KADIR HAS UNIVERSITY
SCHOOL OF GRADUATE STUDIES
DEPARTMENT OF ADMINISTRATIVE SCIENCES

FEDERATED ANOMALY DETECTION FOR LOG-BASED DEFENSE SYSTEMS

UĞUR ÜNAL

DOCTOR OF PHILOSOPHY THESIS

ISTANBUL, APRIL, 2022

FEDERATED ANOMALY DETECTION FOR LOG-BASED DEFENSE SYSTEMS

UĞUR ÜNAL

A thesis submitted to
the School of Graduate Studies of Kadir Has University
in partial fulfilment of the requirements for the degree of
Doctor of Philosophy in
MANAGEMENT INFORMATION SYSTEMS

İstanbul, April, 2022

APPROVAL

This thesis titled FEDERATED ANOMALY DETECTION FOR LOG-BASED DEFENSE SYSTEMS submitted by UĞUR ÜNAL, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in MANAGEMENT INFORMATION SYSTEMS is approved by

Professor Hasan Dağ (Advisor) .....
Kadir Has University

Associate Professor Mehmet Nafiz Aydın .....
Kadir Has University

Assistant Professor Emrullah Fatih Yetkin .....
Kadir Has University

Professor Mustafa Bağrıyanık .....
Istanbul Technical University

Associate Professor Ahmet Cüneyd Tantuğ .....
Istanbul Technical University

I confirm that the signatures above belong to the aforementioned faculty members.

.....
Prof. Dr. Mehmet Timur Aydemir
Director of School of Graduate Studies
Date of Approval: 12/04/2022

DECLARATION ON RESEARCH ETHICS AND PUBLISHING METHODS

I, UĞUR ÜNAL; hereby declare

- that this Ph.D. Thesis that I have submitted is entirely my own work and I have cited and referenced all material and results that are not my own in accordance with the rules;
- that this Ph.D. Thesis does not contain any material from any research submitted or accepted to obtain a degree or diploma at another educational institution;
- and that I commit and undertake to follow the "Kadir Has University Academic Codes and Conduct" prepared in accordance with the "Higher Education Council Codes of Conduct".

In addition, I acknowledge that any claim of irregularity that may arise in relation to this work will result in a disciplinary action in accordance with university legislation.

UĞUR ÜNAL

.....

12/04/2022

To my family

ACKNOWLEDGEMENT

"*Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less.*"

— Marie Curie

First and foremost I am extremely thankful to my supervisor, Prof. Hasan Dağ for his invaluable advice, continuous support, and patience during my PhD study. I am also grateful to him for providing me an opportunity to follow my own research ideas. I would like to thank all the members in the Center for Cybersecurity & Critical Infrastructure Protection. It is their kind support that have made my study and life in the Kadir Has University a joyful time.

Last, but surely not least, I would like to express my infinite gratitude to my parents and my wife. With their tremendous empowerment and encouragement in the past few years, I was able to overcome the challenges, and keep my determination in completion of my study.

FEDERATED ANOMALY DETECTION FOR LOG-BASED DEFENSE SYSTEMS

ABSTRACT

The adaptation of Industry 4.0 and IoT creates a vast network which opens up various new vulnerabilities to systems. Increasing number of cyber attacks becomes more sophisticated which impedes functionality of enterprises and critical infrastructures. Malfunctioning of the services of these systems can cause catastrophic results considering wealth and well-being of a society. Organizations need an intelligent defense system which is adaptable to newer threats to create rapid solutions. Anomaly detection is widely adopted protection step and is significant for ensuring a system security. Logs, which are accepted sources universally, are utilized in debugging, system health monitoring, user authorization and access control systems and intrusion detection systems. Recent developments in Deep Learning (DL) and Natural Language Processing (NLP) show that contextual information decreases false-positives yield in detection of anomalous behaviors.

Additionally, decentralization and exponentially increased number of data sources make traditional machine learning algorithms impractical. Federated Learning (FL) brings a solution to overcome decentralization and privacy issues. It aims to employ participating devices to learn from own data and sending local models for global convergence over secure communication. FL provides data security and decreases communication cost greatly, since local data is not transported to a central server. In a volatile cyber domain, it is a necessity to take a quick precautions for potential threats. The benefits of FL ensure building a defense system which provides real-time detection of cyber attacks.

In this thesis, we propose a novel anomaly detection model and risk-adaptive federated approach. First, AnomalyAdapters (AAs) which is an extensible multi-anomaly task detection model. It uses pretrained transformers' variant to encode log se-

quences and utilizes adapters to learn a log structure and anomaly types. Adapter-based approach collects contextual information, eliminates information loss in learning, and learns anomaly detection tasks from different log sources without overuse of parameters. Moreover, evaluation of this work elucidates the decision making process of the proposed model on different log datasets to emphasize extraction of threat data via explainability experiments. Lastly, Risk-adaptive anomaly detection with federated learning (FedRA) which is based on the idea of Spreading Phenomena. It decentralizes the aforementioned detection approach and adapts weighting of shared parameters to ensure capturing incoming cyber attacks in a timely manner.

Keywords: system logs, natural language processing, federated learning, anomaly detection

LOG TABANLI SAVUNMA SİSTEMLERİ İÇİN FEDERE OLAGANDIŞILIK TESPİTİ

ÖZET

Endüstri 4.0 devrimi ve Nesnelerin İnterneti (IoT) teknolojilerinin ortaya çıkışı sistemlerin büyük ağlara yayılmasını ve yeni tehditlere daha savunmasız duruma getirmiştir. Hızla artan çok yönlü siber saldırılar kurumların servislerini ve kritik altyapılı sistemlerin işlevlerini engellemektedir. Bu sistemlerin arızalanması ya da durması, toplumun varlığı ve sağlığı açısından büyük tehlikelere yol açabilmektedir. Ortaya çıkan saldırılara adapte olması ve hızlı çözümler üretebilmesi için bu kuruluşların akıllı savunma sistemlerine ihtiyacı vardır. Olağandışılık tespiti sistemlerin güvenliğinin sağlaması açısından önemli bir savunma metodudur. Sistem günlükleri evrensel veri kaynakları olup, siber tehditlerin analizleri için en çok kullanılan ve gerçek zamanlı izlenebilen yardımcı veri kaynaklarıdır. Hata ayıklama, sistem sağlını izleme, kullanıcı yetki ve erişim kontrol sistemleri ve saldırı tespit sistemleri, sistem günlüklerinin analiz aracı olarak kullanıldığı örneklerdir. Derin Öğrenme ve Doğal Dil İşleme alanlarındaki gelişmelerle birlikte bağlamsal bilgilerin kullanımı hatalı tespitlerin oranını düşürdüğü görülmektedir.

Ek olarak, merkezi olmayan sistemlerin gelişimi ve hızla artan veri miktarı geleneksel makine öğrenme metotlarının işlevini kısıtlamaktadır. Federe Makine Öğrenmesi (FMÖ) ya da Federe Öğrenme (FÖ) dağıtık sistemler için bu sorunlara ve veri gizliliğine çözüm getirmektedir. FÖ, merkezi olmayan büyük verilerin eğitilmesini sağlayan dağıtık bir makine öğrenmesi yöntemidir. Bu yöntem geleneksel yaklaşımın ötesinde eğitilmiş modellerin parametrelerinin ortaklaşa kullanılmasını sağlar. Merkezi hesaplama ünitesi toplanan parametreleri birleşik hale getirip, sistem içerisindeki diğer elemanlara dağıtır. FÖ mimarisi güvenli toplama prensiplerinin kullanılmasını sağlayarak eğitim sırasında veri gizliliğini arttırmaktadır. Böylece veri aktarım hızını, güvenliği ve gizliği konularında yaşanabilecek olan sorunlara çözüm getirmektedir. Hızla gelişen siber dünyada, potansiyel tehlikelere karşı süratle önlem alınması

bir gerekliliktir. FÖ mimarisinin getirileri sistemlerde bunu sağlamaktadır.

Bu tezin sonucunda, yenilikçi olağandışılık tespiti ve önem tabanlı federe öğrenme yöntemleri önerilmiştir. İlk olarak geliştirilebilir çoklu anomali tespit modeli ,AnomalyAdapters (AAs), önerilmiştir. Bu model önceden eğitimli bir 'transformer' varyantını kullanarak sıralı sistem günlüklerini anlamladırır ve adaptörler aracılığıyla ise, ortaya çıkan aykırılıkları tespit eder. Adaptör tabanlı öğrenme bağlamsal bilgilerin toplanmasını, öğrenme sırasındaki bilgi kaybının engellenmesi ve gerekli olmayan parametrelerin kullanılmamasını sağlar. Bununla birlikte, önerilen model açıklanabilirlik esas alınarak tehdit verilerinin çıkarımı test edilmiştir. Son olarak, Yayılma Olayı'ndan etkilenerek riske bağlı adapte olabilen federe öğrenme (FedRA) sunulmuştur. İlk olarak elde edilen tespit modelini dağıtık öğrenme yapısına geçirip, paylaşılan parametrelerin skor ayarını yapmaktadır. Böylece ortaya çıkan siber saldırılara gerçek zamanlı uyum sağlayabilmektedir.

Anahtar Sözcükler: sistem günlükleri, log, doğal dil işleme, federe makine öğrenmesi, olağandışılık tespiti

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\Phi \vert \psi$	acquired parameters after model training
D_{KL}	Kullback–Leibler divergence
h	hidden size of the model
d	adapter's dimension after activation function
r	residual value from transformer block's feed forward layer
b	transformer block
w	trained model weights
η	learning rate
S	selected clients each round
β	transmission unit time
μ	recovery time
τ^{SIS}	characteristic time of the node to spread in SIS epidemic model
d	degree of node in graph

LIST OF ACRONYMS AND ABBREVIATIONS

AAs	AnomalyAdapters
AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
BPE	Byte-Pair Encoding
CNN	Convolutional Neural Network
CSA	Cyber Situation Awareness
CTI	Cyber Threat Intelligence
DoS	Denial of Service
DL	Deep Learning
DNN	Deep Neural Network
DS	Design Science
DSRM	Design Science Research Methodology
DSRP	Design Science Research Process
DSS	Decision Support System
ES	Expert System
FCNN	Fully Convolutional Neural Network
FL	Federated Learning
GRU	Gated Recurrent Units
HDFS	Hadoop Distributed File Systems
ICS-CERT	Industrial Control Systems Cyber Emergency Response Team
IDS	Intrusion Detection Systems
IG	Integrated Gradients
IMS	Information Management System
IoT	Internet of Things
IoC	Indicator of Compromise
IPS	Intrusion Prevention Systems
IR	Input Reduction
IS	Information Systems
ML	Machine Learning

MLM	Masked Language Model
NOC	Network Operation Center
LAA	Log Anomaly Adapter
LLA	Log Language Adapter
LSTM	Long short-term Memory
SCADA	Supervisory Control and Data Acquisition
SI	Susceptible-Infectious epidemic model
SIEM	Security Information and Event Monitoring
SIS	Susceptible-Infectious-Susceptible epidemic model
SG	Smooth Gradients
SGD	Stochastic Gradient Decent
SOC	Security Operation Center
ROBERTa	Robustly Optimized BERT Pretraining Approach
ReLU	Rectified Linear Unit
TTP	Tactics, Techniques, and Procedures

1. Introduction

The emergence of technological innovations brings sophisticated threats. Industry 4.0 and IoT revolutions lead systems to become more decentralized and interconnected (Colombo et al. 2017). Systems that are managing operations, need to become more sophisticated and intelligent to adapt these innovations. On the other hand, these developments expanded the attack surface and designs which make systems more vulnerable (Alves et al. 2014). Cyberattacks are increasing day by day aligned with these innovations and entails the need for rapid solutions for defense mechanisms. These attacks may hinder enterprise operations or more importantly interrupt critical infrastructure systems that are essential to safety, security, and well-being of a society. Securing a system reveals wide range of specific areas which protective solutions need to be prepared for (Jang-Jaccard and Nepal 2014). Security vulnerabilities and cyber environment are crucial topics to consider when taking precautions. Additionally, systems integrity, confidentiality and availability should be a must in designing these solutions (Council et al. 2007).

Large-scale system becomes more distributed and gained focus by IT industry which provides applications to manage daily work, such as; e-commerce, online banking, communication platforms(social networks) (He, Zhu, He, Li, et al. 2017). Downtime of these systems can have a huge negative impact on end users and significant revenue loss (He, Zhu, He, Li, et al. 2017). Kaspersky's annual report 1.1 presents phishing attack distribution on targeted organizations, as an evidence for threat space. In the report, attackers' choice on organization type is mostly global internet portals, but online stores, payment systems and financial services are the other mainly targeted organization types. In addition, critical infrastructures plays an important role on human life quality, such as; water and wastewater, transportation, healthcare and energy systems(nuclear power). Any interruption or faulty behaviour on these

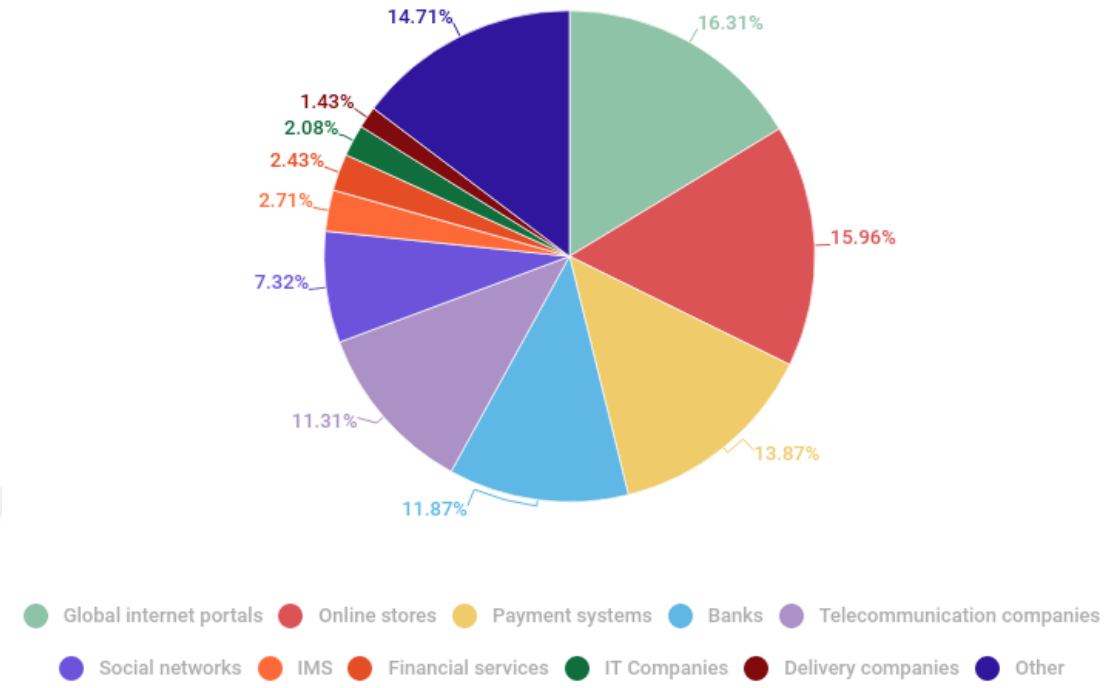

Figure 1.1 Phishing attacks' distribution on various organizations (EU) (Kaspersky 2021).

systems, may result in catastrophic accidents or can cause a crisis. (Miller and Rowe 2012; Denning 2000; Mustard 2005; Nicholson et al. 2012) are well-known incidents caused by cyberattacks which have caused loss of wealth and life on CIs. In addition, ICS-CERT annual reports on found vulnerabilities indicate inclined trend on cyber attacks on various CI sectors (*ICS-CERT Review* 2016), which indicates the need of readiness for various cyber threats.

To prevent upcoming cyberattacks and adapt sophisticated cyber environment, organizations have various system components, such as intrusion detection systems (IDS), intrusion prevention systems (IPS), authorization and access control systems, and security information and event management (SIEM) tools. These components are part of security operations center (SOC) which can be considered as central unit monitoring system activities (Nabil et al. 2017). Several approaches and implementations exist to prevent malicious behaviours in these system components, but anomaly detection techniques have an important role on preventing suspicious

behaviours. One of the most significant problem in observing unusual behaviour of data is anomaly detection. If an information deviates dramatically from usual data behavior in some range, it is classified as an anomaly (Hu et al. 2017). In general, it signifies that an object is distinct from the rest in a group. In other words, *anomaly detection* is a process to distinguish unknown or peculiar events which can be determined by observations (Chandola, Banerjee, and Kumar 2009). There are several difficulties in identifying a normal activity which can be presented in a structured form. The boundary between normal and anomalous behavior is not always clear, the exact anomaly detection varies depending on the field of application, the availability of relevant data for learning, and data can contain noise due to normal behavior is dynamic and constantly evolving (Hu et al. 2017).

In next section, we investigated cyber situation awareness (CSA) via SIEM tool implementations to discover trends and understand the need for applied techniques. SIEM is the envisaged implementation tool for the proposed artefacts in the thesis work. Because, SIEM tool connects to diverse range of devices (or system components) and monitors activities via log events or other calculated and static indicators. Recent advancements on applying security measures are improved with machine learning (ML) and deep learning (DL) applications. Thus, SIEM provides a suitable environment for implementation of these newly designed methodologies. These tools can be designed as enterprise-based, critical infrastructure based, IoT-based or as a plugin component to be included inside system architecture. In the

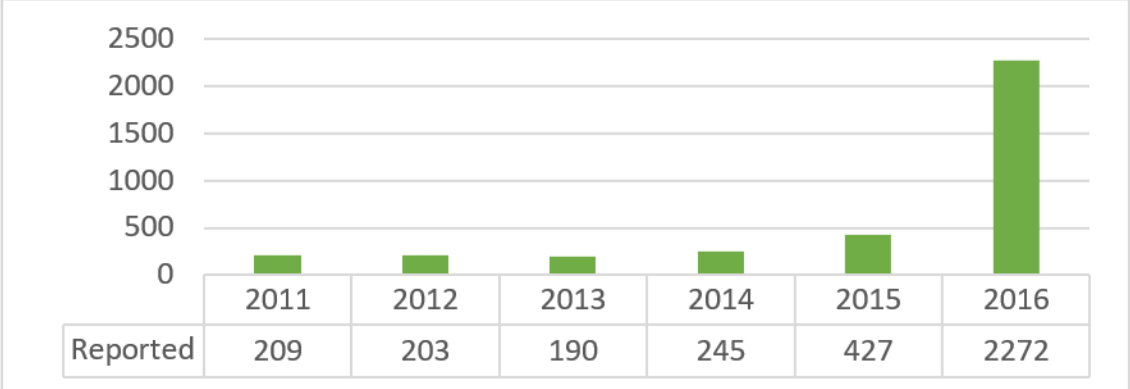

Figure 1.2 ICS-CERT Annual Vulnerability Coordination Report (*ICS-CERT Review* 2016).

context of situation awareness, SIEM tool's implementation analysis allow us to capture the need for readiness in cyber domain. In addition, utilized threat artefacts reveal the need for the type of data considering detection methods.

