكر** :تستخدم ListView.builderلعرض قائمة التذاكر بكفاءة، حيث يتم بناء العناصر فقط عند ظهورها على الشاشة.
- **تفاعل المستخدم** :يمكن للمستخدمين المصرح لهم (الفنيين والمسؤولين) تغيير حالة التذكرة.

- **آلية التحديث** :عند الضغط على زر الحالة، يتم عرض نافذة منبثقة (Dialog) تحتوي على الحالات المتاحة. بعد اختيار حالة جديدة، يتم استدعاء دالة updateTicketStatus من api_service.dart لإرسال التحديث إلى الخادم.
- **التغذية الراجعة** :يتم إعلام المستخدم بنجاح أو فشل العملية عبر SnackBar، ويتم تحديث القائمة تلقائياً لعرض الحالة الجديدة.

بهذه الطريقة، يقدم تطبيق الموبايل واجهة أمامية قوية، تفاعلية، وسهلة الاستخدام، مما يكمل النظام بشكل فعال ويوفر أداة قوية للمستخدمين أثناء التنقل.




الشكل(7-18) واجهة تسجيل الدخول والواجهة الرئيسية




الشكل(7-19) واجهة التذاكر

### 3.3.5-تشغيل تطبيق الأندرويد

1-تشغيل كود app.py

2-تشغيل محاكي الاندرويد المج أو استخدام جهازي حقيقي

3-الانتقال باستخدام سطر الأوامر CLI إلى مسار التطبيق

4-تنفيذ الأمر التالي:

Flutter run

# النتائج والخاتمة

في ختام المشروع ، تم إنجاز وتطوير نظام متكامل وذكي لإدارة عمليات مركز شبكة MTN (NOC).

لا يمثل هذا النظام مجرد أداة تقنية جديدة، بل هو تحول استراتيجي ينقل عمليات مراقبة الشبكة وصيانتها من نموذج العمل التقليدي القائم على رد الفعل إلى نهج استباقي، ذكي، ومبني على البيانات. لقد تم تصميم هذا النظام لمواجهة التحديات الجوهرية المتمثلة في التعامل اليدوي مع الأعطال، وبطء زمن الاستجابة، وصعوبة تتبع الأداء بشكل فعال.

### 1. ملخص الإنجازات والأثر التشغيلي

لقد نجح المشروع في تحقيق أهدافه الأساسية من خلال بنية متكاملة ومترابطة، حيث يقدم كل مكون قيمة فريدة للنظام ككل:

- **محرك الذكاء الاصطناعي(AI Engine) :** هو قلب النظام المبتكر، حيث يقوم بتحويل البيانات الأولية للأعطال إلى رؤى قابلة للتنفيذ. يقوم المحرك تلقائيًا بتحليل آلاف الأعطال الواردة، وتصنيفها، وتحديد أولويتها، واقتراح الفريق المسؤول والحلول المحتملة بناءً على الأنماط التاريخية. الأهم من ذلك، هو قدرته على التنبؤ بالأعطال المستقبلية، مما يتيح فرصة لإجراء صيانة وقائية استباقية، الأمر الذي يقلل من انقطاع الخدمة ويحسن من موثوقية الشبكة ورضا العملاء بشكل مباشر.
- **لوحة التحكم الويب(Dashboard) :** تمثل لوحة التحكم المركز العصبي للنظام، حيث توفر للمشرفين والمشغلين رؤية بانورامية وشاملة لحالة الشبكة في الزمن الحقيقي. تتجاوز اللوحة مجرد عرض البيانات، لتقدم تحليلات ورسومًا بيانية تفاعلية حول مؤشرات الأداء الرئيسية (KPIs)مثل تردد الأعطال، متوسط زمن الإصلاح(MTTR) ، وأداء الفرق. كما تتيح إدارة مركزية للمستخدمين وصلاحياتهم، مما يضمن بيئة عمل آمنة ومنظمة.
- **تطبيق الهاتف المحمول(Mobile App) :** يُعد هذا التطبيق بمثابة الذراع الميدانية للنظام، حيث يُمكّن المهندسين والفنيين من التفاعل مع الأعطال بكفاءة غير مسبوقة. فمن خلال الإشعارات الفورية، والوصول السريع إلى تفاصيل الأعطال، والحلول المقترحة من قبل الذكاء الاصطناعي، يستطيع الفريق الميداني تشخيص المشكلات وإصلاحها بسرعة أكبر. كما أن إمكانية تحديث حالة التذاكر وإضافة الملاحظات مباشرة من الموقع تُسرّع من دورة العمل وتقلل من الأعباء الإدارية.

