Proyecto de Investigación

KASSANDRA: APLICACIÓN DE TÉCNICAS DE ML PARA LA DETECCIÓN DE COMPORTAMIENTOS ANÓMALOS DE USUARIOS

MÁSTER EN INGENIERÍA WEB

TRABAJO DE FIN DE MÁSTER

AUTOR

Alba Cotarelo Tuñón

TUTOR

José Manuel Redondo López

Junio 2021

Agradecimientos

En este año tan complicado para todos en muchísimos términos, quiero comenzar dándole las gracias a Redondo por ser un director de proyecto de 10 en todos los aspectos y por cuidar a sus proyectantes tanto como lo hizo.

A mi familia por haberme sufrido y animado durante todo esto.

A mis amigos y amigos-familia por compartir con ellos los momentos de ocio virtual en cuarentena y luego en la desescalada para desconectar del mundo académico y profesional.

Al OOT Lab por haberme acompañado y aconsejado en los inicios de este proceso llamado "hacer la ciencia".

A mi equipo de baloncesto por devolverme la motivación en un momento clave del año y celebrar conmigo cada vez que terminaba una sección del artículo.

Por último, gracias a Antonio por, aun con todos los altibajos, haberme hecho un poco más fácil estos dos años de máster y año y medio de pandemia. *We didn't start the fire*.

Resumen

Este artículo describe una técnica para implementar eficientemente las funcionalidades de los UEBA utilizando un reciente algoritmo de aprendizaje no supervisado llamado Extended Isolation Forests. El prototipo diseñado para probar esta técnica demostró una adecuada capacidad de detección de anomalías en el comportamiento del usuario analizando datos de peticiones HTTP de usuarios reales sobre un conjunto de casos de uso típicos. Estos casos fueron diseñados para modelar los escenarios habituales en los que se puede realizar un ataque. Los resultados son lo suficientemente prometedores como para extender la técnica a más protocolos, empresas de bajo presupuesto que no pueden permitirse soluciones UEBA comerciales o entornos educativos.

Abstract

This paper describes a technique to efficiently implement UEBA functionalities using a recent unsupervised learning algorithm named *Extended Isolation Forests*. The prototype designed to test this technique demonstrated adequate user behavior anomaly detection capabilities analyzing real-user HTTP request data over a set of typical use cases. These cases were designed to model typical scenarios where an attack may be performed. The results are promising enough to extend the technique to more protocols, low-budget companies that cannot afford commercial UEBA solutions, or educational environments.

Resume

Esti artículu describe una técnica pa implementar eficientemente les funcionalidaes de los UEBA utilizando un nuevu algoritmu d'aprendizaxe non supervisáu nomáu Extended Isolation Forests. El prototipu diseñáu pa probar esta técnica amosó una afayadiza capacidá detección d'anomalíes nel comportamientu l'usuariu analizando datos de peticiones HTTP d'usuarios reales sobro un conxuntu casos d'usu típicos. Estos casos fueron diseñaos pa modelar los escenarios habituales nos que se pue realizar un ataque. Los resultaos son lo suficientemente prometeores como pa estender la técnica a más protocolos, empreses de baxu presupuestu que nun puen permitise soluciones UEBA comerciales o contestos educativos.

Índice de contenido

Índice de Figuras

Índice de Tablas

Capítulo 1 Introducción

1.1 MOTIVACIÓN

La correcta implementación de los estándares internacionales de gestión de la seguridad de la información (ISO 27001, ISMS) [1] debe asegurar una correcta configuración de los sistemas de una infraestructura en lo relativo a seguridad. Sin embargo, la seguridad de los sistemas pasa también por ejercer una constante labor de monitorización y vigilancia de lo que ocurre en cada uno de ellos, de manera que eventos considerados ataques o bien anomalías puedan ser detectadas y notificadas para responder a ellas de manera temprana y prevenir daños mayores.

Habitualmente se identifica esta labor de vigilancia y monitorización con los *Intrusion Detection Systems (IDS)*, que generan alertas una vez se ha producido el ataque, y los *Intrusion Prevention Systems (IPS)*, que, además, monitorizan el tráfico lo bloquean mientras el ataque se produce [2], o también sistemas tipo *Endpoint Detection and Response (EDR)*, que complementan soluciones de gestión y antivirus para endpoints para identificar y eliminar amenazas [3]. No obstante, más recientemente estos sistemas también se complementan con otros denominados *User and Entity Behaviour Analytics (UEBA)*. Los sistemas anteriores están enfocados a la detección de acciones de usuarios ilegítimas y ataques conocidos, sin embargo, son los usuarios legítimos los que cuentan con más privilegios sobre las infraestructuras. Estos sistemas no están equipados para prevenir ataques desde dentro [4]. Los *UEBAs* basan su comportamiento en la detección de comportamientos anómalos en la red que se desvíen de unas líneas base consideradas como patrones de comportamiento habituales.

Las plataformas que implementan *UEBAs* actualmente son habitualmente costosas y requieren una infraestructura que puede no estar al alcance de muchos negocios. Como las soluciones de *Aruba*, *Exabeam* o *Varonis*. Esto hace que no puedan disponer de una capa de seguridad adicional que hoy en día es cada vez más necesaria.

1.2 FINALIDAD DEL PROYECTO

Este proyecto de investigación pretende implementar un sistema tipo *UEBA* capaz de detectar anomalías en el comportamiento de los usuarios usando un algoritmo moderno de aprendizaje no supervisado denominado *Extended Isolation Forest (EIF)* [5], el cual es capaz de aislar las anomalías de un conjunto de datos mediante particiones recursivas reduciendo los falsos positivos respecto a los *Isolation Forest (IF)* tradicionales. Con él, el sistema debe ser capaz de detectar las anomalías que se pudiesen producir en el tráfico que recibe un servidor atendiendo a la hora a la que se reciben dichas peticiones.

El prototipo diseñado tiene un coste de recursos limitado que le permite ser usado en entornos con una menor capacidad de procesamiento y también en entornos educativos, para que los alumnos vean de primera mano las capacidades de un sistema de este tipo, lo que va en beneficio de su futura preparación profesional.

Para probar la viabilidad de esta técnica, hemos limitado el análisis realizado a peticiones HTTP, otros protocolos serán añadidos en el futuro. El prototipo implementado consigue detectar comportamientos anómalos para una serie de casos de uso típicos y en base a ciertas características de las peticiones HTTP. Incluye también un prefiltrado para evitar el coste computacional de la ejecución de los *Extended Isolation Forest* en caso de que alguna de las características no esté presente en los datos de entrenamiento. Por último, calcula una puntuación de peligrosidad en función de cuánto se haya desviado esa petición del comportamiento ordinario registrado y sobre la que el usuario administrador pueda realizar las acciones correspondientes.

Capítulo 2 Fijación de Objetivos

2.1 POSIBLES ÁMBITOS DE APLICACIÓN

Los sistemas de seguridad suelen trabajar analizando la red y el comportamiento de los usuarios para detectar acciones no permitidas por unas restricciones previas para acceder a información, alterar ciertos archivos o ejecutar un determinado software. Estas restricciones se suelen implementar con permisos u otro tipo de autorizaciones, accesos mediante contraseña (u otros mecanismos de autorización) y roles de usuario.

Esta aproximación es muy efectiva, pero existen escenarios donde puede que no resulte como se espera debido a varias razones. Si cada usuario de la intranet tiene roles asignados y rutinas de trabajo, puede que, dependiendo de la hora del día, un usuario pueda tener accesos no detectados a recursos no autorizados en los siguientes casos:

- *Errores de configuración*: IDS e IPS dependen de una serie de reglas que pueden no estar correctamente ajustadas o configuradas, o simplemente contiene errores que les impiden capturar determinadas acciones maliciosas. No sólo nos referimos a acciones que se sabe que son maliciosas, sino también a nuevas amenazas que no se consideran hasta el momento que se demuestra que son maliciosas. Asignar restricciones detalladas puede ser difícil y el error humano es siempre una posibilidad.
- *Limitaciones de la configuración*: Puede haber casos en los que las restricciones no se puedan configurar de forma granular o flexible para modelar el comportamiento aceptable del usuario. O tal vez sí, pero es un proceso muy complejo y propenso a errores. ¿Qué sucede si un usuario tiene acceso a un sistema, pero solo a determinadas horas del día? ¿O solo dos días a la semana? ¿O periódicamente? ¿O solo cuando se le asigna esporádicamente a un proyecto? ¿Son los permisos del sistema capaces de modelar tal complejidad sin requerir una supervisión constante? ¿O son incapaces de modelarlos en absoluto?
- *Comportamientos inesperados*: Incluso si los permisos y las restricciones se pueden hacer cumplir adecuadamente, no podemos descartar la posibilidad de que los usuarios, moviéndose dentro de las limitaciones que tienen, puedan superar estas limitaciones de todos modos. Existen técnicas conocidas para eludir ciertas limitaciones, lo que permite a los usuarios acceder a partes del sistema que no deberían, sin pasar por la vigilancia IDS / IPS. Además, incluso sin la evasión, el comportamiento atípico de un usuario dentro de su radio de acción puede indicar un intento de ataque de escalada de privilegios en curso que vigila cuidadosamente qué recursos son accesibles con la identidad del usuario suplantado. Es posible que el software de control de seguridad no pueda detener estas actividades de los usuarios, ya que no se detectarán como maliciosos porque, poniéndolos en un contexto general, no lo son.

Asumimos que hay muchos escenarios en los que los usuarios realizan actividades repetitivas sobre archivos, programas o sistemas remotos como parte de su rutina de trabajo.

En cuanto a los contextos donde se podría utilizar:

- Las soluciones UEBA que se encuentran en el mercado tienen un coste que puede no ser asumido por pequeñas empresas. Este proyecto pretende simplificar y concretar el funcionamiento de este tipo de sistemas para ofrecer una solución que conlleve el uso de menos recursos.
- Los UEBAs cada vez tienen más importancia para la detección de ataques internos o suplantaciones de identidad. Por ello, se debe formar a los estudiantes en este tipo de herramientas para que sean conocedores de las mismas cuando salgan al mercado laboral. Kassandra se diseña como un sistema que, por su sencillez y bajo consumo de recursos, pueda ser usado por estos estudiantes.

2.2 OBJETIVOS

A continuación, se describen brevemente los objetivos de la investigación.

- El prototipo inicial del proyecto Kassandra estará restringido a la detección de anomalías en las peticiones HTTP. Adicionalmente, aunque solo hablamos de usuarios físicos, se podría aplicar el mismo modelo a las peticiones realizadas periódicamente / habitualmente por máquinas.
- El algoritmo se entrenará con registros de peticiones HTTP realizadas previamente con el comportamiento aceptable de los usuarios. Las anomalías dentro de los datos entrenados también se pueden detectar y aislar, por lo que no afectan la detección de anomalías futuras una vez que Kassandra esté en producción.
- El objetivo de Kassandra no es bloquear comportamientos potencialmente maliciosos (esta es la función de IPS), sino ser una plataforma adicional de alerta temprana de actividades sospechosas que pueden eludir otros métodos de detección. Esto significa que Kassandra también puede desarrollar criterios para establecer cuándo vale la pena advertir un comportamiento y medir la "peligrosidad" del comportamiento en sí, dependiendo de qué tan "lejos" se haya desviado del esperado.

Por lo tanto, Kassandra permite un mayor nivel de control sobre las actividades de los usuarios, complementando los enfoques tradicionales de restricción de acceso a los recursos con información sobre las actividades programadas de los usuarios. Kassandra intenta unir la programación de nivel superior de las tareas del usuario con el control de nivel inferior sobre las actividades técnicas reales del usuario, por lo que los usuarios solo pueden realizar lo que están destinados a hacer o solían hacer, y los gerentes que puedan tener menor experiencia técnica podrían configurar los niveles de peligrosidad. Esto también podría facilitar la ubicación de usuarios fraudulentos, usuarios suplantados, indicaciones de escaladas de privilegios o software fraudulento que realiza peticiones que se hacen pasar por usuarios del sistema.

Capítulo 3 Planificación y Gestión del TFM

3.1 PLANIFICACIÓN DEL PROYECTO

En esta sección se encuentran recogidos los apartados relativos a la gestión del proyecto.

3.1.1 Identificación de interesados

En esta sección se muestra la lista de interesados en el proyecto. Además de la proyectante y del director de proyecto se ha tenido en cuenta otro estudiante en calidad de consultor y los posibles usuarios.

Nombre	Cargo	Departamento
Alba Cotarelo Tuñón	Proyectante	Universidad de Oviedo
Antonio Payá González	Estudiante	Universidad de Oviedo
José Manuel Redondo López	Director de Proyecto	Universidad de Oviedo
Alumnos y profesores de seguridad y administración de sistemas	Posibles usuarios	
Administradores de sistemas	Posibles usuarios	

Tabla 1. Lista de interesados

3.1.2 OBS y PBS

En la siguiente tabla se muestran los recursos del OBS (Organization Breakdown Structure) que forma parte del proyecto. En el diagrama, la estructura del mismo.

Nombre del recurso	Tipo	Iniciales	Capacidad máxima	Tasa estándar	Tasa horas extra	Costo/Uso	Acumular	Calendario base
Alba Cotarelo Tuñón	Trabajo	A	100%	23,00 €/hora	0,00 €/hora	0,00 €	Prorrateo	TFM
José Manuel Redondo López	Trabajo	J	20%	25,00 €/hora	0,00 €/hora	0,00 €	Prorrateo	TFM
Antonio Payá González	Trabajo	P	20%	23,00 €/hora	0,00 €/hora	0,00 €	Prorrateo	TFM

Tabla 2. OBS

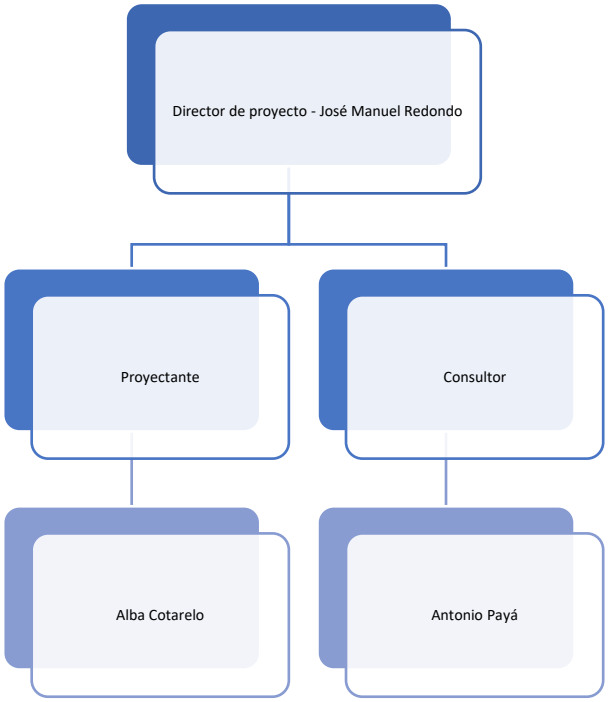


Figura 1. Estructura del OBS

En el siguiente diagrama se presenta el PBS (Product Breakdown Structure), que recoge todos los productos entregables del proyecto.

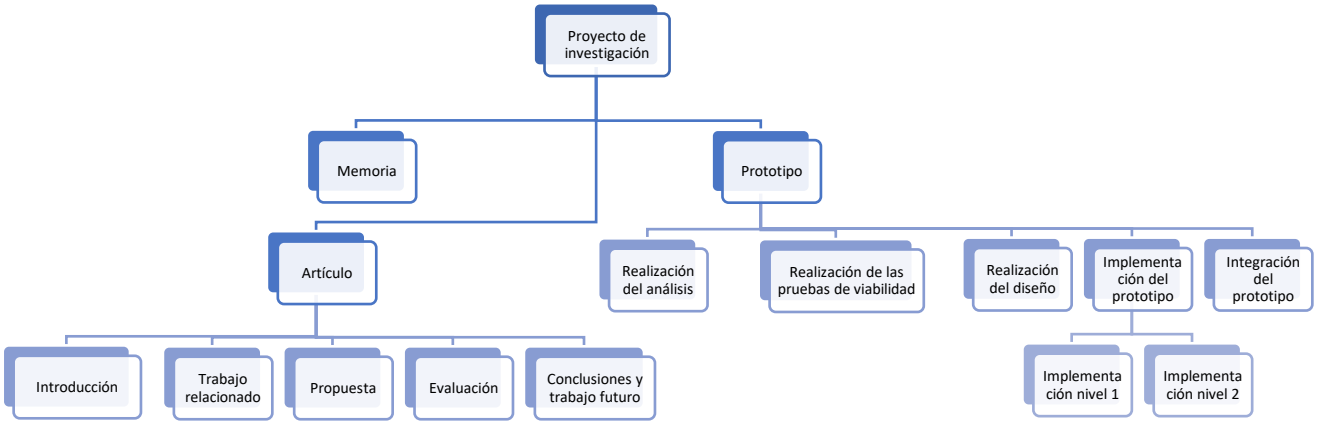


Figura 2. PBS

3.1.3 Planificación inicial. WBS

En este apartado se desglosa el WBS (Work Breakdown Structure) inicial. Para cada tarea se muestra su ID, su duración, fechas de comienzo y fin y tareas predecesoras.

EDT	Tarea	Duración	Comienzo	Fin	Predecesoras
1	**Proyecto de investigación**	**117 días**	**lun 11/01/21**	**jue 03/06/21**	
1.1	Definición de la planificación	0 días	mar 12/01/21	mar 12/01/21	13
1.2	Realización del análisis	0 días	jue 14/01/21	jue 14/01/21	21
1.3	Realización de la gestión de riesgos	0 días	jue 21/01/21	jue 21/01/21	24
1.4	Realización de pruebas de viabilidad	0 días	mar 02/03/21	mar 02/03/21	34
1.5	Realización del diseño	0 días	jue 11/03/21	jue 11/03/21	37
1.6	Implementación del prototipo	0 días	mié 12/05/21	mié 12/05/21	44
1.7	Integración del prototipo	0 días	sáb 15/05/21	sáb 15/05/21	63
1.8	Realización del artículo científico	0 días	sáb 22/05/21	sáb 22/05/21	66
1.9	Redacción de la memoria final del proyecto	0 días	mar 01/06/21	mar 01/06/21	72
1.10	Cierre del proyecto	0 días	mar 01/06/21	mar 01/06/21	77
1.11	Reunión de inicio de proyecto	2 horas	lun 11/01/21	lun 11/01/21	
1.12	Realización de la planificación inicial	2 horas	mar 12/01/21	mar 12/01/21	12
1.13	Realización del presupuesto inicial	2 horas	mar 12/01/21	mié 13/01/21	13
1.14	**Reuniones mensuales**	**97 días**	**jue 04/02/21**	**jue 03/06/21**	
1.14.1	Reuniones mensuales 1	2 horas	jue 04/02/21	jue 04/02/21	
1.14.2	Reuniones mensuales 2	2 horas	jue 04/03/21	jue 04/03/21	
1.14.3	Reuniones mensuales 3	2 horas	jue 01/04/21	jue 01/04/21	
1.14.4	Reuniones mensuales 4	2 horas	jue 06/05/21	jue 06/05/21	
1.14.5	Reuniones mensuales 5	2 horas	jue 03/06/21	jue 03/06/21	
1.15	**Análisis del sistema**	**1,33 días**	**mié 13/01/21**	**jue 14/01/21**	
1.15.1	Diseño de objetivos del proyecto	2 horas	mié 13/01/21	mié 13/01/21	14
1.15.2	Definición de alcance del proyecto	2 horas	jue 14/01/21	jue 14/01/21	22
1.16	**Realización de la gestión de riesgos**	**5 días**	**jue 14/01/21**	**mié 20/01/21**	
1.16.1	Identificación de riesgos	3 horas	jue 14/01/21	vie 15/01/21	21
1.16.2	Análisis y priorización de los riesgos	2 horas	vie 15/01/21	sáb 16/01/21	25
1.16.3	Planificación de los riesgos	2 horas	sáb 16/01/21	lun 18/01/21	26
1.16.4	Definición de planes de contingencia	4 horas	lun 18/01/21	mar 19/01/21	27
1.16.5	Actualización y monitorización de los riesgos	4 horas	mar 19/01/21	mié 20/01/21	28

1.17	**Lectura de documentación**	**20,67 días**	**jue 21/01/21**	**lun 15/02/21**	
1.17.1	Investigación sobre sistemas de análisis de comportamiento e identidad de usuarios	20 horas	jue 21/01/21	jue 28/01/21	24
1.17.2	Investigación sobre algoritmos de detección de anomalías	20 horas	vie 29/01/21	sáb 06/02/21	31
1.17.3	Investigación sobre proyectos existentes de detección de anomalías sobre tráfico HTTP	20 horas	lun 08/02/21	lun 15/02/21	32
1.18	**Pruebas de implementación de viabilidad**	**11,67 días**	**mar 16/02/21**	**mar 02/03/21**	
1.18.1	Implementación preliminar de varios algoritmos de detección de anomalías con cargas de datos no reales	20 horas	mar 16/02/21	mar 23/02/21	30
1.18.2	Comparación de los distintos de algoritmos sobre los casos de negocio	5 días	mié 24/02/21	mar 02/03/21	35
1.19	**Diseño del sistema**	**7,67 días**	**mar 02/03/21**	**jue 11/03/21**	
1.19.1	Diseño de casos de uso	4 horas	mar 02/03/21	mié 03/03/21	34
1.19.2	Definición de los niveles de filtrado del sistema	3 horas	mié 03/03/21	vie 05/03/21	38
1.19.3	Definición de características de una petición HTTP sobre las que actúa el algoritmo	6 horas	vie 05/03/21	lun 08/03/21	39
1.19.4	Definición de políticas de peligrosidad	4 horas	lun 08/03/21	mar 09/03/21	40
1.19.5	**Diseño de integración del sistema en una infraestructura**	**1,33 días**	**mié 10/03/21**	**jue 11/03/21**	
1.19.5.1	Documentación de la integración en la infraestructura	4 horas	mié 10/03/21	jue 11/03/21	41
1.20	**Implementación del prototipo**	**50 días**	**jue 11/03/21**	**mié 12/05/21**	
1.20.1	Elección de lenguaje y tecnologías a utilizar	4 horas	jue 11/03/21	vie 12/03/21	37
1.20.2	Parseado de dataset	20 horas	vie 12/03/21	sáb 20/03/21	45
1.20.3	**Implementación de algoritmo de detección de anomalías**	**42 días**	**lun 22/03/21**	**mié 12/05/21**	
1.20.3.1	**Implementación nivel 1**	**3 días**	**lun 22/03/21**	**mié 24/03/21**	
1.20.3.1.1	Implementación del algoritmo de filtrado del nivel 1	5 horas	lun 22/03/21	mar 23/03/21	46
1.20.3.1.2	Realización de tests del nivel 1 con poca carga de datos	4 horas	mar 23/03/21	mié 24/03/21	49
1.20.3.2	**Implementación nivel 2**	**30,67 días**	**jue 25/03/21**	**sáb 01/05/21**	
1.20.3.2.1	**Implementación del algoritmo sobre cada característica de petición HTTP**	**19,33 días**	**jue 25/03/21**	**sáb 17/04/21**	

1.20.3.2.1.1	Implementación del módulo: Método HTTP	14 horas	jue 25/03/21	mar 30/03/21	50
1.20.3.2.1.2	Implementación del módulo: Directorio	14 horas	mié 31/03/21	mar 06/04/21	53
1.20.3.2.1.3	Implementación del módulo: Extensión del archivo	14 horas	mar 06/04/21	lun 12/04/21	54
1.20.3.2.1.4	Implementación del módulo: Longitud de URL	14 horas	lun 12/04/21	sáb 17/04/21	55
1.20.3.2.2	Realización de tests del nivel 2 con poca carga de datos	4 horas	sáb 17/04/21	lun 19/04/21	52
1.20.3.2.3	Pruebas de entrenamiento para cada característica con datos sintéticos pero cargas de datos reales	10 días	mar 20/04/21	sáb 01/05/21	57
1.20.3.3	Implementación de políticas de peligrosidad	5 días	sáb 01/05/21	sáb 08/05/21	51
1.20.3.4	**Realización de tests unitarios con cargas de datos reales**	**2,67 días**	**sáb 08/05/21**	**mié 12/05/21**	
1.20.3.4.1	Realización de tests sobre el nivel 1	4 horas	sáb 08/05/21	lun 10/05/21	59
1.20.3.4.2	Realización de tests sobre el nivel 2	4 horas	mar 11/05/21	mié 12/05/21	61
1.21	**Integración del prototipo**	**2,67 días**	**mié 12/05/21**	**vie 14/05/21**	
1.21.1	Entrenamiento con cargas de datos reales	3 horas	mié 12/05/21	jue 13/05/21	60
1.21.2	Realización de tests de integración	5 horas	jue 13/05/21	vie 14/05/21	64
1.22	**Redacción de artículo científico**	**6 días**	**sáb 15/05/21**	**sáb 22/05/21**	
1.22.1	Introducción	3 horas	sáb 15/05/21	lun 17/05/21	63
1.22.2	Propuesta	5 horas	lun 17/05/21	mar 18/05/21	67
1.22.3	Evaluación	6 horas	mié 19/05/21	jue 20/05/21	68
1.22.4	Conclusiones y trabajo futuro	2 horas	vie 21/05/21	vie 21/05/21	69
1.22.5	Revisión del artículo	2 horas	vie 21/05/21	sáb 22/05/21	70
1.23	**Redacción de la memoria final del proyecto**	**7 días**	**sáb 22/05/21**	**lun 31/05/21**	
1.23.1	Introducción	6 horas	sáb 22/05/21	mar 25/05/21	66
1.23.2	Fijación de objetivos	6 horas	mar 25/05/21	jue 27/05/21	73
1.23.3	Ejecución del proyecto	6 horas	jue 27/05/21	sáb 29/05/21	74
1.23.4	Cierre del proyecto	3 horas	lun 31/05/21	lun 31/05/21	75
1.24	Cierre de proyecto	2 horas	mar 01/06/21	mar 01/06/21	76

Tabla 3. Planificación inicial. WBS

3.2 RIESGOS

En esta sección se encuentran los apartados relativos a la gestión de riesgos.