1.1 Investigation of Cyber Situation Awareness

Awareness, in the sense of security, builds the backbone of operations understanding the current and future cyber activities. Situation awareness has become the focal point of securing systems due to dynamic nature of cyber domain. Technological advancements cause the volatility to transform into upcoming challenges. Understanding those is the key to keep CSA progression. Earlier studies define required steps to administer situation awareness. These steps (perceive, comprehend, project, and resolve) are also adapted to cyber domain. Rapid technological changes redefine the content of those and thus, it creates demands improving automated tools, which play as systematic factor in nurturing situation awareness. As a system factor, SIEM tools can be basis for comprehending cyber domain. SIEM tool's enhancement is useful to evaluate current state and help predict upcoming challenges for maintaining awareness.

Cyber Situation Awareness (CSA) aims to create *known(s)* for circumstances which threaten health of a system (Chismon and Ruks 2015). Onwubiko depicts situation awareness as a reaction to changes in a system and handing over an appropriate answer (Onwubiko 2016). In this perspective, situation awareness is a requirement for organizations to be prepared for awaiting cyber threats. CSA is considered as an application of awareness (Onwubiko 2016). Reference model for situation awareness is originally defined in Endsly's work (Endsley and Garland 2000) and redesigned in Onwubiko's work (Onwubiko 2012). Continual challenges in cyber domain pushes the reference model to adapt and specialize in the affect of circumstances.

Industry 4.0 and IoT related technological advancements enlarge the threat space. Centralized systems become more decentralized which causes them to become vul-

nerable to newer threats. These threats affect awareness of security analyst negatively and obstruct decision making processes. Additionally, unresolved artefacts impede mitigation of impacts and become much bigger problem for an organization. To overcome this issue, organizations need to adapt recent developments and intelligent techniques. It is also crucial to analyze these techniques applied to the updated models within CSA. SIEM tools are chief security systems of an organization and thus, they can be taken to assess awareness in cyber domain.

1.1.1 Cyber Situation Awareness

Situation awareness mostly defined as collecting inputs from a system which informs surroundings to act upon (Onwubiko 2016). This topic studied and defined in many ways which still vague definition overall (Lif, Granåsen, and Sommestad 2017). As stated on awareness reference model (Fig. 1.3), operators and human cognitive abilities are involved in the process (Onwubiko 2016). Technological developments convey human abilities inefficient and increase the need for automation. Human cognition works in harmony with current system environment and evolve duly. Furthermore, Situation awareness in cyber domain refers distinct type of preparedness integrated with digital space (Ahmad et al. 2021). A report presented in Forbes indicates that Security and Network Operation Centers (SOC/NOC) are rapidly adapting emerging technologies such as DL and ML based analytical applications (Ehrlicher 2020). It allows ease in defending more complicated cyber threats. By this way, enhancing automation on these centers is becoming a necessity to provide secure systems. In this section, we draw attention on digital environment of a system. In other words, we investigate systemic factors of CSA to uncover aimed approaches for thesis work.

1.1.2 SIEM Tools

Security Information and Event Management (SIEM) tools supply comprehensive view of a system and can be taken as basis for CSA. The usage of a SIEM tool

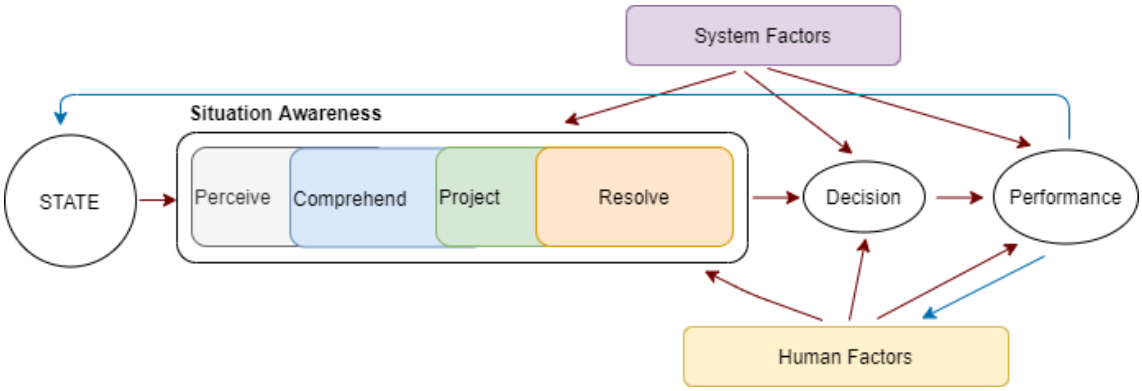

Figure 1.3 Situation awareness reference model (Onwubiko 2016).

has a long history in IT infrastructure. It collects event data from security devices and formulates to a standard form. Moreover, the collected data is analyzed or correlated. As a final step, these tools communicate with an alert system to inform security analyst (Fig. 1.4). Thus, it yields automatic insights corresponding to current cyber domain. It is also building block to SOCs of an organization (Nabil et al. 2017). SIEM plays a role in *perceive* and *comprehend* steps with collecting data and correlation. In addition, it attends in *project* and *resolve* steps with alert management and analyst's decision making process. By its nature these tools need to evolve with current trends and complex cyber threats. According to Gartner's report, over %70 of SIEM tools used by organizations are adjusted with data centric technologies such as artificial intelligence (AI) and ML (Sadowski, Bussa, and Kavanagh 2020).

SIEM tools can be categorized by two (2) main criteria, as thoroughly investigated in Nabil's work (Nabil et al. 2017). Functional features are to measure analytical foundations of the tool. Moreover, it is related to correlation process as seen in Fig. 1.4. The algorithm used in defining pattern and relationships within data is proprietary design choices for an organization. Second feature is technical, which is related to the implementation choices of the tool. By this means, preparing efficient SIEM is related to vendor, ease of deployment, and evolution of the tool (Nabil et al. 2017). In this section, we use latter feature to indicate architectural improvements.

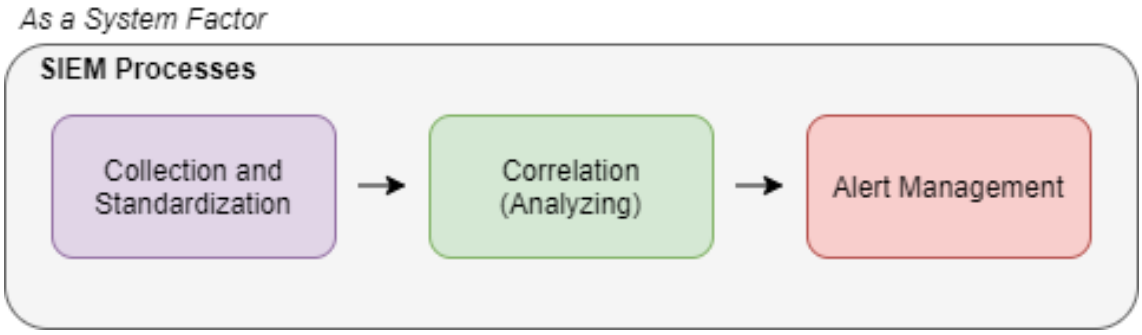

Figure 1.4 SIEM processing steps (Nabil et al. 2017).

1.1.3 Cyber Threat Intelligence

Defining risk factors is essential to ensure system security. Furthermore, understanding these factors is possible with gaining holistic view of vulnerabilities, threats, and impacts (Sekharan and Kandasamy 2017). Cyber Threat Intelligence (CTI) is built by a data-centric series of processes. Data needs to be collected within set of rules to form knowledge and analyzed to present actionable information. There are many works how to define actionable information on decision making (Zhu and Dumitras 2018; Tundis, Ruppert, and Mühlhäuser 2020; Dalziel 2014), but we keep the focus on intelligent artefacts that are used as indicators on the improvement process of SIEM tools.

There are four sub-levels of CTI. These categories defined considering intended audience and its application (Chismon and Ruks 2015). Strategic Intelligence is presented for high level decision makers. They encompass financial impact of threat space. Tactical Intelligence relates to Tactics, Techniques, and Procedures (TTPs), which provides overall information about threats, attackers tactics, and execution procedures. Strategic and Tactical Intelligence are consist of long-term decisions. Operational Intelligence provides thoroughly detailed information about awaiting threats. Technical Intelligence is specific type of data for limiting upcoming threats. Those are also known as Indicator of Compromise (IoC) such as; malicious IP addresses. In SANS's report, organizations prefer technical intelligence with % 41, operational intelligence with %27, tactical intelligence with %18 and strategic intelligence with %13 in cyber threat defense (Brown and Lee 2019).

On the point of accordance with SIEM tool, it provides security analytic solutions by providing the ability to detect incidents occurring in corporate networks in a secure way. These systems involve investigating evidence-based data after a potential threat alert has been received. The detection process is achieved with a large amount of data. In the context of data, new sources can also be mentioned in any field where human observation is mentioned (Mokalled et al. 2019). At this point, it would be convenient to call analyzed information "intelligence" because of its context via information; a piece of analyzed information has become actionable as intelligence. With the analysis environment created by SIEM, information that is based on the observations of security analysts emerges. Steps can be taken to make the SIEM tool more useful for security analysts and even minimize the possibility of analysts making mistakes during decision-making. One of the many actions that can be taken in this regard is to use CTI artefacts (Vielberth, Menges, and Pernul 2019). In gaining more efficient outputs from SIEM, getting benefits from CTI for such targets is a proper step. CTI enables SIEM to perform better by blending internal network alerts with as much context as desired. Thus, a more proactive posture is performed for the resilience of the organization.

Fig. 1.5 provides a sample illustration of how CTI is consumed by SIEM. According to the flow in that figure, data from databases, operating systems, applications, user related data under the enterprise are turned into intelligence through CTI. Thus all the data accumulated in SIEM will be correlated in this way. As an output, analysis reports reaches the security analyst. With the contribution of CTI and the correlation of SIEM, the decision making process is completed with minimum error.

1.1.4 Comparison of studies

In this section, SIEM tools, which have been proposed within recent studies, are examined. We use several indicators to complete understanding of the current state of CSA. Investigating systematic blocks as SIEM tools and its approaches foster decision-making and provides exhaustive insights. In addition, it enhances the qual-

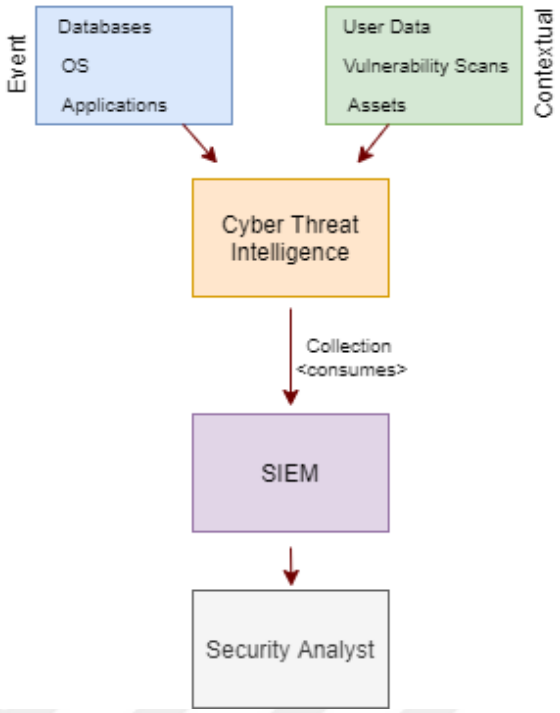

Figure 1.5 CTI role in SIEM tools.

ity of decision-making process in cyber domain. Endsly stresses that becoming aware of possibilities in threat space builds basis decision-making (Endsley and Garland 2000). By this means, we are also creating a basis for research questions and clearly indicate relationship between SIEM tools and CTI.

The rest of the section is organized by implementation area; Plug-in based tools, enterprise based tools, Supervisory Control and Data Acquisition (SCADA; which is control systems in critical infrastructures) based tools, and IoT based tools. The studies focusing only on improvements of SIEM tool are classified as Plug-in. It also means that proposed improvements can be integrated into any generic SIEM tool. We present the examined studies with indicators in Table 1.1.

Plug-in based tools. Diverse source of intelligence offers elaborate ways to approach security concerns. Lah proposes a method to detect the lateral movement attack by analyzing risk scores (Lah, Dziyauddin, and Azmi 2018). User risk scores provide an evaluation of user activities by determined risk factors. This work combines user risk scoring with packet action, transmission time, and frequency for at-

tack discovery. Lateral movement is a persistent attack type that takes long time to discover due to investigation package movement, contents, and destination. Several SIEM vendors develop plugin modules for specifically understanding and analyzing user behaviours. The proposed framework as an extension to SIEM tool that clusters user behaviours via risk scores, then tries to detect lateral movement attacks with aforementioned packet behaviours. By this way, it aims reducing false positive rate in detection.

Bryant and Saiedian's work, called LogRhythm, is based on the difficulty for analysts to find the root-cause of alarms (Bryant and Saiedian 2017). Data aggregation from many sources offers several problems, including handling huge amounts of data from different sensors or trying make sense of unorganized, incompatible data. These challenges have huge effects on identifying the breaches or take corrective action on detect/prevent the attacks from the analyst perspective. The response of security teams to these incidents may be inadequate, since the incidents are usually irregular or consist of complex information. In the study, changes were made on the kill chain models to facilitate the collection of data between different sensors. In this way, it tries to solve the problem of alarms not having enough detail, which is actually one of the main problems in security alerts. It helps the security analysts to make efficient investigation workflows. LogRhythm provides effective way in determining the relationship between alarms and possible threats.

On the other hand, Moukafih's work proposed simple neural networks as weak learners and created high detection capabilities with few computer resources (Moukafih, Orhanou, and El Hajji 2020). In this study, high detection rate and small submodels were combined with the aid of model ensemble technique. Calculations were made on different scenarios for improve the overall accuracy. While making these calculations, weak models were used and then a classification was made according to the nature of the event. Additionally, the paper contributes the investigation of some techniques on how to improve the general accuracy along with the models developed using weak models. The detection model is developed using practices

in machine learning. As an extension to IDS, the analysis method can be easily integrated into SIEM tools.

Enterprise based tools. Burgot's work presents a methodology to process packet captures (Burgot et al. 2020). This approach is a combination of graph theory, data science and cybersecurity. The network logs are trained through an autoencoders to discover patterns of the abnormal behaviours from the logs. Spark, Tensorflow, ElasticSearch, Kibana and Linkurious are used for proposed system architecture. The approach in the research was tested on the CICIDS2017 dataset and was able to detect 11 out of 15 attacks (DoS, DDoS, Port Scans, Web Attacks, etc). It also provided a synthesis of insights into these attacks through explainability indicators for each alert using clustering. This study aims to detect attacks on an IT system based on netflow data with multidisciplinary concept. This provides a transparent view to security analysts on the investigation of anomalies.

Security team needs to transmit data in a secure way. In case, data is exposed to manipulation or disclosure by an attacker locally or remotely, there can be high impacts. By adopting this motivation, In Eswaran's study, the scope of the activities carried out in the network were determined (Eswaran, Srinivasan, and Honnavalli 2021). They focused on details they thought that the verification of those would be faster. They tried to analyze the event data which is initiated by the host and then try to determine if the access attempts are successful or not. They proposed experimental test-bed and collected logs with the *sysmon* and then calculated the command length (cmdlen) of logs. They analyze the malicious activities and examine the higher cmdlen because of malicious activities usually have higher cmdlen. Overall, the algorithm checks GUID (globally unique identifier) of an event with Windows Threat Index. If no result is achieved, then zero-day with controlling cmdlen starts. If an anomaly is detected in the system such as malware, its hash values is calculated and then the event is tagged as threat to create an alarm. Their experimental analysis done by the Splunk SIEM.

Muthuraj's work focuses on securing Active Directory Domain Services (Muthuraj et al. 2020). This work utilizes Splunk SIEM tool and adds functional improvement. SIEM enables to extract information about attacks targeting domain services and how to prevent these attacks. Different types of attack chains have been determined according to user logon types. All the analysis has been carried out by focusing on a specific attack scenario, and steps have been created by considering the attacker mindset in the attack steps. According to the attack scenarios, if the attack is successful, measures are taken against these scenarios. After these scenarios, it is evaluated whether the attacks could be detected with windows logs. It is thought that enterprises can become more resilient to attacks that may occur in the active directory with the proposed measures against attacks.

Moreover, Lee focuses on a DL-based methods for advanced cyber threat detection (Jonghoon Lee et al. 2019). Many security events are collected and transformed into individual event profiles. The proposed SIEM tool is based on AI using various methods such as; Fully Convolutional Neural Network (FCNN), Convolutional Neural Network (CNN), and Long short-term Memory (LSTM). Compared to traditional machine learning approaches, this methodology can better categorize genuine alerts, which means that the number of notifications sent to security analysts might be reduced substantially. It aims to convert security events into individual event profiles for large-scale processing data. This system helps security analysts to respond quickly to cyber threats.