### 2. التحديات والدروس المستفادة

واجه المشروع خلال مراحل التطوير مجموعة من التحديات التي تم التغلب عليها بنجاح، مما أكسب الفريق خبرات قيمة. من أبرز هذه التحديات:

- **تكامل البيانات :**كان توحيد البيانات الواردة من مصادر وأنظمة متعددة ومختلفة (كما يظهر في بنية قاعدة البيانات) تحدياً كبيراً، وتطلب جهدًا في تنظيف البيانات وهيكلتها لضمان جودتها.
- **دقة النماذج التنبؤية :**تطلب بناء نماذج الذكاء الاصطناعي وتدريبها دورات متعددة من الاختبار والتحسين للوصول إلى مستوى دقة مقبول يمكن الاعتماد عليه في بيئة العمل الحقيقية.
- **تصميم تجربة المستخدم (UI/UX)** كان من الضروري تصميم واجهات سهلة الاستخدام لكل من لوحة التحكم وتطبيق الهاتف، لتلبية احتياجات مختلف فئات المستخدمين (من المشغلين إلى الفنيين والمديرين) وضمان تبنيهم السريع للنظام.

في المحصلة، يمثل هذا المشروع أكثر من مجرد نظام لإدارة الأعطال؛ إنه أساس متين لثقافة عمل جديدة في مركز عمليات الشبكة، تعتمد على البيانات والكفاءة والعمل الاستباقي. لقد أثبت النظام قدرته على تعزيز الكفاءة التشغيلية، وتقليل التكاليف، والأهم من ذلك، تحسين جودة الخدمة المقدمة لمشتركيMTN ، مما يجعله ركيزة أساسية لأي مبادرات تحول رقمي مستقبلية في الشركة.

# الأفاق المستقبلية

- **تطوير الذكاء الاصطناعي:**
  - دمج نماذج تعلم معزز (Reinforcement Learning) لتمكين النظام من التعلم ذاتيًا من نتائج الحلول المقترحة، مما يحسن من دقة توصياته المستقبلية.
  - إضافة وحدات معالجة اللغات الطبيعية (NLP) لتحليل الملاحظات النصية التي يدونها الفنيون، واستخلاص حلول ومعارف جديدة بشكل آلي.
- **توسيع التكامل مع الأنظمة الأخرى:**
  - ربط النظام مع أنظمة إدارة المخزون (Inventory Management) لربط الأعطال بقطع غيار ومعدات محددة.
  - التكامل مع نظام إدارة علاقات العملاء (CRM) لربط الأعطال الفنية بشكاوى العملاء بشكل مباشر، وتقديم رؤية أشمل لتأثير الأعطال على تجربة المستخدم.
- **تحسينات وظيفية:**
  - إضافة ميزات التعاون والتواصل المباشر (Chat) داخل تطبيق الهاتف لتمكين الفرق الميدانية من التنسيق فيما بينها أثناء التعامل مع الأعطال المعقدة.
  - تطوير قدرات العمل دون اتصال بالإنترنت (Offline Mode) في تطبيق الهاتف، لضمان استمرارية العمل في المناطق ذات التغطية الضعيفة.
- **تطوير الذكاء الاصطناعي والتعلم الآلي**
- **تحليل السبب الجذري :(Root Cause Analysis – RCA)** الانتقال من مجرد توقع العطل التالي إلى تحليل سلاسل الإنذارات المتزامنة لتحديد "السبب الجذري" الحقيقي للمشكلة. سيؤدي ذلك إلى تقليل "ضوضاء الإنذارات" وتوجيه الفرق الفنية مباشرةً إلى مصدر الخلل الأساسي، بدلاً من التعامل مع الأعراض المتعددة.
- **التذاكر والإصلاح الآلي(Automated Ticketing & Remediation) :** تطوير النظام ليقوم ليس فقط بإنشاء التذاكر، بل بتنفيذ إجراءات الإصلاح الأولية بشكل آلي للمشاكل المتكررة والمعروفة (مثل إعادة تشغيل منفذ شبكة أو خدمة معينة). في حال فشل الإجراء الآلي، يتم تصعيد التذكرة للتدخل البشري مع تزويد الفني بتقرير مفصل عن الخطوات التي تم اتخاذها.