3.2.1 Plan de gestión de riesgos

La documentación correspondiente al plan de gestión de riesgos se encuentra en el anexo 10.1 de este documento.

3.2.2 Identificación de riesgos

Los riesgos identificados se corresponden con un contexto empresarial, no académico. En este contexto, el proyecto se desarrolla como proyecto interno en un entorno I+D de la empresa y el objetivo es la venta del producto al cliente. Entre otros recursos que difieren con lo anteriormente presentado, se utiliza software contratado en la nube.

3.2.3 Registro de riesgos

ID del riesgo	Nombre del riesgo	Responsable	WP	Probabilidad	Impacto				Impacto	0,40	Respuesta
					Presupuesto	Planificación	Alcance	Calidad		Priorización	
1	Retraso de las tareas relacionadas con "Pruebas de implementación de viabilidad"	Responsable de riesgos de investigación	1	Alta	Medio	Alto	Bajo	Bajo	0,39		Mitigar. Daremos el mayor margen de tiempo posible para la realización de las subtareas de la tarea "Pruebas de implementación de viabilidad". De este modo, el retraso tenderá a ser el mínimo posible y, en el mejor de los casos, adelantaremos la planificación.
2	La aplicación resultante de la investigación puede no ser lo suficientemente eficiente.	Director de proyecto	1	Media	Crítico	Bajo	Alto	Crítico	0,45		Mitigar. Es un riesgo que se corre al iniciar una investigación. Si el desarrollo del proyecto fuese para un cliente, se debería adelantar un porcentaje del presupuesto y todo esto debería ser especificado en el contrato a firmar por el cliente, con lo que la empresa salvaría el presupuesto. Como es un proyecto interno, podemos exprimir todos los beneficios que contenga la parte de investigación (artículo, conferencias...) e incluirla dentro de la biblioteca de conocimiento de la empresa como recurso en caso de que en un futuro se pueda seguir trabajando en ello. Sin embargo, los gastos recaerán totalmente sobre la empresa sin poder obtener beneficio económico a corto plazo (el producto no está listo para vender), por lo que antes de poner en marcha el

ID del riesgo	Nombre del riesgo	Responsable	WP	Probabilidad	Impacto				Impacto	0,40	Respuesta
					Presupuesto	Planificación	Alcance	Calidad		Priorización	
											proyecto, la empresa debe evaluar si lo puede asumir.
3	Costes extra en la instalación en un cliente por insuficiencias de su infraestructura	Director de proyecto	1	Media	Medio	Medio	Inapreciable	Medio	0,15		Mitigar. Se intentará cubrir todos los escenarios posibles siguiendo estadísticas del tipo de infraestructuras más extendidas entre las empresas de nuestro mercado.
4	Problemas de disponibilidad con el servidor contratado en la nube	Responsable de riesgos de infraestructuras	1	Baja	Bajo	Bajo	Bajo	Bajo	0,05		Mitigar. Como únicamente afecta a las tareas de entrenamiento del modelo de datos, no es un riesgo crítico, para evitar la pérdida de datos, se harán copias periódicas en servidores locales de los resultados de los trabajos de computación.
5	Cambio de las licencias de uso en las bibliotecas de algoritmos Machine Learning utilizadas	Director de proyecto	1	Muy Baja	Medio	Medio	Bajo	Medio	0,03		Mitigar. Se buscarán alternativas que implementen los mismos algoritmos para tener una segunda opción y se valorarán comparando con el precio de la licencia de las ya utilizadas.
6	Poca documentación sobre las bibliotecas de algoritmos utilizadas	Responsable de riesgos de investigación	1	Baja	Medio	Medio	Bajo	Medio	0,09		Mitigar. Se dará margen de tiempo en las tareas que impliquen desarrollo con estos algoritmos para poder dedicarse en profundidad a estas bibliotecas.
7	Retraso en las tareas relacionadas con "Implementación del nivel 1"	Responsable de riesgos de investigación	1								
8	Retraso en las tareas relacionadas con "Implementación del nivel 2"	Responsable de riesgos de investigación	1	Baja	Medio	Medio	Medio	Medio	0,09		Mitigar. Se dará margen de tiempo en las tareas de la implementación de este nivel

ID del riesgo	Nombre del riesgo	Responsable	WP	Probabilidad	Impacto				Impacto	0,40	Respuesta
					Presupuesto	Planificación	Alcance	Calidad		Priorización	
											que conlleven más dificultad o no sean triviales.
9	Retraso en las tareas relacionadas con "Integración del prototipo"	Responsable de riesgos de investigación	1	Media	Alto	Medio	Medio	Medio	0,28		Mitigar. Se dará margen de tiempo en las tareas de la implementación de este nivel que conlleven más dificultad o no sean triviales. Realizar pruebas de viabilidad previas antes de decantarse por un algoritmo.
10	En los datos de log utilizados para las pruebas no hay patrones respecto a los horarios laborales	Responsable de riesgos de investigación	1	Media	Medio	Medio	Medio	Medio	0,15		Mitigar. Se dará margen de tiempo en las tareas que conlleven más dificultad o no sean triviales.
11	El sistema no cubre de forma positiva los casos de uso de detección de anomalías modelados durante el diseño de la experimentación	Responsable de riesgos de investigación	1	Baja	Bajo	Bajo	Bajo	Bajo	0,05		Mitigar. Preparar datos sintéticos que sí contengan esos patrones e introducir los casos de uso que queremos probar.
12	Poca información de sistemas similares para incluir en trabajo relacionado. Ya que muchas soluciones pueden ser privadas	Responsable de riesgos de investigación	1	Baja	Bajo	Bajo	Crítico	Alto	0,27		Asumir. Si tras todas las pruebas y modificaciones correspondientes, el sistema no cumple con los objetivos marcados (casos en los que detecta anomalías), aceptamos ese hecho y escribimos las conclusiones pertinentes.
13	Retraso en las tareas relacionadas con el parseado de datos	Responsable de riesgos de investigación	1	Media	Bajo	Bajo	Bajo	Medio	0,15		Mitigar. Buscar soluciones open source si la información proporcionada por los principales productos es insuficiente.
14	Retraso en la redacción final del artículo	Responsable de riesgos de investigación	1	Media	Bajo	Medio	Bajo	Bajo	0,15		Mitigar. Se dará margen de tiempo en las tareas de la implementación que conlleven más dificultad o no sean triviales. Puede ser que el enfoque

ID del riesgo	Nombre del riesgo	Responsable	WP	Probabilidad	Impacto				Impacto	0,40	Respuesta
					Presupuesto	Planificación	Alcance	Calidad		Priorización	
											tomado para el parseo de los datos no sea el adecuado y se tarde en encontrar el mejor.
15	Retraso en la fecha de fin del proyecto a consecuencia del COVID-19	Director de proyecto	1	Baja	Bajo	Bajo	Inapreciable	Bajo	0,05		Mitigar. El artículo puede pasar por más revisiones de las esperadas antes de que se considere listo para enviar a la revista.

Tabla 4. Registro de riesgos

3.3 PRESUPUESTO INICIAL

En esta sección se encuentran los apartados relativos al presupuesto del proyecto.

3.3.1 Presupuesto de Costes

En esta sección se encuentra el presupuesto de costes. Las horas de reuniones se suman a la partida de planificación inicial.

Las tarifas para cada recurso se muestran en la siguiente tabla.

Perfil	Precio Coste	Precio Venta	Horas invertidas
Alba Cotarelo Tuñón	23,00 €	55,00 €	354
José Manuel Redondo López	25,00 €	60,00 €	16
Antonio Payá González	23,00 €	55,00 €	14

Tabla 5. Tarifas de los recursos

Para calcular el precio de venta se han estimado los siguientes costes. Los gastos e ingresos del personal se calculan sobre las horas que dedican a este proyecto.

Concepto	Ingresos	Gastos
Personal	21.200,00 €	8.864,00 €
Sedes	0,00 €	6.000,00 €
Equipos y Licencias	0,00 €	1.650,00 €
Servicios	0,00 €	200,00 €
Formación	0,00 €	0,00 €
Costes financieros (1,00% de los ingresos)	0,00 €	212,00 €
Otros costes	0,00 €	200,00 €
	21.200,00 €	17.126,00 €
BENEFICIOS	**4.074,00 €**	

Tabla 6. Costes

Aplicación de técnicas de ML para la detección de comportamientos anómalos de usuarios										
I 1	***I 2***	***I 3***	***Descripción***	***Cantidad***	***Unidades***	***Precio***	***Precio total***	***Subtotal (3)***	***Subtotal (2)***	***Total***
										20.760,00 €
1			**Realización de la planificación inicial**					2.150,00 €	2.150,00 €	
	1		Alba Cotarelo Tuñón	16	horas	55,00 €	880,00 €			
	2		José Manuel Redondo López	12	horas	60,00 €	720,00 €			
	3		Antonio Payá González	10	horas	55,00 €	550,00 €			
2			**Gestión de riesgos**					825,00 €	825,00 €	
	1		Alba Cotarelo Tuñón	15	horas	55,00 €	825,00 €			
3			**Investigación**						5.905,00 €	
	1		**Análisis del sistema**					680,00 €		
		1	Alba Cotarelo Tuñón	4	horas	55,00 €	220,00 €			
		2	José Manuel Redondo López	4	horas	60,00 €	240,00 €			
		3	Antonio Payá González	4	horas	55,00 €	220,00 €			
	2		**Lectura de documentación**					3.300,00 €		
		1	Alba Cotarelo Tuñón	60	horas	55,00 €	3.300,00 €			
	3		**Pruebas de implementación de viabilidad**					1.925,00 €		
		1	Alba Cotarelo Tuñón	35	horas	55,00 €	1.925,00 €			
4			**Diseño del sistema**						1.155,00 €	
	1		**Diseño del algoritmo de detección**					935,00 €		
		1	Alba Cotarelo Tuñón	17	horas	55,00 €	935,00 €			
	2		**Diseño de integración del sistema en la infraestructura**					220,00 €		
		1	Alba Cotarelo Tuñón	4	horas	55,00 €	220,00 €			
5			**Implementación del prototipo**						8.030,00 €	
	1		**Preparación de datos**					1.320,00 €		

		1	Alba Cotarelo Tuñón	24	horas	55,00 €	1.320,00 €			
	2		**Implementación de algoritmo de detección de anomalías**					6.710,00 €		
		1	Alba Cotarelo Tuñón	122	horas	55,00 €	6.710,00 €			
6			**Integración del prototipo**						440,00 €	
	1		**Entrenamiento del modelo de datos**					165,00 €		
		1	Alba Cotarelo Tuñón	3	horas	55,00 €	165,00 €			
	2		**Pruebas de integración**					275,00 €		
		1	Alba Cotarelo Tuñón	5	horas	55,00 €	275,00 €			
7			**Resumen de la investigación**						990,00 €	
	1		**Redacción de artículo científico**					990,00 €		
		1	Alba Cotarelo Tuñón	18	horas	55,00 €	990,00 €			
8			**Redacción de la memoria final del proyecto**						1.265,00 €	
	1		**Redacción de la memoria final del proyecto**					1.265,00 €		
		1	Alba Cotarelo Tuñón	23	horas	55,00 €	1.265,00 €			

TOTAL										**20.760,00 €**

Tabla 7. Presupuesto inicial

3.3.2 Presupuesto de Cliente

En esta sección se muestran los presupuestos detallado y resumido del cliente.

3.3.2.1 *Presupuesto de Cliente Detallado*

El presupuesto destinado a la gestión de riesgos se incluye en las partidas de investigación, diseño e integración.

Aplicación de técnicas de ML para la detección de comportamientos anómalos de usuarios				
Partida	***Item***	***Partida***	***Importe***	***Total***
1		**Realización de la planificación inicial**	2.150,00 €	2.150,00 €
2		**Investigación**		6.180,00 €
	1	Análisis del sistema	2.330,00 €	
	2	Pruebas de implementación de viabilidad	3.575,00 €	
3		**Diseño del sistema**		1.430,00 €
	1	Diseño del algoritmo de detección	935,00 €	
	2	Diseño de integración del sistema en la infraestructura	220,00 €	
4		**Implementación del prototipo**		8.030,00 €
	1	Preparación de datos	1.320,00 €	
	2	Implementación de algoritmo de detección de anomalías	6.710,00 €	
5		**Integración del prototipo**		715,00 €
		Entrenamiento del modelo de datos	165,00 €	
		Pruebas de integración	275,00 €	
6		**Resumen de la investigación**		990,00 €
	1	Redacción de artículo científico	990,00 €	
7		**Redacción de la memoria final del proyecto**		1.265,00 €
	1	Redacción de la memoria final del proyecto	1.265,00 €	

TOTAL CLIENTE	**20.760,00 €**

Tabla 8. Presupuesto de cliente inicial detallado

3.3.2.2 *Presupuesto de Cliente Resumido*

Aplicación de técnicas de ML para la detección de comportamientos anómalos de usuarios		
Cod.	***Partida***	***Total***
1	Realización de la planificación inicial	2.150,00 €
2	Investigación	6.180,00 €
3	Diseño del sistema	1.430,00 €
4	Implementación del prototipo	8.030,00 €
5	Integración del prototipo	715,00 €
6	Resumen de la investigación	990,00 €
7	Redacción de la memoria final del proyecto	1.265,00 €

TOTAL CLIENTE	**20.760,00 €**

Tabla 9. Presupuesto de cliente inicial resumido

3.4 EJECUCIÓN DEL PROYECTO

3.4.1 Plan de Seguimiento de Planificación

En este apartado se describen las tres líneas base del proyecto: inicial, a mitad del proyecto y al final.

En cuanto a la línea inicial, se definió la planificación inicial junto con los recursos implicados. Para cada tarea se indica su duración, su fecha de inicio, fecha de fin y los recursos asignados. También se definió la lista de interesados en el proyecto, que aprobó la planificación inicial tras su revisión. Se acordó la notificación mensual del cumplimiento de plazos para informar a los interesados en caso de retrasos en el término de las tareas.

En cuanto a la línea a mitad de proyecto, la planificación se actualiza mensualmente en las reuniones anteriormente descritas, comunicando dichas actualizaciones a los interesados en el proyecto.

Por último, tras el término del proyecto se realizará un informe de incidencias, la revisión de la última versión de la planificación y el presupuesto final. Además, se incluirá el seguimiento de los riesgos identificados, así como las medidas tomadas para solventarlos en caso de que hayan ocurrido.

3.4.2 Bitácora de Incidencias del Proyecto

En esta sección se enumeran las incidencias sucedidas a lo largo del desarrollo del proyecto.

- El primer dataset de entrenamiento carecía de algunos de los patrones diarios de comportamiento, por lo que se modificó dando lugar a datos sintéticos. Sin embargo, más

tarde el modelo se reentrenó con otro dataset diferente que sí tenía los patrones acordes al contexto de los casos de uso.

- El parseado de datos sufrió mucho versionado hasta encontrar el enfoque adecuado. El tiempo inicial estimado para ello no fue suficiente y se tuvo que modificar la planificación retrasando las tareas siguientes.
- Cuando encontramos el algoritmo adecuado para la detección de anomalías, se decidió que podía ser paralelizado modificando la librería inicial. Se añadió dicha tarea a la planificación, ya que no estaba contemplada.

3.4.3 Riesgos

ID del riesgo	Nombre del riesgo	Seguimiento
1	Retraso de las tareas relacionadas con "Pruebas de implementación de viabilidad"	El tiempo extra asignado a esta tarea fue suficiente para cubrir la incertidumbre de la viabilidad del proyecto debido a la naturaleza de la investigación. Incluso se terminaron las tareas con margen de tiempo y se pudo adelantar el inicio de las siguientes.
2	La aplicación resultante de la investigación puede no ser lo suficientemente eficiente.	Los resultados finales indicaron un tiempo de ejecución aceptable.
3	Costes extra en la instalación en un cliente por insuficiencias de su infraestructura	La infraestructura del cliente se correspondía con las recogidas en el estudio estadístico de mercado realizado previamente.
4	Problemas de disponibilidad con el servidor contratado en la nube	Se realizaron copias de seguridad en caso de perder la conexión con los servidores, pero no hubo incidencias.
5	Cambio de las licencias de uso en las bibliotecas de algoritmos Machine Learning utilizadas	No hubo cambios en las licencias de las librerías utilizadas.

Tabla 10. Seguimiento de riesgos

3.5 CIERRE DEL PROYECTO

3.5.1 Planificación final

En la planificación final se observa un desplazamiento en el tiempo de algunas tareas, ya que las relacionadas con las pruebas viabilidad se terminaron antes de lo planificado. Sin embargo, hubo otras que se retrasaron por incidencias surgidas y otras que se tuvieron que añadir conforme se desarrollaba el prototipo.

EDT	Tarea	Duración	Comienzo	Fin	Predecesoras

1	**Proyecto de investigación**	**118,33 días**	**lun 11/01/21**	**vie 04/06/21**	
1.1	Definición de la planificación	0 días	mar 12/01/21	mar 12/01/21	13
1.2	Realización del análisis	0 días	jue 14/01/21	jue 14/01/21	21
1.3	Realización de la gestión de riesgos	0 días	jue 21/01/21	jue 21/01/21	24
1.4	Realización de pruebas de viabilidad	0 días	mar 02/03/21	mar 02/03/21	34
1.5	Realización del diseño	0 días	jue 11/03/21	jue 11/03/21	37
1.6	Implementación del prototipo	0 días	mié 12/05/21	mié 12/05/21	44
1.7	Integración del prototipo	0 días	sáb 15/05/21	sáb 15/05/21	65
1.8	Realización del artículo científico	0 días	mié 26/05/21	mié 26/05/21	68
1.9	Redacción de la memoria final del proyecto	0 días	vie 04/06/21	vie 04/06/21	75
1.10	Cierre del proyecto	0 días	sáb 05/06/21	sáb 05/06/21	80
1.11	Reunión de inicio de proyecto	2 horas	lun 11/01/21	lun 11/01/21	
1.12	Realización de la planificación inicial	2 horas	mar 12/01/21	mar 12/01/21	12
1.13	Realización del presupuesto inicial	2 horas	mar 12/01/21	mié 13/01/21	13
1.14	**Reuniones mensuales**	**97 días**	**jue 04/02/21**	**jue 03/06/21**	
1.14.1	Reuniones mensuales 1	2 horas	jue 04/02/21	jue 04/02/21	
1.14.2	Reuniones mensuales 2	2 horas	jue 04/03/21	jue 04/03/21	
1.14.3	Reuniones mensuales 3	2 horas	jue 01/04/21	jue 01/04/21	
1.14.4	Reuniones mensuales 4	2 horas	jue 06/05/21	jue 06/05/21	
1.14.5	Reuniones mensuales 5	2 horas	jue 03/06/21	jue 03/06/21	
1.15	**Análisis del sistema**	**1,33 días**	**mié 13/01/21**	**jue 14/01/21**	
1.15.1	Diseño de objetivos del proyecto	2 horas	mié 13/01/21	mié 13/01/21	14
1.15.2	Definición de alcance del proyecto	2 horas	jue 14/01/21	jue 14/01/21	22
1.16	**Realización de la gestión de riesgos**	**5 días**	**jue 14/01/21**	**mié 20/01/21**	
1.16.1	Identificación de riesgos	3 horas	jue 14/01/21	vie 15/01/21	21
1.16.2	Análisis y priorización de los riesgos	2 horas	vie 15/01/21	sáb 16/01/21	25
1.16.3	Planificación de los riesgos	2 horas	sáb 16/01/21	lun 18/01/21	26
1.16.4	Definición de planes de contingencia	4 horas	lun 18/01/21	mar 19/01/21	27
1.16.5	Actualización y monitorización de los riesgos	4 horas	mar 19/01/21	mié 20/01/21	28
1.17	**Lectura de documentación**	**20,67 días**	**jue 21/01/21**	**lun 15/02/21**	
1.17.1	Investigación sobre sistemas de análisis de comportamiento e identidad de usuarios	20 horas	jue 21/01/21	jue 28/01/21	24

1.17.2	Investigación sobre algoritmos de detección de anomalías	20 horas	vie 29/01/21	sáb 06/02/21	31
1.17.3	Investigación sobre proyectos existentes de detección de anomalías sobre tráfico HTTP	20 horas	lun 08/02/21	lun 15/02/21	32
1.18	**Pruebas de implementación de viabilidad**	**8,33 días**	**mar 16/02/21**	**mar 23/02/21**	
1.18.1	Implementación preliminar de varios algoritmos de detección de anomalías con cargas de datos no reales	15 horas	mar 16/02/21	vie 19/02/21	30
1.18.2	Comparación de los distintos de algoritmos sobre los casos de negocio	10 horas	vie 19/02/21	mar 23/02/21	35
1.19	**Diseño del sistema**	**7,67 días**	**mar 23/02/21**	**mié 03/03/21**	
1.19.1	Diseño de casos de uso	4 horas	mar 23/02/21	mié 24/02/21	34
1.19.2	Definición de los niveles de filtrado del sistema	3 horas	mié 24/02/21	jue 25/02/21	38
1.19.3	Definición de características de una petición HTTP sobre las que actúa el algoritmo	6 horas	jue 25/02/21	sáb 27/02/21	39
1.19.4	Definición de políticas de peligrosidad	4 horas	lun 01/03/21	mar 02/03/21	40
1.19.5	**Diseño de integración del sistema en una infraestructura**	**1,33 días**	**mar 02/03/21**	**mié 03/03/21**	
1.19.5.1	Documentación de la integración en la infraestructura	4 horas	mar 02/03/21	mié 03/03/21	41
1.20	**Implementación del prototipo**	**52 días**	**mié 03/03/21**	**mié 12/05/21**	
1.20.1	Elección de lenguaje y tecnologías a utilizar	4 horas	mié 03/03/21	vie 05/03/21	37
1.20.2	Parseado de dataset	40 horas	vie 05/03/21	mar 16/03/21	45
1.20.3	**Implementación de algoritmo de detección de anomalías**	**37,33 días**	**mié 17/03/21**	**mié 12/05/21**	
1.20.3.1	**Implementación nivel 1**	**3 días**	**mié 17/03/21**	**vie 19/03/21**	
1.20.3.1.1	Implementación del algoritmo de filtrado del nivel 1	5 horas	mié 17/03/21	jue 18/03/21	46
1.20.3.1.2	Realización de tests del nivel 1 con poca carga de datos	4 horas	jue 18/03/21	vie 19/03/21	49
1.20.3.2	**Implementación nivel 2**	**26 días**	**sáb 20/03/21**	**lun 19/04/21**	
1.20.3.2.5	Paralelización de la librería de EIF	6 horas	sáb 20/03/21	mar 23/03/21	50
1.20.3.2.1	**Implementación del algoritmo sobre cada característica de petición HTTP**	**15,33 días**	**mar 23/03/21**	**sáb 10/04/21**	
1.20.3.2.1.1	Implementación del módulo: Método HTTP	14 horas	mar 23/03/21	lun 29/03/21	52

1.20.3.2.1.2	Implementación del módulo: Directorio	10 horas	lun 29/03/21	vie 02/04/21	54
1.20.3.2.1.3	Implementación del módulo: Extensión del archivo	10 horas	vie 02/04/21	mar 06/04/21	55
1.20.3.2.1.4	Implementación del módulo: Longitud de URL	10 horas	mié 07/04/21	sáb 10/04/21	56
1.20.3.2.2	Realización de tests del nivel 2 con poca carga de datos	4 horas	sáb 10/04/21	lun 12/04/21	53
1.20.3.2.3	Pruebas de entrenamiento para cada característica con datos sintéticos pero cargas de datos reales	20 horas	mar 13/04/21	lun 19/04/21	58
1.20.3.2.4	Pruebas de entrenamiento para cada característica con datos y cargas de datos reales	2 horas	lun 19/04/21	lun 19/04/21	59
1.20.3.3	Implementación de políticas de peligrosidad	15 horas	mar 20/04/21	sáb 08/05/21	51
1.20.3.4	**Realización de tests unitarios con cargas de datos reales**	**2,67 días**	**sáb 08/05/21**	**mié 12/05/21**	
1.20.3.4.1	Realización de tests sobre el nivel 1	4 horas	sáb 08/05/21	lun 10/05/21	61
1.20.3.4.2	Realización de tests sobre el nivel 2	4 horas	mar 11/05/21	mié 12/05/21	63
1.21	**Integración del prototipo**	**2,67 días**	**mié 12/05/21**	**vie 14/05/21**	
1.21.1	Entrenamiento con cargas de datos reales	3 horas	mié 12/05/21	jue 13/05/21	62
1.21.2	Realización de tests de integración	5 horas	jue 13/05/21	vie 14/05/21	66
1.22	**Redacción de artículo científico**	**8,67 días**	**sáb 15/05/21**	**mié 26/05/21**	
1.22.1	Introducción	3 horas	sáb 15/05/21	lun 17/05/21	65
1.22.6	Trabajo relacionado	8 horas	lun 17/05/21	mié 19/05/21	69
1.22.2	Propuesta	5 horas	jue 20/05/21	vie 21/05/21	70
1.22.3	Evaluación	6 horas	vie 21/05/21	lun 24/05/21	71
1.22.4	Conclusiones y trabajo futuro	2 horas	mar 25/05/21	mar 25/05/21	72
1.22.5	Revisión del artículo	2 horas	mar 25/05/21	mié 26/05/21	73
1.23	**Redacción de la memoria final del proyecto**	**7 días**	**mié 26/05/21**	**vie 04/06/21**	
1.23.1	Introducción	6 horas	mié 26/05/21	vie 28/05/21	68
1.23.2	Fijación de objetivos	6 horas	vie 28/05/21	lun 31/05/21	76
1.23.3	Ejecución del proyecto	6 horas	lun 31/05/21	mié 02/06/21	77
1.23.4	Cierre del proyecto	3 horas	mié 02/06/21	vie 04/06/21	78
1.24	Cierre de proyecto	2 horas	vie 04/06/21	vie 04/06/21	79

Tabla 11. Planificación final

3.5.2 Informe final de riesgos

Los riesgos más probables fueron suficientemente mitigados, mientras que los más peligrosos en cuanto a alcance y calidad del proyecto no sucedieron. Las principales acciones que fueron clave para mitigar los primeros fueron dar un margen de tiempo suficiente a aquellas tareas que, por su naturaleza investigadora, no podíamos estimar de forma ajustada su duración. Sin embargo, otros riesgos como los relacionados con el retraso de tareas de parseado de datos, no fueron mitigados de forma suficiente. Aun así, el tiempo recortado en unas tareas compensó la duración de otras y el efecto sobre la planificación final fue leve en términos de tiempo y coste.