In Sornalakshmi's study, a SIEM tool has been proposed to detect the DoS attack that occurs frequently by monitoring and managing the logs in the server (Sornalakshmi 2017). Considering which alarms may occur during a DoS attack; Different rule combinations have been determined to generate an alarm. A solution is also provided for the detection of changes that may occur in the system due to malicious activities. To prevent the determined attack, the requests received by the client will be recorded in the access log and stored in the log files. The logs will be checked according to the rules written, and when there is a match between these

two, an alarm will be generated. The determination of the written rules has changed according to the attack types of the targeted attacks. The generated alarms will be recorded for later checks. The proposed tool improvements aims to reduce the possible false-negative alarm ratio in enterprises.

Moreover, Mulyadi, presents a framework which focuses on the containerized version of Elasticsearch, Logstash, and Kibana stack (ELK) which is supporting various features and lightweight (Mulyadi et al. 2020). Docker technology also reduces snapshot size and startup time. The motivation is to adopt creating a regular alarm system in a container environment and detect threats by writing customized rules. Two rule sets are used in the developed system and created rules aim to provide explicit security to the local environment. To prove the efficiency of their work, they compared two(2) types of systems which are the standalone and the containerized application. The efficiency of use was measured based on Elasticsearch latency and server/host utilization. Standalone version is not helpful for the huge amount of data and detecting anomalies. Dockerized version provides real-time search and also scalability features. Additionally, Wazuh plugin is used as second security agent in proposed framework.

SCADA based tools. There are numerous challenges for critical infrastructures considering cybersecurity domain and integration of SIEM tools on SCADA systems become crucial. Singh focuses on providing a unique framework for dealing with insufficient alert information in order to create an effective log management system (Singh, Callupe, and Govindarasu 2019). They used the kill-chain concept, which may be used by an advanced persistent attacker (APA) to conduct cyber-attacks on the power system. Simply, cyber-kill chain model presents attackers' procedures and steps to deploy a cyber attack. The main motivation of this study is versatility of threats on the SCADA systems. Log management has significant role for securing critical infrastructure. In this way log management tools provide secure ways to monitor and manage a network. Security onion (SecOn) open source tool is used as a security management and intrusion detection. For reporting, Kibana is used as

visualization tool.

Other than SIEM tools' integration in SCADA, adaption of current trends to SIEM tools also shows promising results in cyber physical environment. Hindy discusses on this topic and proposes a ML-based anomaly detection methods (Hindy et al. 2018). This work aims to analyze sensor data gathered from Programmable Logic Controllers (PLC) in water system. Six different (Logistic Regression, Gaussian Naïve Bayes, k-Nearest Neighbours, Support Vector Machine, Decision Trees, and Random Forests) ML models are used in their experiments. There are four experiments considering output of ML models accordance to measure security analyst's decision. These experiments are investigated as binary classification and multi-label classification problem. In addition, single/two threat scenario suggestion is presented by probabilities after classification and multiple threat scenario suggestion is provided over a confidence level. By this way, this work provides a way to understand mitigation of multiple processes in case of cyber incidents.

IoT based tools. IoT integration makes systems more distributed which also increasing the risk of cyber attacks on various devices. Moreover, vast communication network with multiple devices creates huge amount of data. Andrés Pardo presents a blockchain based tool which ensures security of events by its nature (Mármol 2019). Blockchain technology is a digital ledger that allows cryptocurrencies to securely move between digital wallets (Mas'ud et al. 2021). It is important that security events are captured as a whole, as false alarms may occur in events as an example of data manipulation. Blockchain eases cooperation with multiple devices and provides protective authorization with Smart Contracts. This work proposes different elements, such as IoT sentinels and SIEM miners. Sentinels are guarding elements for IoT devices and responsible for processing events into transactions. Miners combines transactions into blocks and detect abnormal transaction by analyzing blocks. With this motivation, a blockchain-based system called βSIEM-IoT has been presented as security management tool for IoT environment. In this way, events from different sources can be linked and distributed attacks can be detected. On the

other hand, this methodology allows to investigate internal and external intelligence respect to transaction in blockchain environment.

Vasilyev and Shamsutdinov focus on a challenge in Multi-Agent Systems which is to find a common way to represent distributed knowledge among agents (Vasilyev and Shamsutdinov 2020). The proposed SIEM tool was developed by statistical methods and merged with Artificial Immune System (AIS) for IDS. Also, the system has agents to provide data manipulation processes such as data collection via sniffing the network, storing that data, and transmitting them to IDS agents. All data on network attacks was a black box on the AIS side, as it was trained based on normal network activity data. Then, it is also trained to classify known attacks on wireless sensors. This approach makes it possible not only to detect anomalies but also to predict them if possible. The resulting report is displayed in the admin console with visualizations. This method's foremost opportunity is; it has become possible for analysts to evaluate the system's performance under conditions where not all events are known.

Furthermore, Hwoij introduces an architecture to overcome needs of smart cities. A SIEM tool has been proposed to protect and securely process data from IoT devices (Hwoij, Khamaiseh, and Ababneh 2021). The study focused on trying to improve security concerns for IoT devices and addressed the problems that any smart city may experience. The prominent contribution in this study is the integration of SIEM tool with the IoT environment. The proposed architecture consists of 3 main parts; smart environment, SIEM, and SOC. Collecting data, parsing and filtering are handled by Splunk SIEM tool. These main sections are distributed to cover the entire area of a smart city in a geographic approach. This structure makes it possible to take action in real-time, able to manage security operations, and reporting against a threat or attack in a smart city.

Table 1.1 Investigation of SIEM tools

	Architecture Improvements	**Enhanced SIEM process**	**Implementation area**	**Utilized CTI artefact**	**Solution**		
					Infrastructure	Programmatic	Neural
Bryant and Saiedian 2017	Functional	CS, Co and A	Plug-in	Technical	✗	✓	✗
Lah, Dziyauddin, and Azmi 2018	Functional	Co	Plug-in	Tactical	✗	✓	✗
Sornalakshmi 2017	Functional	CS, Co, and A	Enterprise	Technical	✗	✓	✗
Mármol 2019	Technical	CS and Co	IoT	Tactical	✓	✓	✗
Hindy et al. 2018	Technical & Functional	CS and Co	SCADA	Technical	✓	✓	✓
Jonghoon Lee et al. 2019	Technical & Functional	CS, Co and A	Enterprise	Technical Tactical	✓	✓	✓
Singh, Callupe, and Govindarasu 2019	Technical & Functional	CS, Co and A	SCADA	Technical Tactical	✓	✓	✗
Moukafih, Orhanou, and El Hajji 2020	Functional	CS and Co	Plug-in	Technical	✗	✓	✓
Muthuraj et al. 2020	Functional	CS, Co and A	Enterprise	Tactical	✗	✓	✗
Vasilyev and Shamsutdinov 2020	Technical & Functional	CS and Co	IoT	Technical	✓	✓	✓
Mulyadi et al. 2020	Technical	CS, Co and A	Enterprise	Technical	✓	✓	✗
Eswaran, Srinivasan, and Honnavalli 2021	Functional	CS and Co	Enterprise	Technical	✗	✓	✗
Hwoij, Khamaiseh, and Ababneh 2021	Technical	CS and A	IoT*	Technical	✓	✓	✗
Burgot et al. 2020	Technical & Functional	CS, Co and A	Enterprise	Technical	✓	✓	✓

Collecting & Standardization (CS), Correlation (Co), Alert Management (A)

(*) Combination with critical infrastructures

1.1.5 Discussion

This section aims to provide a holistic view of a CSA and investigates systematic building blocks via SIEM tools. The developed solutions discussed under three categories are infrastructure, programmatic, and neural. It is to reveal the purpose and contribution of the studies in the most straightforward way; in other words, we provide a classification of the solutions in which they produced and where they belong inside these categories. Infrastructure section is pointing out whether there is an architectural design choice in the study. The programmatic section addresses if there is a development in the software-related areas. Under the neural section, it is stated whether there are ML/AI based methods. Each study offers new perspectives on the systematic factor of situations awareness. Although reviews do not show certain direction in advancements, it is seen that dynamic nature of cyber domain affects on proposed results. As a basis for CSA, SIEM tools enhancements indicate

that implemented area is crucial for ensuring awareness.

More accurate and effective usage of security-related data allows to carry out risk analysis in a cleaner way. The cooperation between CTI and SIEM tools enables enterprises and critical infrastructures to support secure environment. SIEM tools mostly profit from technical and tactical intelligence artefacts in overall implementations. Proposed solutions also indicate that organizations tend to develop the systematic factors of situation awareness in respect to upward trends such as; ML and AI, Blockchain technologies, and IoT-based infrastructures, besides focusing on specific issues which can be used as an extension of tools.

The section above is a part of a published conference paper.
DOI: 10.1109/UBMK52708.2021.9558964

1.2 Problem Identification and Motivation

In this research, identification of the problem and motivation of proposed methods are two folds:

Monitoring unknown, peculiar and undiscovered patterns in a dataset or in a system network traffic is crucial. Because, this type of evidences are used in system performance analysis, debugging, monitoring health of a system, intrusion or anomaly detection. These examples are also called as an anomaly or outlier. The real time detection of anomalies or unknown events is important to preserve system security. Investigation of vast amount data (log events) obstructs finding these type of patterns. To achieve that, data-centric solutions is a necessity to monitor and identify the data sources. In addition, contextual understanding of system logs can reveal patterns which enables systems to discover suspicious data points. Also, acquired data points expand threat intelligence which can mitigate in an organization, locally or globally.

Rapid developments in deep learning and increasing quality of data collected from

edge devices presents new opportunities for decentralization. In a centralized cloud based application, client requests or updates to central server can block processing due to latency issues. In a critical environment, such as anomaly detection system, latency issues or halting requests can create irreversible damage. Therefore, decentralization of protection systems of critical system components provides durability. Advancements via Industry 4.0 and IoT rises processing power of edge devices which prepares the readiness and establishes the need of enhancing communication cost and data security.

1.3 Research Aim and Questions

This thesis work aims to build anomaly detection model which is based on contextual information of system logs via NLP methods and adapt produced model into decentralized system via federated optimization techniques.

- *RQ1*: The fact that system logs can be related to each other sequentially, contextually and collectively makes it difficult to detect anomalies. Can a new language model based on system logs be developed to solve this problem?
- *RQ2*: Can language models adapt efficiently to discover anomalies?
- *RQ3*: Is it possible to adapt language models in a decentralized system based on importance of a participant to rapidly detect threats in information systems?

2. Research Methodology

Design Science Methodology provides an important paradigm to improve efficiency on conducting applicable research (Peffers et al. 2007). Design Science (DS) paradigm aims to extend capabilities of organization and build new designed artefact, since Information Systems (IS) is an important factor on guiding organizational activities. The important task in DS paradigm is to comprehend problem domain (Hevner et al. 2004). The DS methodology evaluates IT artefacts to solve identified organizational problems (Pfeffers et al. 2006). The DS research provides a link in IS and its practice. IS research outcomes are important in practice and DS enables to show impact of outcome artefact. These artefacts can have various forms, such as software structures or rigorous mathematics (Peffers et al. 2007). Moreover, consistency in IS research and design science research process (DSRP) shows applicability of designed artefact.

Design Science Research Methodology (DSRM) proposes three objectives; consistency with prior literature, carry out research with sequence of nominal processes, and presents mental model to evaluate research output (Peffers et al. 2007). Moreover, DSRM presents conceptual principles of DS research, practical rules, and a nominal process for carrying out and presenting the research (Peffers et al. 2007). By this way, presented IS research has proper template which makes repeatable and explainable research processes (Peffers et al. 2007). There are a lot of options in determining software artefacts. Nominal process helps us drawing a road map to accomplish intended goals of the project. However, it is not the only solution for the research mechanism; it suggests a better way to do DS research. A mental model is a model of reality, which shows characteristics of research outputs. Mental models can be the presentation of a design, a model, or an approach in imaginary situation. The mental model provides a context, that researchers can understand and evalu-

ate the outcome. It aims to give information for other researchers to prepare their researches effectively (Peffers et al. 2007).

As Peffers states, achieving DSRM processes requires development of the methodology (Pfeffers et al. 2006). Consensus building of methodology among other researchers presents DSRM activities how to conduct a research. Additionally, nominal processes are consist of six steps: *problem identification and motivation*, *define the objectives for a solution*, *design and development*, *demonstration*, *evaluation* and *communication* (Pfeffers et al. 2006). However, researcher can select entry points according to aim for the solution of the problem (Peffers et al. 2007).

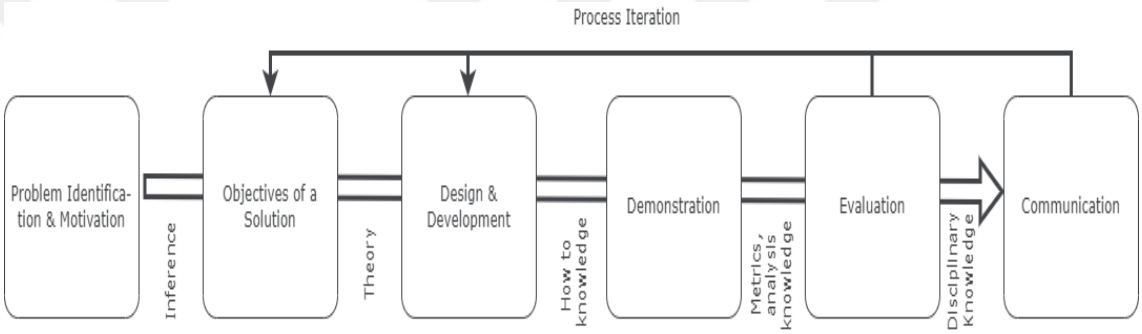

Figure 2.1 Design Science Research Process Model (Peffers et al. 2007).

In this research, we investigate problems in anomaly detection and ensuring security in decentralized systems. Throughout the document we propose two(2) software artefacts which are appropriate for envisaged solutions. First *artefact* identifies problems in log semantic-based anomaly detection models. It proposes an efficient transfer learning method implementation to overcome issues according to prior research. Second *artefact* identifies problems in anomaly detection and security of a system in a federated environment to overcome issues considering communication cost and adaptability to state of the system. Both artefacts address problem-centered approach as an entry point in the DSRP.

Nominal process iteration for *first artefact*: Parameter-efficient Multi-Anomaly Task Detection

- *Problem identification and motivation*: Recent developments in NLP studies

show that contextual information decreases false-positives yield in detecting anomalous behaviors. Transformers and their adaptations to various language understanding tasks exemplify the enhanced ability to extract this information. Deep network based anomaly detection solutions use generally feature-based transfer learning methods. It is unfeasible and a redundant way considering adapting various type of log sources.

- *Define the objectives for a solution*: AnomalyAdapters, log semantic-based anomaly detection model which is based on adapter transfer learning and transformer architecture.
- *Design and development*: Design a ROBERTa-based adapter transfer learning method (AnomalyAdapters) for anomaly detection
- *Demonstration*: Demonstrate trained anomaly detection model on HDFS and Firewall datasets in detection anomalous log event.
- *Evaluation*: Evaluate anomaly detection model using Recall, Precision and F1-score metrics and show explainability of threat data.
- *Communication*: One journal paper is published *(Article DOI: 10.1109/ACCESS.2022.3141161)*

Nominal process iteration for *second artefact*: Risk-Adaptive anomaly detection with federated learning

- *Problem identification and motivation*: Different federated optimization methods applied for anomaly detection. In case of a system security with influence of Spreading Phenomena, susceptible and infected nodes' parameters needed to be weighted. This will allow crucial information to added to global updates faster which allows adapting security measures in timely manner.
- *Define the objectives for a solution*: FedRA, risk-adaptive federated optimization method for anomaly detection. Global model updates are influenced by Spreading Phenomena and configured using SIS epidemic model.
- *Design and development*: Design a federated optimization based on network epidemics.

- *Demonstration*: Demonstrate trained model on HDFS dataset for detecting anomalous log event.
- *Evaluation*: Evaluate trained model by comparing convergence speed with existing federated approach.
- *Communication*: To be submitted.

3. Parameter-efficient Multi-Anomaly Task Detection

System security poses a big step for enterprises, governments, and safety critical systems. Adaptation of Industry 4.0 and IoT concepts open up more vulnerabilities, because the systems become more interconnected. In large-scale systems misidentifying an action can obstruct operations and negatively affect the maintenance of their services. Monitoring and analyzing threats is crucial as the state of technology grows rapidly. The more complex a system becomes, the harder it is to detect threats' behavior. Thus, scalable and flexible security solutions are required for an organization (Panetta 2021). Anomaly detection systems are a part of Intrusion Detection or Prevention Systems (IDS/IPS), which are connected to different sources. A common practice is to use rule-based applications with the help of system administrators that are responsible for investigating events based on the threat intelligence. These types of approaches tend to fail, due to joined sources in a system yielding excessive data. Identifying anomalous behavior differs in sources and also is challenging considering streaming data in an online setting. Therefore, detecting anomalous events accurately and timely is crucial (Nedelkoski et al. 2020). Log is accepted as an universal indicator of events for debugging and analysis purposes. They are designed to deliver information about an action and its related variables of a system. System logs are the main source of monitoring cyber incidents in real-time (Lin et al. 2016). Continuous expansion of configurations of logs with each update to a system complicates sustaining the stability of defense mechanism.

Anomaly detection is the process of revealing undefined and abnormal actions in the system according to movements that are usually detrimental, predefined, or determined by an observation (Chandola, Banerjee, and Kumar 2009). This is a data-driven technique for investigating unexpected behaviors (Lin et al. 2016). A log is a unstructured text that is designed for debugging and monitoring. It is

stored in a text form for readability and convenience. Creating logs for readability produces an excessive number of instances and increases the difficulty of automation (He, Zhu, He, Li, et al. 2016). Moreover, it makes detecting anomalies harder with a combination of many sources (Yuan et al. 2010).

