تفصيل أجزاء النظام: لدينا العديد من الأجزاء مثل النماذج المدربه مثل XGBoost.pkl, Random\_forest.pkl

وملفات الاكواد مثل predict.py, iot\_detection.zeek, dashboard.py

حيث أن zeek هو لمراقبة الشبكة خيث يقوم بتجميع الحزم المتفرقه الى اتصالات ومن ثم يقوم بانشاء سطر لكل اتصال هذا السطر يمثل السمات وقيمها ويكتب هذا السطر في ملف io\_t\_detection.log

ولدينا الأمر tail -f io\_t\_detection.log > /tmp/zeek\_pipe وهو يقوم بقؤاءة الاسطر الجديده من ملف io\_t\_detection.log وارسالها الى قناة pipe

ولدينا كود التنبؤ predict.py الذي يقوم بتحميل نموذج تعلم الاله المدرب ويفوت بحلقة لقراءة الاسطر الخاصه بالقناة zeek\_pipe ومن أجل كل سطر يقوم بمعالجته المعالجة المطلوبه لبصيح الدخل للمودل مثل البيانات التي تدرب عليها، ومن ثم يتنبأ بماهية الاتصال ان كان benign أو malicious باستخدام المودل المحمّل المدرّب.

ويوجد كود dashboard.py ةهة يقةم بقراءة خرج الpredict.py ويعرض النتائج بشكل اوضح للمستخدم

تم الدمج (integration) بينهم كما التالي:

حزمة بيانات الشبكة ← أداة ←zeek ملف السجلio\_t\_detection.log ← قناة الاتصال ←(Pipe) كود التنبؤ ← طباعة النتيجة ← سكربت واجهة العرض dashboard

## خطة الاختبارات

Integration tests: إنّ المحاكاة التي تم توظيف المودل بها تشمل اختبارات الدمج لأن المحاكاة تدمج المودل مع اكواد مراقبة البيانات وتحليلها zeek وكود الكشف predit.py وعرض النتائج dashboard مع الهجوم

Working tests: أيضا المحاكاة التي قمنا بها تسمح بعمل اختبارات لمعرفة اذا كان كود الكشف قادر على التنبؤ بشكل صحيح ام لا اعتمادا على النماذج المدرّبة

# النموذج المقترح

Figure 1 System block diagram

يوضح هذا المخطط سير العمل الشامل لمشروعنا هذا حيث يمكن تقسيمه الى مرحلتين أساسيتين هم مرحلة التدريب ومرحلة المحاكاة

حيث فب المرحلة الأولى: قمنا بتنزيل ملفات conn.log.labeled ثم طبقنا عليهم كود بايثةن لاستخراج ملفات الداتاست منهم بصيغة .csv ثم طبقنا كود بايثون على كل ملف .csv من أجل معالجة البيانات قبل ادخالها للتدريب والاختبار وعندها نتج لدينا 16 ملف لبيانات معالجه فقمنا بدمجهم جميعهم الى ملف بيانات كلي تحت اسم ‘combined\_dataset.csv’ ثم قمنا بتدريب 4 أنواع من المصنفات منهم 3 تعلم الي (machine learning) ومصنف تعلم الي عميق (deep learning) وهم:

Random forest, lightgbm, xgboost, neural network

وبالطبع قمنا بحفظ المودل المدرب لكل مصنف منهم.

وفي المرحلة الثانية: شغلنا 2 ubuntu virtual machines واحده للكشف وواحدة للهجوم حيث الة الهجوم جرت هجوما على الة الكشف التي تحتوي النموذج المدرب ليتنبأ بماهية الاتصالات القادمة من الattacking machine ونرى فعاليته بكشف الهجومات في الوقت الحقيقي.