3.5.3 Presupuesto final de costes

La actualización del presupuesto se corresponde con los cambios realizados en la planificación.

Aplicación de técnicas de ML para la detección de comportamientos anómalos de usuarios

I 1	*I 2*	*I 3*	*Descripción*	*Cantidad*	*Unidades*	*Precio*	*Precio total*	*Subtotal (3)*	*Subtotal (2)*	*Total*
										8.772,00 €
1			**Realización de la planificación inicial**					898,00 €	898,00 €	
	1		Alba Cotarelo Tuñón	16	horas	23,00 €	368,00 €			
	2		José Manuel Redondo López	12	horas	25,00 €	300,00 €			
	3		Antonio Payá González	10	horas	23,00 €	230,00 €			
2			**Gestión de riesgos**					345,00 €	345,00 €	
	1		Alba Cotarelo Tuñón	15	horas	23,00 €	345,00 €			
3			**Investigación**						2.239,00 €	
	1		**Análisis del sistema**					284,00 €		
		1	Alba Cotarelo Tuñón	4	horas	23,00 €	92,00 €			
		2	José Manuel Redondo López	4	horas	25,00 €	100,00 €			
		3	Antonio Payá González	4	horas	23,00 €	92,00 €			
	2		**Lectura de documentación**					1.380,00 €		
		1	Alba Cotarelo Tuñón	60	horas	23,00 €	1.380,00 €			
	3		**Pruebas de implementación de viabilidad**					575,00 €		
		1	Alba Cotarelo Tuñón	25	horas	23,00 €	575,00 €			
4			**Diseño del sistema**						483,00 €	
	1		**Diseño del algoritmo de detección**					391,00 €		
		1	Alba Cotarelo Tuñón	17	horas	23,00 €	391,00 €			
	2		**Diseño de integración del sistema en la infraestructura**					92,00 €		

		1	Alba Cotarelo Tuñón	4	horas	23,00 €	92,00 €			
5			**Implementación del prototipo**						3.496,00 €	
	1		**Preparación de datos**					1.012,00 €		
		1	Alba Cotarelo Tuñón	44	horas	23,00 €	1.012,00 €			
	2		**Implementación de algoritmo de detección de anomalías**					2.484,00 €		
		1	Alba Cotarelo Tuñón	108	horas	23,00 €	2.484,00 €			
6			**Integración del prototipo**						184,00 €	
	1		**Entrenamiento del modelo de datos**					69,00 €		
		1	Alba Cotarelo Tuñón	3	horas	23,00 €	69,00 €			
	2		**Pruebas de integración**					115,00 €		
		1	Alba Cotarelo Tuñón	5	horas	23,00 €	115,00 €			
7			**Resumen de la investigación**						598,00 €	
	1		**Redacción de artículo científico**					598,00 €		
		1	Alba Cotarelo Tuñón	26	horas	23,00 €	598,00 €			
8			**Redacción de la memoria final del proyecto**						529,00 €	
	1		**Redacción de la memoria final del proyecto**					529,00 €		
		1	Alba Cotarelo Tuñón	23	horas	23,00 €	529,00 €			

TOTAL										**8.772,00 €**

Tabla 12. Presupuesto final

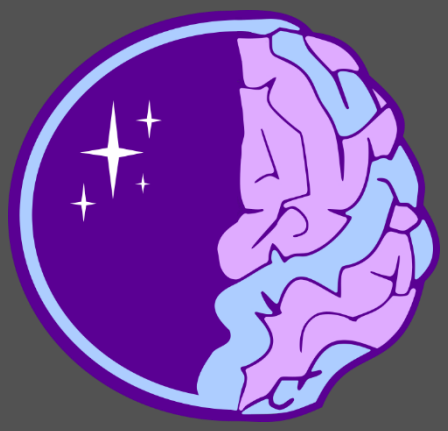
3.5.4 Informe de lecciones aprendidas

En este apartado se enumeran las lecciones aprendidas durante el desarrollo de este proyecto.

En primer lugar, no debimos subestimar el tiempo necesario para el preparado de los datos para que sean consumidos por módulos de Machine Learning. Fue un punto clave en desarrollo que se demoró más de lo previsto. Por otro lado, lo que al comienzo del desarrollo nos parecía tiempo sobreestimado para las tareas de búsqueda de documentación, estuvo bastante acertado. Es necesario dejar tiempo con margen para este tipo de tareas en contextos de proyecto de investigación como es nuestro caso.

Por último, aunque con la búsqueda de documentación se integraba también la búsqueda de sistemas similares, no se abordó en detalle esta parte en detalle hasta la redacción del artículo. Si al tiempo que recogíamos información general hubiésemos tomado constancia de otros productos similares, hubiésemos ahorrado tiempo de cara al final de la planificación.

Capítulo 4 Estado Actual de los Conocimientos Científico-Técnicos



Kassandra es un complemento de seguridad enfocado a la detección de anomalías en el comportamiento normal de usuarios o máquinas durante la interacción con un servidor. Los datos base dependen de cada caso, ya que se entrena con los logs del servidor que se vaya a proteger. Por ello, se utilizan técnicas de aprendizaje no supervisado, las cuales no necesitan una clasificación previa de los datos de entrenamiento. Además, esta aproximación permite la detección de ataques de día cero que no estuviesen registrados anteriormente y que supusiesen un desvío del comportamiento usual de las entidades implicadas [6].

4.1 ALGORITMOS USADOS PARA LA DETECCIÓN DE ANOMALÍAS

Falcão et al. [6] recogen una comparación de algoritmos de detección de anomalías para la detección de intrusos. El resultado de sus experimentos indica que los algoritmos que mejor distinguen los comportamientos anómalos son *Isolation Forest (IF)* [7] y *One-Class SVM* [8]. Los siguientes mejores algoritmos clasificados en este trabajo son *Connectivity-based Outlier Factor y Outlier detection using Indegree Number*.

La detección de anomalías en contextos de seguridad relativos a comportamientos de usuarios ha sido explorada previamente mediante el uso de Isolation Forest. Este algoritmo se adecúa a este contexto por varias razones. En primer lugar, no requieren ejemplos de anomalías en su dataset, lo que supone una ventaja a la hora de entrenar sobre un escenario nuevo. Por otro lado, *One-Class SVM* es muy sensible a las anomalías, por lo que el dataset de entrenamiento debería estar libre de ellas, cosa que tampoco podemos asegurar en nuestro caso sin un análisis previo adaptado al dominio [9] [10].

Sun et al. [11] proponen la aplicación de los IF sobre un conjunto de logs de una empresa real, construyendo una línea base de comportamiento por cada usuario. Las variables que tienen en cuenta son:

- *Match rule*. Reglas predefinidas en el sistema usadas para la autenticación.
- *Signature check*. Indica si la firma del dispositivo coincide con su ID.
- *Device check*. Indica si el dispositivo está asociado al usuario.
- *Navegador*. El navegador desde el cual el usuario se conecta.
- *Hora de conexión del usuario*. Incluye mes, día, año, horas, minutos y segundos.

Indican que los IF fueron diseñados inicialmente para datasets de datos categóricos. Su implementación de los IF incluye el parseo de datos categóricos a datos cuantitativos mapeando los mismos en un orden aleatorio. Consiguen un 98,92% de exhaustividad y un 50,77% de precisión. Sin embargo, concluyen que el sistema no distingue entre los comportamientos de diferentes usuarios debido a que los datos categóricos extraídos tienen muy pocas variaciones entre ellos.

Bülow et al. [12] también utilizan IF sobre el dataset abierto CERT[1] que clasifica los datos en amenazas o no amenazas, concretamente sobre los datos de login y logout de usuarios. Las variables que tienen en cuenta son:

- Día de la semana.
- Mes.
- Hora.
- Usuario.
- PC.
- Actividad.

Para parsear estos datos, utilizan la propuesta de Kaleko[2], la cual ofrece un método para la representación de variables cíclicas como son las referentes a la fecha. Mientras, las variables categóricas son parseadas con Label Encoding. Sin embargo, la propia codificación de estas variables introduce una propiedad no deseada en este caso: un orden inherente en función de los IDs mapeados que en realidad no existe y que tiene un impacto negativo sobre su modelo. Detectan el 76% de las amenazas y un 7% de datos normales clasificados como tales de forma errónea.

4.2 Isolation Forest y Extended Isolation Forest

Los IF [7] basan su funcionamiento sobre la idea de que las anomalías son “pocas y diferentes”. El algoritmo genera árboles realizando particiones aleatorias sobre las instancias del dataset hasta que todas se encuentren aisladas o se alcance una determinada profundidad del árbol. Para predecir el valor de anomalía de una nueva instancia, como los árboles se construyen de forma aleatoria, se genera un número determinado de ellos y se calcula la distancia media desde la raíz hasta dicha instancia. La investigación comprueba que las anomalías se aíslan con un menor número de particiones y que un mayor número de árboles converge hacia un único resultado.

[1] CERT dataset https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099 [Accedido por última vez el 10 de mayo de 2021]

[2] Feature Engineering - Handling Cyclical Features, David Kaleko. http://blog.davidkaleko.com/feature-engineering-cyclical-features.html [Accedido por última vez el 10 de mayo de 2021]

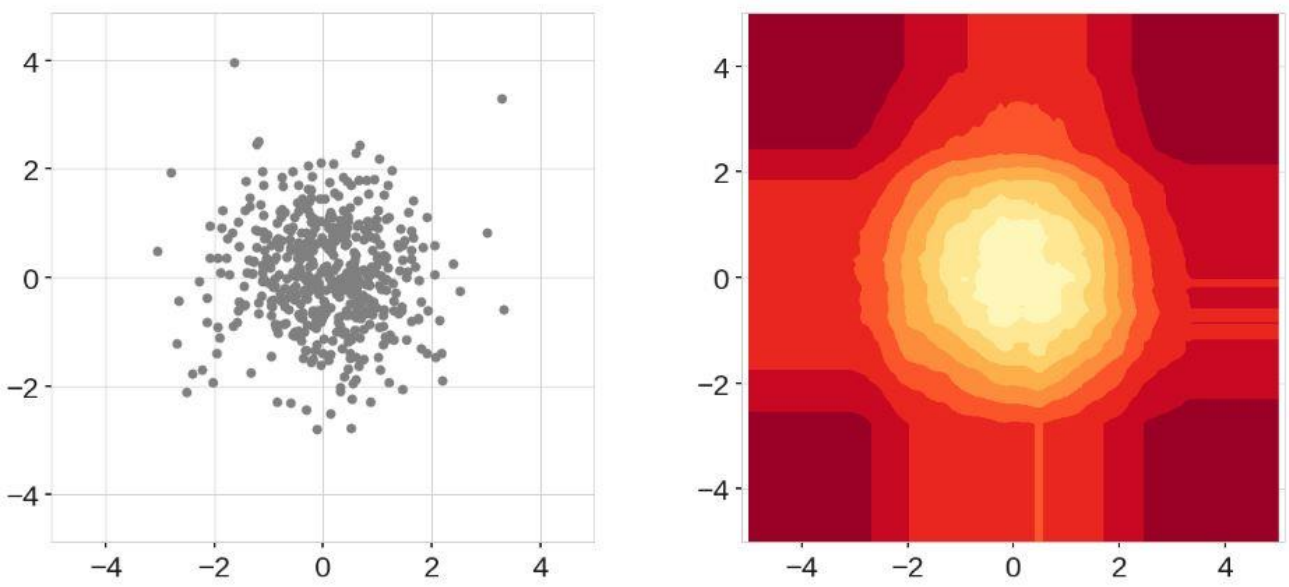


Figura 3. Conjunto de datos normalmente distribuidos [5]

La Figura 3 izquierda muestra un conjunto de datos normalmente distribuidos. La Figura 3 derecha muestra el mapa de valores de anomalías sobre dichos datos. Se observa que los colores cercanos al amarillo se corresponden con la zona que ocupan la mayor parte de los puntos, datos no considerados anomalías. Mientras que los colores cercanos al rojo oscuro son las zonas donde se encuentran los valores anómalos (más separados del clúster central).

Los IF obtienen buenos resultados con o sin anomalías presentes en el dataset de entrenamiento. Además, son capaces de manejar grandes cantidades de datos de forma eficiente, lo que permite su uso en aplicaciones en tiempo real.

Sin embargo, dependiendo de cómo se distribuyan los clústers de puntos en el espacio, pueden aparecer zonas “fantasma” no anómalas como se muestra en la Figura 4.

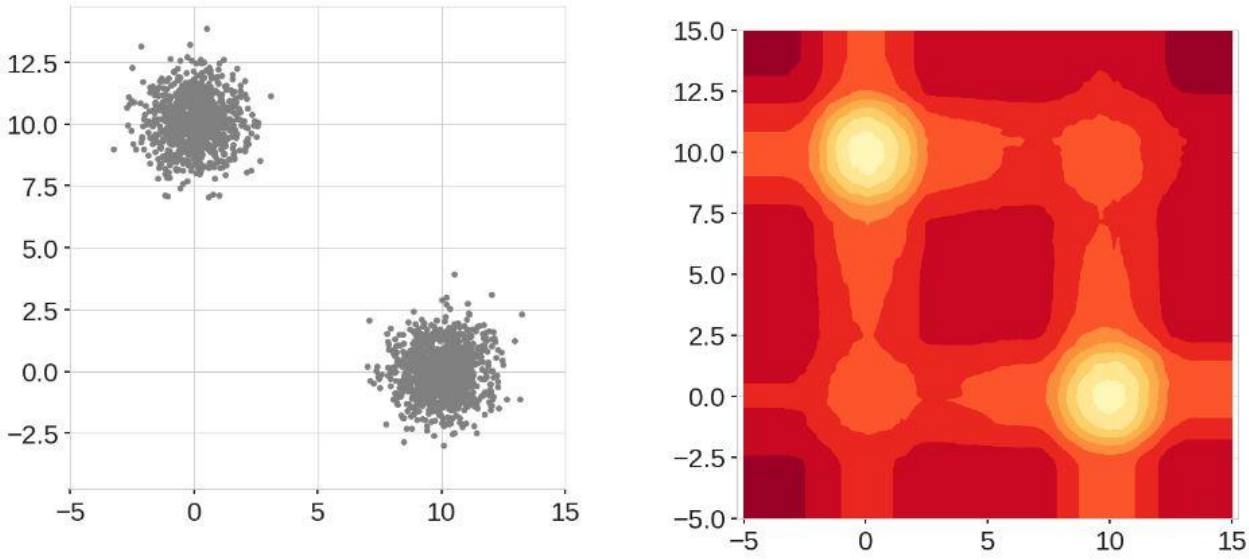


Figura 4. Conjunto de datos normalmente distribuidos en dos clústers [5]

Hariri et al. [5] proponen una modificación de los IF llamada Extended Isolation Forest (EIF). Mientras que las particiones de los IF se realizaban de manera horizontal o vertical como se muestra en la Figura 5, los EIF introducen una pendiente aleatoria en las líneas que delimitan dichas particiones (Figura 6).

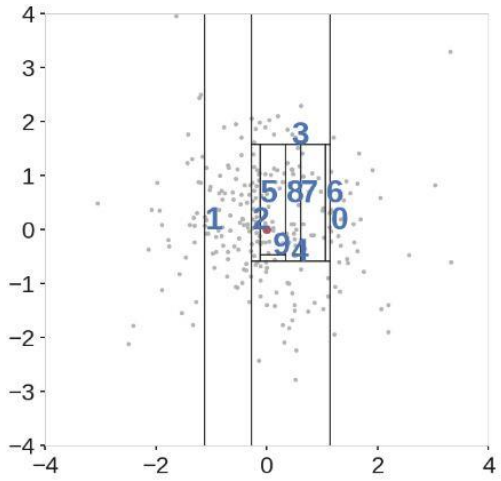


Figura 5. Particiones realizadas por IF [5]

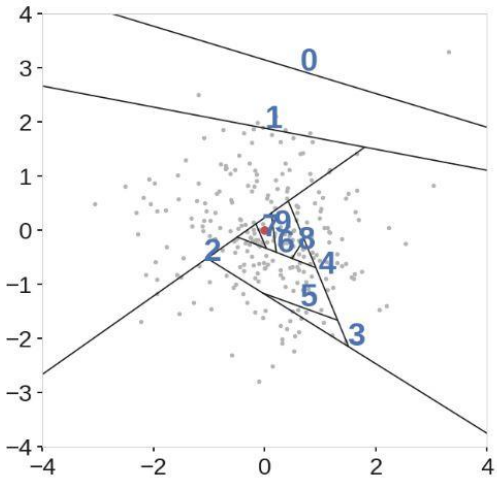


Figura 6. Particiones realizadas por EIF [5]

De esta forma, consiguen resolver el problema de las zonas "fantasma" que clasificarían como no anómalos posibles valores alejados de los clústeres (Figura 7). En Kassandra, estamos utilizando EIF para los modelos de detección de anomalías en lugar de la implementación clásica de IF.

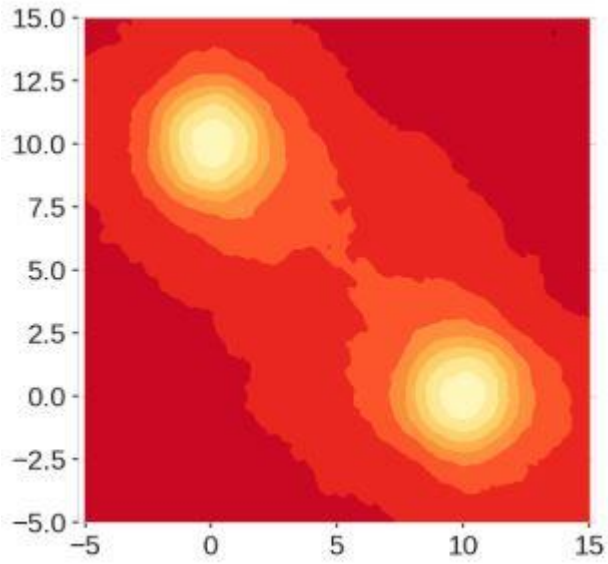


Figura 7. Mapa de anomalías de EIF de dos clústers de datos normalmente distribuidos [5]

4.3 USER AND ENTITY BEHAVIOUR ANALYTICS

Un *UEBA (User Entity Behavior Analytics)* es un sistema de detección de amenazas internas basado en monitorización de comportamientos de usuarios en una organización o red [4]. Se basan en la creación de unas líneas base consideradas como comportamientos ordinarios o no anómalos. Los datos que se utilizan para crear estas líneas base pueden ser logs de sistema, de aplicaciones, logs de dispositivos de red… A partir de su desviación sobre éstas se calculan los valores de anomalía de los nuevos comportamientos entrantes.

Los *UEBAs* comparten las siguientes características [4]:

- Los sistemas deberían poder usar todas las fuentes de datos disponibles.
- Se deben priorizar los sistemas críticos.
- Se han de calcular valores de peligrosidad en función de las potenciales desviaciones del comportamiento ordinario de los usuarios, a los cuales se les asignan pesos de acuerdo con el nivel de restricción adecuado para cada escenario.
- El sistema ha de poder detectar ataques no registrados previamente.

En cuanto al diseño de los sistemas:

- *Parseado de los datos de input*, como logs, para que sean tratados por los diferentes algoritmos.
- *Analíticas de seguridad*. Comprende los módulos de machine learning, tanto de aprendizaje supervisado como no supervisado, y la correlación de eventos, capaz de detectar anomalías a partir de un conjunto de eventos.
- *Creación de perfiles de usuario* sobre un determinado periodo de tiempo.
- *Gestión de las amenazas*.

Existen diferentes soluciones en el mercado. La siguiente tabla recoge varias de las señaladas por Guercio en eSecurity Planet [13] junto con sus casos de uso, características especiales y las técnicas de machine learning que utilizan.

UEBA	**Casos de uso**	**Características especiales**	**Técnicas machine learning**
Aruba[3]	Industrias de alto riesgo y reguladas	Análisis integrado del tráfico de red	Supervisado y no supervisado. Detección de familias de ataques (ramsonware).
Cynet[4]	Equipos de operaciones de seguridad que buscan una administración más amplia de aplicaciones y dispositivos	Integra control de acceso, administración de aplicaciones y gestión de endpoints	Capacidades de corrección automatizadas.
Exabeam[5]	Grandes organizaciones, agencias federales	Detección y prevención de ransomware	Clusterización de usuarios en grupos por comportamiento.

[3] Aruba. https://blogs.arubanetworks.com/solutions/turbocharge-your-siem-with-aruba-introspect-machine-learning-analytics-and-ueba/ [Accedido por última vez el 5 de mayo de 2021]

[4] Cynet. https://www.cynet.com/wp-content/uploads/2017/06/Cynet-360-Brochure.pdf [Accedido por última vez el 5 de mayo de 2021]

[5] Exabeam. https://www.exabeam.com/product/ueba/ [Accedido por última vez el 5 de mayo de 2021]

Forcepoint[6]	Equipos de operaciones de seguridad	Puntuaciones de riesgo consolidadas para individuos; repeticiones de vídeo de las pantallas de los usuarios	N/A.
Fortinet[7]	Bancos, fabricantes y desarrolladores de juegos	Monitoriza los puntos de conexión incluso cuando está fuera de la red	Aprendizaje no supervisado sobre el comportamiento del usuario.
Gurucul[8]	Operaciones de seguridad corporativas	Gran biblioteca de algoritmos de aprendizaje automático	Modelos de aprendizaje automático personalizables.
LogRhythm[9]	Industrias de alto riesgo y altamente reguladas	Orquestación, automatización y respuesta integradas	Análisis estadístico y técnicas supervisadas y no supervisadas. Análisis automatizado de riesgos.
One Identity[10]	Dirigido a cuentas privilegiadas de alto riesgo	Detección de amenazas en tiempo real, biometría conductual	Aprendizaje sin supervisión. Línea base del comportamiento normal.
Palo Alto [14]	Equipos de operaciones de seguridad que buscan amplias protecciones	Investigación automatizada de alertas, análisis de impacto, búsqueda de amenazas	No supervisado: grupos de comportamiento.Supervisado: clasificación del dispositivo de acceso.Reglas personalizadas.
RSA[11]	Equipos de operaciones de seguridad que buscan automatización	Detección de anomalías no supervisada y aprendizaje automático	Casos de aprendizaje y uso no supervisados. No se necesita análisis previo.
Securonix[12]	Equipos de operaciones de seguridad, especialmente en empresas muy grandes	Informes de fraude; vigilancia comercial; análisis de datos de pacientes	Patrones de comportamiento.

[6] Forcepoint. https://www.forcepoint.com/sites/default/files/resources/files/presentation_ueba_webinar_slides_en.pdf [Accedido por última vez el 5 de mayo de 2021]

[7] Fortinet. https://www.fortinet.com/products/ueba#models-specs [Accedido por última vez el 5 de mayo de 2021]

[8] Gurucul. https://gurucul.com/ [Accedido por última vez el 5 de mayo de 2021]

[9] LogRythm https://gallery.logrhythm.com/data-sheets/userxdr-datasheet-2020.pdf [Accedido por última vez el 5 de mayo de 2021]

[10] One Identity https://www.oneidentity.com/products/one-identity-safeguard-for-privileged-analytics/ [Accedido por última vez el 5 de mayo de 2021]

[11] RSA https://www.rsa.com/content/dam/en/data-sheet/rsa-netwitness-ueba.pdf [Accedido por última vez el 5 de mayo de 2021]

[12] Securonix https://www.securonix.com/web/wp-content/uploads/2019/06/Securonix-UEBA-Datasheet.pdf [Accedido por última vez el 5 de mayo de 2021]

Splunk[13]	Equipos de operaciones de seguridad	Línea base de comportamiento multidimensional; exploración de anomalías	Modelos de aprendizaje automático personalizados con propio SDK (Splunk UBA 4.0 – Machine Learning Software Development Kit).
Varonis[14]	Equipos de operaciones de seguridad	"Máquina del tiempo de seguridad" analiza los datos anteriores; detección ransomware	N/A.
Veriato Cerebral[15]	Equipos de operaciones de seguridad y departamentos de recursos humanos	Análisis psicolingüístico; instantáneas de pantalla; grabación de pulsaciones de teclas	Grupos de comportamiento. Bajos falsos positivos. Procesamiento de lenguaje natural para prevenir futuras amenazas internas.

Tabla 13. Comparación de productos UEBA

Varonis [15] señala que los *UEBAs* tienen ciertas limitaciones en cuanto a los falsos positivos. Puede haber casos de uso legítimos que se consideren anómalos y provoquen consecuencias en el funcionamiento de una empresa. Pone como ejemplo un empleado que tenga que trabajar un fin de semana para cumplir con una fecha de entrega.

En nuestra investigación nos centraremos en el parseado de los datos de input y el módulo de aprendizaje no supervisado. No distinguiremos entre usuarios, ya que sólo atendemos a una pequeña parte de la red de una empresa, que son los logs de una aplicación. El sistema debe de identificar un conjunto de casos anómalos y minimizar los falsos positivos sobre el resto de peticiones.

El precio varía en función del número de entidades vigiladas, de las máquinas protegidas o del almacenamiento utilizado al mes. Esto puede impedir el despliegue de este tipo de medidas de seguridad en empresas con bajo presupuesto o en algunos entornos educativos, lo que permite formar a los futuros ingenieros informáticos en este tipo de sistemas de seguridad.

En el caso de los *UEBA* de código abierto, existen varias soluciones. *OpenUBA* [16] es un marco para modelar el comportamiento de los usuarios. Su principal objetivo es hacer accesible a los usuarios no técnicos las razones por las que un comportamiento se considera anómalo. Los usuarios también pueden utilizar modelos predefinidos o desarrollar los suyos propios. Utilizan bibliotecas como *Tensorflow* [17] o *Scikit Learn* [18]. Sin embargo, tal y como se ha mostrado en la literatura anterior, la implementación de Scikit Learn de uno de los algoritmos más adecuados para predecir anomalías, *Isolation Forest*, presenta los problemas descritos anteriormente.