- **التخصيص الذكي للموارد(Intelligent Resource Allocation) :** استخدام التعلم الآلي لتحليل أداء الفنيين والفرق المختلفة، ومن ثم إسناد التذاكر تلقائيًا إلى المورد (فني أو فريق) الأكثر كفاءة وخبرة في التعامل مع نوع معين من الأعطال، مما يحسن من سرعة وجودة الإصلاح ويقلل من التكاليف.

- **الكشف عن الحالات الشاذة(Anomaly Detection) :** تدريب نماذج قادرة على اكتشاف سلوكيات غير طبيعية في الشبكة لا تُصدر إنذارات قياسية، مثل الارتفاع التدريجي في زمن الاستجابة (Latency) أو فقدان الحزم(Packet Loss) ، مما يسمح بمعالجة المشاكل المحتملة قبل أن تؤثر على جودة الخدمة.

- **التكامل المعمّق مع الأنظمة والبنية التحتية (Deeper Integration with Systems & Infrastructure)**

- **بنية الخدمات المصغّرة :(Microservices Architecture)** إعادة هيكلة التطبيق الخلفي (Backend) إلى خدمات مصغّرة ومستقلة (مثل خدمة للمصادقة، خدمة للتذاكر، خدمة للإنذارات). هذا التحول سيعزز من قابلية التوسع والمرونة، ويسهل صيانة وتطوير كل جزء من النظام على حدة دون التأثير على الأجزاء الأخرى.

- **النشر السحابي :(Cloud-Native Deployment)** ترحيل النظام إلى بيئة سحابية مثل AWS, Azure, or GCPباستخدام تقنيات الحاويات (Docker) ومنصات التنسيق (Kubernetes)سيضمن ذلك التوافرية العالية، المرونة في التوسع حسب الحاجة، والاستفادة من الخدمات السحابية المدارة لقواعد البيانات والذكاء الاصطناعي.

- **التكامل المباشر مع أنظمة إدارة الشبكة(NMS/EMS) :** إنشاء روابط مباشرة مع أنظمة إدارة عناصر الشبكة الخاصة بالموردين مثل Huawei MAE, Ericsson OSS

للحصول على بيانات آنية وتنفيذ أوامر التشخيص والإصلاح مباشرة من داخل النظام، مما يقلل الاعتماد على الواجهات المتعددة.

إن تبني هذه الرؤية المستقبلية سيحول المشروع من نظام إدارة أعطال فعال إلى منصة عمليات شبكة ذكية واستباقية، تشكل العمود الفقري الرقمي لعمليات MTN وتضمن تقديم خدمات عالية الموثوقية والجودة.

# المراجع

**المراجع العربية:**


- عرودكي، هشام .(2011) **نظم الاتصالات الخلوية: أسس ومبادئ** .دمشق: دار الفكر.
- آستور، سيرا .(2025) **مقرر شبكات الحاسوب** .الجامعة الافتراضية السورية.
- الخطيب، باسل .(2025) **مقرر تقنيات الويب 1-2-المتقدم** .الجامعة الافتراضية السورية.
- خوري، عبده .(2025) **مقرر نظم قواعد البيانات** .الجامعة الافتراضية السورية.
- سلمان، عصام .(2025) **مقرر تقنيات تعلم الآلة** .الجامعة الافتراضية السورية.
- المصطفى، محمد مازن .(2025) **مقرر تطوير تطبيقات الموبايل** .الجامعة الافتراضية السورية.
- طعيمة، ع. (تاريخ غير متوفر) .**تعلم الآلة وعلم البيانات: الأساسيات والمفاهيم والخوارزميات والأدوات)** .