Log mining, parsing, and anomaly detection techniques must evolve to capture a decisive intelligence. Anomaly detection studies can be divided into two categories: log key-based and semantic-based, according to how they use log data. As key-based methods, earlier works focused on static indicators or kill-chain analysis methods utilizing logs such as; PCA (W. Xu et al. 2009), invariant mining (Lou et al. 2010), and workflow monitoring (Yu et al. 2016). DeepLog (Du et al. 2017) approaches logs as an unstructured text and adopts text processing techniques to extract log templates(or keys) with a parsing tool (Du and Li 2016). It uses LSTM model to predict next log keys via learning the current normal log event sequence from antecedent events. Furthermore, advances in deep networks started to lead anomaly detection studies. More recent studies oriented toward NLP techniques are able to extract contextual information. These are also called log semantic-based methods. LogAnomaly (Meng et al. 2019) and LogRobust (Zhang et al. 2019) both utilize semantic information of log sequences with combination of their templates. Transformer architecture brings promising results in various domains' problems and tasks, especially in text data (Vaswani et al. 2017). Thus, it suitable to be experimented in anomaly detection studies. HitAnomaly indicates the instability of log parsing tools and combines semantic information with log's parameter values (Huang et al. 2020). Another recent work, Logsy removes the need of log parsing tools to prevent information loss in yielding templates and uses transformer model with a multi-head attention mechanism (Nedelkoski et al. 2020). To achieve that, these semantic-based works utilize a pretrained embedding to transfer knowledge into anomaly detection task. Transfer learning methods are not signified between anomaly task domains; however the method of implementation can improve tasks in the existing environment. To that extent, we believe that anomaly detection studies based on log data can be improved via semantic information, which is enabled by

transformer architecture. Besides, it can be optimized and adapted for applications in which multiple models need to be trained for anomaly tasks in an online setting.

We approach anomaly detection as a data-driven application and propose a task-based anomaly detection method considering a central system that manages multiple sources. To achieve that, we utilize an adapter-based learning in the detection model. Adapters were first introduced as transfer learning method for detecting visual representation (Rebuffi, Bilen, and Vedaldi 2017), and later introduced for language processing for transformers (Houlsby et al. 2019). Additionally, as discussed in related works (Chandola, Banerjee, and Kumar 2009; Chalapathy and Chawla 2019), we study the types of anomalies in three categories: *point*, *conditional*, and *collective*. We are motivated by the advantages and versatility of transformer-based language models and propose a model for host-based anomaly detection systems. Considering each log as a sentence and system-calls as a language; our aim is to gain semantic information through adapters to distinguish anomalies. Using the nature of language models, we aim to use a multi-purpose approach, which is expandable to new sources without loss of information and overuse of parameters.

3.1 Background and Related Work

Anomaly detection is the activity to distinguish unmatched, peculiar, or unknown examples from the data (Chandola, Banerjee, and Kumar 2009). This type of detection techniques are used in different applications such as; fraud detection in finance, intrusion detection in cyber security, fault detection in safety critical systems, and access control models (PV and Sandhu 2016) in critical infrastructures. These defense applications have a system-wide priority, since it is crucial to maintain their services. Analyzing system logs is also a way to understand runtime behavior. As an example, a peculiar network traffic flow at a workstation points out a port scan attack, which is an investigation attack by hackers to find open ways or check the state of security of an organization. In addition, a vast number of logs are created by complex systems constrain analyzes manually (Fu et al. 2013). System opera-

tors usually investigate state of a system, but large number of attributes included in logs generate complexity prohibiting the understanding contextual information. Most solutions for anomaly detection are for a specific domain or problem, because the availability of the data for stating anomalous behavior is a problem (Jyothsna, Prasad, and Prasad 2011). As in the definition, detection of anomalies are simple; however, in application domain, it is very challenging. Key components of anomaly detection are detection techniques, problem characteristics, and the application source (Colombo et al. 2017).

There are several categorization of the existing anomaly detection techniques, but one can confine them into; log template or key based, log semantic-based under the hood of supervised, and unsupervised methods (Du et al. 2017; Huang et al. 2020; Meng et al. 2019; Zhang et al. 2019; Nedelkoski et al. 2020). Key-based methods use log parsing tools to overcome free text problem and identify structured versions of logs as a template. There are two parsers that have been tested in recent works. Spell is an unsupervised parsing method which operates based on longest common sub-sequence. Drain, named Drain3 with Python3 compatibility update[1] , is an online tree based parser with specific written rules (He, Zhu, Zheng, et al. 2017). Several setbacks appear in utilizing parser: requiring manual configurations and controlling rules become complexier, wrong parsed logs create false alarms due to the inability in capturing parameter values or actions (Huang et al. 2020), and acquired templates can cause loss of information (Nedelkoski et al. 2020). Recent studies have mainly focused on capturing semantics from logs using pretrained embeddings to overcome these problems. It also means less processing requirement before a preparing detection model.

Considering anomaly detection as an NLP task, using pretrained word or sub-word embeddings greatly increases the accuracy instead of a sparse definition such as a one-hot representation. Word2vec (Mikolov et al. 2013) and fastText (Bojanowski et al. 2017) are shallow deep network based language models used in the area. There

1. https://github.com/IBM/Drain3

are two types of usages in anomaly detection: pretrained embeddings for encoding directly or utilizing related algorithms to create a variant from scratch. Word Embeddings for Anomaly Classification (WEAC) method extracts features from event logs through word embeddings, which indicate abnormal behaviors (Pande and Ahuja 2020). Skip-gram and Continuous Bag of Words are used in training from scratch. So, Word2vec algorithm was used to gather vector representation of words. On the contrary, WEAC does not discard infrequent words, because it is important not to omit those for anomaly detection. LogAnomaly (Meng et al. 2019) presents template2vec algorithm which is based on the distributional lexical-contrast embedding (dLCE)'s method (Nguyen, Walde, and Vu 2016) to define word representation based on log sources from scratch. Produced vector representations are the inputs fed into LSTM model to detect anomalies. LogRobust uses pretrained fastText embeddings, which is already trained on the Wikipedia dump [2] (Zhang et al. 2019). It attempts to capture semantic information of log events and eliminates more parsing errors, due to provide better similarity in embedding space.

Natural language understanding methods have improved with the introduction of transformer-based LMs. BERT (Devlin et al. 2018) is a pioneer language representation model trained on English Wikipedia and BooksCorpus in the pretraining stage. It is a masked language model that efficiently provides bidirectional semantics. It is greatly contributed in various NLP tasks, due to its fine-tuning ability to adapt downstream tasks. BioBERT (Jinhyuk Lee et al. 2020), SciBERT (Beltagy, Lo, and Cohan 2019) and NeuroBERT (Toneva and Wehbe 2019) are examples of variants of transferring knowledge in different domains. In anomaly detection studies, HitAnomaly (Huang et al. 2020) uses BERT for gathering word vector representations to build log sequence embeddings, then uses the information to distinguish anomalies within hierarchical transformer blocks. Logsy uses its own tokenization method and creates a log vector token that is similar to '[CLS]' token presented in BERT paper (Nedelkoski et al. 2020). It represents a summary of a log event and identifies anomalous behavior with a transformer model.

2. https://dumps.wikimedia.org/

There are two examples of transfer learning methods: feature-based and fine-tuning. The anomaly detection methods, we investigated, utilize feature-based transfer learning. They profit from pretrained embeddings to define log sequences' representations and are adapted into proposed deep learning architectures (LSTM, Bi-LSTM and Transformer). In procuring security of a complex system, central log monitoring tools are responsible for analyzing sequences from multiple and nonidentical log sources. Proposed deep networks need to adapt each different source, which relates to different tasks based on the source. In this process, both feature-based and fine-tuning present new updated weights for each task. This is an inefficient way considering transferred model's degree of sharing parameters. if we are up to create new models for each source or update learned weights, the processes cause loss of information also known as catastrophic forgetting (Zenke, Poole, and Ganguli 2017). In an online setting, streaming vast amount of log sources create a necessity to train new model for a new source sequentially without retraining shared models.

In anomaly detection model, we focus on log semantic-based methods and improve anomaly detection as a downstream task. We utilize pretrained ROBERTa language model. In contrast to its predecessor (BERT), it uses a dynamic changing masking pattern, is able to support longer sequences and discards next sentences prediction task in pretraining (Yinhan Liu et al. 2019). By this way, the model indicates enhanced performance in post-training methods and downstream tasks in experiments (Yinhan Liu et al. 2019). To learn datasets and anomalies, we deploy adapter-based (Houlsby et al. 2019) transfer learning to create scalable and parameter-efficient model which is applicable to various log sources at once. We aimed to build a compact model, considering stream of log sequences as an input.

3.2 Experiments

The anomaly detection model is constructed in a pipelined flow. First, log events are gathered from system logs and prepared for language model training, then we prepare log language adapters for learning synthetic structure. Second, we prepare

data structure of log sequences according to definition of anomalies, then we build structured logs for anomaly adapters. Third, we combine anomaly adapters (AAs) for multi-anomaly task objective. Lastly, we evaluate our experiments with related metrics and compare with recent studies, but *importantly* we test single-source and multi-source pipelines with explainability methods to understand model decisions and acquire feedback on treat data.

Our experiments are performed using a local AI-powered machine. We used a Volta-type architecture GPU with 16GB memory (3xNVIDIA RTX A4000-16GB) and Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz. Volta architecture allows mixed precision ability in execution and enables faster iterations in our experiments. 'O1' option -Mixed Precision- is used (NVIDIA-Automatic Mixed Precision library (Nvidia 2022)). It means tensor-type calculations is made on FP16 (fixed precision, 16) which are called white-listed operations. Moreover, black-listed operations are executed in FP32 (fixed precision, 32) such as softmax. By this way, large-scale of logs trained and adapted to tasks more efficiently and timely (He, Zhu, He, and Lyu 2016).

Source code of experiments can be found on the github page [3] .

3.2.1 Datasets

An unusual behavior of data or anomaly is a change differs from previous observations and source of those is thought to be different mechanism (Ahmed et al. 2015). These unusual behaviours are divided into three:

Point Anomaly (a). They are outliers that arise due to the deviation in the values of the samples in the data (Chalapathy and Chawla 2019). The location address obtained from the system logs during connection to an unauthorized machine can be seen as an example of this type of inconsistency.

3. https://github.com/uunal/anomaly-adapters

Conditional Anomaly (b). These are anomalies that occur when interconnected processes or dependent variables differ over time (Kosek 2016). We can examine the workflow priority anomalies or events interconnected with a cause-effect relationship in that category. As an example to this type of anomaly, port scanning is a method used by attackers to investigate targeted system environment by sending requests to list of ports of a host in the system. With acquired responses, attacker tries to exploit utilized services and gather information about the system.

Collective Anomaly (c). It is an anomaly where the samples are not seen as an anomaly on their own, but when a group of samples differ from the total data in relation to each other (Ahmed, Mahmood, and Hu 2016). In addition, group of these events defines the type of attack. DoS attack is a collective attack which the perpetrator aims to slow down or shut down the services of the system. The group of sent requests from an attacker defines the type of abnormal behaviour.

Firewall logs: The firewall dataset consists of 14,277,447 logs. Three days activity in a corporate network are simulated. We have used all log sequence except for the first day, which includes a DoS attack. We have extracted %0.01 of the abnormal event. Most of the data in first day is predominated by DoS attack, which we omitted and edited data without changing timeline of log events, since attack focuses on only several workstations in the network. 172,135 number of normal logs and 16,902 number of anomalous logs, which consist of DoS, Port scanning, worms and unknown machine connections. This dataset was also mentioned in finding a DoS attack at (Du et al. 2017). This dataset is particularly simulated for IEEE Visual Analytics Science and Technology (VAST) 2011 MiniChallenge-2.

We chose to introduce this dataset because of the explainability motivation aligned with existing Use of Policy Rules in documentation. Additionally, the dataset presents new type of anomalous events different than HDFS dataset, such as point and collective anomalies which also fits the expected scenario.

HDFS: Hadoop Distributed File Systems (HDFS) dataset was first presented in

mining console logs (W. Xu et al. 2009). It consist of 11,175,629 logs gathered from Amazon EC2 nodes. A total of 10,887,379 logs are tagged normal and 288,250 logs are tagged abnormal. Dataset can be found in LogHub, which is collection of system log datasets for AI-based analytics (He et al. 2020). The activities in that datasets are defined by 'blockID' attribute which acts collectively or as a single event.

Both datasets include ground truth information about anomalous and normal behaviors. HDFS dataset includes labeled block IDs indicating which block's log sequence is anomalous. Firewall dataset can be found in challenge called Computer Network Operations at All Freight Corporation[4] . Reviewer documents and Use of Policy Rules for All Freight Corporation provide ground truth related to attacks in Firewall and other log files(such as; PCAP and IDS logs).

3.2.2 Cleaning Data

Log sources are for controlling and analyzing system events. Those are prepared by system developers in nature of free text for readability concerns (Nedelkoski et al. 2020). It is crucial to clean duplicated terms and augment symbolic information in the text without losing information. This process helps build a better knowledge base for the anomaly detection model.

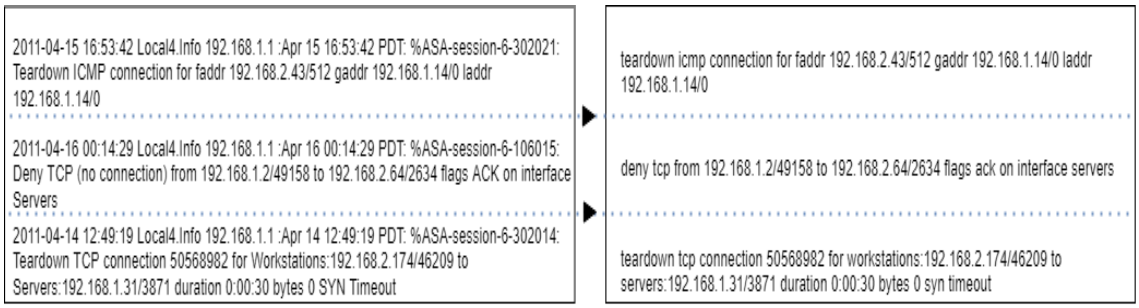

Figure 3.1 Cleaning firewall log examples.

In the firewall dataset, *message_codes* are inserted into log events for identification, as an index. Some event logs include source and destination IPs. They are written in parenthesis. Also, hex coded information can be found included in brackets. This

4. http://vacommunity.org/Computer+Networking+Operations+at+All+Freight+Corporation

represents duplication of information. We removed redundant text content and kept semantics intact. Symbolic presentation of event actions, e.g., '→', is converted to 'to' in verbally describable form. In the HDFS dataset, event logs consist of headers which its content also is included in readable form. '*INFO dfs.FSNamesystem: BLOCK* ..*' and '*WARN dfs. PendingReplicationBlocksPendingReplicationMonitor: ..*' are some examples which are removed to prevent duplication. In this dataset, block information scripted in different forms, we merged block identifiers *'blk_-'* and *'blk_'* to *'blk'* for text regularization. These domain specific cleaning steps are applied to sources before building log vector representations. Figures 3.1 and 3.2 shows samples of cleaning from Firewall and HDFS datasets.

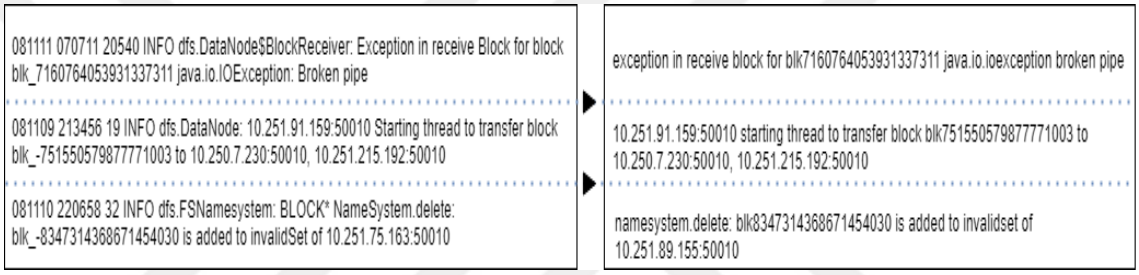

Figure 3.2 Cleaning HDFS log examples.

3.2.3 Processing

Logs can be considered unstructured or semi-structured type of text. We aim to gather much broader contextual information. To achieve that, processing data in our setup is two-folds; First, we prepare data for a log language model. Second, we prepare data for a log sequence anomaly detection model. In log language model, we maintained *line by line* arrangement of the log events in firewall and HDFS datasets and applied cleaning steps. In this manner, we can learn contextual structure of an event log.

In anomaly detection, datasets' timeline and order of logs need to keep intact during preprocessing, since log order has a huge impact on defining anomalous events. In our definitions, see Figure-3.3, timeline is used to point out order, not specifically time that log occurs. Anomalous events differs in their data structure. In point anomaly *a)*, log events formed as $T = [t_1, t_2, .., t_N]$ such that, t_{n-x} is an event consists

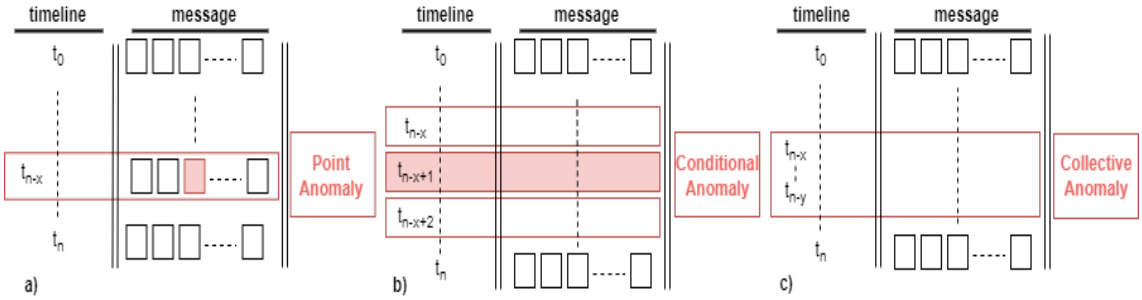

Figure 3.3 Processing log sources by anomaly types.

of semantic features. On timeline $n - x$, a log event has a abnormal token or token groups or whole log event. In conditional anomaly *b)*, log events are structured as $T = [t_1, t_2, .., t_N]$, such that t_{n-x+1} describes an event in the context of t_{n-x} and t_{n-x+2}. On the timeline, event flow should not step on t_{n-x+1} unless it is abnormal. In collective anomaly *c)*, log events are structured as $T = [t_1, t_2, .., t_N]$ such that, log events collectively create unwanted behavior for system health between t_{n-x} and t_{n-y}. Contextual signs reveal anomalous behavior which spread through log sequences in point, conditional and collective anomalies.