[13] Splunk https://www.splunk.com/en_us/blog/conf-splunklive/splunk-uba-the-ultimate-lego-for-machine-learning-models.html [Accedido por última vez el 5 de mayo de 2021]
[14] Varonis https://www.varonis.com/products/datalert/ [Accedido por última vez el 5 de mayo de 2021]
[15] Veriato Cerebral https://www.veriato.com/solutions/technology/artificial-intelligence [Accedido por última vez el 5 de mayo de 2021]

Khaliq et al. [4] también presenta una lista de otros *UEBA* de código abierto como *ELK Stack*, *SIEMonster* o *Alien Vault OSSIM*. Algunos de ellos tienen limitaciones de licencia libre sobre las características o el escalado. Según los autores, ninguno de ellos tiene la funcionalidad de *UEBA* en su versión gratuita.

Los centros educativos no pueden permitirse una solución *UEBA* privada. Además, su despliegue en un contexto educativo y controlado podría ser complejo. Por otro lado, las soluciones de código abierto existentes pueden no tener las características deseadas para probar la funcionalidad de *UEBA* o pueden ser demasiado amplias para explicar el concepto.

Capítulo 5 Descripción del Sistema

Kassandra actúa como un filtro para las peticiones HTTP. Estas peticiones deberán pasar a través de uno o dos niveles de filtrado, el análisis se para en el primer nivel si la petición ya es considerada una anomalía para mejorar el rendimiento.

- El **Nivel 1** se basa en un sistema experto basado en reglas [19], un conjunto de condiciones que terminan por decir si la IP, el método HTTP y el directorio accedido se reconocen como previamente presentes en los datos de entrenamiento y, por lo tanto, se etiquetan como peticiones normales. Si el Nivel 1 etiqueta una petición como "normal", continua su filtrado hacia el Nivel 2.
- El **Nivel 2** consta de un conjunto de subfiltros en los que se analiza una característica particular de la petición HTTP. Cada subfiltro tiene su propio modelo de aprendizaje automático entrenado, que mide el valor de peligrosidad de una petición. Estos módulos utilizan *Extended Isolation Forests* para predecir esos valores.

Después de pasar por esta estructura de filtrado, una petición recibe un valor de peligrosidad en función de si se considera una anomalía o no. Como usuarios de Kassandra, podemos configurar el umbral de lo que consideramos una potencial amenaza o no, o incluso dar más importancia a algunas características de la petición HTTP que podrían ser más peligrosas.

5.1 DATASET

El primer prototipo de nuestra investigación analiza las peticiones HTTP, por lo que los datos utilizados en el entrenamiento se obtendrían en producción a partir de los archivos de registro del servidor centralizado. Esto se extenderá a otros protocolos en el futuro. Para el trabajo que se presenta aquí, los conjuntos de datos utilizados en el entrenamiento y las pruebas fueron obtenidos de un servicio de logs de una aplicación de un dominio concreto de una empresa real.

Los datos que obtenemos de los logs son los siguientes:

- IP del usuario.
- Método HTTP.
- Nombre del primer directorio de la URL.
- Extensión del archivo.
- Longitud total de la URL.

Consideramos que estos datos pueden proporcionar información sobre el comportamiento habitual de los usuarios en el contexto de una aplicación empresarial común en un entorno LAN, que es el escenario principal en el que vamos a centrar nuestra investigación. De este modo, un conjunto restringido de usuarios e IPs accederá a ella, por lo que el conjunto de IPs de origen será normalmente

restringido. Los métodos HTTP utilizados están definidos por las peticiones anteriores, en los logs de los endpoints, una desviación del acceso rutinario puede indicar un comportamiento peligroso. En el caso de los nombres de directorios y extensiones de archivos, es posible que sólo se acceda a ciertas carpetas en determinados momentos del día. Por último, los parámetros maliciosos que aumentan anormalmente la longitud de la URL pueden utilizarse para comprobar ciertas vulnerabilidades, por lo que también se tiene en cuenta su número de caracteres.

5.2 PARSEADO DE DATOS

Los *Extended Isolation Forests* son buenos clasificadores de datos cuantitativos [11], por lo que tuvimos que transformar las peticiones HTTP en puntos en un plano. Cada petición se representa como un punto en un plano 2D. El foco de nuestra detección de anomalías está en lo que los usuarios normalmente no deberían hacer en un determinado momento del día, por lo que la representación gráfica mostrará esta información como una circunferencia, como un reloj, pero con las 24 horas completas. Esta representación de características periódicas se basa en la propuesta de Bülow et al. [12] (ver Figura 8). En nuestro sistema las 12 a.m. horas se representan en el punto (2,0) y las 6 a.m. en el (0,2), las horas del día continúan en sentido antihorario. Siguiendo esta disposición circular se consigue relacionar en distancia geométrica la última hora de un día con la primera de la siguiente, lo que no se consigue con una línea recta. Realizamos esta transformación sobre las características definidas en la sección anterior, excepto la IP del usuario. Para explicar el proceso, utilizaremos como ejemplo la longitud de la URL. Considera como eje *X* la circunferencia y, sobre él, un eje *Y* por cada minuto (1440 ejes *Y*).

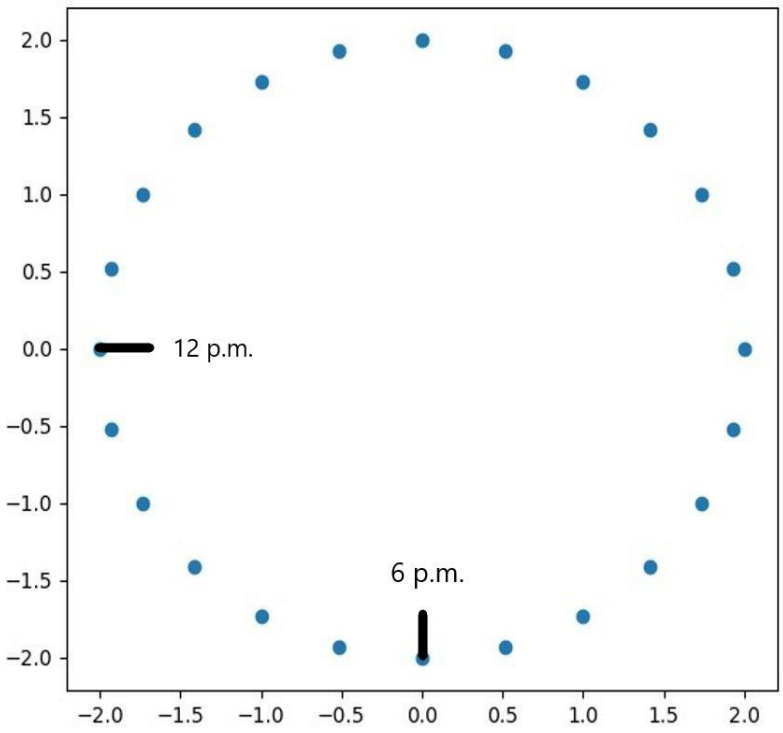


Figura 8. Representación de las horas de un día completo

A partir de esto, diseñamos un sistema de coordenadas que nos permita incluir esta información horaria. Llamaremos *SC* al sistema cartesiano ordinario y *SC'* al sistema cartesiano sobre cada minuto. En la Figura 9 se representan los minutos sobre el eje *X* y los posibles valores de una característica sobre el eje *Y*, por ejemplo, las diferentes longitudes de URL que aparecen en los datos de entrenamiento.

Calcularemos como ejemplo los puntos *A* y *B* de la Figura 9, que podrían corresponder a las longitudes de URL de dos peticiones diferentes.

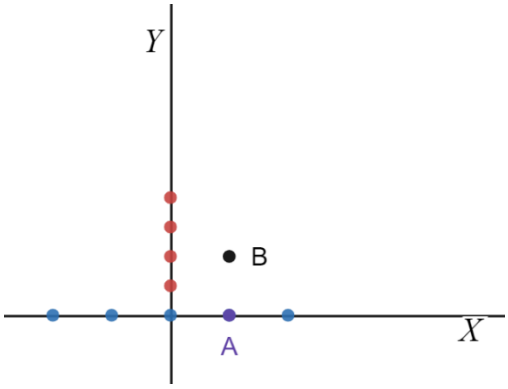


Figura 9. Representación del sistema SC' y de los puntos A y B

La circunferencia que pasa por *A* marcará el radio base (en *SC'*, *Y=0*) sobre la que calcularemos el resto de los puntos. En la Figura 10 se muestran los puntos *A* y *B* en *SC*. El objetivo es traducir los puntos de *SC'* a *SC* para que puedan ser analizados por los EIF.

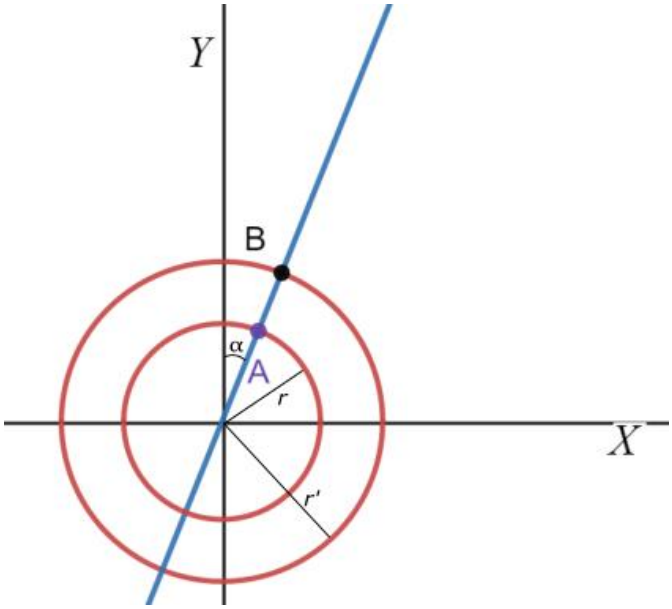


Figura 10. Representación del sistema SC y de los puntos A y B

Los puntos *A* y *B* se calculan de la siguiente forma:

Si en SC':

$$A = (X_a, Y_a)$$

$$B = (X_b, Y_b)$$

La primera ecuación de las siguientes se corresponde con la recta azul que corta la circunferencia, es decir, nuestro eje *Y* en *SC'*. La segunda se corresponde con la circunferencia de radio *r*. Calculamos *α* con la siguiente ecuación, siendo *minuto* el valor de *X* en *SC'* (la unidad de minuto a la que se recibe la petición).

$$\alpha = \text{minuto} * \frac{360}{1440}$$

$$y = \tan(\alpha)\, x$$

$$x^2 + y^2 = r$$

Definimos un valor para *r* y resolvemos el sistema de ecuaciones para obtener el punto A en SC.

Para calcular *B*, sólo tenemos que sustituir *r* por *r'*. Tanto *r* como *r'* son valores conocidos que definiremos más adelante. Para movernos por el eje *Y*, aumentaremos en *r'- r* las unidades que queramos en función el valor de la característica.

Tendremos un "reloj" (modelo) para cada característica de la petición que queramos analizar. Kassandra comprueba las siguientes características para cada petición:

- Nombre del primer directorio de la URL (raíz en caso de que se acceda a un archivo en "/")
- Método HTTP de la petición
- Extensión del archivo al que se accede (si se accede a un archivo de forma explícita)
- Longitud de la URL

En cada modelo, cada valor cualitativo se ordena de mayor a menor según su frecuencia de aparición en los datos de entrenamiento y se le asigna un identificador único (en ese orden) para ser representado gráficamente. Por lo que el gráfico constará de un conjunto de puntos dispuestos sobre circunferencias concéntricas donde las más cercanas al centro (y con puntos más agrupados entre sí) estarán formadas por los valores más comunes (ver Figura 11, este gráfico se explicará en detalle en la sección 5.3.2). De esta manera, ya no se introduce un orden en los datos sin significado en el dominio como comentaban Sun et al. [11] y Bülow et al. [12].

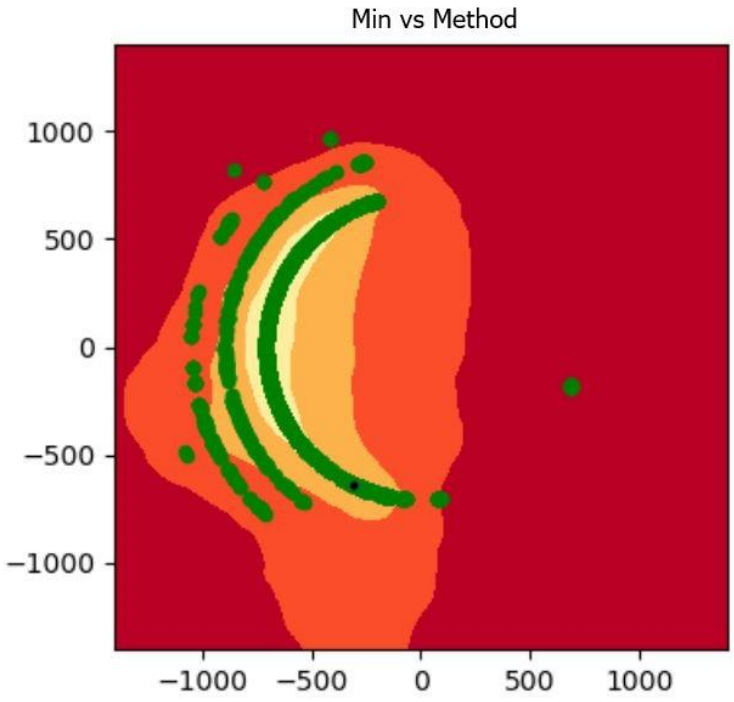


Figura 11. Ejemplo de output

Por ejemplo, si en el dataset se encuentran las siguientes URLs:

- */login* con 25 apariciones
- */home* con 40 apariciones

Para el análisis del directorio accedido se ordenarán y se les asignará un identificador de la siguiente manera:

- */home* con ID: 0
- */login* con ID: 1

Como comentamos, en las coordenadas de *SC'*, *X* es el minuto en el cual se accede a la URL */home* e *Y* se corresponde con el ID. Para cada característica definimos *r* (*Y=0*) y el offset entre cada unidad (*offset = r'- r*) (Tabla 14). El valor de *r* y los offsets se han definido mediante experimentación eligiendo aquellos con los que los EIF obtengan mejores resultados, es decir, se diferencien mejor las anomalías.

Característica	r	offset
Nombre del primer directorio de la URL	500000	300000
Método HTTP	500000	300000
Extensión del archivo	500000	300000
Longitud de la URL	500000	100000

Tabla 14. Valores de r y offsets para cada característica

5.3 ESTRUCTURA DE KASSANDRA

Kassandra consta de dos niveles de filtrado. Una petición debe pasar por la ruta marcada con línea discontinua en la siguiente imagen y, si para el punto en que llega al *stage 1* aún se considera una petición normal, continúa al Nivel 2. El segundo nivel permite un filtrado más fino. De esta manera, en caso de que se detecte como anomalía en el Nivel 1, se ahorra la ejecución del segundo. La Figura 12 muestra la estructura de Kassandra.

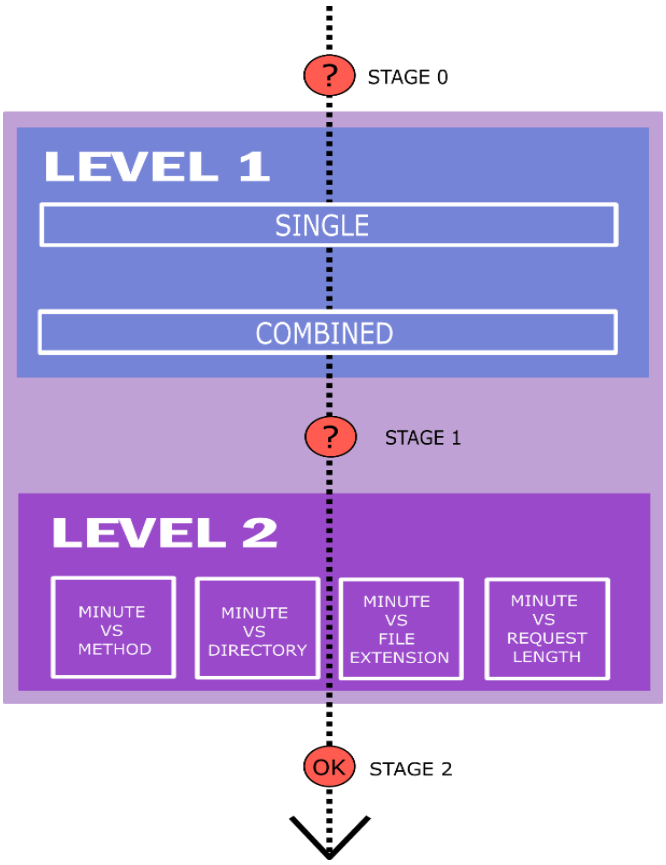


Figura 12. Estructura de Kassandra

5.3.1 Nivel 1

El Nivel 1 de Kassandra constituye el conjunto de reglas que una petición debe cumplir para no ser anómala en el *stage 1*. El Nivel 1 también tiene dos subniveles: *Single* (único) y *Combined* (combinado) (ver Figura 12). En el subnivel *Single*, Kassandra comprueba si las siguientes partes de la petición están presentes en el conjunto de datos de entrenamiento y da un valor de peligrosidad parcial del 0% al 75%:

- IP
- Método HTTP
- Directorio accedido

En el subnivel *Combined*, Kassandra comprueba si las siguientes combinaciones están incluidas en algún registro del conjunto de datos de entrenamiento, es decir, esos pares o tríos aparecen juntos. Este subnivel devuelve un valor de peligrosidad parcial del 0% al 25%.

- IP y método HTTP
- IP y directorio
- Método HTTP y directorio

- IP, método HTTP y directorio

El valor de peligrosidad para la petición en el *stage 1* será la suma de los valores ponderados de los subniveles simples y combinados. Por defecto, se utilizan los pesos presentados en la Tabla 15. Pesos para el Nivel 1 para la ponderación. El 75% de la ponderación se distribuye uniformemente entre las características del subnivel *Single* (los valores que no están en el entrenamiento son los más peligrosos) y el resto se distribuye se distribuye entre el subnivel *Combined*.

Subnivel	**Característica**	**Valor de ponderación**
Single	IP	0,25
	Método HTTP	0,25
	URL	0,25
Combined	IP y método HTTP	0,0625
	IP y directorio	0,0625
	Método HTTP y directorio	0,0625
	IP, método HTTP y directorio	0,0625

Tabla 15. Pesos para el Nivel 1

5.3.2 Nivel 2

En el Nivel 2, Kassandra profundiza el análisis de las peticiones que atraviesan el Nivel 1. Consiste en un conjunto de subfiltros donde cada uno tiene un módulo con un *EIF* entrenado que determinará si el valor de una determinada característica de la petición es anómalo o no a la hora del día en que ingresa la petición. Éstas son las características a las que nos referimos:

- Nombre del primer directorio de la URL (raíz en caso de que se acceda a un archivo en "/")
- Nombre del directorio raíz de la URL
- Extensión del archivo al que se accede (si se accede a un archivo de forma explícita)
- Longitud de la URL

5.3.2.1 *Extended Isolation Forest*

Para la implementación de los módulos de *EIF* utilizamos la librería propuesta por Hariri et al. [5] añadiendo concurrencia tanto en el entrenamiento como en el cálculo del valor de anomalía. Esta paralelización se realiza en la generación de árboles, ya que son independientes unos de otros, y en el cálculo de la distancia hasta la instancia a predecir desde la raíz de cada árbol. El número de hilos es configurable, lo que permite ajustar el rendimiento de los módulos a las especificaciones de la máquina donde se ejecute.

La librería modificada puede ser consultada en: https://github.com/albact7/eif.

5.3.2.2 *Módulos*

Los módulos implementados también están representados en la Figura 12 y se describen de la siguiente forma:

- Módulo 1. *Minuto vs Método*: en este módulo, Kassandra decide si es anómalo que un determinado método HTTP esté presente en una petición en un momento determinado del día. Devuelve un valor de peligrosidad de 0 a 1, siendo 1 una gran anomalía.
- Módulo 2. *Minuto vs Directorio*: en este módulo, Kassandra mira el primer directorio de la URL de la petición y decide si es normal o no que se acceda a este directorio en ese momento. Devuelve un valor de peligrosidad de 0 a 1, siendo 1 una gran anomalía.
- Módulo 3. *Minuto vs extensión de archivo*: en este módulo, Kassandra decide si acceder a un archivo con una extensión determinada en un momento determinado es anómalo o no. Devuelve un valor de peligrosidad de 0 a 1, siendo 1 una gran anomalía.
- Módulo 4*. Minuto vs longitud de la URL de la petición*: en este módulo, Kassandra considera la cantidad de caracteres que forman la URL de la petición. Devuelve un valor de peligrosidad de 0 a 1, siendo 1 una gran anomalía.

Como dijimos, cada módulo devuelve un valor de 0 a 1, experimentalmente podemos decir que los valores normales suelen quedarse en el rango [0.3,0.5] y los valores anómalos en el rango [0.5,0.8]. Para obtener un valor final de anomalía, ponderamos los resultados para los módulos 1, 2, 3 y 4 de en base a los pesos definidos en el archivo de configuración, por lo que el valor total de anomalía será la media ponderada de ellos. Pero un valor muy anómalo debería pesar más, por lo que estamos agregando un 20% más de importancia a ese valor en caso de que esto suceda (consideramos como anomalía un valor mayor del 65%).

Por defecto se utilizan los siguientes valores para la ponderación.

Módulo	**Valor de ponderación**
Minuto vs Método	0,25
Minuto vs Directorio	0,25
Minuto vs Extensión de archivo	0,25
Minuto vs Longitud de URL	0,25
Valor extra de peligrosidad	0,2

Tabla 16. Pesos para el Nivel 2

Después de eso, de acuerdo con la siguiente tabla, el valor de anomalía completo de la petición recibe una etiqueta para darle al usuario una retroalimentación sobre cuán peligroso es. El usuario puede utilizar esta información para decidir si atender a la alerta o no dependiendo de la política de restricción que siga para ese escenario.

ETIQUETAS DE PELIGROSIDAD					
0.0-0.5	*0.5-0.55*	*0.55-0.60*	*0.60-0.65*	*0.65-0.70*	*0.70-1.0*
0	1	2	3	4	5

Tabla 17. Etiquetas de peligrosidad y sus valores correspondientes

5.3.3 Diseño de políticas para la detección de anomalías

Cada caso es diferente y cada usuario de Kassandra puede tener diferentes necesidades. Para adaptar Kassandra a un conjunto de datos específico para lograr la máxima efectividad en la detección de anomalías, existen algunos parámetros que el usuario puede cambiar y experimentar hasta obtener el mejor resultado. Además, también sirve para estudiar la peligrosidad de cada característica en función de los requisitos del escenario.

5.3.3.1 *Parámetros de EIF*

Cambiar la cantidad de árboles al generar un EIF y el tamaño del subconjunto de datos de entrenamiento para cada árbol del EIF puede ayudar a ajustar las predicciones a un determinado conjunto de datos de entrenamiento y ajustar la precisión. En la Figura 11 se muestra un ejemplo de imagen de un módulo generada tras la detección de una anomalía. Los puntos verdes representan los datos presentes en el entrenamiento. Las zonas amarillas claras se corresponden con los valores considerados normales (más frecuentes). Cuanto más se acerca al color rojo oscuro, significa que esos valores aparecen en ese tramo horario con menor frecuencia y se consideran más anómalos. Por lo que podemos probar el modelo hasta que la predicción sea lo suficientemente correcta y las áreas de color se ajusten a nuestros datos.

5.3.3.2 *Pesos del cálculo de valor de peligrosidad*

Otra posible configuración está en el umbral de lo que consideramos una anomalía o no. Cada valor de anomalía parcial para ambos niveles tiene un peso en el resultado total del valor de peligrosidad. Este peso es configurable si deseamos darle más importancia a, por ejemplo, una IP extraña o un método HTTP que no debería estar ahí. Como dijimos antes, el valor de peligrosidad para el Nivel 1 debe ser del 0% para pasar al Nivel 2, el filtro previo evita el consumo extra de computación cuando una petición es claramente peligrosa (no pasa del Nivel 1). En el Nivel 2 hay un peso extra para que aquellos valores de peligrosidad por encima de cierto umbral sean más importantes para el resultado final.

La configuración de los pesos permite adaptar Kassandra al escenario requerido y obtener los correspondientes resultados adaptados.

5.3.4 Integración de infraestructura

En la Figura 13 se observa la integración de Kassandra con el servidor a proteger. Entre el usuario y el servidor se configura un proxy reverso que redirige los logs de las peticiones a Kassandra. Kassandra cuenta para ello con un servidor UDP en escucha y que cuando recibe una petición comienza a analizarla a través de los niveles de filtrado.