**المراجع الأجنبية:**


- Aidarous, S., & Plevyak, T. (1998). Telecommunication Network Management. IEEE Press.
- Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow, 2nd Edition. O'Reilly Media.
- Sangaiah, A. K., et al. (2022). Automatic Fault Detection and Diagnosis in Cellular Networks and Beyond 5G: Intelligent Network Management. Algorithms, 15(11), 432.
- Li, Y., et al. (2021). A Nationwide Study on Cellular Reliability: Measurement, Analysis, and Enhancements. Proceedings of the ACM SIGCOMM 2021 Conference.
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow* (2nd ed.). O'Reilly Media.
- Sangaiah, A. K., et al. (2022). Automatic Fault Detection and Diagnosis in Cellular Networks and Beyond 5G: Intelligent Network Management.
- Aidarous, S., & Plevyak, T. (1998). *Telecommunication Network Management*. IEEE Press.
- Grinberg, M. (2018). *Flask Web Development, 2nd Edition*. O'Reilly Media.

**المواقع الإلكترونية:**


- The Pallets Projects. (2023). *Flask Documentation*. Available at: https://flask.palletsprojects.com/
- Thales Group. Generations of mobile networks explained. Available at: https://www.thalesgroup.com/en/markets/digital-identity-and-security/inspired/basics-of-mobile-networking/milestones
- Thales Group. Generations of mobile networks explained. Available at: https://www.thalesgroup.com/en/markets/digital-identity-and-security/inspired/basics-of-mobile-networking/milestones
- Asentria. Mobile Network Failures – Some Causes to Think About. Available at: https://www.asentria.com/mobile-network-failures-causes/
- Wikipedia. Cellular network. Available at: https://en.wikipedia.org/wiki/Cellular_network

# الملحق

# Appendix

يحتوي هذا الملحق على المكونات البرمجية الأساسية التي تمثل العمود الفقري لنظامMTN NOC ، وتشمل مخطط قاعدة البيانات، والكود المصدري للواجهة الخلفية، وأمثلة من واجهات المستخدم.

### 1. قاعدة البيانات(Database Schema)

الوصف :يوضح هذا الجزء الكود الكامل لإنشاء قاعدة البيانات mtn_noc_systemبجميع جداولها (users, alarms, tickets, sites...)، بالإضافة إلى العلاقات بينها والمفاتيح والفهارس التي تم إنشاؤها لضمان تكامل البيانات وتحسين أداء الاستعلامات.

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
CREATE TABLE `alarms` (
  `root_alarm` varchar(500) DEFAULT NULL,
  `severity` varchar(500) DEFAULT NULL,
  `alarm_id` int(11) NOT NULL,
  `name` varchar(500) DEFAULT NULL,
  `ne_type` varchar(500) DEFAULT NULL,
  `site_code` varchar(500) DEFAULT NULL,
  `alarm_source` text DEFAULT NULL,
  `mo_name` text DEFAULT NULL,
  `location_info` text DEFAULT NULL,
  `additional_info` text DEFAULT NULL,
  `occurred_on` datetime DEFAULT NULL,
  `cleared_on` datetime DEFAULT NULL,
  `acknowledged_on` varchar(500) DEFAULT NULL,
  `acknowledged_by` varchar(500) DEFAULT NULL,
  `rru_name` varchar(500) DEFAULT NULL,
  `bbu_name` varchar(500) DEFAULT NULL,
  `enodeb_id` varchar(500) DEFAULT NULL,
  `log_serial_number` varchar(500) DEFAULT NULL,
  `user_label` varchar(500) DEFAULT NULL,
  `equipment_alarm_serial` varchar(500) DEFAULT NULL,
  `maintenance_status` varchar(500) DEFAULT NULL,
  `subnet` varchar(500) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

CREATE TABLE `alarms_ai` (
  `ID` int(11) NOT NULL,
  `alarm_id` int(11) NOT NULL,
  `Severity` varchar(20) DEFAULT NULL,
  `Name` varchar(100) NOT NULL,
  `SiteCode` varchar(20) NOT NULL,
  `OccurredOn` datetime NOT NULL,
  `Technology` varchar(10) DEFAULT NULL CHECK (`Technology`
in ('2G','3G','4G','UNKNOWN')),
  `Responsible` varchar(50) DEFAULT NULL,
  `HumanIntervention` varchar(1) DEFAULT '0',
  `ProposedSolution` text DEFAULT NULL,
```

```
  `Priority` varchar(255) DEFAULT NULL,
  `RepairDuration` float DEFAULT NULL,
  `NextFault` varchar(100) DEFAULT NULL,
  `CreatedAt` datetime NOT NULL,
  `LastUpdated` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