From the point of language processing, each log line is processed in a distinct context based on anomaly type. In a simpler context, each line in log data set as $L = [f_1, f_2, .., f_N]$ such that $f_i, i \in [i, .., N]$. N is number tokens created by Byte-Pair Encoding (BPE)(Sennrich, Haddow, and Birch 2015) and has similarities to WordPiece (Schuster and Nakajima 2012) algorithm used in original BERT paper. Original BPE algorithm was used for compressing bytes. In this version of the algorithm, it combines most frequent characters to form n-grams till whole words. Using vocabulary of ROBERTa language model, we prepared chunks of 512 tokens (maximum) via BPE for corresponding log sequence. Sequence of tokens is defined by behavior of log event. In HDFS dataset, this is determined using block ID. In firewall dataset, this is determined by normal and various anomalous events. For example, if f_x describes port scan attack, all continuation logs included in the chunk until max tokens are reached without splitting a log event.

3.3 Anomaly Detection Model

Anomaly detection system is a part of intrusion detection or SIEM tools. Also, anomalous events are not predefined or not expected patterns in the normal activity (Chandola, Banerjee, and Kumar 2009). The detection system analyzes log events within diverse range of sources and indicate anomalous patterns. To detect these patterns, we can explicate the problem as binary classification (Steinwart, Hush, and Scovel 2005; Malaiya et al. 2018).

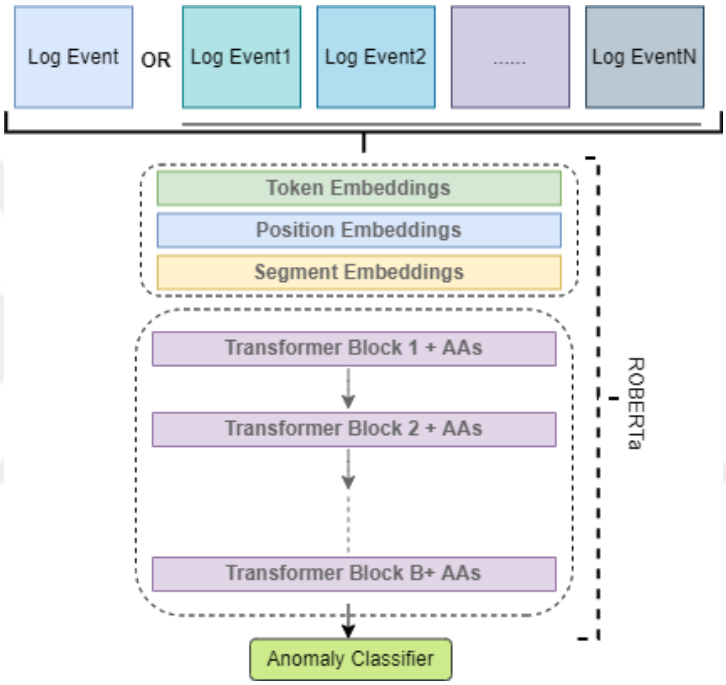

Figure 3.4 Overview of anomaly detection model.

Earlier log semantic-based approaches utilize mainly a feature-based transfer learning. Transformer-based variants' are good at learning from huge chunks of data and produce millions of parameters. Considering explosion of logs and nature of analysis, detection models need to adapt different (ab)normal behavior without retraining for each source. By this way, we prevent creating new parameters and forgetting information of the latter for each task (Pfeiffer, Kamath, et al. 2020).

Adapter fine-tuning is introduced for transformers architecture (Houlsby et al. 2019), which aims to create a bottleneck in transformer block to restrain created parameters and ease sharing. We utilize ROBERTa as base model which has Θ parameters.

This will be our shared parameters across learning log sources and anomaly detection tasks. Each task adapter introduces new parameters Φ and attached to corresponding transformer block n such that $n \in \{1, 2, .., T\}$, T is the number of transformer block used pretrained model (in our case, T=12). To formulate, Φ is trained with loss function as L and used source data as D for each task, see Equation 3.1. By this way, each task presents new set of parameters which contains %1-3.4 of the base model (Houlsby et al. 2019). For task t=0:

$$\Phi_0 \leftarrow \arg\min_{\Phi} L_0(D_0; \Theta, \Phi) \tag{3.1}$$

As in described in processing step, there two types of log data structure is created. First, we kept each log event separately in order to capture syntax in log language modeling. This process is only implemented in training language adapters for further composition with log anomaly adapters. Second, we formalise log sequences according to defined anomaly types, see Figure 3.3. Streamed log sequences are encoded with BPE tokenizer and fed into detection model.

We propose AnomalyAdapters which is a flexible, modular and parameter-efficient transformer-based model which provides transferring knowledge without losing learned parameters and sharing among tasks with adapter-tuning (Houlsby et al. 2019). Our anomaly detection approach is two folds for a log source: log source language learning and anomaly task learning. Lastly, we propose multi-anomaly task detection with AdapterFusion (Pfeiffer, Kamath, et al. 2020) method to analyze multiple sources simultaneously.

3.3.1 Log Language Adapters

Language modeling is required to comprehend distribution of a log source (Saunshi, Malladi, and Arora 2020). Masked Language Model (MLM) training improves base model to represent syntactic structure of a downstream task (Sinha et al. 2021).

Therefore, building log source's language model expedites comprehending semantics of log events. In MLM objective, randomly selected tokens in the log event. From that selected tokens, %80 of them replaced with *[MASK]* special token, %10 of them unchanged and %10 of them changed with token in the vocabulary (Yinhan Liu et al. 2019). In MLM training, cross-entropy loss function is used for optimization of the model. In Equation 3.2, it aims to learn q distribution from inputed log event to true distribution of p in log source. D_{KL} denotes Kullback–Leibler (KL) divergence from p to q, and training attempts to minimize divergence (Saunshi, Malladi, and Arora 2020).

$$\begin{aligned} H(p,q) &= -\sum_{x} P(x) log P(x) - \sum_{x} P(x) log \frac{q(x)}{p(x)} \\ &= H(p) + D_{KL}(p|q) \end{aligned} \tag{3.2}$$

In log language adapter (LLA) training, we kept original ROBERTa model implementation from Huggingface (Wolf et al. 2020) and add adapter modules into transformer blocks using Adapters' library (Pfeiffer, Rücklé, et al. 2020). We are using language adapter which introduced in (Pfeiffer, Vulić, et al. 2020). It is able to learn language specific transformations, and we utilizing to adapt various log types. Adapter modules are optimized and actual weights of base model are frozen during training. This way we efficiently create less parameters in tuning. In Figure 3.5, we have shown how log language adapter module is added into transformer block. We aim to transfer the information into distinguishing anomalous activities.

3.3.2 Log Anomaly Detection

In this section, we provide an architectural addition to adapt anomaly detection in log sequence representations. Adapters are able to create composition blocks in order to share information at ease, see Figure 3.6. Language adapters are intended to capture source specific knowledge. Furthermore, task adapters aim to learn downstream task. In our setup, anomaly detection is the second-order downstream task

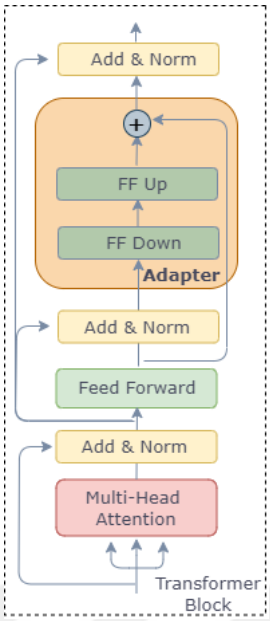

Figure 3.5 Log source's language adapter inside transformer block (Pfeiffer, Kamath, et al. 2020).

which adapting behavior of log sequences (Pfeiffer, Vulić, et al. 2020). Anomaly adapters learns these behaviors in a binary classification setup. In this step of training, only log anomaly adapter (LAA) is activated and optimized. Thus, Log LA and transformer weights are kept frozen.

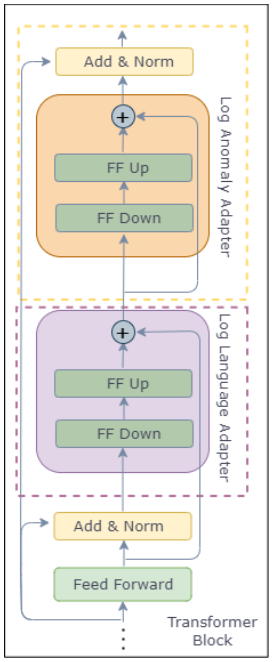

Figure 3.6 Log sequence's anomaly task adapter inside transformer block.

In Equation 3.3, LLA includes a down-projection to hxd where h is the hidden size of

the model and d is the adapter's dimension with a ReLU activation afterwards. Finally an up-projection to dxh is applied. The output of the log LA is fed into a down projection again with following a swish activation function. Then, up-projection is applied again to match dimensions with h layers. In addition, r indicates residual value from transformer block's feed forward layer. Each value represents adapter components in corresponding transformer block b.

$$\begin{aligned} LLA_b(h_b, r_b) &= U_b(ReLU(D_b(h_b))) + r_b \\ LAA_b(h_b, r_b) &= U_b(swish(D_b(LLA_b))) + r_b \end{aligned} \tag{3.3}$$

3.3.3 Multi-Anomaly Task Detection

In real-life log monitoring and analysis tools, log instances are gathered from various machines in a system. To extend the applicability of the approach, we propose multi-anomaly task detection with creating composition of different LLA and LAA stacks. We introduce a new ψ number of parameters to learn how to cooperate stacks together on solving multiple anomalies from different sources. In Equation 3.4, for combined task t we learn Ψ_t parameters for n different task such that $n \in \{1, 2, ..., N\}$.

$$\Psi_t \leftarrow \arg\min_{\Phi} L_t(D_t; \Theta, \phi_1, ..., \phi_N, \Phi) \tag{3.4}$$

In this approach, presented ψ parameters consist of Query(Q_b), Key(K_b) and Value(V_b) that b indicates corresponding transformer block. In each block, output of feed forward layer fed into Q_b and adapter's output use as input for K_b and V_b. In this way, we utilize attention-based learning to decide which stack should be responsible for incoming log sequence.

We calculate output of values from each adapters and transformer block:

$$z'_{b,n} = z^T_{b,n} V_b$$
$$Z'_b = [z'_{b,0}, ..., z'_{b,N}] \tag{3.5}$$

Key and query values are input into a softmax function to learn which LAA is suitable for that log sequence. Then, it is multiplied with AAs values create output.

$$s_b = softmax(h^T_b Q_b \otimes z^T_{b,n} K_b)$$
$$o_b = s^T_b Z'_b \tag{3.6}$$

In this multi-anomaly task training, we combined Firewall LAA and HDFS LAA under the fusion module explained above. Combination of AAs in fusion structure is shown in Figure 3.7 that represents each transformer block in the base model.

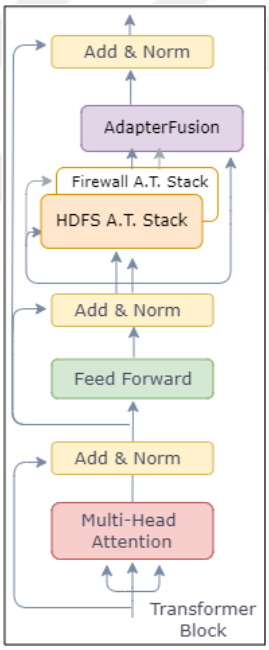

Figure 3.7 Multi-anomaly task detection block in each transformer block.

3.4 Evaluation

In the experiments we applied the processing steps required for both Firewall and HDFS datasets in Section 3.2. First we prepared log sources for language adapter training. Then, we selected half of the datasets for language modeling. In this selection we kept distribution of normal and abnormal log events. In firewall dataset,

type of events are found via attacks that cause anomalies. In HDFS dataset, it is determined by the distribution of normal and abnormal blockIDs. Stratified sampling was used in the process of the data splitting. In log sequence anomaly adapter training, log events are transformed into normal and anomaly definitions as described in 3.3. In both processing, normal events structured collectively. Additionally, we have used %80 of data for training and %20 of data for testing in each training phase. For additional training hyper-parameters, see Appendix A.2.

Evaluation Metrics. Anomaly detection is a binary classification problem. False Positive (FP) rates indicates wrongfully detected anomalies and False Negative (FN) shows missed anomaly ratio in detection from existing anomalous log events. To maximize the performance, FP and FN rates should be minimized. For this reason, we utilize Precision, Recall and F1-score measures in evaluation.

$$\texttt{Precision} = \frac{TP}{TP + FP} \qquad \texttt{Recall} = \frac{TP}{TP + FN}$$

$$\texttt{F1-Score} = \frac{2 \cdot \texttt{Precision} \cdot \texttt{Recall}}{\texttt{Precision} + \texttt{Recall}} \tag{3.7}$$

		Precision	Recall	F1-score
Key-based	PCA	0.98	0.67	0.79
	DeepLog	0.9	0.96	0.93
Semantic-based	LogRobust	0.98	0.93	0.95
	LogAnomaly	0.96	0.94	0.95
	HitAnomaly	0.94	0.95	0.95
	Single AAs	0.97	0.94	**0.95**
	Multi AAs	0.96	0.93	0.945

Figure 3.8 Evaluation on HDFS dataset. Evaluation metrics for Single AAs for Firewall datasets are: *Precision*:0.99 ,*Recall*:0.98,*F1-score*:0.98.

In both training process and dataset, we used pretrained ROBERTa language model, as a transformer variant, to encode and adapt defined anomaly types through adding a bottleneck element. We have used several baselines to compare log key-based and

log semantic-based anomaly detection methods. In log key-based approaches, we compared with two studies, PCA (W. Xu et al. 2009) which analyzes log representation as count vectors, DeepLog (Du et al. 2017) which uses LSTM model to predict next log key in workflow. In log semantic-based approaches, LogAnomaly (Meng et al. 2019) creates feature-based learning via dLCE log vector representation in LSTM model. LogRobust (Zhang et al. 2019) is another solution which initiates log representation with shallow deep embeddings and facilitates from Bi-LSTM model in detection. HitAnomaly (Huang et al. 2020) uses BERT-based log and parameter embeddings with hierarchical transformer architecture. As a counterpart, AnomalyAdapters is a novel way to train on various log sources with an efficiency. And we are able build composable and scalable anomaly detection model. As a result, we have selected HDFS dataset as a common comparator and utilized firewall logs to establish diversity in sources.

Additionally, we investigated the amount of newly introduced parameters for log language and anomaly adapters. ROBERTa model has 120M parameters which we share among different anomaly tasks and sources. Single AAs solution presents; %1.47 in LLA,%2.66 in LAA of the base model's parameters in Firewall logs, %1.47 in LLA,%3.38 in LAA of the base model's parameters in HDFS logs. Multi AAs fusion solutions presents additional %30 of base model's parameters for detecting anomalies from multiple sources. In comparison to methods used in log semantic-based anomaly detection models, we generated %2-4 base model parameters for the anomaly detection model on a single log source instead of creating %100 or more task specific parameters. In overall, we achieved on-par results with recent studies with less parameters in Single AAs model for the HDFS dataset. For the Firewall dataset, we achieved acceptably high scores, especially in F1-score (0.98) in Single AAs model. In combination of both log datasets, multi-anomaly task detection model achieves considerably high F1-score (0.945) with highly shared parameters without compromising contextual information. This approach also establishes competitive advantage on building extensible models for anomaly detection in an online setting.

3.5 Explainability of Model Decision and Threat Data

In recent years, understanding deep neural network becomes necessity with acquiring good results. Complex models can create precise decision making on trained tasks, but lack of comprehending how. Yet, not showing importance of model functionality in domain applications impedes further advancements in deep networks (F. Xu et al. 2019). There are many domains that need an explainability of a model decision such as; health, education and security (Doshi-Velez and Kim 2017). In cyber security domain, using algorithms to test a model function is beneficial in perspective of CTI life cycle (Samtani et al. 2020). These algorithms builds comprehensive visuals to unbox decision making by deep networks. Doshi-Velez states that lack of problem formulation creates 'incompleteness' (Doshi-Velez and Kim 2017). We believe that rapidly changing technological advancements obstruct adaptability of model function to a problem in cyber domain, in consequence of incompleteness.

Transformer architecture and its applications to different domain problems are considered as complex or black-box model (F. Xu et al. 2019). In cyber security, deep neural network (DNN) based solutions to anomaly or intrusion detection have lack of presenting a way to explain inference results. In general, experiments are based on trusting a model decision via only evaluation metrics. Using attributing techniques can reveal the affect of input features on decision making and more importantly enlightens cyber threat data. By this means, it can be used to improve proposed solutions.