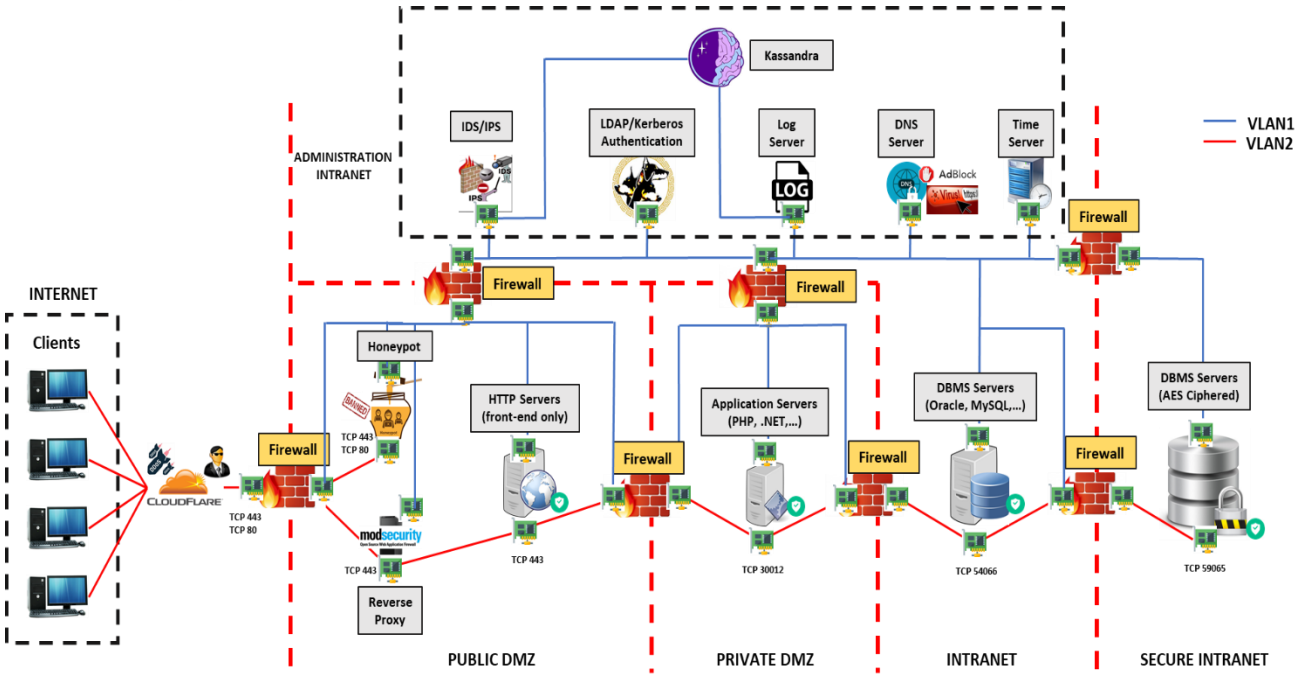


Figura 13. Integración de Kassandra

Genera como resultado un informe de los valores de anomalía para el Nivel 1 y, si continúa para el Nivel 2, otro resumen de dichos valores junto con una imagen que representa la petición analizada sobre el conjunto de valores de entrenamiento. En la Figura 14 se muestra una captura se muestra el informe de una petición no anómala. En las dos últimas líneas se observan el valor final de anomalía y la etiqueta de peligrosidad correspondiente, en este caso, 0.

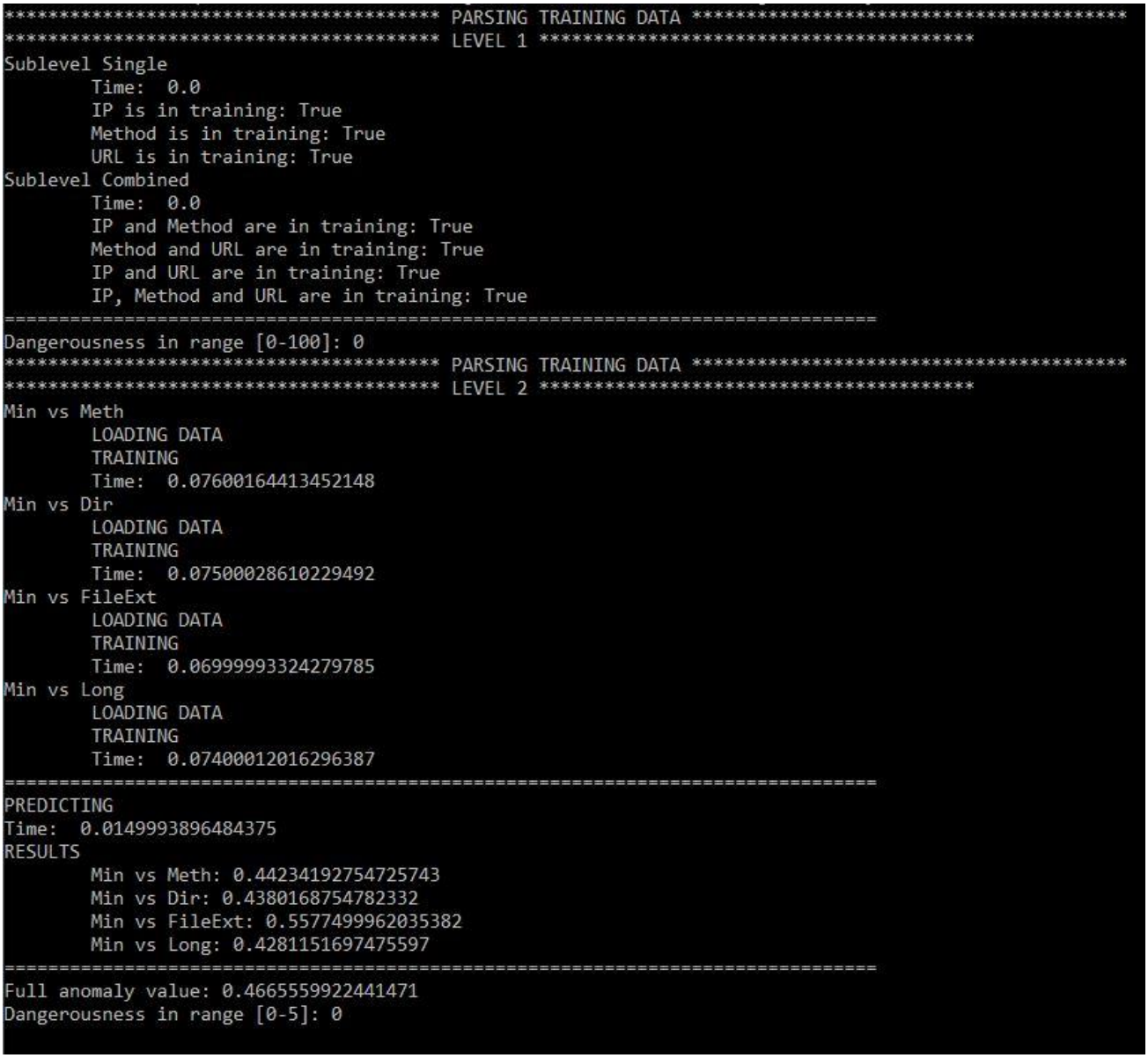


Figura 14. Captura de la salida por consola de Kassandra

En la Figura 15 se muestran las representaciones de los EIF para los cuatro módulos que intervienen en el Nivel 2. El punto de color negro se corresponde con la localización de la petición analizada. Podemos ver que los puntos verdes (peticiones HTTP en los datos de entrenamiento) se agrupan entre las 7 a.m. y las 6 p.m., el rango de horario laboral de la empresa. También observamos algunas anomalías entre las 11 p.m. y las 12 a.m., como dijimos, la presencia de anomalías en el dataset de entrenamiento no influye en las predicciones de los EIF.

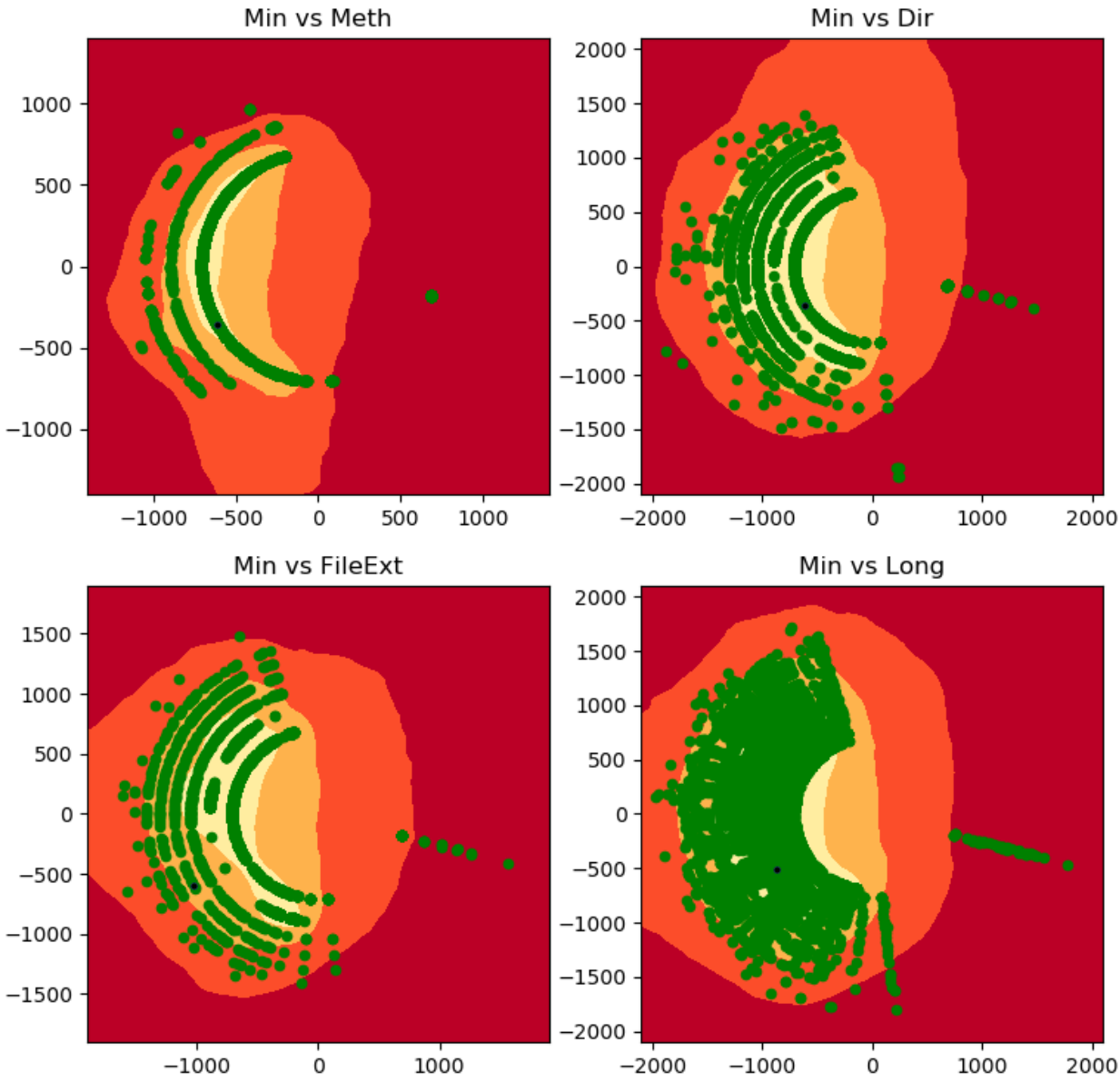
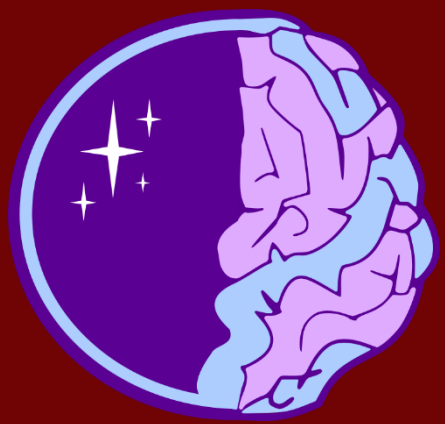


Figura 15. Imagen generada por Kassandra para el Nivel 2

Capítulo 6 Metodología del Trabajo



En esta sección se describirá la metodología seguida para la evaluación de este trabajo. Para ello se han diseñado una serie de casos de uso que sirven para entender el funcionamiento de un *UEBA*.

6.1 SISTEMA DE PRUEBAS

Los experimentos han sido realizados en una máquina Windows 10 Pro, 32GB de RAM, AMD Ryzen 7 3700X 8-Core Processor 3.6 GHz. Hemos configurado la librería modificada de EIF con 16 hilos (ver sección 5.3.2).

6.2 CASOS DE USO

Se han diseñado una serie de casos de uso típicos para probar Kassandra. Estos escenarios del modelo que un *UEBA* debe cubrir en un contexto de aplicación empresarial analizando las peticiones HTTP de un servidor. Además, estos intentan modelar los casos base que, en un contexto académico, un estudiante debería conocer para entender cómo funcionan los *UEBA*. A continuación, se describen los casos que suponen una anomalía. Se diseña uno para cada característica, ya sea del Nivel 1 o del Nivel 2 cuyo valor pueda cambiar de forma anómala. Las peticiones anómalas que se comprueban casos del Nivel 2, han superado el Nivel 1 sin puntuación anómala. Ahora se analizará la hora a la que se han realizado esas peticiones. La Tabla 18 muestra los casos diseñados con un ejemplo de ataque real potencial.

Nivel	**Subnivel**	**Nº de caso**	**Descripción**	**Ataques potenciales a detectar**
Nivel 1	Single	1	Una IP que pertenece a la red pero nunca ha accedido al servidor hasta ahora (no está en los datos de entrenamiento) accede a los recursos.	Intrusión en la red
		2	Un método HTTP nunca antes utilizado se utiliza en una petición al servidor	Testing de servidor web: herramienta de hackeo.
		3	Se accede a un directorio al que nunca se ha accedido antes (o se ha intentado acceder a él).	Enumeración de directorios [20].
	Combined	4	Una IP (un usuario) que nunca envía peticiones POST envía una.	Ataque interno o suplantación de identidad.

		5	An IP (a user) accesses a directory it has never accessed before.	Ataque interno o suplantación de identidad.
		6	Una petición aplica un método POST en un directorio al que solo llegan peticiones GET.	Testing de servidor web: herramienta de hackeo.
Nivel 2		7	*Minute vs method*. Una petición se recibe con un método que normalmente no se registra en ese momento.	Ataque interno o suplantación de identidad.
		8	*Minute vs directory*. Se recibe una petición a un directorio al que normalmente no se accede en ese momento.	Ataque interno o suplantación de identidad.
		9	*Minute vs file extension*. Se recibe una petición que accede a un archivo con una extraña extensión de archivo en ese momento.	Herramientas de escaneo para la realización de enumeraciones (paths comunes o archivos conocidos).
		10	*Minute vs request length*. Una petición se recibe con una dirección URL cuya longitud no está registrada o no está registrada en ese momento.	URL injection [21].
Ejemplo de configuración de pesos		11	Configuración de pesos para la adaptación de Kassandra a un escenario en el que un equipo de trabajadores de una empresa trabaja con una aplicación durante sus horas de trabajo, la aplicación maneja información crítica y su base de datos no puede ser alterada.	Si alguno de los ataques anteriores es particularmente peligroso debido a las características del escenario.

Tabla 18. Casos de uso

6.3 QUÉ CONSIDERAMOS COMO ANOMALÍA

Kassandra ofrece al usuario una puntuación de anomalía y una etiqueta de peligrosidad basadas en la desviación de la petición entrante respecto a la línea base construida a partir de los datos de entrenamiento. El usuario debe de tener en cuenta el escenario en el que Kassandra está funcionando e interpretar el resultado de la forma correspondiente. Las etiquetas arrojan información sobre esta posible interpretación. Además, si la petición llega al Nivel 2, el usuario puede

visualizar la distancia que hay entre la petición analizada y el resto. Tras esto, tiene suficiente información sobre la solicitud para considerarla peligrosa (y hacer las acciones de protección necesarias) o no. Esto depende del umbral de peligro que el escenario pueda tolerar.

Capítulo 7 Resultados Obtenidos

En esta sección se describen los resultados correspondientes a los casos de uso descritos en la sección anterior. Para cada caso de uso se muestra una tabla con la puntuación total de anomalía obtenida en cada nivel y el tiempo que tarda en realizar cada análisis. Respecto al Nivel 1, se muestra si esa característica o esa combinación están presentes en el conjunto de datos de entrenamiento y su puntuación de anomalía correspondiente en la ponderación. Respecto al Nivel 2, se indica de la misma forma la puntuación de anomalía que se obtiene tras analizar la petición con cada módulo.
Al final, también se muestra la etiqueta de peligrosidad asignada a la petición.
Tenemos en cuenta que el horario laboral es de 7 a.m. a 6 p.m. horas por lo que la mayoría del tráfico se encontrará en ese rango.

7.1 NIVEL 1

En las tablas que muestran los resultados del Nivel 1 se indica si se cumple dicha regla (*True/False*) y, en caso de que no se cumpla, el valor parcial de anomalía que supone sobre el total.

7.1.1 Single

7.1.1.1 *Caso 1*

```
1.1.1.1 - - [19/Mar/2021:13:51:10 +0000] "GET /caracterizacion/pycnometers HTTP/1.1" 200 70380
"http://10.250.30.72/characterization/pycnometer" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36"
```

Level 1	Single	IP	False	0.25
		METHOD	True	0
		DIRECTORY	True	0
	Combined	IP and METHOD	False	0.0625
		METHOD and DIRECTORY	True	0
		IP and DIRECTORY	False	0.0625
		IP, METHOD and DIRECTORY	False	0.0625
Dangerousness in range [0-100]				43.75

Tabla 19. Caso 1

7.1.1.2 *Caso 2*

```
10.250.121.111 - - [19/Mar/2021:13:51:10 +0000] "HEAD /caracterizacion/pycnometers HTTP/1.1" 200
70380 "http://10.250.30.72/characterization/pycnometer" "Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36"
```

Level 1	Single	IP	True	0
		METHOD	True	0.25
		DIRECTORY	True	0
	Combined	IP and METHOD	False	0.0625
		METHOD and DIRECTORY	True	0
		IP and DIRECTORY	False	0.0625
		IP, METHOD and DIRECTORY	False	0.0625
Dangerousness in range [0-100]				43.75

Tabla 20. Caso 2

7.1.1.3 *Caso 3*

```
10.250.121.111 - - [19/Mar/2021:13:51:10 +0000] "GET /anomaly HTTP/1.1" 200 70380
"http://10.250.30.72/characterization/pycnometer" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36"
```

Level 1	Single	IP	True	0
		METHOD	True	0
		DIRECTORY	False	0.25
	Combined	IP and METHOD	True	0
		METHOD and DIRECTORY	False	0.0625
		IP and DIRECTORY	False	0.0625
		IP, METHOD and DIRECTORY	False	0.0625
Dangerousness in range [0-100]				43.75

Tabla 21. Caso 3

7.1.2 Combined

7.1.2.1 *Caso 4*

```
10.250.56.169 - - [19/Mar/2021:13:51:10 +0000] "POST /caracterizacion/pycnometers HTTP/1.1" 200
70380 "http://10.250.30.72/characterization/pycnometer" "Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36"
```

Level 1	Single	IP	True	0
		METHOD	True	0
		DIRECTORY	True	0
	Combined	IP and METHOD	False	0.0625
		METHOD and DIRECTORY	True	0
		IP and DIRECTORY	True	0
		IP, METHOD and DIRECTORY	False	0.0625
Dangerousness in range [0-100]				12.5

Tabla 22. Caso 4

7.1.2.2 *Caso 5*

```
10.250.56.169 - - [19/Mar/2021:13:51:10 +0000] "GET /logfiles/info HTTP/1.1" 200 70380
"http://10.250.30.72/characterization/pycnometer" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36"
```

Level 1	Single	IP	True	0
		METHOD	True	0
		DIRECTORY	True	0
	Combined	IP and METHOD	True	0
		METHOD and DIRECTORY	True	0
		IP and DIRECTORY	False	0.0625
		IP, METHOD and DIRECTORY	False	0.0625
Dangerousness in range [0-100]				12.5

Tabla 23. Caso 5

7.1.2.3 *Caso 6*

```
10.250.56.86 - - [22/Mar/2021:14:16:22 +0000] "POST /tresdprinter HTTP/1.1" 304 0
"http://10.250.30.72/dashboard" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36 Edg/89.0.774.57"
```

Level 1	Single	IP	True	0
		METHOD	True	0
		DIRECTORY	True	0
	Combined	IP and METHOD	True	0
		METHOD and DIRECTORY	False	0.0625
		IP and DIRECTORY	True	0
		IP, METHOD and DIRECTORY	False	0.0625
Dangerousness in range [0-100]				12.5

Tabla 24. Caso 6

En las pruebas de este apartado, el filtrado del Nivel 2 no se llega a ejecutar debido a que el valor de peligrosidad en el Nivel 1 es mayor que 0. Según los pesos que hemos asignado, si una de las características del subfiltro *Single* no está presente en el conjunto de entrenamiento, lo que significa que la regla no se cumple, será más peligrosa que una regla *Combined* no cumplida. Observamos que las solicitudes que no cumplen una regla *Single* tienen un valor de peligro más alto. Tras recibir los resultados, el administrador puede bloquear la IP o el método extraño, eliminar u ocultar ese directorio al que nunca se ha accedido o localizar la IP que ha incumplido una regla y a la que está asignada.

7.2 NIVEL 2

La petición ha de obtener un valor 0 de anomalía en el Nivel 1 para pasar al 2, por lo que os resultados para el Nivel 1 son los mismos para los siguientes casos. Estos resultados se corresponden con la siguiente tabla.

Level 1	Single	IP	True	0
		METHOD	True	0
		DIRECTORY	True	0
	Combined	IP and METHOD	True	0
		METHOD and DIRECTORY	True	0
		IP and DIRECTORY	True	0

		IP, METHOD and DIRECTORY	True	0
Dangerousness in range [0-100]				0

Tabla 25. Resultados del Nivel 1 para los casos de uso de esta sección

En el Nivel 2 se tiene en cuenta la hora a la que se recibe la petición. En nuestro contexto empresarial, el horario laboral de los empleados entra en un rango de 7 a.m. a 6 p.m.

En las tablas con los resultados del Nivel 2 se indican los módulos y los valores de anomalía que arrojan (sin ponderar). Además, se muestra el valor total de anomalía teniendo en cuenta la ponderación de todos los pesos y la etiqueta de peligrosidad. El valor que se obtiene de los módulos es el calculado por los EIF, experimentalmente observamos que los valores menores de 0,5 no constituyen una anomalía mientras que los valores anómalos se encuentran en el rango [0,5 – 0,8].

Junto con la tabla con los valores numéricos se muestra el gráfico correspondiente. Se genera un mapa con valores anómalos por módulo que aplique a la petición, sin embargo, sólo mostraremos el mapa del módulo a estudiar en cada caso de uso.

7.2.1 Caso 7

```
10.250.30.142 - - [22/Mar/2021:17:00:00 +0000] "POST /runs_data/results HTTP/1.1" 201 454
"http://10.250.30.72/atomizations/runs_data/374" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36"
```

Level 2	Min vs Method	0.606139347491226
	Min vs Directory	0.5043299340972276
	Min vs File Extension	None
	Min vs Length	0.42358875578858196
Full anomaly value		0.5113526791256785
Dangerousness in range [0-5]		1

Tabla 26. Caso 7

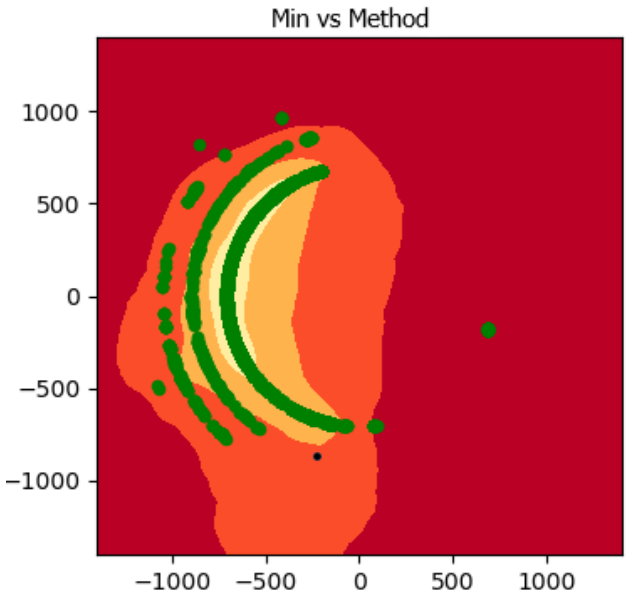


Figura 16. Caso 7

Debido a la distribución del trabajo, los usuarios de la red suelen enviar peticiones POST entre las 8 a.m. y las 5 p.m., porque es cuando se suelen cargar datos a la base de datos. Sin embargo, se recibe una petición de ese tipo a 6 p.m. Esto supone una anomalía y se refleja en el valor obtenido en el módulo *Min vs Method* (0,6). En el mapa vemos la petición (punto negro) sobre el color naranja, lo que indica que es una zona horaria con altos valores de anomalía para este método POST. Si, en cambio, fuese un método GET, se encontraría en la circunferencia verde del interior sobre la zona de color amarillo oscuro, lo que indica un valor más pequeño de anomalía. Una posible respuesta del administrador del sistema sería identificar al usuario (IP) que ha cargado datos en un rango de tiempo anormal y descubrir por qué.

7.2.2 Caso 8

```
10.250.120.43 - - [26/Mar/2021:16:00:10 +0000] "GET /login HTTP/1.1" 200 1747 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36"
```

Level 2	Min vs Method	0.48602944208973614
	Min vs Directory	0.652887406988883
	Min vs File Extension	None
	Min vs Length	0.5917812781761074
Full anomaly value		0.5996957851227678
Dangerousness in range [0-5]		2

Tabla 27. Caso 8

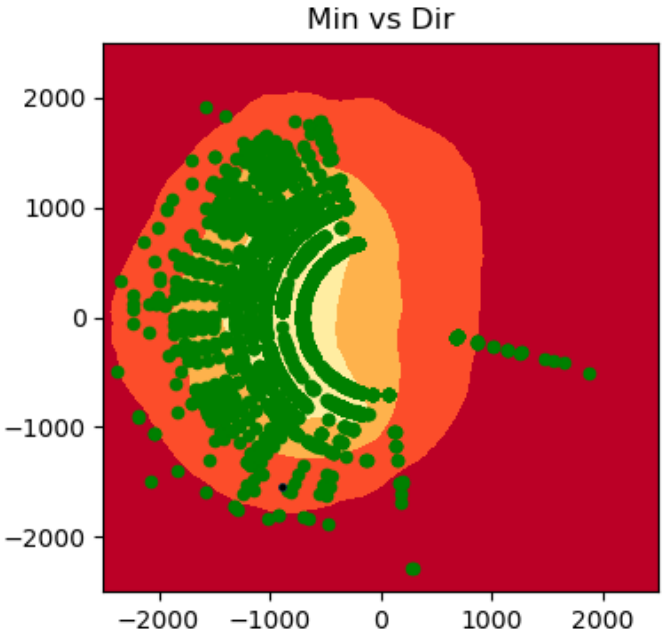


Figura 17. Caso 8

En este caso, observamos que tanto el módulo *Min vs Length* como el *Min vs Directory* arrojan valores anómalos sobre la petición analizada. En el caso de *Min vs Directory* se debe a que los trabajadores suelen iniciar sesión mediante la URL */login* de 7 a.m. a 3 p.m. Pero esta petición se recibe a las 4 p.m. Por otro lado, *Min vs Length* también señala una anomalía ya que es posible que no haya más peticiones con una URL de igual longitud a esa hora entre los datos de entrenamiento. Si la URL a la que se accede es */login*, significa que todos los empleados se conectan al sistema en esa franja horaria y el administrador debería comprobar por qué hay alguien que se conecta a una hora extraña, ya que una persona que no está trabajando en ese momento podría haber sido suplantada por alguien ajeno al horario de la empresa.