CREATE TABLE `tickets` (
  `id` int(11) NOT NULL,
  `alarm_ai_id` int(11) NOT NULL,
  `status` enum('Open','In Progress','Resolved','Closed')
NOT NULL DEFAULT 'Open',
  `assigned_to_user_id` int(11) DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT
current_timestamp(),
  `updated_at` timestamp NOT NULL DEFAULT
current_timestamp() ON UPDATE current_timestamp(),
  `resolved_at` timestamp NULL DEFAULT NULL,
  `resolution_notes` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;
CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `username` varchar(50) NOT NULL,
  `password` varchar(255) NOT NULL,
  `full_name` varchar(100) NOT NULL,
  `email` varchar(100) DEFAULT NULL,
  `phone` varchar(20) DEFAULT NULL,
  `role`
enum('admin','noc_operator','technician','supervisor') NOT
NULL DEFAULT 'technician',
  `department` varchar(50) DEFAULT NULL,
  `is_active` tinyint(1) DEFAULT 1,
  `created_at` timestamp NOT NULL DEFAULT
current_timestamp(),
  `updated_at` timestamp NOT NULL DEFAULT
current_timestamp() ON UPDATE current_timestamp(),
  `last_login` timestamp NULL DEFAULT NULL,
  `login_count` int(11) DEFAULT 0
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

ALTER TABLE `alarms` ADD PRIMARY KEY (`alarm_id`);
ALTER TABLE `alarms_ai` ADD PRIMARY KEY (`ID`), ADD UNIQUE
KEY `alarm_id` (`alarm_id`);
ALTER TABLE `tickets` ADD PRIMARY KEY (`id`), ADD KEY
`alarm_ai_id` (`alarm_ai_id`), ADD KEY `assigned_to_user_id`
(`assigned_to_user_id`);
ALTER TABLE `users` ADD PRIMARY KEY (`id`), ADD UNIQUE KEY
`username` (`username`);
```

```
ALTER TABLE `alarms` MODIFY `alarm_id` int(11) NOT NULL
AUTO_INCREMENT;
ALTER TABLE `alarms_ai` MODIFY `ID` int(11) NOT NULL
AUTO_INCREMENT;
ALTER TABLE `tickets` MODIFY `id` int(11) NOT NULL
AUTO_INCREMENT;
ALTER TABLE `users` MODIFY `id` int(11) NOT NULL
AUTO_INCREMENT;

ALTER TABLE `alarms_ai`
  ADD CONSTRAINT `alarms_ai_ibfk_1` FOREIGN KEY (`alarm_id`)
REFERENCES `alarms` (`alarm_id`);
ALTER TABLE `tickets`
  ADD CONSTRAINT `tickets_ibfk_1` FOREIGN KEY
(`alarm_ai_id`) REFERENCES `alarms_ai` (`ID`) ON DELETE
CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `tickets_ibfk_2` FOREIGN KEY
(`assigned_to_user_id`) REFERENCES `users` (`id`) ON DELETE
SET NULL ON UPDATE CASCADE;