In our experiments we have tested three gradient-based algorithms to explain inference results in our evaluation. Integrated Gradients (IG) (Sundararajan, Taly, and Yan 2017) method tries to understand inference of a deep network with its input features. Gradients are, simply, the coefficients learned by DNN. It can create cause-effect relationship on the model inference stage. Acquiring IG is to accumulate gradients along with a path considering input x and $x^{'}$. In Equation 3.8, we can see calculation of integrated gradient for i^{th} dimension for x considering F is

the model function.

$$\mathrm{IG}_i(x) = (x_i - x_i^{'}) \times \int_{a=0}^{1} \frac{\partial F(x^{'} + \alpha \times (x - x^{'})}{\partial x_i} d\alpha \tag{3.8}$$

Smooth Gradients (SG) method yields gradients and acts on them as *saliency* or *sensitivity maps*. This method brings noise into gradient calculation and can be combine with other gradient map techniques. By label(or class), it is known that sensitivity maps correlates with decision boundaries (Smilkov et al. 2017). Especially, it is working with image classification very well and comprehensible by human perception. Expert knowledge and experience are needed to interpret a specialized domain such as; cyber security domain and anomaly detection task. In (3.9), SG_c calculates the effect of minimum change on class decision.

$$\hat{\mathrm{SG}}_c(x) = \sum_{1}^{n} \mathrm{SG}_c(x + \eta(0, \sigma^2)) \tag{3.9}$$

Lastly, Input Reduction (IR) is different way to analyze interpretation. In contrast to saliency interpreters we discussed before, it examines the importance via counterfactual way (Feng et al. 2018). Importance is defined by difference in confidence change after altering input values. In (3.10) shows the calculation of importance on input perturbation. This gradient-based methodology also enlightens the pathological behavior of a model. In the reduction process, we may see one or two tokens to be selected at the end and the method protects the original result. By this way it also reveals adversarial examples for a model.

$$\mathrm{IR}(x_i|x) = f(y|x) - f(y|x_{-i}) \tag{3.10}$$

In Figure 3.9 and 3.10, we have presented an example sequence from HDFS and

firewall logs. We chose a log sequence which alarms the detection model as anomaly. For brevity, we omitted part of log sequence, as methods indicate less importance or lower gradient-based value on the model decision. In Figure 3.9, we are looking at an anomalous behavior of a block in HDFS logs. IG method focuses context between sequences and shows the most impacting phrase as 'not belong' (event action) and its context. SG method slightly differs from others and focuses to create boundary on a starting point of the action such as; 'request received' or 'added invalid-set'. Subwords that are highlighted in grey show omitted inputs without changing model decision. IR method focuses on the same phrase again as in IG to decide anomalous behaviour. The result also depicts adversarial example for the log sequence. In Figure 3.10, we investigated a port scan activity on workstations. IG method emphasises overall context of a log sequence, but indicates 'tcp connection' for creating an abnormal event on the workstation. SG method again focuses on the action word 'built' of an event boundary, but also points out IP range (.175) defined in the network. IR method singled out 'tcp' and '.175' which is a good example of pathological behaviour of a model, but we can comprehend that connection type and source IP are the indicators of an anomaly. To sum up, overall results are logical, methods focus on workstations which are infected and port scanning other systems in their sub-network. Additionally, *.175* is not in the range of defined IPs in the Use of Policy Rules for the tested network and sequences conditionally point out port scan attack.

Overall in our explanation tests, we used proposed models for Single and Multi AAs (see Appendix A.3) and examine model decision without providing any context information, policy rules for the network or configuration file of a log type prior to training a model. Comparing facts from HDFS and Firewall dataset, our proposed model understand the reasoning behind an anomaly and can match useful threat data. Also, models exposes their pathological behaviors to us that some tokens in context have high importance in decision making. This also leaves a gap for improving the current stage.

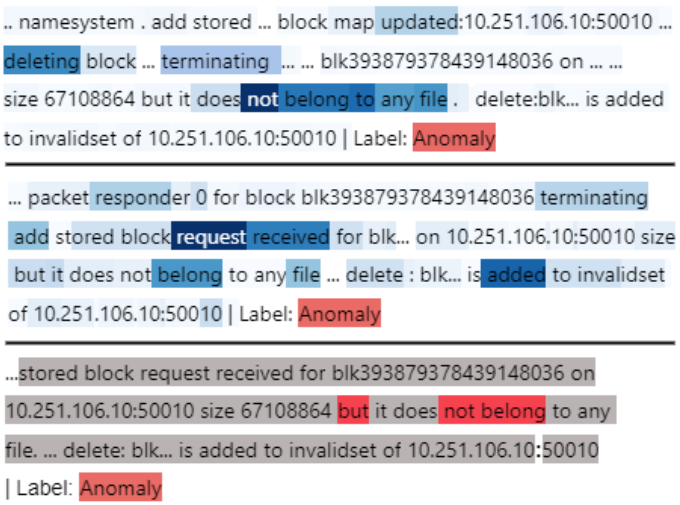

Figure 3.9 Model decision on HDFS logs by Integrated/Smooth Gradients and Input Reduction methods.

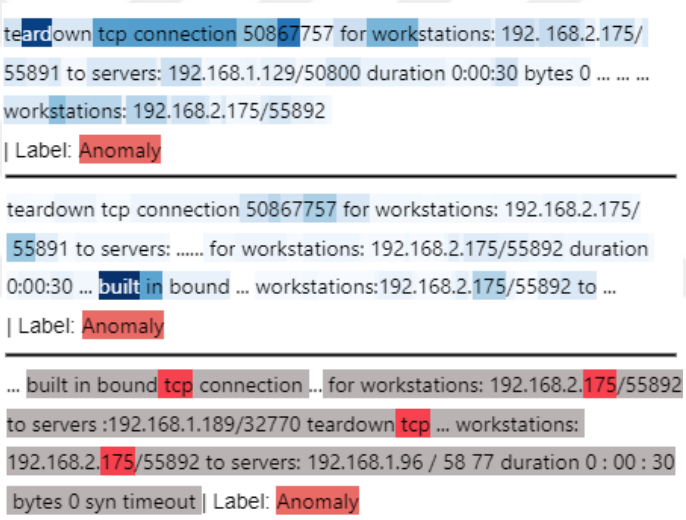

Figure 3.10 Model decision on firewall logs by Integrated/Smooth Gradients and Input Reduction methods.

3.6 Discussion

Security applications are a necessity for systems in different domains, such as enterprises and critical infrastructures. Anomaly detection is the crucial part of these systems for ensuring security of the continuous activities. Logs are the first source to consult when analyzing events in a system. By this means, system administrators and security professional put log monitoring systems into center of security operations centers. In addition to that, SIEM tools are the preferred implementation

space for security enhancements.

Log events are recorded in free form or unstructured text. System developers prefers to build readable log events in exchange to ease manual monitoring (Bertero et al. 2017). It also opens up a problem when considering the complex nature of systems. Manual labor can not match in existing problem space, hence there are many suggested solutions based on automating log analysis in anomaly detection systems. There are different categorization of presented solutions. If we simplified solution proposal under security domain, we can divide them into two: log key-based and semantic-based anomaly detection methods. Semantic-based methods mainly elaborates contextual knowledge of logs from pretrained deep or shallow networks. These findings also reveal the need of researching learning methods considering applicability to the domain needs.

Under this hood, we build AnomalyAdapters, which provides an extensible and modular approach for anomaly detection. It brings a competitive advantage on yielded parameters and simultaneous adaptability to different log sources. Addition to that, adapter's bottleneck architecture improves sharing information without catastrophic forgetting issues. In our experiments, we have compared our work with other recent studies in the field and also tested model decisions to get feedback in a readable form. Explainability is a known issue for black-box models, thus it also enables threat intelligence actively in the log semantic-based learning which opens a new direction for enhancing solution of anomaly detection problem.

4. FedRA: Risk-Adaptive anomaly detection with federated learning

Recent developments in the ML offers high performance solutions for various tasks in many domain. There are three factors help improving the performance of ML solutions; vast amount of data and its availability, enhancements of computational power and advancements in new deep learning architectures (Alazab et al. 2021). But, many domains still are not able to utilize these advancements in real time. The biggest problem of all is not to exploit vast amount of data at the edge devices effectively. In that scenario, we are assuming edge devices as computationally sufficient sources. To achieve learning with edge devices bring up complications together. Transportation of data to analyze in a dedicated system components has several problems, such as; data leakage, security and privacy (Preuveneers et al. 2018). In addition, data size is the determinant for network transfer rate while moving data blocks to these components. It is important to optimize data size in order to achieve faster solutions in a traditional scenario. The emergence of interconnected systems produces a need for decentralized control mechanism for investigating data blocks. Federated Learning (FL) as an optimization mechanism for decentralized environment, enables participant devices to learn collaboratively without sharing data. Therefore, FL can present solutions for transporting data in accordance with communication cost, ensuring reliability and integrity of learning system and privacy of the data from participants compared to traditional centralized learning (Alazab et al. 2021).

FL becomes a popular choice with recent innovations to analyze, learn data habits and make inference according to envisaged tasks. FL is a way of building a model in a decentralized setup (Bonawitz et al. 2017). Growing number of enhanced processing ability at remote devices enables this type of learning mechanism. FL is utilized

in various domain problems, such as next word prediction in messaging applications on mobile devices (H. B. McMahan et al. 2017), health monitoring wearable devices for early diagnosis of several diseases (Y. Chen et al. 2020) and on-vehicle learning mechanism to maintain latest model for best intelligent performance (Pokhrel and Choi 2020). Unlike traditional learning approaches, FL method is able to coordinate and distribute learned parameters from participant models. Central control mechanism of FL combines learned parameters with a specific strategy, then share among participants. By this way, overall system can learn knowledge from unseen data sources. Additionally, FL mechanism enables using differential privacy and secure aggregation techniques to ensure security and privacy of the parameters that each participant shares (McMahan and Ramage).

Cybersecurity is the process through which an organization attempts to secure its interlinked and Internet-connected systems against various cyberthreats. Cybersecurity also seeks to protect sensitive data from external threats. Businesses use the methodologies to protect their data centers, includes organizational and client-specific data, from illegal access and prevent disruptions on its services. Individuals also utilize security applications to protect their private information from threats. A cyber-attack is an attempt by an individual or organization to breach information in order to obtain some benefit from the victim's system being disrupted (Alazab et al. 2021). Businesses are targeted by cyber attacks, and cybercrime is on the rise, furthermore large-scale and complex system security becomes more crucial and speed of developing cyber environment opens a case for FL approaches, due to aforementioned problems. Considering these problems, integrity, confidentiality and availability should be considered on designing security solutions (Council et al. 2007). Completeness of data and consistency is pivotal role in ensuring integrity of a system. In case of a cyberattack, intruders can manipulate data instances to mislead central system, but FL ensures sensitive data contained in local environment. Unauthorized access to data causes leaking confidential information, since FL provides a secure environment utilized by authorized access. Availability of a system is crucial in case of a cyberattack. With FL, local models and global model

are kept available. These three pillars are building block of CIA triad. CIA model was utilized for building many security applications. In addition, security professionals evaluate impact of vulnerabilities based on these pillars of a system (Zhuang, Zamir, and Liang 2020). By this point, FL mechanism presents a holistic view via enabling these pillars on cybersecurity applications (Nguyen et al. 2018).

There are various efforts on cybersecurity to apply FL solutions. Lim's work indicates increasing computation power and storage capabilities in mobile edge computing and investigates the current problems in applicability of FL (Lim et al. 2020). Also it points out the advantages of FL in mobile networks such as; efficient use of bandwidth, privacy and low latency. On the other hand, Kim's work presents incentive mechanism to enhance quality of collaboration in FL setup (Kim et al. 2018). It is a Blockchain based strategy which enables update verification among participants. In addition, participant, which verify these updates, are rewarded by the Blockchain network. Incentive mechanism are most useful when applied network is a human-controlled use case. Moreover, Weng's work indicates vulnerabilities in FL communication (Weng et al. 2019). In global model updates, transmitted intermediate gradients still reveals important information. Also, this work points out that dishonest participator still risks security measures in FL mechanism. To ensure auditability of training process, it presents a Blockchain-based FL mechanism (Weng et al. 2019). Most of the business processes are built on top of cloud computing infrastructure affects distributed services that are based on IoT. Abeshu and Chilamkurti investigates security problems about edge computing in a cloud-based environment and propose DL solution in cybersecurity for FL, considering edge devices and highly distributed networks (Abeshu and Chilamkurti 2018). Moreover, the survey states that convergence of a model, statistical heterogeneity and communication cost are important topics to investigate in FL (Lim et al. 2020), especially for cybersecurity applications.

In this chapter, we investigate communication cost and convergence speed of a model training in a hostile environment. We propose a novel federated approach for

anomaly detection which is a continuation work for AnomalyAdapters. Parameter-efficient learning provided by adapter-tuning is also advantageous on sharing parameters in FL mechanism. Additionally, we present a risk-adaptive federated learning with influence of Spreading Phenomena in the applied system network.

4.1 Background and Related Work

Decentralization of systems and increasing complexity impede monitoring and analyzing each local data block in a centralized system. The emergence of IoT, cloud computing and increasing computing power on edge devices orient proposed solutions into decentralized approaches. The amount of sensitive data yield by edge devices increases via enhanced computing power and data storage. There are several problems occur in context of DL model solutions in the current state. Collecting distributed data from a system to build a knowledge base, in our case a DL model, costs time and needs higher computing power to build a model from vast amount of data without using edge devices. As a solution, FL provides suitable system mechanism to govern highly distributed systems. With rapid growth in decentralization, individuals' and businesses' sensitive data is constantly under attack from dangerous hackers and invaders (Alazab et al. 2021). In case of system security, a cyberattack can block operations of a system and central mechanism should mitigate these problem in a timely manner. Organizations need a reliable defense tools in order to protect and analyze each data source in a network. As described earlier, not being prepared to overcome these threats can cause catastrophic incidents which negatively impact safety, health and well-being of a society. IDSs, IPSs and SIEM tools are the most common defence systems used in prevention and protection from targeted cyberattacks. Recent studies show ML based solutions provides a decent solution on detecting known patterns, such as signatures of intruders. These type of solutions are built from known indicators of an attack and presents promising result in perspective of protection systems, but these solutions have high false detection rates and not able to detect novel attack types (Hu et al. 2017). In that case, recent ML/DL-based anomaly detection techniques are more advantageous in

adapting abnormal or unknown events.

Anomaly detection is crucial to preserve system security (Chandola, Banerjee, and Kumar 2009). It is one of the most significant topic used in data analysis. An information object is termed an anomaly if it deviates significantly from usual data behavior in some domain. In general, it signifies that the object is distinct from the rest in a given data array (Hu et al. 2017). Mitigation of these peculiar events is important to protect sustainability of an organization or even the corresponding domain. Continuous growth of information systems and decentralization creates an open area for utilizing unique abilities of FL. In the light of aforementioned problems, FL presents a novel way to analyze local data from individual devices and enables to build customized solutions. With the usage of FL, DL based anomaly detection solution can protect devices individually and globally. FL enables adaptation of detection techniques in various decentralized system setup. In Li's work, it proposes a collective IDS mechanism via including participants from multiple industrial cyber physical systems in the same domain (B. Li et al. 2020). The proposed model, DeepFed, utilizes CNN integrated with GRU component in order to detect intrusion behaviours. In concern for privacy issues, this work utilizes secure communication via Pallier cryptosystem to provide federated learning over a cloud server that is secure and privacy-preserving. In Chen's work, proposed an IDS mechanism, which is called FedAGRU, for securing wireless edge networks (Z. Chen et al. 2020). This work also focuses on reducing communication cost via selecting more important local updates through an attention mechanism. In Liu's work, it proposes a FL mechanism based on deep network, which utilizes CNN-LSTM for detecting anomalies in time series (Yi Liu et al. 2020). CNN is utilized for learning features from the dataset, and LSTM enables to predict efficiently considering time series dataset. In addition, this work uses gradient compression in order to reduce communication cost.

FedDetect is a federated optimization approach for IoT platforms (Zhang et al. 2021). It utilizes router as security gateway to prevent incoming attacks, such as DDoS attack, to IoT devices. The security gateway, a local training, uses Deep Autoen-

coder (Rumelhart, Hinton, and Williams 1985) in detecting anomalous behaviour. FedDetect adapts federated averaging with cycling learning rate which adapts to aggregation rounds (Zhang et al. 2021). MT-DNN-FL (Zhao et al. 2019) is a multi-task anomaly detection approach which is tested on CICIDS2017(Sharafaldin, Lashkari, and Ghorbani 2018), ISCXVPN2016(Draper-Gil et al. 2016) and ISCX-Tor2016(Lashkari et al. 2017) datasets. It utilizes averaging optimization from local updates. Another proposed model uses a two stage learning in detecting anomalies. First stage happens in local machines, second stage is to build anomaly classifier in global optimization (Zhao et al. 2020). In this process, part of the model layers keep frozen to prevent information loss. Another IoT-based work (Mothukuri et al. 2021), utilizes ensemble of four Gated Recurrent Units (GRU) via random forest decision classifier. In evaluation of the proposed model, this work uses Modbus dataset(Frazão et al. 2018). In context of federated optimization, it utilizes averaging method. In the state of FL implementations in cybersecurity, FedAVG (B. McMahan et al. 2017) is highly used optimization method which weights participating parameters via number of data instances used.