7.2.3 Caso 9

```
10.250.30.142 - - [31/Mar/2021:14:35:36 +0000] "GET /logfiles/346C.txt?n=1617172971389 HTTP/1.1"
200 2358608 "http://10.250.30.72/atomizations/logfiles" "Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36"
```

Level 2	Min vs Method	0.4717708620641212
	Min vs Directory	0.44585501989086523
	Min vs File Extension	0.5094796821658436
	Min vs Length	0.48469759318351907
Full anomaly value		0.4779507893260873
Dangerousness in range [0-5]		0

Tabla 28. Caso 9

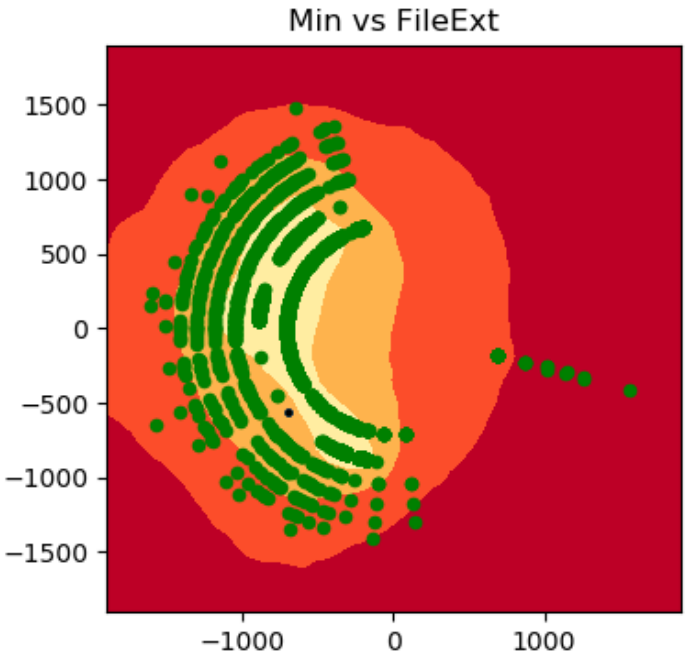


Figura 18. Caso 9

Se recibe una petición a un fichero con la extensión *.txt* a las 2 p.m., sin embargo, los trabajadores acceden a los ficheros de log *.txt* en horas concretas, pero no suelen hacerlo de 12 a.m. a 4 p.m. El valor de anomalía para este módulo es de 0,5. No constituye una anomalía muy peligrosa a pesar de que en ese horario no haya peticiones a *.txt* en los datos de entrenamiento. Esto se debe a que, en general, a esas horas el servidor presenta mucho tráfico (como se observa en los puntos verdes alrededor de nuestra petición) y a que es una extensión recurrente en el día a día (se encuentra en la segunda circunferencia verde desde el centro, es decir, es la segunda extensión más corriente). Si hay un proceso automático que analiza los archivos de registro, alguien podría haber modificado la configuración de su ejecución y el administrador debería comprobar por qué se ha ejecutado en ese momento.

7.2.4 Caso 10

```
10.250.56.169 - - [19/Mar/2021:13:51:10 +0000] "GET
/caracterizacion/pycnometersdgfdgdfhgf?asdhjadss HTTP/1.1" 200 70380
"http://10.250.30.72/characterization/pycnometer" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36"
```

Level 2	Min vs Method	0.42968298074313704
	Min vs Directory	0.4257355117151063
	Min vs File Extension	None

	Min vs Length	0.6500609045840737
Full anomaly value		0.5462967973517627
Dangerousness in range [0-5]		1

Tabla 29. Caso 10

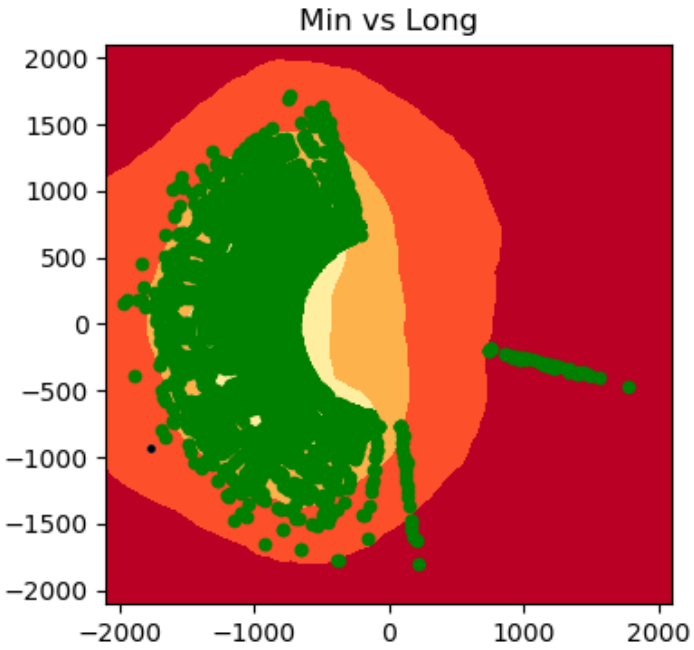


Figura 19. Caso 10

Durante la jornada laboral se suelen enviar peticiones de URL de longitudes similares. Se suele acceder a los mismos recursos o a archivos que sigan un código de nombrado determinado, de esta forma comparten la longitud de la URL. En la misma jornada laboral se registra una petición con una URL de 48 caracteres. Observamos que el módulo *Min vs Length* arroja un valor anómalo. Esto se debe a que en los datos de entrenamiento no hay más URLs con esa misma longitud. Aunque la longitud de URL es un valor cuantitativo, los datos que consume el EIF están modelados según el orden de frecuencias explicado anteriormente. La representación de los datos en forma de circunferencias concéntricas lleva a que los datos situados cerca del centro (por estar más cerca, ya que se distribuyen en los ejes Y de forma radial) tengan menor valor de anomalía. La representación por frecuencias impide que, por ejemplo, URLs de poca longitud, pero poco frecuentes, sean consideradas ordinarias. Se podría haber detectado una inyección de URL, por lo que el administrador debería comprobar si la URL es peligrosa y quién está asignado a la IP de origen.

7.2.5 Caso 11

Para este caso utilizamos la misma petición que para el caso 7, así podremos ver cómo afecta la redistribución de los pesos al valor total de anomalía.

```
10.250.30.142 - - [22/Mar/2021:17:00:00 +0000] "POST /runs_data/results HTTP/1.1" 201 454
"http://10.250.30.72/atomizations/runs_data/374" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36"
```

La nueva configuración de pesos es la siguiente.

Módulo	**Valor de ponderación**
Minuto vs Método	0.6
Minuto vs Directorio	0.1333
Minuto vs Extensión de archivo	0.1333
Minuto vs Longitud de URL	0.1333
Valor extra de peligrosidad	0.2

Tabla 30. Pesos para el caso 11

Level 2	Min vs Method	0.6093968103557534
	Min vs Directory	0.5078636245700898
	Min vs File Extension	None
	Min vs Length	0.4227425841885193
Full anomaly value		0.5581661476860799
Dangerousness in range [0-5]		2

Tabla 31. Caso 11

Podemos observar que obtenemos resultados similares al caso 7 pero con mayor valor de anomalía debido a que ha aumentado peso del módulo que arroja el valor más anómalo. La imagen generada se corresponde también con la del caso 7.

Debido al escenario descrito, la gran mayoría de peticiones que envían al servidor son GET para consultar información. Sin embargo, aunque en menor medida, también envían peticiones POST, pero en un rango horario más restrictivo: de 7 a.m. a 4 p.m. La aplicación empresarial acepta una serie de endpoints que normalmente sólo incluyen métodos GET. En el dataset de entrenamiento aparecen peticiones POST, en menor medida que el método anterior, y en un rango horario de 7 a.m. a 4 p.m. aproximadamente. Entra en el servidor una petición POST a las 5 p.m. y los nuevos pesos aumentan el valor total de anomalía, subiendo la etiqueta de peligrosidad a 2. El valor de la etiqueta anima al administrador a no pasar por alto la alerta porque la solicitud constituye una anomalía crítica.

Capítulo 8 Conclusiones y Trabajo Futuro

8.1 CONCLUSIONES Y TRABAJO FUTURO

Kassandra es capaz de detectar anomalías en el comportamiento de los usuarios o entidades con respecto a la interacción HTTP con un servidor. A continuación, los administradores del sistema pueden analizar qué parte de la petición es peligrosa y en qué grado. Distingue entre dos formas de filtrado: sólo Nivel 1 si la petición se considera anómala con un solo filtro o Nivel 1 y Nivel 2 si requiere un análisis más profundo. Se puede configurar para dar más importancia a ciertas características de las peticiones si el escenario en el que se despliega lo requiere. Kassandra ofrece el comportamiento de los sistemas *UEBA* adaptados a entornos educativos y de bajo volumen computacional.

El sistema consigue buenos resultados en la detección de anomalías para los casos de uso esperados. El sistema experto del Nivel 1 identifica en primer lugar las anomalías que más difieren de las líneas base. Mientras, los EIF del Nivel 2 consiguen aislar con éxito los puntos lejanos a donde se encuentran las agrupaciones de las peticiones de los datos de entrenamiento. Además, la visualización de los resultados con el desglose de valores y los gráficos generados permiten analizar por qué una petición no se parece a las habituales y en qué medida.

En el futuro se implementarán otros protocolos además de HTTP. El prototipo actual no diferencia entre los usuarios, pero otro proyecto a trabajar sería la construcción de un modelo para cada usuario a monitorizar. De esta manera se podría tener control sobre las líneas base de cada usuario en contacto con la red. Otro punto a trabajar sería también distinguir entre los días de la semana y las vacaciones o días lectivos, ya que puede darse el caso de que los comportamientos habituales cambien dependiendo de estas variables. También se puede desarrollar una funcionalidad complementaria para generar un informe para estudiar los logs (conjunto de datos de entrenamiento) con el fin de detectar irregularidades e identificar anomalías pasadas. Esta herramienta también está planeada para ser incorporada como parte de los materiales educativos de la asignatura de Seguridad de Sistemas Informáticos de la Escuela de Ingeniería Informática de la Universidad de Oviedo [22].

El código fuente está disponible en https://github.com/Egida-Kassandra/kassandra.

8.2 DIFUSIÓN DE RESULTADOS

Se ha enviado el artículo que describe este proyecto, bajo el título *Kassandra: Efficient application of Extended Isolation Forests to detect anomalous user behaviours*, a la revista *ACM Transactions on Intelligent Systems and Technology*[16].

Kassandra en una de sus primeras versiones consiguió, junto con Egida[17], llegar a la semifinal de la Primera Edición del INNCYBER Innovation Award[18].

Po último, Kassandra se incorporará dentro del material de la asignatura de Seguridad de Sistemas Informáticos del Grado en Ingeniería de Software de la Escuela de Ingeniería Informática de Oviedo [22].

[16]ACM Transactions on Intelligent Systems and Technology https://dl.acm.org/journal/tist/ [Accedido por última vez el 13 de junio de 2021]
[17]Egida & Kassandra https://github.com/Egida-Kassandra [Accedido por última vez el 10 de junio de 2021]
[18]Inncyber Innovation Hub https://www.inncyberinnovationhub.com/ [Accedido por última vez el 10 de junio de 2021]

Capítulo 9 Bibliografía

REFERENCIAS BIBLIOGRÁFICAS

[1] J. S. Broderick, "ISMS, security standards and security regulations," *Inf. Secur. Tech. Rep.*, vol. 11, no. 1, pp. 26–31, Jan. 2006, doi: 10.1016/j.istr.2005.12.001.

[2] A. S. Ashoor and S. Gore, "Difference between Intrusion Detection System (IDS) and Intrusion Prevention System (IPS)," in *Communications in Computer and Information Science*, 2011, vol. 196 CCIS, pp. 497–501, doi: 10.1007/978-3-642-22540-6_48.

[3] F. Montenegro, "Expanding Machine Learning Applications on the Endpoint," 2018. Accessed: 31-May-2021. [Online]. Available: https://www.blackberry.com/content/dam/cylance/documents/pdf/451ResearchExpandingMLApplicationsontheEPReport.pdf.

[4] S. Khaliq, Z. U. Abideen Tariq, and A. Masood, "Role of User and Entity Behavior Analytics in Detecting Insider Attacks," in *1st Annual International Conference on Cyber Warfare and Security, ICCWS 2020 - Proceedings*, 2020, doi: 10.1109/ICCWS48432.2020.9292394.

[5] S. Hariri, M. Carrasco, and R. J. Brunner, "Extended Isolation Forest."

[6] F. Falcão *et al.*, "Quantitative comparison of unsupervised anomaly detection algorithms for intrusion detection," in *Proceedings of the ACM Symposium on Applied Computing*, 2019, vol. Part F1477, pp. 318–327, doi: 10.1145/3297280.3297314.

[7] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.

[8] B. Schölkopf, S. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution." Accessed: 12-May-2021. [Online]. Available: http://direct.mit.edu/neco/article-pdf/13/7/1443/814849/089976601750264965.pdf.

[9] "sklearn.ensemble.IsolationForest — scikit-learn 0.23.1 documentation." https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html (accessed Jun. 10, 2020).

[10] M. Goldstein and S. Uchida, "A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data," *PLoS One*, vol. 11, no. 4, p. e0152173, Apr. 2016, doi: 10.1371/journal.pone.0152173.

[11] L. Sun, S. Versteeg, S. B. Bozta¸s, and A. Rao, "Detecting Anomalous User Behavior Using an Extended Isolation Forest Algorithm: An Enterprise Case Study."

[12] J. Bülow and M. Scherman, "Insider Threat detection using Isolation Forest," 2018.

[13] K. Guercio, "Best User and Entity Behavior Analytics (UEBA) Tools," 2020. https://www.esecurityplanet.com/products/best-user-and-entity-behavior-analytics-ueba-tools/ (accessed May 10, 2021).

[14] "Cortex XDR Whitepaper - Palo Alto Networks." https://www.paloaltonetworks.com/resources/whitepapers/cortex-xdr (accessed May 10, 2021).

[15] J. Petters, “What is UEBA? Complete Guide to User and Entity Behavior Analytics | Varonis,” 2020. https://www.varonis.com/blog/user-entity-behavior-analytics-ueba/ (accessed May 10, 2021).

[16] J. L. Pharr, “OpenUBA: A SIEM-agnostic, Open Spurce Framework for Modeling User Behavior,” 2020. https://github.com/GACWR/ouba-paper/blob/master/openuba.pdf.

[17] “TensorFlow.” https://www.tensorflow.org/ (accessed May 26, 2021).

[18] “Scikit Learn.” https://scikit-learn.org/stable/ (accessed May 27, 2021).

[19] A. Abraham, “Rule-Based Expert Systems,” in *Handbook of Measuring System Design*, John Wiley & Sons, Ltd, 2005.

[20] C. Sima, “Start Secure. Stay Secure.™ Are your web applications vulnerable?,” 2005.

[21] S. Kumar, R. Mahajan, N. Kumar, and S. K. Khatri, “A study on web application security and detecting security vulnerabilities,” in *2017 6th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2017*, 2018, vol. 2018-Janua, pp. 451–455, doi: 10.1109/ICRITO.2017.8342469.

[22] J. M. Redondo, “Improving Concept Learning Through Specialized Digital Fanzines,” 2021, pp. 134–143, doi: 10.1109/icse-seet52601.2021.00023.

CAPÍTULO 10 ANEXOS

10.1 PLAN DE GESTIÓN DE RIESGOS

10.1.1 Metodología

El presente Plan de Gestión de Riesgos recoge el total de los riesgos identificados para el proyecto *Kassandra: Aplicación de técnicas de Machine Learning para la detección de comportamientos anómalos de usuarios*. Estos riegos podrían hacer que el proyecto no llegara a concluirse, suponiendo una importante pérdida económica, de modo que resulta imprescindible categorizar y valorar el impacto que podrían causar cada uno de ellos.

En este documento se incluyen todos los riesgos identificados en la fase de análisis, aunque a lo largo del desarrollo del proyecto estos evolucionarán o cambiará su probabilidad de ocurrencia.

10.1.1.1 *Metodología General*

La metodología que se empleará durante la redacción de este Plan de Gestión de Riesgos toma como base la descrita por Boehm, la cual contiene dos fases:

1. En la primera de las fases se comienza por identificar, analizar y priorizar los riesgos que pueden producirse en el proyecto que se procederá a desarrollar.
2. La segunda fase se encarga de definir las estrategias para tratar y controlar los riesgos definidos en la fase anterior.

Partiendo de esta base, la metodología empleada incorpora nuevos conceptos obtenidos del PMBOK, tales como la matriz de riesgos.

La planificación de riesgos será realizada de acuerdo con una estrategia proactiva, es decir, se realizará una evaluación previa de todos los riegos susceptibles de producirse en el proyecto, evaluando de igual modo las consecuencias que estos podrían acarrear.

10.1.1.2 *Metodología de Gestión de Riesgos*

De acuerdo con la metodología general planteada en el apartado anterior se plantea una metodología de gestión de riesgos que se divide en dos etapas independientes: la evaluación de los riegos y el control de los mismos.

1. Evaluación de los riesgos.
   1.1. Identificación de los riesgos: Permite identificar todas las adversidades que se podrían originar en el contexto de ejecución del proyecto.
   1.2. Análisis de los riesgos: Determinar la probabilidad de que el riesgo se produzca y el impacto que este pudiera originar en el desarrollo del proyecto.
   1.3. Priorización de riesgos: Priorizar aquellos riegos que suponen una verdadera amenaza para el proyecto y que por tanto deberían ser tratados con mayor urgencia.
2. Control de los riesgos.

2.1. Planificación de los riesgos: Estudio de cómo tratar de evitar o reducir el impacto causado por el riesgo en el proyecto o cómo potenciar los beneficios si el riesgo se entiende como una oportunidad para el proyecto.
2.2. Solución a los riesgos: Definición de planes de contingencia para evitar o definir cómo proceder si el riesgo se produce verdaderamente.

2.3. Monitorización de los riesgos: Actualización y definición de planes para tener bajo control los riesgos que se identificaron en las primeras fases.

10.1.2 Herramientas y Tecnologías

Para la identificación de los riesgos se tomarán las siguientes herramientas:

10.1.2.1 *Tormenta de ideas*

Cada uno de los individuos trabaja de forma independiente y comparte con el resto del equipo solamente aquellas ideas que considera más relevantes para el proyecto, de modo que todos los miembros tienen las mismas oportunidades de expresar las ideas.

10.1.2.2 *Checklists*

Se realiza por medio de la evaluación de un conjunto de puntos que son aplicables a diferentes proyectos y que permiten, a partir de la experiencia, evitar que se comentan errores graves y que se producen de manera recurrente.

10.1.2.3 *Estudio de incidencias*

Consiste en el análisis de las causas que han originado riesgos en proyectos pasados a fin de poder detectar la aparición de alguno de esos riesgos en el contexto del proyecto actual.

10.1.2.4 *Método Delphi*

Entrevista estructurada realizada a un conjunto de expertos que preverán los riegos susceptibles de originarse en un determinado contexto del proyecto.

10.1.3 Roles y Responsabilidades

10.1.3.1 *Director de proyecto*

Dirige y realiza el seguimiento de los riesgos además de resolver los problemas surgidos durante su manejo e integración de su gestión en la gestión del proyecto. Gestiona los riesgos no asignados a los siguientes responsables.

10.1.3.2 *Responsable de riesgos de investigación*

Se encarga de supervisar y gestionar los riesgos que atienden a las fases de investigación de la gestión y desarrollo de los proyectos.

10.1.3.3 *Responsable de riesgos de infraestructuras*

Supervisa y gestiona los riesgos relacionados con las infraestructuras cuyo uso se incluye en sus proyectos, ya sean propias o contratadas.

10.1.4 Presupuesto

El presupuesto de Gestión de riesgos se muestra a continuación.

Item	Concepto	Asignación (€)
1	Identificación de riesgos	69€
2	Análisis y priorización de los riesgos	46€
3	Planificación de los riesgos	46€
4	Definición de planes de contingencia	92€
5	Actualización y monitorización de los riesgos	92€
	TOTAL	345 €

10.1.5 Calendario

A continuación, se presentan las fechas de entrega/finalización de cada punto de la gestión de riesgos.

Hito/Actividad	Fecha
Identificación de riesgos	15/01/21
Análisis y priorización de los riesgos	16/01/21
Planificación de los riesgos	18/01/21

Definición de planes de contingencia	19/01/21
Actualización y monitorización de los riesgos	20/01/21

10.1.6 Categorías de Riesgo

A fin de poder identificar los riesgos y conocer la estructura de los mismos se categorizan dentro de una de las siguientes categorías, pudiendo pertenecer cada uno de ellos a más de una categoría:

1. Técnico
   1.1. Requisitos
   1.2. Tecnología
   1.3. Prestaciones y fiabilidad
   1.4. Calidad
2. Organizacional
   2.1. Dependencias del proyecto
   2.2. Recursos
   2.3. Financiación
   2.4. Personal.
3. Gestión del proyecto
   3.1. Estimación
   3.2. Planificación
   3.3. Control
   3.4. Comunicación
4. Externo:
   4.1. Proveedores
   4.2. Usuario
   4.3. Tiempo
   4.4. Fenómenos

10.1.7 Definiciones de probabilidad

Nombre	Porcentaje Equivalente	Descripción
Muy baja	0% - 20%	La probabilidad de que el riesgo se materialice es muy poco probable
Baja	20% - 40%	La probabilidad de que el riesgo llegue a ocurrir es baja
Media	40% - 60%	Existe cierta probabilidad de que el riesgo llegue a producirse
Alta	60% - 80%	Hay altas probabilidades de que la situación que describe el riesgo se de en el contexto del proyecto
Muy alta	80% - 100%	El riesgo afectará al proyecto con una alta probabilidad

10.1.8 Definiciones de impacto por objetivos

Impacto sobre los objetivos principales					
Objetivos de proyecto	*Escalas relativas o numéricas*				
	Muy bajo	*Bajo*	*Moderado*	*Alto*	*Muy alto*
Alcance	Reducciones del alcance poco apreciables	Reducciones en módulos poco relevantes del proyecto	Reducciones en áreas importantes para el proyecto	Reducción en módulos del proyecto que impidan lograr los requisitos de alto nivel	Reducciones en el proyecto que provocan que el proyecto deje de ser funcional
Calidad	Reducción de la calidad de la solución planteada en temas menores, como el diseño de componentes visuales	Reducción de la calidad en navegadores que no soportan los últimos avances en los estándares	Reducción de la calidad que requeriría una aprobación del director del proyecto	Reducción de la calidad que desestime la presentación del proyecto ante tribunal	Reducción de la calidad que implique la no usabilidad completa del producto
Coste	Incremento poco notable en el coste	Incremento en el coste menor al 5%	Incremento en el coste de entre el 5% y el 10%	Incremento en el coste de entre el 10% y el 20%	Incremento en el coste superior al 20%
Tiempo	Incremento en el tiempo de menos de 10 horas	Incremento de menos de un 5% de lo planificado	Incremento en tiempo de entre el 5% y el 10%	Incremento en el tiempo de entre el 10% y el 20%	Incremento de tiempo superior al 20%

10.1.9 Matriz de Probabilidad e Impacto

Very High	***0.90***	0.05	0.14	0.27	0.50	0.81
High	***0.70***	0.04	0.11	0.21	0.39	0.63
Medium	***0.50***	0.03	0.08	0.15	0.28	0.45
Low	***0.30***	0.02	0.05	0.09	0.17	0.27
Very Low	***0.10***	0.01	0.02	0.03	0.06	0.09
		0.05	***0.15***	***0.30***	***0.55***	***0.90***
		Very Low	**Low**	**Medium**	**High**	**Very High**

10.1.10 Tolerancias

El umbral de riesgo se define en una cota de 0,40. Sobrepasado ese valor se estimará que los riesgos suponen una verdadera amenaza para el proyecto. Los valores de impacto menores de dicho valor no serán tomados como relevantes en un primer término.

10.1.11 Planes de Contingencia

10.1.11.1 *Presupuesto*

En los momentos en los que resulte imprescindible una alteración del presupuesto definido y acordado antes del inicio del proyecto, estos aspectos serán cuantificados y valorados por el director del proyecto y el jefe de proyectos y no deberán exceder, en ningún término un 5% del coste del proyecto.

10.1.11.2 *Planificación*

La fecha final de cierre del proyecto está prevista para el 4 de junio de 2021, sin embargo, se tienen en cuenta los riesgos que conlleva un proyecto de investigación en cuanto a la planificación y se puede ampliar mientras el trabajo no supere el 5% del presupuesto acordado en el apartado anterior.

10.1.12 Formatos de la Documentación

Para la gestión de la documentación se tomarán como referencia las siguientes normas:

- UNE-ISO 31000:2010: Gestión del riesgo
- UNE-EN 31010:2011: Gestión del riesgo. Técnicas de apreciación del riesgo

10.1.13 Seguimiento

Para el seguimiento de los riesgos se establece la siguiente política:

- Los riesgos serán analizados y recalculado su impacto cada mes de transcurso del proyecto.
- De forma mensual se realizará un análisis de nuevas amenazas que pudieran surgir en el contexto del proyecto.
- La forma en que se revisarán los riesgos es mediante la reunión de los stakeholders que se verán afectados por tales riesgos.