CREATE INDEX `idx_cleared_on` ON `alarms`(`cleared_on`);
CREATE INDEX `idx_occurred_on` ON `alarms_ai`(`OccurredOn`);
```

2. الكود المصدري لتدريب النماذج train models

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier,
RandomForestRegressor
from sklearn.linear_model import LogisticRegression
from sklearn.multioutput import MultiOutputClassifier
import xgboost as xgb
import joblib

# 1. تحميل البيانات ودمجها
df_historical = pd.read_csv("HistoricalData.csv")
df_alarmstr = pd.read_csv("alarmstr.csv")
df = pd.merge(df_historical, df_alarmstr, on='Name',
how='left')

# 2. معالجة البيانات واستخراج الميزات
df['ResolutionTime_Minutes'] = (pd.to_datetime(df['Cleared
On (NT- (['(
                                pd.to_datetime(df['Occurred
On (NT)'])).dt.total_seconds() / 60
df['Technology'] = df.apply(lambda row: "2G/3G/4G/UNKNOWN",
axis=1)
target_columns = ['Technology', 'Responsible', 'Human
Intervention', 'Proposed Solution', 'Priority']
```

```
df['text'] = df[['Name','NE Type','Responsible','Proposed
Solution']].fillna('-').agg(' '.join, axis=1)

# 3. تحويل النصوص والتقسيم
vectorizer = TfidfVectorizer(max_features=500)
X_sparse = vectorizer.fit_transform(df['text'])
Y = df[target_columns].apply(LabelEncoder().fit_transform)
y_res = df['ResolutionTime_Minutes']
X_train, X_test, y_train, y_test, y_train_res, y_test_res =
train_test_split(X_sparse, Y, y_res, test_size=0.2)

# 4. تدريب نماذج التصنيف المتعدد
models = {
'RandomForest': RandomForestClassifier(n_estimators=100,
max_depth=10, min_samples_leaf=4, random_state=42),
'LogisticRegression': LogisticRegression(max_iter=300,
C=0.5, solver='saga', random_state=42),
'XGBoost': xgb.XGBClassifier(n_estimators=150,
learning_rate=0.1, max_depth=5, use_label_encoder=False,
eval_metric='mlogloss', random_state=42)
}
for name, base_model in models.items():
    X_train = X_train_sparse
    X_test = X_test_sparse

    try:
        multi_model = MultiOutputClassifier(base_model)
        multi_model.fit(X_train, y_train)
        y_pred = multi_model.predict(X_test)

        f1_scores = []
        print(f"\n=== model: {name} ===")
        for i, col in enumerate(target_columns):
            labels =
list(range(len(label_encoders[col].classes_)))
            report = classification_report(
                y_test.iloc[:, i],
                y_pred[:, i],
                labels=labels,
                target_names=label_encoders[col].classes_,
                zero_division=0,
                output_dict=True
            )
            f1 = report['weighted avg']['f1-score']
            f1_scores.append(f1)
MultiOutputClassifier(base_model).fit(X_train, y_train)

# 5. نموذج التنبؤ بزمن الحل
resolution_time_model = RandomForestRegressor().fit(X_train,
y_train_res)
```

```
# 6. نموذج التنبؤ بالعطل التالي
all_unique_faults = pd.concat([sequence_df['CurrentFault'],
sequence_df['NextFault']]).unique()
le_next_fault.fit(all_unique_faults)

X_next_fault =
le_next_fault.transform(sequence_df['CurrentFault']).reshape
(-1, 1)
y_next_fault =
le_next_fault.transform(sequence_df['NextFault'])

X_train_next, X_test_next, y_train_next, y_test_next =
train_test_split(X_next_fault, y_next_fault, test_size=0.2,
random_state=42)

next_fault_model = RandomForestClassifier(n_estimators=100,
max_depth=10, random_state=42)
next_fault_model.fit(X_train_next, y_train_next)

# 7. حفظ النماذج
joblib.dump(multi_model, 'best_multi_model.pkl')
joblib.dump(vectorizer, 'tfidf_vectorizer.pkl')
joblib.dump(resolution_time_model,
'resolution_time_predictor.pkl')
```

3. الكود المصدري للواجهة الخلفية (Backend - Flask)