Algorithm 1 Federated optimization algorithm – FedAVG (B. McMahan et al. 2017)

w_0 initialize values
K : num of participants
C : ratio of selected participants
for $t = 0, 1, 2, ...$ **do** ▷ t : each round
    $m \leftarrow \max([C \cdot K], 1)$
    S_t = select random m participant
    for $k \in S_t$ **do** ▷ k : participant
        w_{t+1}^k = **ClientUpdate**(k, w_t) ▷ w^k local updates
    $w_{t+1} = \sum_{k \in S_t} \frac{n_k}{n_\sigma} w_{t+1}^k,\ n_\sigma = \sum_{k \in S_t} n_k$ ▷ w global model updates

FedAVG is introduced in B. McMahan et al. (2017) which exemplifies federated optimization techniques. In this optimization method, central aggregator distributes w_g to each client/participant, then clients perform optimization on their local dataset

D_c of size n_c, total length of datasets as n. After each client applies stochastic gradient decent (SGD) $f_c = \eta \nabla F_c(w_g)$ with a learning rate η on its loss function F_c as $w_{g+1}^c \leftarrow w_g^c - f_c$ and share its learned weights w_{g+1}^c with the central server to improve global model w_{g+1}. So we have $w_{g+1} \leftarrow \sum_{c=1}^{C} \frac{n_c}{n} w_{g+1}^c$ at the end of round. The server combines these parameters via weighted averaging and builds global model. In each round, the global model is shared with each client and repeats the process until model converges or desired federated rounds (B. McMahan et al. 2017). Aside from that, there are federated optimization approaches other than using weighted averaging for global parameters. It is also good to mention them in order to comprehend adaptability issues in case of convergence problem in the global model. Similarly, to protect communication cost, adaptive methods used on central aggregator, and clients has same SGD optimization in updates. FedADAM is an adaptive federated optimization algorithm which transformed from ADAM (Reddi et al. 2020). FedADAGRAD (Reddi et al. 2020) is a version of ADAGRAD optimization algorithm and FedYOGI (Reddi et al. 2020) is the version of YOGI optimization algorithm. This work presents FedOPT (see Algorithm 2) algorithm which encapsulates FedAVG and separates client and server optimization distinctively. Each client, and server utilizes gradient-based optimization which includes aforementioned optimizations above. According to findings in Reddi et al. (2020), these algorithms also have similar communication cost compare to FedAVG and are more suitable for heavy-tailed data distributions for ensuring convergence. In federated anomaly detection studies, weighted averaging is the most preferred and prominent optimization method.

4.2 Risk Adaptive Participant Selection and Weighting with Network Theory

In federated optimization, weighted local updates is used to increase the effectiveness of global model (Tao and Li 2018; Mingzhe Chen et al. 2020; Mingqing Chen et al. 2019), since the data among participants not independent identically distributed in real life scenario (Nilsson et al. 2018). Therefore, it is important to choose which

Algorithm 2 Adaptive Federated Optimization – FedOPT(Reddi et al. 2020)

w_0, η ClientOPT, ServerOPT ▷ inputs,η:learning rate
for $t = 0, 1, 2, ...$ **do** ▷ t : each round
    Sample a subset S of clients
    $w_{0,i}^t = w_t$
    for each client $i \in S$ **do**
        for $k = 0, ..., K-1$ **do**
            Compute SGD on client
            $w_{i,k+1}^t = ClientOPT(...)$
        $\Delta_i^t = w_{i,K}^t - w_t$
    $\Delta_t = \frac{1}{S} \sum_{i \in S} \Delta_i^t$
    $w_{t+1} = ServerOPT(w_t, -\Delta_t, \eta, t)$ ▷ η:learning rate

participant enables model to converge quicker according to envisaged task. Complex networks methods can be adapted and applied into many interdisciplinary studies owing to intuitive basis (Albert and Barabási 2002). Moreover, basic properties of a network can calculated and visualized rapidly and easy to comprehend (Caldarelli and Catanzaro 2012). Avalanche-like growth in data sources necessitates to extract fruitful information to build intelligent solutions, and alternative approaches are enabled in coordination with network theory. Using this theory as a mathematical basis, we can able to define a system network and make inferences using network properties (Albert and Barabási 2002). To define a network, we need some elaborate definition of how these component interact with each other. Like social (etc. Twitter, Facebook) or biological networks (etc. COVID-19 pandemic, SARS), digital networks can be orchestrated using network theory. In a hostile cyber environment, digital viruses or harmful software utilize connections between components to spread. In IS infrastructures, each machine can be described as nodes and connections as edges. Considering network as G and the set of sub units present in the system as V and any interaction between sub units presented with set E. By this way, we can create mathematical understanding how each machine interact with each other, see Equation 4.1 (Albert and Barabási 2002).

$$G = (v_i, v_j),\ v_i\, and\, v_j \in V\ and\ (v_i, v_j) \in E \tag{4.1}$$

Spreading Phenomena is an important topic under network theory which investigates how diseases spread in communities or network (Albert and Barabási 2002). Nowadays, trying to find notions for spread mechanism of diseases skyrocketed, due to ongoing COVID-19 pandemic (*Coronavirus disease (COVID-19): How is it transmitted?*). This phenomena is investigated under network epidemics in order to comprehend and predict impact of such diseases. Additionally, this specific research area enables analytical reasoning to build cause-and-effect analysis and forecast impacts of infectious diseases or digital viruses (Caldarelli and Catanzaro 2012). Epidemic modelling, is a framework defined by network epidemics, assists to analyze the spread of the pathogens in the network. As a definition, epidemics represent the unusually large outbreaks which is also affected population in a short term (Newman 2002) which is also suitable definition for spread of digital viruses (Albert and Barabási 2002). Epidemic modelling has different concepts for managing state of the susceptible network. There are two approaches, in our concern, to define state of the nodes in a network for this proposed method 4. These models are Susceptible(S)-Infected(I) and Susceptible(S)-Infected(I)-Susceptible(S) (Albert and Barabási 2002). First model (SI), accepts that at start state ($t = 0$ as time) all nodes are vulnerable to get infected by diseases and models the network in a way to forecast when all nodes in the network will be infected in a network at $t = t_e$ as endemic state. In consideration of digital network, an IS infrastructure, cyber incident management teams will interrupt spread to prevent endemic state and maintain organizational services kept intact with resetting network to start state. SIS model is more suitable in this case, since infected nodes become disease free(susceptible) again at unit time μ. We assume that μ is the reaction time of cybersecurity teams for cleaning infected machine.

So far we understand that SIS model is a matching approach in case of cyber threats. In addition, communication cost is the most effective advantage in distributing pa-

rameters in federated learning. To increase its effectiveness of the cost, we can optimize it with the importance of the participant node's shared weights. Tao and Li (2018) proposed an algorithm which selecting the most effective intermediate gradients in order to decrease communication cost more. The important point is to find gradients which are the most useful for global model convergence. To discover that, we elaborate *characteristic time* which is presented by epidemic models. Characteristic time is a measure to define how fast a disease can spread to other nodes in the network (Caldarelli and Catanzaro 2012). In SIS model, using susceptible(S), infectious(I), degree of node(d), transmission unit time(β) and recovery time (μ), we calculate τ^{SIS} as characteristic time of the node to spread, see Equation 4.2. As definition states, degree of node and characteristic time is inversely proportional. Such that, if a degree of node is increased, time to spread diseases shortens. By this way, the order of the importance of node, participants, is determined by the amount diversity of the data. Characteristic time effectively filters out less influential node. Thus, ranking them according to their risks can be done with Equation 4.2.

$$\tau^{SIS} = \frac{d}{\beta d^2 - \mu d} \tag{4.2}$$

Assumptions. Epidemic modelling considers a network based on two fundamental hypotheses (Albert and Barabási 2002). We should present these assumptions in order to build IS network for aforementioned approaches.

`Assumption-1`: Compartmentalization, which means each node in defined network has three states. Susceptible(S) describes healthy node which can be infected by a virus, in our case a cyber threat. Infectious(I) describes an infected node which can spread virus to other nodes in the network. Lastly Recovered(R), describes nodes which is infected before and recovered from virus. In our case, we choose not to use recovered, since other epidemic models can define recovered as immune node (Caldarelli and Catanzaro 2012). Considering dynamic cyber threat environment, infected node can be cleared from a cyber attack, but it is still be susceptible for

other type cyber threats.

`Assumption-2`: Homogeneous Mixing considers nodes have equal chance to become infected from a harmful threat. In IS infrastructural design, some nodes in the network ,such as central database, is highly protected, but it eliminates where disease come from precisely and assumes any node in the network can infect another (Albert and Barabási 2002).

`Assumption-3`: In addition to these fundamental hypotheses, we choose to rule out recovery time($\mu = 1$) in order to simplify mechanism. This also means that, cyber security teams have a fix time to recover cyber attack in order to continue its services or threat elimination.

4.3 Experiments

The proposed federated optimization is prepared by using AnomalyAdapters, see Section 3.3.2. It is a decentralized control mechanism for the model. We experimented on the aforementioned HDFS dataset. Our experiment is designed to compare convergence speed in accordance with the network epidemics. The network epidemics is defined by SIS model (Albert and Barabási 2002) for applicability in dynamic cyber threat domain. SIS model used as a mathematical basis for simulating experimental network topology and providing risk-based metrics to test effect on convergence speed.

Cleaning and processing are applied to the HDFS dataset according to the Section 3.2. To explicate federated simulation, we prepared 3-folds of subsets from HDFS dataset. We split the dataset using Stratified K-Fold algorithm to keep distribution ratio of normal and abnormal log events via ground truth information, *labels*. Additionally data is distributed favorably by degree distribution. In our setup, we used a Volta-type architecture GPU with 16GB memory (3xNVIDIA RTX A4000-16GB) and Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz. Each dedicated GPUs acted as a participant machine, and CPU acted as a central server for governing

federated mechanism. From 11,175,629 line of logs, each fold created for a simulated machine, 80% of the each dataset is used for the training, 20% is used for the evaluation. Client-based evaluation is used to compare results in a federated setup. In this setup, we assume a participant that have enough processing power and dedicated security tool which is responsible for prevention of abnormal events. Other machines included in the network are acted as sources and their data is collected at main participant in federated learning.

Source code of experiments can be found on the github page [5] .

4.4 Federated Anomaly Detection

Federated learning mechanism consist of three main stages. First, it determines the initial stage of nodes, calculating degree of each node using adjacency matrix. Adjacency matrix is used to describe a graph network to implement mathematical calculations (Caldarelli and Chessa 2016). It explicitly shows connections between nodes using 0s and 1s. Each degree of a node is calculated using adjacency matrix in order to specify characteristic time of nodes. Degree of a node simply defines number of connections. Infection rate(β) is selected 0.1 in experiments to balance characteristic time range in the network. In overall approach, still number of selected participants are defined by a portion C from total K participants. Resulting value m is used to select number of nodes from highest to lower degree distribution of defined network (see Algorithm 3). This stage follows with local model updates, and global model updates (B. McMahan et al. 2017). Risk-adaptive model(FedRA) is a FedAVG-based optimization method, and difference are highlighted, see Algorithm 3. Risk adaptive participant selection and weighted aggregation is added via characteristic time attribute of a node. According to inverse characteristic time of participant local updates are weighted in the training process, see Algorithm 3.

5. https://github.com/uunal/anomaly-adapters-fedra

Algorithm 3 Influence of Spreading Phenomena in FedAVG // FedRA

A :adjacency matrix of network

$\beta \leq 0.1$

for $n = 1, 2, ...$ **do** ▷ for each node in network

  $d_n \leftarrow \sum_j a_{i,j}$ ▷ i and j indexes of A

  $\tau \leftarrow \frac{d_n}{\beta d_n^2 - \mu d_n}$ ▷ characteristic time to spread

for $t = 0, 1, 2, ...$ **do** ▷ t : each round

  $m \leftarrow \max([C \cdot K], 1)$

  $S_t = maxD(m, d_N)$ ▷ select m highest d_n from N nodes

  for $k \in S_t$ **do** ▷ k : participants

    w_{t+1}^k = **ClientUpdateLLA**(k, w_t) ▷ w^k local updates

    w_{t+1}^k = **ClientUpdateLAA**(k, w_t)

  $w_{t+1} = \sum_{k \in S_t} \frac{n_k \times \tau^{-1}}{n_\sigma} w_{t+1}^k$, $n_\sigma = \sum_{k \in S_t} n_k \times \tau^{-1}$ ▷ w global updates

Algorithm 4 Train LLA

ClientUpdate(k, w): → on client k

$b \leftarrow$ split D_k into batches of size B

for $e = 0, 1, 2, ...$ **do** ▷ e : each epoch

  for $i \in b$ **do** ▷ i : batch

    $w = w - \eta \nabla l_{LLA}(w; i)$

return w to server

Algorithm 5 Train LAA

ClientUpdate(k, w): → on client k

$b \leftarrow$ split D_k into batches of size B

for $e = 0, 1, 2, ...$ **do** ▷ e : each epoch

  for $i \in b$ **do** ▷ i : batch

    $w = w - \eta \nabla l_{LAA}(w; i)$

return w to server

Risk-adaptive federated anomaly detection using AnomalyAdapters consist of 4 sub-steps in overall federated optimization (see Fig:4.1). We have numbered these steps in accordance with Algorithm 3. Phase 0 is for Log Language Adapter(LLA see section 3.3.1) and Phase 1 is for Log Anomaly Adapters(LAA see section 3.3.2) training. Pseudo algorithms for client(participant) updates are also shown in Algorithms 4 and 5. Steps 3 and 4 are the aforementioned preparations for selecting nodes and calculating network-based metrics. In Step 1, LLA is trained over participants to capture syntactical information from the sources in a federated way. Step 1 is also an inactive stage and can be trained only with normal log activities. At

the start of Step 2, Steps 3 and 4 can be repeated if any changes necessary in the network. In Step 2, LAA is trained with selected participants for anomaly detection. These four steps also shows the pipeline of federated mechanism. 3,4 and 1 depicts language model training, 3,4 and 2 depicts anomaly detection training. At the end of each round, local updates(weights) send to central server. Server uses the information calculated in steps 3 and 4 and utilizes weighted aggregation with inverse characteristic time of participants (line:12 at Algorithm 3).

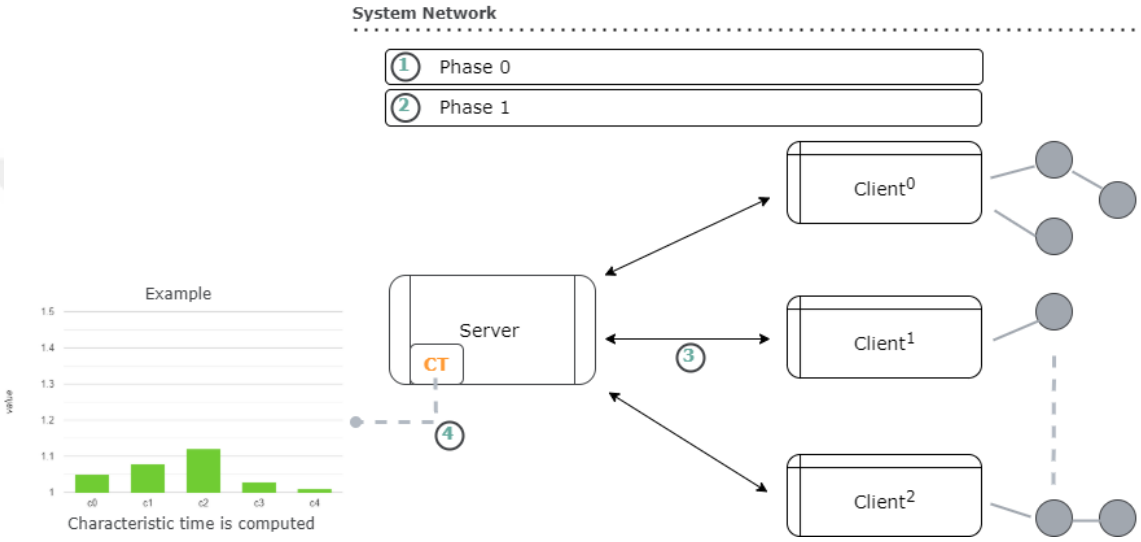

Figure 4.1 Overview of the risk-adaptive anomaly detection with federated learning

4.5 Evaluation

Participant selection and weighting approaches can speed up the converge process of the federated model. In addition, filtering participant decreases the communication cost greatly. Another advantage of the AnomalyAdapter approach is to be easily transferable with federated learning. Adapter sizes varies between 3-7MB which is a huge gain compared to actual size of the model(more than 512MB). For the evaluation of the model, we used Flower framework (Beutel et al. 2020) with NDlib framework (Rossetti et al. 2018) for the simulation of the SIS model. For network topology, we acquired real IS infrastructure and anonymize data nodes and attributes properly due to privacy issues. In Phase 0, server shares its pretrained ROBERTa model to each client, which is not repeated in Phase 1 unless base model needs

to be updated. We call global model as w_t. Each participant gathers w_t and add language adapters to the model structure. In adapter training, we are freezing w_t layers and only train on adapters parameters, we call them as w_{LLA}. In each round of preparing log language model, clients update w_{LLA} and send to the central server. In the evaluation, server combines w_t with w_{LLA} to create LLA as shown in Figure 3.5, then it sends back to each client. In Phase 1, all participant have the base model(w_t) and language adapters (w_{LLA}) already. When training for anomaly detection starts each client builds model structure for LAA as shown in Figure 3.6 to create final model structure as shown in Figure 3.4. In anomaly detection training, each client updates LAA adapters, we call them as w_{LAA}. After that same procedures applies as in Phase 0 in coordination with LAA training.

Various approaches utilize gradient compression techniques in order to reduce communication cost such as in Liu's work (Yi Liu et al. 2020). Compression overall causes the information loss and these works tries to minimize the accuracy loss in desired task. Salehkalaibar and Rini (2022) analyzes compression methods to show how much accuracy change in accuracy results and also proposes an alternative way to overcome this problem. With adapter-based training, we eliminate information loss gained by the adapters and communicate with %2-4 base model parameters for the anomaly detection model on a single log source instead of transporting %100 of model parameters. Additionally, selection of participants for federated learning effects convergence speed of the trained model. In our experiments, FedRA reaches faster convergence compared to FedAVG as seen in Figure 4.2. With 6 round of training with the experiment setup, weighting parameters with inverse characteristic time improves convergence speed. Moreover, our test show that convergence is guaranteed in experiment setup with FedRA approach. On the other hand, evaluation results shows that central learning mechanism reaches higher results compare to federated setup, see Figure 4.3, but FedRA outperforms FedAVG with small margin in anomaly detection setup. Drop in evaluation results is also an expected outcome, due to lack of anomalous events in the log sources. This is an another problem federated learning that heterogeneity of the data source can cause drop on evaluation

results (T. Li et al. 2020).