GNU FREE DOCUMENTATION LICENSE

Version 1.3, 3 November 2008

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not

explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. *Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.*
B. *List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.*
C. *State on the Title page the name of the publisher of the Modified Version, as the publisher.*
D. *Preserve all the copyright notices of the Document.*
E. *Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.*
F. *Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.*
G. *Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.*
H. *Include an unaltered copy of this License.*
I. *Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.*
J. *Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.*

K. *For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.*
L. *Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.*
M. *Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.*
N. *Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.*
O. *Preserve any Warranty Disclaimers.*

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique

number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those

notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See https://www.gnu.org/licenses/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with … Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Kassandra: Efficient application of Extended Isolation Forests to detect anomalous user behaviours

ALBA COTARELO, University of Oviedo, Spain
ANTONIO PAYA, University of Oviedo, Spain
JOSE MANUEL REDONDO, University of Oviedo, Spain

This paper describes a technique to efficiently implement UEBA functionalities using a recent unsupervised learning algorithm named Extended Isolation Forests. The prototype designed to test this technique demonstrated adequate user behaviour anomaly detection capabilities analysing real-user HTTP request data over a set of typical use cases. These cases were designed to model typical scenarios where an attack may be performed. The results are promising enough to extend the technique to more protocols, low-budget companies that cannot afford commercial UEBA solutions, or educational environments.

CCS Concepts: • **Security and privacy → Intrusion/anomaly detection and malware mitigation**;

Additional Key Words and Phrases: Extended Isolation Forests, User Behaviour, Anomaly detection, UEBA

ACM Reference Format:
Alba Cotarelo, Antonio Paya, and Jose Manuel Redondo. XXXX. Kassandra: Efficient application of Extended Isolation Forests to detect anomalous user behaviours. *J. ACM* XX, X, Article XXX (January XXXX), 20 pages. https://doi.org/X

1 INTRODUCTION

The correct implementation of international Information Security Management Standards (ISO 27001, ISMS) [6] must ensure the correct configuration of an infrastructure's systems in terms of security. Additionally, infrastructure security also requires constant monitoring and surveillance of what is happening in each system, so that events that are considered attacks or anomalies can be detected and reported in order to respond to them early and prevent further damage. This surveillance and monitoring work is usually identified with *Intrusion Detection Systems (IDS)*, which generate alerts once an attack has occurred, and *Intrusion Prevention Systems (IPS)*, which also monitor traffic and block it while the attack is occurring [5], or *Endpoint Detection and Response (EDR)* systems, which complement management and antivirus solutions for endpoints to identify and eliminate threats [15]. More recently, these systems are also complemented by *User and Entity Behaviour Analytics (UEBA)*. *IDS* and *IPS* are focused on detecting illegitimate users actions and known attacks. However, legitimate users have "natural" access to the services and infrastructures as part of their work, but they can be impersonated or misbehave due to many reasons, causing "insider attacks". Typically, these systems are not specifically designed to prevent them [12]. *UEBAs* base their behaviour on the detection of anomalous behaviour in the network that deviates from baselines considered as usual behaviour patterns.

Authors' addresses: Alba Cotarelo, alba164@gmail.com, University of Oviedo, Department of Computer Science, Oviedo, Spain; Antonio Paya, antoniopaya@outlook.com, University of Oviedo, Department of Computer Science, Oviedo, Spain; Jose Manuel Redondo, redondojose@uniovi.es, University of Oviedo, Department of Computer Science, Oviedo, Spain.

0004-5411/XXXX/1-ARTXXX $15.00
https://doi.org/X

Platforms that currently implement *UEBAs* are typically expensive and require infrastructure that may not be affordable for many businesses. For example, solutions from *Aruba*, *Exabeam* or *Varonis*. This means that these businesses cannot typically take advantage of the additional layer of security that is increasingly necessary today [18]. This research project aims to implement a *UEBA-type system* capable of detecting anomalies in user behaviour using a modern unsupervised learning algorithm called *Extended Isolation Forest (EIF)* [11], which is able to isolate anomalies in a dataset through recursive partitioning, reducing false positives compared to traditional *Isolation Forest*. With it, the system is able to detect anomalies that may occur in the traffic received by a server according to the time at which these requests are received. The prototype designed has a limited resource cost that allows it to be used in environments with a lower processing capacity and also in educational environments, so that students can see the capabilities of a system of this type first-hand, which will benefit their future professional training. To test the viability of this technique, we limited the analysis to HTTP requests, and other protocols will be added in the future. The implemented prototype is able to detect anomalous behaviour for a number of typical use cases and based on certain characteristics of HTTP requests. In case any of the features are not present in the training data, it also includes a pre-filtering to avoid the computational cost of running the Extended Isolation Forest. Finally, it calculates a danger score based on how much that request has deviated from the ordinary behaviour recorded and so the administrator can make decisions based on this score.

2 RELATED WORK

In this section we present related work of the research.

2.1 Algorithms Used for Anomaly Detection

Falcao et al. [8] collect a comparison of anomaly detection algorithms for intruders detection. The result of their experiments indicates that the algorithms that best distinguish abnormal behaviours are *Isolation Forest (IF)* [14] and *One-Class SVM* [20]. The next best algorithms classified in this work are *Connectivity-based Outlier Factor* and *Outlier detection using Indegree Number*. Detection of anomalies in security contexts related to user behaviour has been previously explored through the use of *Isolation Forest*. This algorithm is appropriated to this context for several reasons. First, they do not require examples of anomalies in their dataset, which is an advantage when training on a new scenario. On the other hand, *One-Class SVM* is very sensitive to anomalies, so the training dataset should be free of them, but we cannot ensure this in our case without a previous analysis adapted to the domain [2] [9]. Sun et al. [22] propose the application of IF on a set of logs from a real company, building a baseline of behaviour for each user. The variables they take into account are:

- *Match rule*. Predefined system rules used for authentication.
- *Signature check*. Indicates whether the device signature matches your ID.
- *Device check*. Indicates whether the device is associated with the user.
- *Browser*. The browser from which the user connects.
- *User connection time*. Includes month, day, year, hour, minute and second.

They indicate that *IF* were initially designed for categorical data datasets, but its implementation of IF includes parsing categorical data to quantitative data by mapping it in a random order. They achieve 98.92% recall and 50.77% accuracy. However, they conclude that the system does not distinguish between the behaviours of different users because the extracted categorical data has very few variations between them. Bülow et al. [7] also use *IF* over the open CERT[1] dataset,

[1]https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099 last accessed on May 10, 2021

specifically about user login and logout data, that classifies data into threats or non-threats. The variables they take into account are:

- Day of the week
- Month
- Hour
- User
- PC
- Activity

To parse this data, they use Kaleko's proposal, which provides a method for rendering cyclic variables such as those relating to date. Meanwhile, categorical variables are parsed with Label Encoding. However, the encoding of these variables itself introduces an unwanted property in this case, as it happened to Sun et al. [22]. This is an inherent order based on the mapped IDs that do not actually exist and has a negative impact on their model. They detect 76% of threats and 7% of normal data classified as such erroneously. In our proposal, we are trying to avoid the addition of domain-meaningless information to data by assigning that order by frequency appearance. Studied systems focus on log on activities while Kassandra analyses all HTTP traffic in a server.

2.2 Isolation Forest and Extended Isolation Forest

IF [14] base their behaviour on the idea that anomalies are "few and different". The algorithm generates trees by randomly partitioning the dataset instances until they are all isolated, or a certain depth of tree is reached. As trees are built randomly, to predict the anomaly value of a new instance a certain number is generated and the average distance from the root to that instance is calculated. Research verifies that anomalies are isolated with fewer partitions and that a big number of trees converge to a unique result.

Figure 1 up shows a set of normally distributed data and the map of anomaly values about that data. It is observed that the colours close to yellow correspond to the area occupied by most of the points, data considered not anomalous. And colours close to dark red are the areas where the abnormal values are located (further separated from the central cluster). *IF* gets good results with or without anomalies in the training dataset. In addition, they are able to handle large amounts of data efficiently, allowing them to be used in real-time applications. However, depending on how point clusters are distributed in space, non-anomalous "ghost" zones may appear as shown in Figure 1 down.

Hariri et al. [11] proposes a modification of the *IF* called *Extended Isolation Forest (EIF)*. While *IF* partitions were performed horizontally or vertically as shown in Figure 2, *EIF* introduces a random slope on the lines that delimit those partitions (Figure 3).

This way, they manage to solve the problem of "ghost" zones that would classify as not anomalous potential dangerous values far from clusters (Figure 4). In Kassandra, we are using *EIF* for anomaly detection models instead of classic *IF* implementation.

2.3 User and Entity Behaviour Analytics

A *User Entity Behaviour Analytics (UEBA)* is an internal threat detection system based on monitoring user behaviour in an organization or network [12]. They are based on the creation of baselines considered to be ordinary or non-abnormal behaviours. The data used to create these baselines can be system logs, applications logs, network device logs... The anomaly values of the new incoming behaviours are calculated from their deviation over these baselines. To be considered a *UEBA*, a system should [12]:

- Be able to use all available data sources.

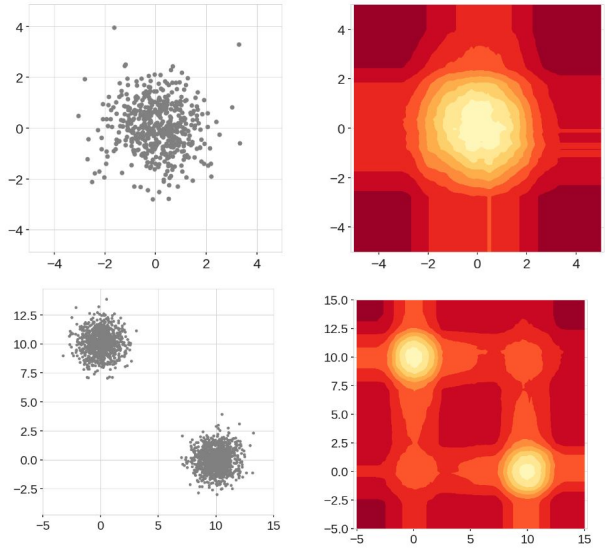


Fig. 1. Dataset normally distributed with one and two clusters and anomaly map score [11]

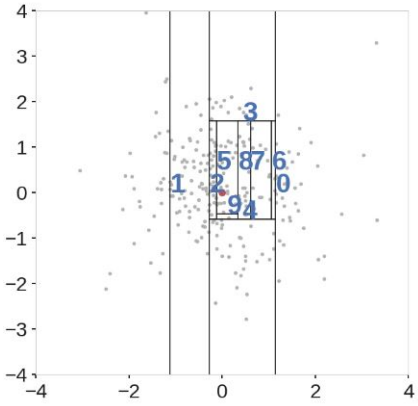


Fig. 2. Partitions done by IF [11]

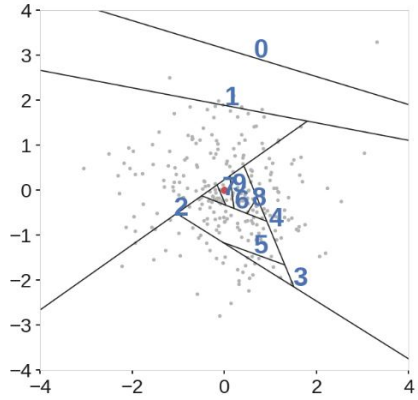


Fig. 3. Partitions done by EIF[11]

- Prioritize critical systems.
- Calculate dangerousness values based on potential deviations from ordinary user behaviour, which are assigned weights according to the appropriate level of restriction for each scenario.
- Be able to detect previously unregistered attacks.

Regarding the design of the systems, there are some common characteristics:

- *Parsed input data*, such as logs, so that it is processed by different algorithms.
- *Security analytics*. It includes machine learning modules, both supervised and unsupervised learning, and event correlation, capable of detecting anomalies from a set of events.
- *Create user profiles* over a certain period of time.
- *Threat management*.

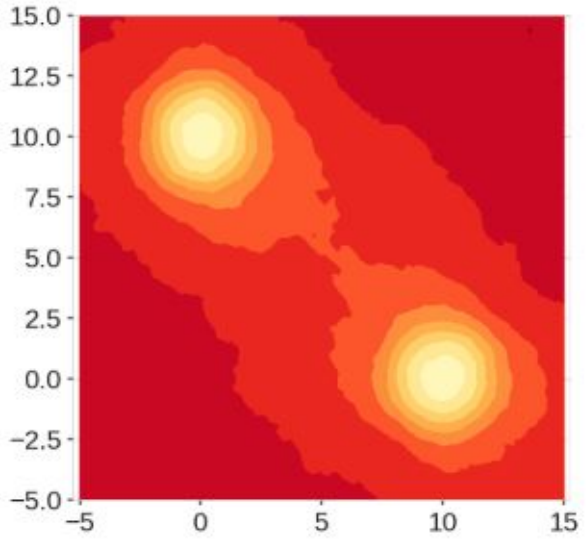


Fig. 4. EIF anomaly map of two normally distributed data clusters[11]

Table 1 shows popular *UEBA* solutions in the market pointed out by Guercio in eSecurity Planet [10] along with its use cases, special features, and the machine learning techniques they use.

Varonis [16] states that *UEBAs* have certain limitations on false positives. There may be legitimate use cases that are considered abnormal and have consequences for the operation of a company. An example is an employee who has to work over a weekend to meet a delivery date. Pricing varies in terms of number of entities monitored, protected machines or used storage per month. This may prevent deploying this kind of security measures in companies with low budget. This also applies to certain educational environments, so future computer engineers could not be trained in this type of security system.

In case of open source *UEBAs*, there are several solutions. *OpenUBA* [17] is a framework for modelling user behaviour. Its main objective is to make accessible to non technical users the reasons of why a behaviour is considered anomalous. Users can also use predefined models or develop their own. They use libraries like *Tensorflow* [3] or *Scikit Learn* [1]. However, as previous literature showed, the Scikit Learn implementation of one of most suitable algorithms to predict anomalies, Isolation Forest, has the problems described before. Khaliq et al. [12] also introduces a list of other open source *UEBAs* like *ELK Stack*, *SIEMonster* or *Alien Vault OSSIM*. Some of them have free license limitations about features or scaling. According to authors, none of them has *UEBA* functionality in its free version.

Educational centres may not be able to afford a private *UEBA* solution. Also, its deployment on a educative and controlled context could be complex or not possible due to lack of computing resources. On the other hand, existent open source solutions may not have the desired features to test *UEBA* functionality, or they may be too complex to explain concepts at certain course levels.

Table 1. UEBA product comparison

UEBA Vendor	Use Cases	Special Features	Machine Learning Techniques
Aruba	High-risk and regulated industries	Integrated network traffic analysis	Supervised and Unsupervised. Detecting attack families (ramsonware).
Cynet	Security operations teams seeking broader app and device management	Integrates access control, application management and endpoint management	Automated remediation capabilities.
Exabeam	Large organizations, federal agencies	Ransomware detection and prevention	Cluster users in groups by behaviour.
Forcepoint	Security operations teams	Consolidated risk scores for individuals; video replays of users' screens	N/A.
Fortinet	Banks, manufacturers and game developers	Monitors endpoints even when off network	Unsupervised learning over user behaviour.
Gurucul	Corporate security operations	Large library of machine learning algorithms; fuzzy logic-based link analysis	Customizable machine learning models.
LogRhythm	High-risk and highly regulated industries	Embedded orchestration, automation and response	Statistical analysis and supervised and unsupervised techniques. Automated risk analysis.
One Identity	Aimed at high-risk privileged accounts	Real-time threat detection, behavioural biometrics	Unsupervised learning. Baseline of normal behaviour.
Palo Alto	Security operations teams seeking broad protections	Automated alert investigation, impact analysis, threat hunting	Unsupervised: groups of behaviour. Supervised: access device classification. Custom rules.
RSA	Security operations teams seeking automation	Unsupervised anomaly detection and machine learning	Unsupervised learning and use cases. No need of analyst tuning.
Securonix	Security operations teams, especially in very large enterprises	Fraud reporting; trade surveillance; patient data analytics	Behaviour patterns.
Splunk	Security operations teams	Multi-dimensional behaviour baseline; anomaly exploration	Custom machine learning models with own SDK (Splunk UBA 4.0 – Machine Learning Software Development Kit).
Varonis	Security operations teams	"Security Time Machine" analyses past data; ransomware detection	N/A.
Veriato Cerebral	Security operations teams and HR departments	Psycholinguistic analysis; screen snapshots; keystroke recording	Groups of behaviours. Low false positives. Natural Language Processing to prevent future inside threats.

3 PROPOSAL

Kassandra acts as a filter for HTTP requests. These requests will have to go through one or two levels of filtering. Analysis is stopped at first level if a request is already considered anomalous to improve performance.

- Level 1 is based on rule-based expert systems [4], a set of conditions which end up telling if the IP, HTTP method and accessed directory are recognized as previously present in training data and hence labelled as normal requests. If Level 1 labels a request as "normal", it is filtered by Level 2.
- Level 2 consist of a set of sub-filters in which one particular characteristic of the HTTP request is analysed. Each subfilter has its own trained machine learning module, which measures the value of dangerousness of a request. These modules use Extended Isolation Forests to predict those values.

After going through this filtering structure, a request receives a value of dangerousness depending on if it is considered an anomaly or not. As users of Kassandra, we can configure the threshold of what we consider a potential threat or not, or even give more importance to some characteristics of the HTTP request that could be more dangerous.

3.1 Dataset

The first prototype of our research analyses HTTP requests, so the data used in training would be obtained in production from the centralized server log files. This will be extended to other protocols in the future. We used data collected from a real company application. The data we collect from the logs are:

- User IP
- HTTP method
- Name of the first directory in the URL (root directory in case a file is accessed in "/")
- File extension (if a file is explicitly accessed)
- Total URL length

We believe that this data may provide information about typical user behaviour in the context of a common business application in a LAN environment, which is the main scenario in which we are going to focus our research. This way, a restricted set of users and IPs will access the resources, so the origin IPs set will typically be restricted. The HTTP methods used are defined by previous requests, in endpoint logs a deviation from routine access can indicate dangerous behaviour. For directory names and file extensions, certain folders may be accessed only at certain times of the day. Finally, malicious parameters that abnormally increase the length of the URL can be used to test for certain vulnerabilities, so their number of characters is also taken into account.

3.2 Data parsing

Extended Isolation Forests are good quantitative data classifiers [22], so we had to transform HTTP requests into points on a plane. Each request is represented as a point on a 2D plane. The focus of our anomaly detection is on what users should not normally do at a certain time of day, so the graphical representation will display this information as a circumference, such as a clock, but with all 24 hours. This representation of periodic characteristics is based on the proposal of the Bülow et al. [7] (see Figure 5). In our system 12 a.m. hours are represented in the point *(2,0)* and 6 a.m. in the *(0,2)*, the hours of the day continue counter-clockwise. Following this circular arrangement, we are able to relate in geometric distance the last hour of a day with the first of the next, which is not achieved with a straight line. We are performing this transformation on each characteristic mentioned on section 3.1 (except user IP). To explain the process, we are using as an example the URL length. First, consider the circumference as the X axis and, on it, one Y axis for every minute (1440 Y-axis). From this, we design a coordinate system that allows us to include this period information. We will call *SC* to the ordinary Cartesian system and *SC'* to the Cartesian system over every minute. Figure 6 depicts the minutes on the X axis and the possible values of a feature on the

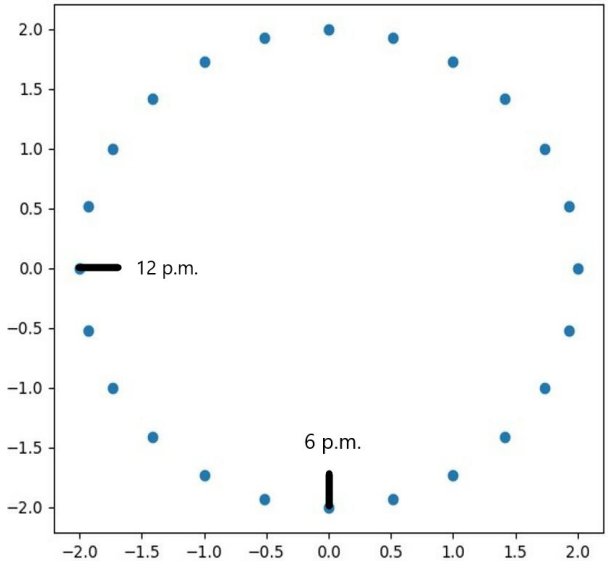


Fig. 5. Representation of 24 hours on a full day

Y axis, for example, the different URL lengths that appear. We are calculating as an example points A and B in Figure 6, which could correspond to the URL lengths of two different requests.

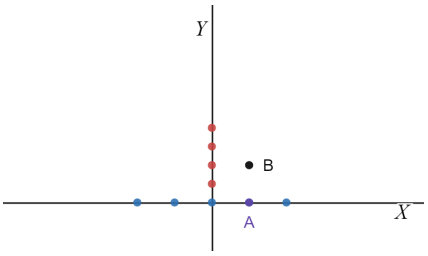


Fig. 6. Representation of the *SC'* system and points *A* and *B*

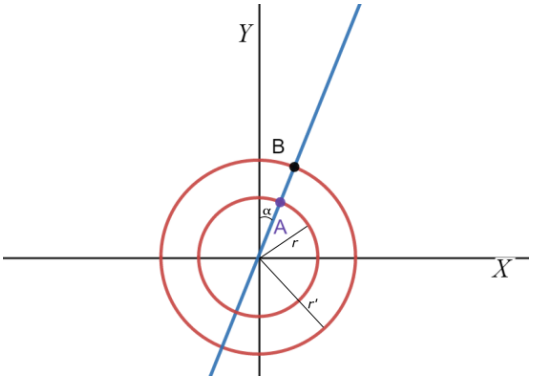


Fig. 7. Representation of the *SC* system and points *A* and *B*

The circumference that passes through A will mark the base radius (in *SC'*, *Y=0*) on which we will calculate the rest of the points. Figure 7 shows points A and B in *SC*. The objective is to translate the points from *SC'* to *SC* so that they can be analysed by the *EIF*.

Points A and B are calculated as follows: The first equation of the following corresponds to the blue line that cuts the circumference, that is, our Y axis in *SC'*.

$$y = tan(\alpha)x \tag{1}$$

The second corresponds to the circumference of radius r.

$$x^2 + y^2 = r \tag{2}$$

We calculate α with the following equation, minute is the value of X in *SC'* (the unit of minute in which the request is received).

$$\alpha = minute\frac{360}{1440} \tag{3}$$

We define a value for r and solve the system of equations to get point A in *SC*. To calculate B, we replace r with r'. Both r and r' are known values that we will define later. To move through the Y axis, we will increase in r-r' the units we want based on the value of the characteristic.

As we said before, we have a "clock" (model) for each feature of the request that we want to analyse. Kassandra checks the characteristics enumerated in section 3.1 excepting the IP.

In each model, each qualitative value is sorted from highest to lowest based on its frequency of occurrence in the training data. It is also assigned a unique identifier (in that order) to be represented graphically. So, the diagram will consist of a set of points arranged on concentric circumferences where those closest to the centre (and with more grouped points) will be made up of the most common values (see Figure 8, this output will be explained in section 3.3.2). This way, a meaningless order is no longer entered into the data as Sun et al. [22] and Bülow et al. [7] commented.

For example, if the following URLs are found in the dataset:

- */login* with 25 appearances
- */home* with 40 appearances

For analysis of the accessed directory, they will be sorted and assigned an identifier as follows:

- */home* with ID: 0
- */login* with ID: 1

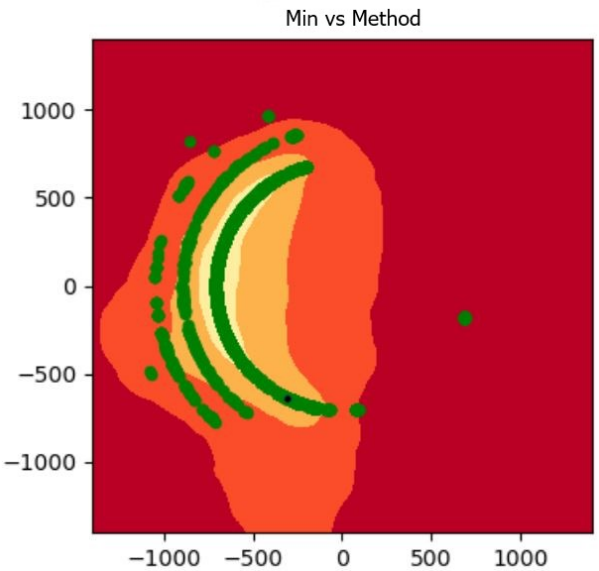


Fig. 8. Example of output of EIF

As we mentioned, in the coordinates of *SC'*, X is the minute in which the URL */home* is accessed and Y corresponds to the ID. For each characteristic we define r *(Y =0)* and the offset between each unit (offset = r'- r) (Table 2). The value of r and offsets have been defined by experimentation by choosing those with which *EIF* gets the best results, i.e. the anomalies are better differentiated.

Table 2. *r* and offsets for each characteristic

Feature	r	offset
Name of the first directory in the URL	500000	300000
HTTP method	500000	300000
File extension	500000	300000
URL length	500000	100000

3.3 Structure

Kassandra has two levels of filtering and its structure is showed in Figure 9. A request must pass through the path marked with a dashed line and, if it is still considered a normal request by the time it reaches stage 1, it continues to Level 2. This way, if the request is detected as an anomaly at Level 1, Level 2 is not executed, improving performance

3.3.1 Level 1. Kassandra Level 1 constitutes the set of rules that a request must fulfill in order not to be anomalous at stage 1. Level 1 also has two sub-levels: *Single* and *Combined* (see Figure 9). In the *Single* sub-level, Kassandra checks if the following parts of the request are present in the training dataset, and gives a partial danger value from 0% to 75%:

- IP
- HTTP method
- Accessed directory

In the *Combined* sub-level, Kassandra checks if the following combinations are included in any records of the training dataset, i.e. those pairs or trios appear together. This sub-level returns a partial dangerousness value from 0% to 25%.