الوصف :يحتوي هذا الملف (app.py) على كامل منطق الخادم. وهو مسؤول عن تعريف جميع نقاط الوصول (API Endpoints) التي يتصل بها تطبيق الويب وتطبيق الموبايل، ومعالجة طلبات المستخدمين، وإدارة الجلسات، والتفاعل مع قاعدة البيانات لجلب البيانات وتحديثها.

```
from flask import Flask, render_template, request, redirect,
url_for, session, flash, jsonify
from flask_mysqldb import MySQL
from werkzeug.security import check_password_hash,
generate_password_hash
from datetime import datetime
from functools import wraps
import os
import subprocess
import sys

# Create a Flask app instance
app = Flask(__name__)
# Set a secret key for session management
app.secret_key = os.urandom(24)

# Configure MySQL database connection
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = '' # Add your password here
if you have one
```

```
app.config['MYSQL_DB'] = 'mtn_noc_system'
app.config['MYSQL_CURSORCLASS'] = 'DictCursor' # Returns
rows as dictionaries

# Initialize MySQL
mysql = MySQL(app)

# Decorator to ensure a user is an admin
def admin_required(f):
    @wraps(f)
    def decorated_function(*args, **kwargs):
        if 'role' not in session or session.get('role') !=
'admin':
            flash('You do not have permission to access this
page.', 'danger')
            return redirect(url_for('dashboard'))
        return f(*args, **kwargs)
    return decorated_function
@app.route('/', methods=['GET', 'POST'])
def login():
  if request.method == 'POST':
     username = request.form['username']
     password = request.form['password']
     cur = mysql.connection.cursor()
     cur.execute("SELECT * FROM users WHERE username = %s",
     (username,))
      user = cur.fetchone()
      cur.close()
       if user:
         try:
           if
check_password_hash(user['password'],password):
             if user.get('is_active', 0) == 1:
               cur = mysql.connection.cursor()
                cur.execute("""UPDATE user SET last_login =
%s,
                 login_count = login_count + 1
                  WHERE id = %s """, (datetime.now(),
user['id']))
                mysql.connection.commit()
                cur.close()
                session['user_id'] = user['id']
                session['username'] = user['username']
                session['full_name'] = user['full_name']
                session['role'] = user['role']
                flash('Login successful!', 'success')
                return redirect(url_for('dashboard'))
                    else:
                        flash('Your account is disabled.
Please contact administrator.', 'danger')
                else:
```

```
                    flash('Invalid username or password',
'danger')
            except ValueError:
                # This handles cases where the password in
the database is not a valid hash.
                flash('Invalid username or password',
'danger')
        else:
            flash('Invalid username or password', 'danger')

    return render_template('login.html')


@app.route('/api/login', methods=['POST'])
def api_login():
    """
    API endpoint for mobile app login.
    Accepts JSON, returns JSON.
    """
    # Get credentials from JSON request body
    data = request.get_json()
    if not data or not data.get('username') or not
data.get('password'):
        return jsonify({"status": "error", "message":
"Username and password are required"}), 400

    username = data['username']
    password = data['password']

    cur = mysql.connection.cursor()
    cur.execute("SELECT * FROM users WHERE username = %s",
(username,))
    user = cur.fetchone()
    cur.close()

    if user and check_password_hash(user['password'],
password):
        if user.get('is_active', 0) == 1:
            # Update last login time
            cur = mysql.connection.cursor()
            cur.execute(
                "UPDATE users SET last_login = %s,
login_count = login_count + 1 WHERE id = %s",
                (datetime.now(), user['id'])
            )
            mysql.connection.commit()
            cur.close()

            # Prepare user data to send back (without the
password hash)
            user_data = {
```

```
                'id': user['id'],
                'username': user['username'],
                'full_name': user['full_name'],
                'role': user['role']
            }
            return jsonify({"status": "success", "message":
"Login successful!", "user": user_data}), 200
        else:
            return jsonify({"status": "error", "message":
"Your account is disabled. Please contact administrator."}),
403
    else:
        return jsonify({"status": "error", "message":
"Invalid username or password"}), 401

@app.route('/api/tickets', methods=['GET'])
def get_tickets_data():
    """
    API endpoint to fetch tickets data from the database.
    This joins tickets, alarms_ai, and users tables.
    """
    try:
        cur = mysql.connection.cursor()

        # Fetch tickets data with all necessary joins
        cur.execute("""
            SELECT
                t.id,
                t.status,
                t.resolved_at,
                t.resolution_notes,
                ai.SiteCode,
                ai.Name AS AlarmName,
                ai.OccurredOn,
                ai.Priority,
                u.full_name AS assigned_to
            FROM tickets AS t
            JOIN alarms_ai AS ai ON t.alarm_ai_id = ai.ID
            LEFT JOIN users AS u ON t.assigned_to_user_id =
u.id
            ORDER BY t.created_at DESC
        """)
        tickets_data = cur.fetchall()
        cur.close()

        # Convert datetime objects to strings for JSON
serialization
        for ticket in tickets_data:
            for key, value in ticket.items():
                if isinstance(value, datetime):
                    ticket[key] = value.isoformat()
```

```
        return jsonify(tickets=tickets_data)

    except Exception as e:
        print(f"Database connection error in /api/tickets:
{e}")
        return jsonify({"error": f"Database connection
failed: {str(e)}"}), 500

# ... (Other routes like dashboard, charts, users
management, etc.)

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

4. مثال على واجهة الويب(Web Frontend)

الوصف :يوضح هذا الكود (login.html) بنية صفحة تسجيل الدخول في بوابة الويب. تم استخدام HTML مع قوالب Jinja2 ({{ ... }}) للتفاعل مع خادم Flask وعرض الرسائل الديناميكية للمستخدم (مثل رسائل الخطأ أو النجاح).

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Network Alarm System - MTN Syria NOC Portal</title>
    <link rel="stylesheet" href="{{ url_for('static',
filename='style.css') }}">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-
beta3/css/all.min.css">
</head>
<body>
    <div class="background"></div>
    <div class="login-container">
        <div class="login-header">
            <div class="logo-container">
                <img src="{{ url_for('static',
filename='logo.png') }}" alt="MTN Syria Logo" class="logo">
            </div>
            <h1>Network Alarm System</h1>
            <p>MTN Syria - NOC Portal</p>
        </div>

        {% with messages =
get_flashed_messages(with_categories=true) %}
            {% if messages %}
                {% for category, message in messages %}
                    <div class="alert alert-{{ category }}">
```

```
                        <i class="fas fa-{{ 'check-circle' if
category == 'success' else 'exclamation-circle' if category ==
'danger' else 'info-circle' }}"></i>
                        {{ message }}
                    </div>
                {% endfor %}
            {% endif %}
        {% endwith %}

        <form method="POST" action="/">
            <div class="form-group">
                <label for="username">Username</label>
                <div class="password-input-container">
                    <input type="text" name="username"
id="username" placeholder="admin" required>
                </div>
            </div>

            <div class="form-group password-group">
                <label for="password">Password</label>
                <div class="password-input-container">
                    <input type="password" name="password"
id="password" placeholder="*********" required>
                    <button type="button" class="toggle-password"
aria-label="Show password">
                        <i class="fas fa-eye"></i>
                    </button>
                </div>
            </div>

            <div class="separator"></div>

            <button type="submit" class="login-button">SIGN
IN</button>
        </form>

        <div class="login-footer">
            <p>&copy; 2025 MTN Syria. All rights reserved.</p>
        </div>
    </div>

    <script>
        document.querySelector('.toggle-
password').addEventListener('click', function() {
            const passwordInput =
document.getElementById('password');
            const icon = this.querySelector('i');

            if (passwordInput.type === 'password') {
                passwordInput.type = 'text';
                icon.classList.remove('fa-eye');
                icon.classList.add('fa-eye-slash');
            } else {
                passwordInput.type = 'password';
                icon.classList.remove('fa-eye-slash');
                icon.classList.add('fa-eye');
```

```
            }
        });
    </script>
</body>
</html>
```