- IP and HTTP method
- IP and directory
- HTTP method and directory
- IP, HTTP method and directory

The dangerousness value for the request at stage 1 shall be the sum of the weighted values of the *Single* and *Combined* sub-levels. By default the values presented on Table 3 are used for weighting. 75% of the weight is evenly distributed to *Single* sub-level characteristics (values not in training are the most dangerous) and the rest is distributed to *Combined* sub-level.

3.3.2 Level 2. In Level 2, Kassandra deepens the analysis of the requests that go past Level 1. It consists of a set of sub-filters where each one has a module with a trained *EIF* that will determine whether the value of a certain feature of the request is anomalous or not at the time of day the request is produced. These are the characteristics we refer to:

- Name of the first directory of the URL (root in case of accessing a file in "/")
- Name of the root directory of the URL
- Extension of the file being accessed (if accessing a file explicitly)
- Length of the URL

Extended Isolation Forest

For the implementation of the *EIF* modules we used a modified version of the library proposed by Hariri et al. [11]. We added concurrency both in the training and in the calculation of the anomaly

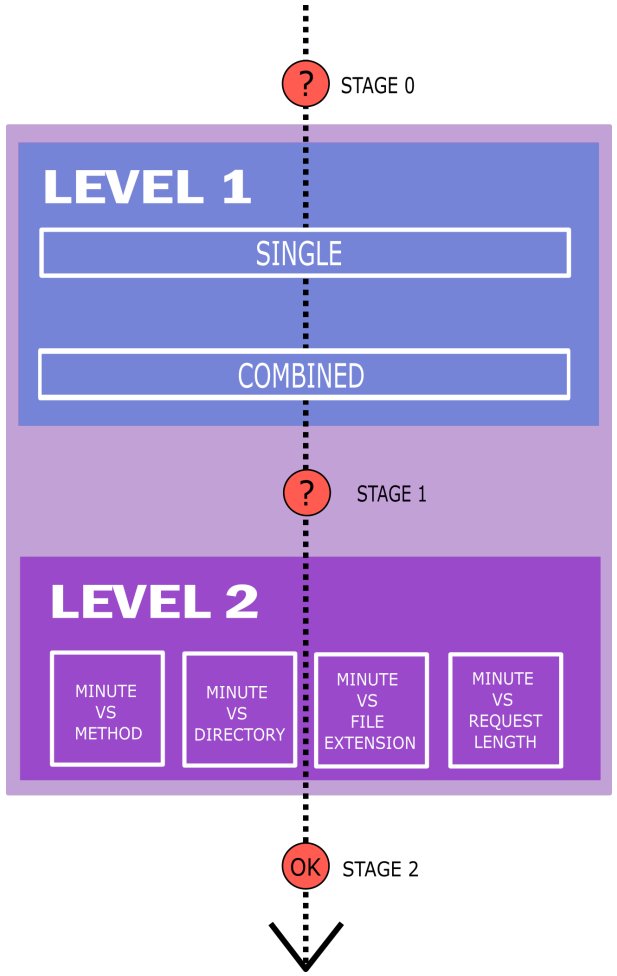


Fig. 9. Kassandra structure

value[2]. This parallelization is performed in the generation of trees, as they are independent of each other, and in the calculation of the distance to the instance to be predicted from the root of each tree. The number of threads is configurable, which allows the performance of the modules to be adjusted to the specifications of the machine where it is executed.

Modules

The implemented modules are also depicted in Figure 9 and are described as follows:

- *Module 1.* Minute vs Method: in this module, Kassandra decides whether it is anomalous for a given HTTP method to be present in a request at a given time of day. It returns a danger value from 0 to 1, with 1 being a major anomaly.
- *Module 2.* Minute vs Directory: in this module, Kassandra looks at the first directory in the request URL and decides whether or not it is normal for this directory to be accessed at that time. It returns a danger value from 0 to 1, with 1 being a major anomaly.

[2]https://github.com/albact7/eif last accessed on May 17, 2021

Table 3. Weights for Level 1

Sub-level	Feature	Weighting value
Single	IP	0,25
	HTTP method	0,25
	URL	0,25
Combined	IP and HTTP method	0,0625
	IP and directory	0,0625
	HTTP method and directory	0,0625
	IP, HTTP method and directory	0,0625

Table 4. Weights for Level 2

Module	Weighting value
Minute vs Method	0,25
Minute vs Directory	0,25
Minuto vs File extension	0,25
Minuto vs Length	0,25
Dangerouss extra value	0,2

- *Module 3.* Minute vs file extension: in this module, Kassandra decides whether accessing a file with a given extension at a given time is abnormal or not. It returns a danger value from 0 to 1, with 1 being a major anomaly.
- *Module 4.* Minute vs request length: in this module, Kassandra considers the number of characters that make up the URL of the request. It returns a danger value from 0 to 1, with 1 being a major anomaly.

As we said, each module returns a value from 0 to 1. Experimentally we observed that normal values usually stay in the range [0.3,0.5] and anomalous values in the range [0.5,0.8]. To obtain a final anomaly value, we weight the results for modules 1, 2, 3 and 4 based on the weights defined in the configuration file, so the total anomaly value will be the weighted average of them. But a very anomalous value should weigh more, so we are adding 20% more importance to that value in case this happens (we consider an anomaly value bigger than 65%). By default the values presented on Table 4 are used for weighting.

After that, according to Table 5, the full anomaly value of the request receives a label to give the user feedback on how dangerous it is. The user can use this information to decide whether to listen to the alert, or not depending on the restriction policy he follows for that scenario.

Table 5. Dangerousness labels and corresponding values

Dangerousness labels						
Range	0.0-0.5	0.5-0.55	0.55-0.60	0.60-0.65	0.65-0.70	0.70-1.0
Label	0	1	2	3	4	5

3.4 Designing policies for anomaly detection

Each case is different and each Kassandra administrator may have different needs. In order to adapt Kassandra to a specific dataset for maximum effectiveness in anomaly detection, there are some parameters that the user can change and experiment until the best result is obtained. In addition, it is also useful to study the danger of each feature depending on the requirements of the scenario.

EIF parameters

Changing the number of trees when generating an *EIF* and the size of the training data subset for each tree in the *EIF* can help to adjust the predictions to a given training data set and adjust the accuracy. Figure 8 shows an example of a module image generated after anomaly detection. The green dots represent the data present in the training. The light yellow areas correspond to the values considered normal (most frequent). The closer it is to the dark red colour, the less frequently those values appear in that time slot and are considered more anomalous. So we can test the model until the prediction is sufficiently correct and the coloured areas fit our data.

Weights of dangerousness value calculation

Another possible setting is at the threshold of what we consider an anomaly or not. Each partial anomaly value for both levels has a weight in the total result of the danger value. This weight is configurable if we want to give more importance to, for example, a strange IP or an HTTP method that should not be there. As we said before, the danger value for Level 1 must be 0% to pass to Level 2, and this pre-filter avoids extra computation consumption when a request is clearly dangerous (it does not pass Level 1). In Level 2 there is an extra weight so that those dangerousness values above a certain threshold are more important for the final result. The configuration of the weights allows to adapt Kassandra to the required scenario and to obtain the corresponding adapted results.

3.4.1 Infrastructure integration. Figure 10 shows the integration of Kassandra with the server infrastructure to be protected on a realistic and production environment. For testing purposes, we used a reverse proxy. It is configured between the user and the server that redirects the logs of the requests to Kassandra. Kassandra has for this purpose a UDP server that listens and when it receives a request, it starts to analyse it through the filtering levels. It generates as a result a report of the anomaly values for Level 1 and, if it continues for Level 2, another summary of these values together with an image representing the analysed request over the set of training values.

Figure 8 shows an *EIF* representation. Kassandra generates one for each module involved in Level 2. The black dot corresponds to the location of the analysed request. We can see that the green dots (HTTP requests in the training data) are clustered between 7 a.m. and 6 p.m., the working hours range of the company. We also observe some anomalies between 11 p.m. and 12 a.m., as we said in section 3.1, the presence of anomalies in the training dataset does not influence the *EIF* predictions.

4 METHODOLOGY

This section will describe the methodology followed for the evaluation of this work. For this purpose, a series of use cases have been designed to understand the functioning of a *UEBA*. The

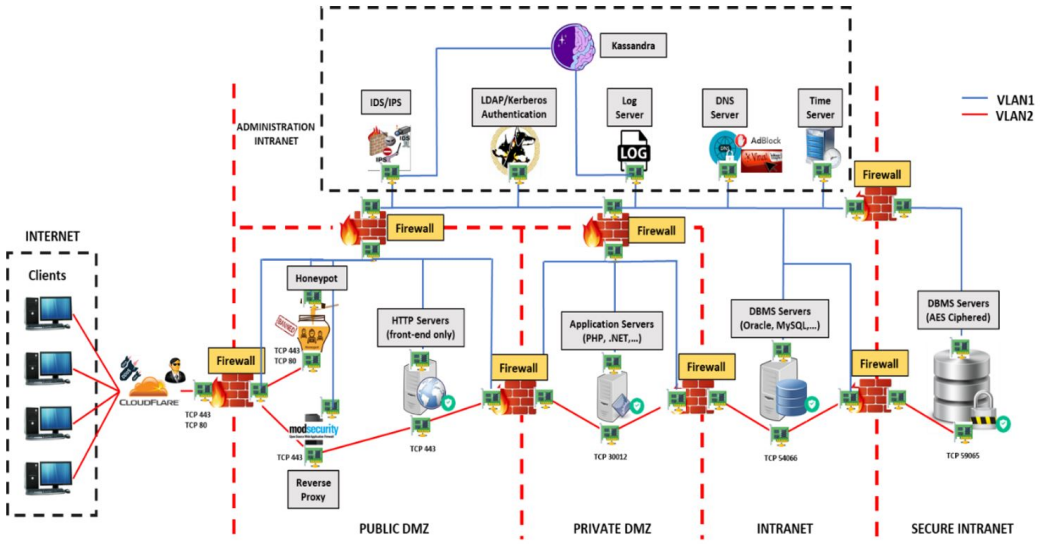


Fig. 10. Integration of Kassandra

experiments have been performed on a Windows 10 Pro machine, 32GB RAM, AMD Ryzen 7 3700X 8-Core Processor 3.6 GHz. We configured *EIF* modified library with 16 threads (see section 3.3.2).

4.1 Use cases

A series of typical use cases have been designed to test Kassandra. These model scenarios that a *UEBA* should cover in a business application context analyzing HTTP requests from a server. Additionally, these try to model base cases that, in an academic context, a student should know in order to understand how *UEBAs* work. The cases that represent an anomaly are then described. The designed cases are described on Table 6, with an example of a potential real attack.

4.2 What do we consider an anomaly

Kassandra provides the user with an anomaly score and a danger label based on the deviation of the incoming request from the baseline constructed from the training data. The user must take into account the scenario in which Kassandra is operating and interpret the result accordingly. The labels provide information about this possible interpretation. Furthermore, if the request reaches Level 2, the user can visualise the distance between the analysed request and the rest. Then, they have enough information about the request to consider it dangerous (and do the necessary protection actions) or not. This depends on the danger threshold the scenario can tolerate.

5 EVALUATION

This section describes the results for the use cases described in the previous section. For each use case, a table is shown with the total anomaly score obtained at each level, and the time taken to perform each analysis. For Level 1, it is shown whether that feature or combination is present in the training dataset and its corresponding anomaly score in the weighting. For Level 2, the anomaly score obtained after analysing the request with each module is indicated in the same way. At the end, the dangerousness label assigned to the request is also shown. For the work presented here, the datasets used in training and testing were obtained from a log service from an application in

Table 6. Use cases

Nivel	Sub-level	Case number	Description	Potential attacks to be detected
Level 1	Single	1	An IP that belongs to the network but has never accessed the server so far (it is not in the training data) accesses the resources.	Network intrusion.
		2	A HTTP method never used before is used in a request to the server	Web server testing: hacking tool.
		3	A directory never accessed before is accessed (or attempted to be accessed).	Directory enumeration [21].
	Combined	4	An IP (a user) that never sends POST requests sends one.	Insider attack or impersonation.
		5	An IP (a user) accesses a directory it has never accessed before.	Insider attack or impersonation.
		6	A request applies a POST method on a directory to which only GET requests arrive.	Web server testing: hacking tool.
Level 2		7	*Minute vs method.* A request is received with a method that is not usually registered at that time.	Insider attack or impersonation.
		8	*Minute vs directory.* A request is received with a directory that is not usually accessed at that time.	Insider attack or impersonation.
		9	*Minute vs file extension.* A request is received that accesses a file with a strange file extension at that time.	Scanning tools performing enumeration (common paths or known files).
		10	*Minute vs request length.* A request is received with a URL whose length is not registered or is not registered at that time.	URL injection [13].
Weight configuration		11	Example of configuration of weights for the adaptation of Kassandra to a scenario in which a team of workers in a company works with an application during their working hours, the application handles critical information and its database cannot be altered.	If any of previous attacks is particularly dangerous due to the characteristics of the scenario.

a particular domain of a company. The office hours are in a range from 7 a.m., to 6 p.m. We are assuming that this schedule is the same for every day of the year.

5.1 Level 1

Table 7 shows the Level 1 results for the use cases proposed in the previous section. *S* corresponds to *Single*, *C* to *Combined*, *I* to *IP*, *M* to *Method* and *D* to *Directory*. The range of dangerousness value for this level is [0-100]. In the tests in this section, Level 2 filtering is not executed because

Table 7. Level 1 results for cases 1 to 6

		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
S	I	0.25	0	0	0	0	0
	M	0	0.25	0	0	0	0
	D	0	0	0.25	0	0	0
C	I & M	0.0625	0.0625	0	0.0625	0	0
	M & D	0	0	0.0625	0	0	0.0625
	I & D	0.0625	0.0625	0.0625	0	0.0625	0
	I & M & D	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625
Total		**43.75**	**43.75**	**43.75**	**12.5**	**12.5**	**12.5**

the danger value in Level 1 is greater than 0. According to the weights we have assigned, if one of the characteristics in the *Single* sub-filter is not present in the training set, which means that the rule is not fulfilled, it will be more dangerous than a not-fulfilled *Combined* rule. We observe that those requests that do not comply with a *Single* rule have a higher danger value. After receiving the results, the administrator can block the strange IP or method, delete or hide that never accessed directory or locate the IP which has broken a rule and to whom it is assigned.

5.2 Level 2

The request has to obtain an anomaly value of 0 at Level 1 to go to Level 2, so the results for Level 1 are the same for the following cases. These results correspond to the Table 8. Level 2 takes into

Table 8. Level 1 results for cases 7 to 11

		Cases 7-11
S	I	0
	M	0
	D	0
C	IP & M	0
	M & D	0
	I & D	0
	I & M & D	0
Total		**0**

Table 9. Level 2 results for cases 7 to 11

	Case 7	Case 8	Case 9	Case 10	Case 11
M	0.6061	0.4860	0.4718	0.4297	0.6094
D	0.5043	0.6529	0.4459	0.4257	0.5079
F	None	None	0.5095	None	None
L	0.4236	0.5918	0.4847	0.6501	0.4227
F.V.	**0.5114**	**0.5997**	**0.4780**	**0.5463**	**0.5582**
Lb	**1**	**2**	**0**	**1**	**2**

account the time at which the request is received. In our business context, the working hours of

the employees fall within the range of 7 a.m. to 6 p.m. In the tables with the Level 2 results, the modules and their (unweighted) anomaly values are shown. In addition, it also shows the total anomaly value taking into account the calculation of all weights and the dangerousness label. The value obtained from the modules is the one calculated by the *EIF*, as we said in section 3.3.2 values lower than 0.5 do not constitute an anomaly while the anomalous values are in the range [0.5 - 0.8]. Together with the table with the numerical values, the corresponding graph is shown. A map with outliers is generated for each module that applies to the request, however, we will only show the map of the module to be studied in each use case. In Table 9, *M* corresponds to *Minute vs Method*, *D* to *Minute vs Directory*, *F* to *Minute vs File extension*, *L* to *Minute vs Length*, *F.V.* to full anomaly value and *Lb* to the danger label.

5.2.1 Case 7. Due to the distribution of work, network users usually send POST requests between 8 a.m. and 5 p.m. because it is when they upload data to the database. However, such a request is received at 6 p.m. This is an anomaly and is reflected in the value obtained in the *Minute vs Method module* (0.6). In the map (Figure 11) we see the request (black dot) on the orange colour, which indicates that it is a time zone with high anomaly values for this POST method. If, on the other hand, it were a GET method, it would be in the inner green circle on the dark yellow area, indicating a smaller anomaly value. A possible response of the system administrator would be identify the user (IP) who has uploaded data in an abnormal time range and discover why.

5.2.2 Case 8. In this case, we observe that both the *Min vs Length* and *Min vs Directory* modules show anomalous values for the analysed request. In the case of *Min vs Directory*, this is due to the fact that workers usually log in via the URL */login* from 7 a.m. to 3 p.m. But this request is received at 4pm. On the other hand, *Min vs Length* also points to an anomaly as there are no more requests with a URL of the same length at that time in the training data (Figure 12). If the accessed URL is */login*, this means that all employees log on the system in that time range and administrator should check why there is someone logging on at a strange hour because a person who is not currently working could have been impersonated by someone who ignores the timetable of the company.

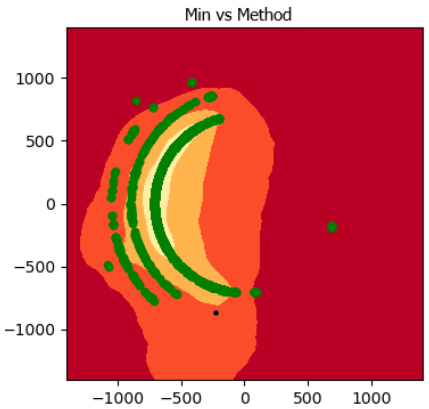


Fig. 11. Case 7. Minute vs Method

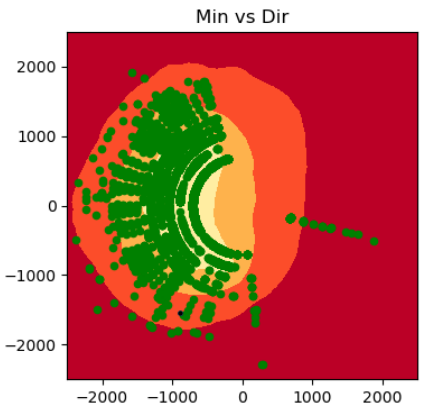


Fig. 12. Case 8. Minute vs Directory

5.2.3 Case 9. A request to a file with the extension *.txt* is received at 2 p.m., however, workers or processes access the *.txt* log files at specific times, but do not usually do so from 12 a.m. to 4 p.m. The anomaly value for this module is 0,5. This is not a very dangerous anomaly despite the fact that there are no requests to *.txt* in the training data at these times (Figure 13). This is because, in

general, at these hours the server is very busy (as seen in the green dots around our request) and because it is a recurrent extension on a daily basis (it is in the second green circle from the centre, i.e. it is the second most common extension). If there is an automatic process which analyses log files, someone could have modified the configuration of its execution and administrator should check why it has been executed at that time.

5.2.4 Case 10. During the working day, URL requests of similar length are often sent. They tend to access the same resources or files that follow a certain naming code, so they share the length of the URL. In the same working day, a request with a 48-character URL is logged. We observe that the *Min vs Length* module returns an anomalous value. This is because there are no other URLs with the same length in the training data. Although the URL length is a quantitative value, the data consumed by the *EIF* is modelled according to the frequency order explained above. The representation of the data (Figure 14) in the form of concentric circles leads to the fact that the data located near the centre (because they are closer, since they are distributed on the Y-axes radially) have a lower anomaly value. The frequency representation prevents, for example, URLs of short length, but infrequent, from being considered ordinary. An URL injection could have been detected, so the administrator should check if the URL is dangerous and who is assigned to the origin IP.

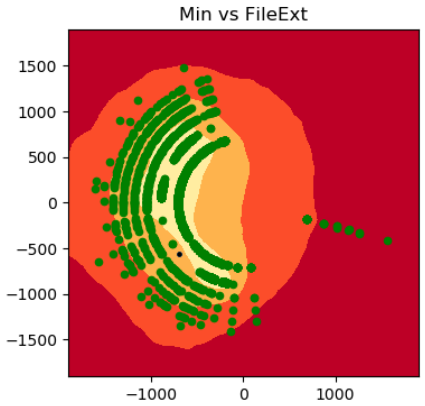


Fig. 13. Case 9. Minute vs File Extension

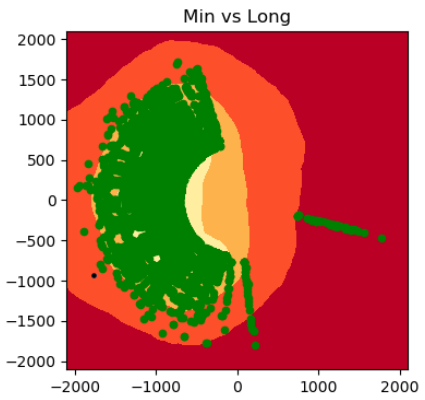


Fig. 14. Case 10. Minute vs Length

5.2.5 Case 11. We can see that we obtain similar results to case 7 (Figure 12) but with a higher anomaly value due to the increased weight of the module with the most anomalous value. The image generated also corresponds to that of case 7. Due to the scenario described, most requests sent to the server are GET to query information. However, although to a lesser extent, they also send POST requests, but in a more restrictive time range: from 7 a.m. to 4 p.m. The company application accepts several endpoints that normally only include GET methods. POST requests appear in the training dataset, but less than the previous method, and in a time range from 7 a.m. to 4 p.m. approximately. A POST request enters the server at 5 p.m. and the new weights increase the total anomaly value, raising the danger label to 2. The value of the label is encouraging the administrator to not overlook the alert because request constitutes a critical anomaly.

6 CONCLUSIONS AND FUTURE WORK

Kassandra is able to detect user/entity behaviour anomalies in the interaction with HTTP servers in the designed use cases, modelled after typical attack scenarios. Then, system administrators can

analyse to which degree a request is dangerous. To improve analysis performance, it implements two filtering phases. A rule-based Level 1 filtering ensures that only these requests requiring deeper analysis pass to Level 2, rejecting the rest. It can be configured to give more importance to certain request characteristics if the scenario in which it is deployed requires it. Kassandra offers the behaviour of UEBA systems adapted to low computational volume and educational environments.

Support for other protocols will be implemented in the future. The current prototype does not currently differentiate users, but in the future this model could be extended to create per-user monitoring. Other new characteristic to incorporate to the model is to consider if certain days correspond to weekends or known holidays, as the behaviour of users may significantly change on these occasions. A complementary functionality to generate a report to study logs (training dataset) in order to detect irregularities and identify past anomalies can also be developed. This tool is also planned to be incorporated as part of the teaching materials of the Computer Security course of the School of Computer Engineering of the University of Oviedo [19]. The source code of Kassandra is available at https://github.com/Egida-Kassandra/kassandra.

7 CITATIONS AND BIBLIOGRAPHIES

REFERENCES

[1] Scikit Learn. Last accessed on May 26, 2021.

[2] Scikit Learn, IsolationForest. Last accessed on May 10, 2021.

[3] TensorFlow. Last accessed on May 26, 2021.

[4] Ajith Abraham. Rule-Based Expert Systems. In *Handbook of Measuring System Design*. John Wiley & Sons, Ltd, jul 2005.

[5] Asmaa Shaker Ashoor and Sharad Gore. Difference between Intrusion Detection System (IDS) and Intrusion Prevention System (IPS). In *Communications in Computer and Information Science*, volume 196 CCIS, pages 497–501. Springer, Berlin, Heidelberg, 2011.

[6] J. Stuart Broderick. ISMS, security standards and security regulations. *Information Security Technical Report*, 11(1):26–31, jan 2006.

[7] Joakim Bülow and Maja Scherman. Insider Threat detection using Isolation Forest. Technical report, 2018.

[8] Filipe Falcão, Anderson Santos, Tommaso Zoppi, Baldoino Fonseca, Andrea Bondavalli, Caio Barbosa Viera Silva, and Andrea Ceccarelli. Quantitative comparison of unsupervised anomaly detection algorithms for intrusion detection. In *Proceedings of the ACM Symposium on Applied Computing*, volume Part F147772, pages 318–327, New York, NY, USA, apr 2019. Association for Computing Machinery.

[9] Markus Goldstein and Seiichi Uchida. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *PLOS ONE*, 11(4):e0152173, apr 2016.

[10] Kyle Guercio. Best User and Entity Behavior Analytics (UEBA) Tools, 2020.

[11] Sahand Hariri, Matias Carrasco, and Robert J Brunner. Extended Isolation Forest. Technical report.

[12] Salman Khaliq, Zain Ul Abideen Tariq, and Ammar Masood. Role of User and Entity Behavior Analytics in Detecting Insider Attacks. In *1st Annual International Conference on Cyber Warfare and Security, ICCWS 2020 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., oct 2020.

[13] Sandeep Kumar, Renuka Mahajan, Naresh Kumar, and Sunil Kumar Khatri. A study on web application security and detecting security vulnerabilities. In *2017 6th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2017*, volume 2018-January, pages 451–455. Institute of Electrical and Electronics Engineers Inc., apr 2018.

[14] Fei Tony Liu, Kai Ming Ting, and Zhi Hua Zhou. Isolation forest. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 413–422, 2008.

[15] Fernando Montenegro. Expanding Machine Learning Applications on the Endpoint. Technical Report September, 2018.

[16] Jeff Petters. What is UEBA? Complete Guide to User and Entity Behavior Analytics | Varonis, 2020.

[17] Jovonni L. Pharr. OpenUBA: A SIEM-agnostic, Open Spurce Framework for Modeling User Behavior, 2020. Last accessed on May 10, 2021.

[18] Ponemon. Cost of Insider Threats: Global Report 2020. Technical report, 2021.

[19] Jose Manuel Redondo. Improving concept learning through specialized digital fanzines. pages 134–143, 2021.

[20] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[21] Caleb Sima. Start Secure. Stay Secure.™ Are your web applications vulnerable? Technical report, 2005.
[22] Li Sun, Steven Versteeg, Serdar Bozta¸s Bozta¸s, and Asha Rao. Detecting Anomalous User Behavior Using an Extended Isolation Forest Algorithm: An Enterprise Case Study. Technical report.