# Abstract

Cellular communication networks are the cornerstone of our modern interconnected world, having revolutionized the way individuals and businesses communicate. Since the emergence of the first generation (1G) in the late 1970s, these networks have evolved tremendously, transitioning from providing simple analog voice calls to offering high-speed data services that support complex applications like high-definition video, augmented reality, and the Internet of Things (IoT). Mobile phones and other wireless devices have become an integral part of our daily lives, making reliance on these networks critically important. As this dependency grows, so does the importance of understanding the failures and challenges facing these complex systems.

Failures in cellular communication networks can lead to service disruptions, significant economic losses, and negative impacts on the user experience. Telecommunication companies like MTN Syria face significant challenges in managing network failures due to the increasing complexity of the infrastructure and the high volume of alarms generated by the Network Operations Center (NOC).

This project aims to develop an intelligent, integrated system that uses artificial intelligence techniques to analyze and manage fault alarms from the Network Operations Center (NOC) at MTN Syria. It will link these alarms directly with a live Ticketing System to expedite the response and resolution process. The project also seeks to represent faults on an interactive map using Geographic Information Systems (GIS) to provide an accurate and immediate visual of fault locations. Additionally, it will send real-time notifications to technical support staff via an Android application, enabling them to view details and take appropriate action. The project's significance lies in its step towards automating fault management and reducing reliance on manual procedures, thereby enhancing service quality and reducing recovery time (MTTR).

**Syrian Arab Republic**
**Ministry of Higher Education**
**Syrian Virtual University**
**Master in Computer Science**

# Design and development of an intelligent system for analyzing and managing cellular network failure alerts in MTN Syria operating environment

A project prepared for a Master's degree in Computer Science

**By**
Eng.Obada Ezz Al-Din

**Supervised by**
Dr. Eng. Sira Astour

**Academic year**
2024-2025