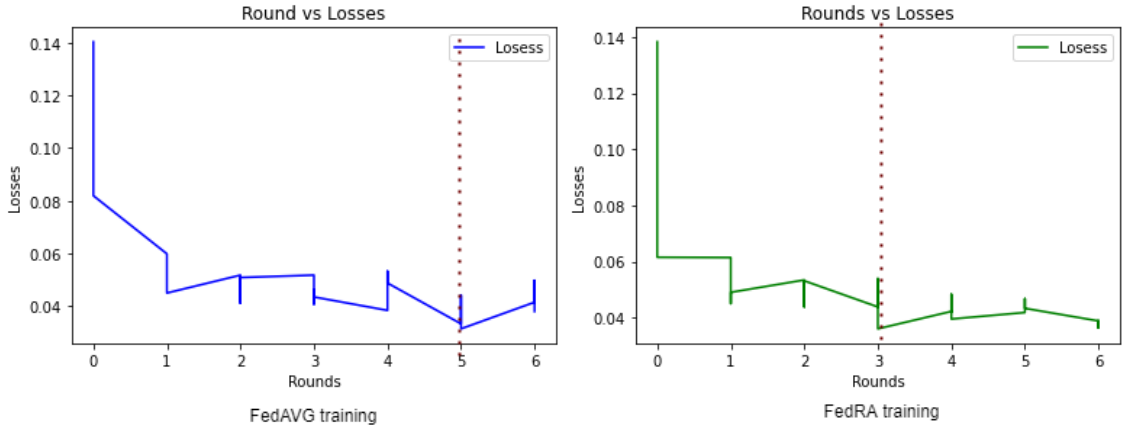

Figure 4.2 Training convergence comparison between FedAVG and FedRA

4.6 Discussion

Federated learning enables building decentralized learning systems, which promises a lot of different aspects for future research. In this chapter, we present a risk-adaptive federated learning system for anomaly detection with enabling network epidemics approaches. SIS epidemic model is used to define network and its attributes. In federated optimization algorithms, filtering participants surely helps convergence of the global model and allows the system to react anomalous behaviors in a timely manner. For current experiment setup, weighting model parameters using inverse characteristic time positively effected the convergence of global model. Overall effect of weighting scheme using network epidemics surely needs more experiments on different use cases, especially for cybersecurity domain.

There are also various drawbacks we had in our proposed method. Anomaly detection is a complex case in statistical analysis. And distribution of data effects model decision especially in federated learning. Selected weighted aggregation for FedRA, still ensures convergence in experimented setup, due to its similar nature to FedAVG. The influence of Spreading Phenomena presents a mathematical basis for understanding and adapting to cybersecurity domain. Anomaly detection in federated setup is still growing research area, including FL. Alternative optimization methods presented in Reddi et al. (2020) can be considered in coordination

	Precision	Recall	F1-score
FedAVG	0.92	0.75	0.826
FedRA	0.94	0.76	0.840

Figure 4.3 Evaluation results comparison between FedAVG and FedRA

with network epidemics in the future studies to prevent drawback in federated data distribution.

5. Analysis of Outcomes in Management Information Systems' Perspective

The increasing use of information technologies in the modern world results in a gradual increase in the amount of data circulating in the information systems (Lakhno 2019). This necessitates the urgent establishment of large-scale IS which adapts and analyzes the growing data points (Panaousis et al. 2014). This general problem also indicates the need of IS development on cybersecurity domain in the course of the emergence of numerous new threats that are difficult to detect (Goztepe 2012). Organizations are facing problems of protecting their valuable assets such as; privacy of user information, company specific data or preventing holds in business operations due to adapting issues on upcoming cyber threats. By this way, businesses need to prioritize cyber protection systems to minimize the negative outcomes from cyber threats. Internal or external organizational developments presents a challenge for security-based teams in managing cyber threats (Kleij et al. 2022). Increased interconnectivity and Industry 4.0 revolution causes growing number of cyber attacks and increases threat complexity in implementation. Cyber incident management and mitigation in organizations proceeds with lack of development and intensely focused on technological aspects (Mahdavifar and Ghorbani 2020). Moreover, rapidly changing threat environment yields vast amount of threat indicators, data sources and CTI artefacts which are conjoined in SOCs. SOC operators or security analyst need take an urgent step on comprehending and analyzing these threat sources since current automated threat prevention solutions have lack of availability in action (Kleij et al. 2022). Another aspect is to manage these threat sources to mitigate internally and externally, prioritization of threat types and organizational expenses (Panaousis et al. 2014). From this point of view, organizations need to add intelligent agents or tools to decrease threat response time and financial damages in order to prevent unrecoverable outcomes.

Decision support systems (DSS) are a computerized information systems that help organization in decision-making, managing and planning operations (Lakhno, Petrov, and Petrov 2017). DSS is also supported with a strong knowledge base, such as databases. Additionally, we can list some other benefits , not all but least, of DSSs as (Turban and Watkins 1986):

- improves organizational control
- enhances communication between teams
- decreases problem solving time, which expands time for teams to work on innovative approaches
- can provide novel evidences in business processes

In addition to traditional IT security, cybersecurity is a mutually inclusive domain with organizational management. Identifying newest cyber threats and collaborating with external entities are important to maintain stable and secure cyber environment (Lakhno, Petrov, and Petrov 2017). Threat mitigation and collaborative work can be provided within known DSS infrastructure. In other words, well founded management of cybersecurity domain should be based on intelligent DSS. Plug-in intelligent artefacts or approaches to a system can generally reduces the volume of work done by human operators and also able to achieve desired task in a timely manner. Expert Systems (ES) defined as an application or a computer program which handles specific task in lieu of human expert (Turban and Watkins 1986). Reasoning, interactive data acquisition, solution justification, and modular structure are the four (4) main attributes of ES for solving specific domain problems (Lakhno 2019). An ES should explain outcomes with reasoning even with lack of data and gather data intelligently filtering unnecessary information and stabilizing information gain with normalizing data points. Moreover, an ES should show explainability of results in rationale way, in some cases *If-then* rules are used to refine logic behind (Mahdavifar and Ghorbani 2020). Additionally, modular design of an ES should include a knowledge base, decision making and user interface (Mahdavifar and Ghorbani 2020). ES have successfully implemented in various domains such

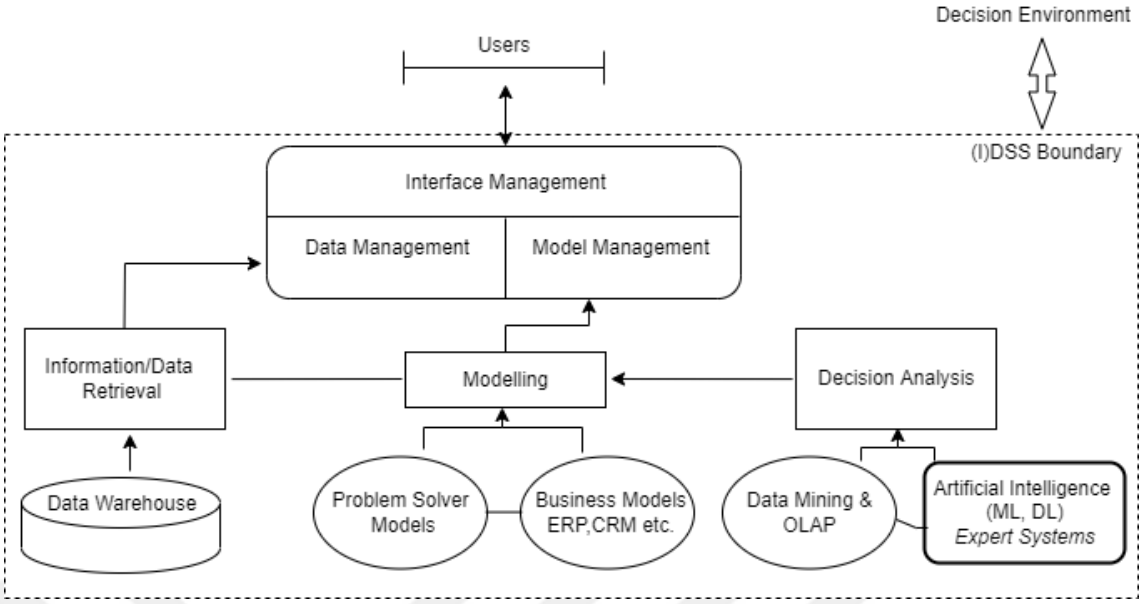

Figure 5.1 Intelligent Multi-perspective DSS Framework (S. Liu et al. 2010)

as diagnosis of heart diseases, diabetes, anemia and most recent works proposing diagnosis of COVID-19 in medical domain, increasing efficiency in yielding crops in agriculture and helping businesses at reaching target audience (Hodhod, Wang, and Khan 2018).

Many studies accept an expert system as an intelligent DSS due to similarities in their definitions (Turban and Watkins 1986). Aside from that, they differentiate in fundamental aspects too. A DSS should be adjustable to dynamic environment of an organization to extend its favors to management (Kleij et al. 2022). On the other hand, an ES has a narrowed view of specifications and domain boundaries (Mahdavifar and Ghorbani 2020). By this view, these information systems are eligible to be integrated to produce better organizational decision-making. DSSs give full permission to control over data acquisition, evaluation and decision-making, but in management aspect, human biases can mislead in complex problems (Turban and Watkins 1986). ES can create a reasoning and evaluation free from biases which procures a synergy in decision made (Mahdavifar and Ghorbani 2020). In figure 5.1, we see an example of integration of an expert system to a DSS.

Cybersecurity professionals need to exhibit strong situation awareness, such as in-

vestigating network logs, monitoring network activities and calculating risks in the long run (Kleij et al. 2022). Cybersecurity teams make use of intelligent tools to analyze internal network activities with existing CTI artefacts and visualize analyzes. Enabling these information systems in cooperation and coordination are crucial for situation awareness and mitigation of threat space. To procure CSA, organizations obtain various cyber intelligent systems and tools, such as Intrusion Detection Systems (IDS), Intrusion Protection Systems (IPS), Security Information and Event Management Systems (SIEM) (Kleij et al. 2022). These intelligent tools produce qualitative analysis reports about network activities, found vulnerabilities, anomalous events, and also provide automatic prevention from cyber attacks. These systems also presents visual elements such as graphs, charts and calculated metrics in order to explain decision making to human experts (Mahdavifar and Ghorbani 2020). Specifically, SIEM tools plays an important part in maintaining CSA. These expert systems can be well-adapted into cybersecurity infrastructure and can have crucial role as a systematic factor in awareness reference model (Fig. 1.3). SIEM tools serve as automated threat identification and present organizational view in the cyber domain. Various implementations and structural differences are discussed in Section 1.1. Moreover, ESs are also in combination with ML and DL approaches and outcomes enhances ability to resolve in desired tasks (Graf, Skopik, and Whitebloom 2016). In general, ES uses written rules based on constructed knowledge base. ML and DL approaches uses knowledge base to learn and understand similar behaviour automatically yield results for a desired task. In this sense, these mechanisms can be combined for intelligent ES (IES). Moreover, IES can adapt dynamic cyber domain in order to ease decision-making process and integrate with DSS to build organizational holistic view. By this way, cybersecurity or cyber incident management teams have a strong CSA.

In this thesis work, we first analyze and investigate the applicability of proposed approaches in an organizational space. To extend this notion, we build our analysis in parallel to maintaining CSA with expert tools, more suitably, a SIEM tool. We observed our research outcomes and found that SIEM tools are important ele-

ment in adaptation of current cyber threat domain. The implementation differences of these ESs enable rapid accommodation to threat space with integrated ML/DL approaches. Both approaches presented in the thesis work are suitable to be implemented into a customized SIEM tool. By this way, the outcomes serve well as an IES which can be integrated into existing DSS to complement organizational management, sustain cyber situation awareness with recent trends and help cybersecurity teams at SOC act better decisive action at detecting anomalous cyber events.

6. Conclusion and Contributions

Defense systems need to adapt new technologies in order to ensure the system's security. Various implementations of security applications with innovative research backgrounds are also necessary for the cyber security domain. Anomaly detection is prominent topic in cybersecurity due to avalanche-like increase in cyber threat space. To enhance protection of organizational services intelligent information systems should be integrated to eliminate human-based decision making processes. Emergence of recent AI-based technologies and approaches presents a vast amount of open questions, especially in cybersecurity domain. Additionally, implementation of tools are bound to existing data sources and types. Text-based data are the most helpful sources in logging and enable cybersecurity teams keep track of historical events and detect abnormal events in the system.

In this thesis, we investigate recent anomaly detection methods based on log contextual information and focus on efficiency of learning models in centralized and federated setup. Increased inter-connectivity of systems and computing power of remote devices expose decentralized learning approaches. Federated learning is the most suitable approach to move learning models to the edge devices. Furthermore, in complex IS infrastructures, prevention of cyber threats is becoming harder day by day. Thus, federated approaches are more adaptable in concern with building rapid cyber incident response tools. Moreover, interdisciplinary topics can be applied to existing AI methodologies to enhance efficiency of desired security tasks.

Transformers is already shown its success in many aforementioned task. Adapters are an additional components to create efficient transfer learning in transformer architecture. Rebuilding parameters are hard to manage considering data storage and its transportation. Adapters can work well with text-based task and provide solutions to these problems. In addition, component-wise integration to transformer

architecture makes transferring knowledge straightforward with adapters. In federated setup, usage of adapters present great reduction in communication cost. Even with a small drop of accuracy, still there is a gap to improve existing approaches. Still, intelligent systems reduce cost of human-power and enhance quality of result with eliminating false-positives in the envisaged task.

Lastly, the contributions of this thesis are;

- We utilize ROBERTa(Yinhan Liu et al. 2019) English language model as a knowledge base, which is a robust version of the BERT architecture. In contrast to related studies, we use Byte-Pair Encoding (Sennrich, Haddow, and Birch 2015) instead of WordPiece (Schuster and Nakajima 2012) in tokenization.
- Instead of a fully fine-tuning model, we have designed language and anomaly adapters for system logs to transfer knowledge without loss of information.
- We experimented on widening the applicability of anomaly detection in the systems. We designed multi-anomaly task detection using a combination of multiple adapters.
- We also presented explainability on our evaluation through gradient-based algorithms and visualized model decisions for investigation of cyber threat data which makes AI-solutions closer to intelligent ES.
- We built a novel risk-adaptive anomaly detection method in federated learning. The proposed architecture is a interdisciplinary work with the influence of Spreading Phenomena.
- We adapted SIS epidemic model to increase convergence speed of federated anomaly detection.
- We provided evidence for integrating proposed methods into Decision Support Systems as an intelligent ES.

Bibliography

Abeshu, Abebe, and Naveen Chilamkurti. 2018. "Deep learning: the frontier for distributed attack detection in fog-to-things computing." *IEEE Communications Magazine* 56 (2): 169–175.

Ahmad, Atif, Sean B Maynard, Kevin C Desouza, James Kotsias, Monica T Whitty, and Richard L Baskerville. 2021. "How can organizations develop situation awareness for incident response: A case study of management practice." *Computers & Security* 101:102122.

Ahmed, Mohiuddin, Adnan Anwar, Abdun Naser Mahmood, Zubair Shah, and Michael J Maher. 2015. "An investigation of performance analysis of anomaly detection techniques for big data in scada systems." *EAI Endorsed Trans. Ind. Networks Intell. Syst.* 2 (3): e5.

Ahmed, Mohiuddin, Abdun Naser Mahmood, and Jiankun Hu. 2016. "A survey of network anomaly detection techniques." *Journal of Network and Computer Applications* 60:19–31.

Alazab, Mamoun, Swarna Priya RM, M Parimala, Praveen Kumar Reddy Maddikunta, Thippa Reddy Gadekallu, and Quoc-Viet Pham. 2021. "Federated Learning for Cybersecurity: Concepts, Challenges, and Future Directions." *IEEE Transactions on Industrial Informatics* 18 (5): 3501–3509.

Albert, Réka, and Albert-László Barabási. 2002. "Statistical mechanics of complex networks." *Reviews of modern physics* 74 (1): 47.

Alves, Pedro Guedes, et al. 2014. "A Distributed Security Event Correlation Platform for SCADA." PhD diss., Universidade de Coimbra.

Beltagy, Iz, Kyle Lo, and Arman Cohan. 2019. "SciBERT: A pretrained language model for scientific text." *arXiv preprint arXiv:1903.10676.*

Bertero, Christophe, Matthieu Roy, Carla Sauvanaud, and Gilles Trédan. 2017. "Experience report: Log mining using natural language processing and application to anomaly detection." In *2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE),* 351–360. IEEE.

Beutel, Daniel J, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, Pedro PB de Gusmão, and Nicholas D Lane. 2020. "Flower: A friendly federated learning research framework." *arXiv preprint arXiv:2007.14390.*

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. "Enriching word vectors with subword information." *Transactions of the association for computational linguistics* 5:135–146.

Bonawitz, Keith, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. "Practical secure aggregation for privacy-preserving machine learning." In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security,* 1175–1191.

Brown, Rebekah, and Robert M Lee. 2019. "The evolution of cyber threat intelligence (CTI): 2019 SANS CTI survey." *SANS Institute. Available online: https://www. sans. org/white-papers/38790/(accessed on 12 July 2021).*

Bryant, Blake D, and Hossein Saiedian. 2017. "A novel kill-chain framework for remote security log analysis with SIEM software." *computers & security* 67:198–210.

Burgot, Romain, Alric Gaurier, Louis Hulot, and Léo Isaac-Dognin. 2020. "Unsupervised methodology to detect anomalies in network communications." In *In Actes de la conférence CAID,* 39.

Caldarelli, Guido, and Michele Catanzaro. 2012. *Networks: A very short introduction.* Vol. 335. Oxford University Press.

Caldarelli, Guido, and Alessandro Chessa. 2016. *Data science and complex networks: real cases studies with Python.* Oxford University Press.

Chalapathy, Raghavendra, and Sanjay Chawla. 2019. "Deep learning for anomaly detection: A survey." *arXiv preprint arXiv:1901.03407.*

Chandola, Varun, Arindam Banerjee, and Vipin Kumar. 2009. "Anomaly detection: A survey." *ACM computing surveys (CSUR)* 41 (3): 1–58.

Chen, Mingqing, Ananda Theertha Suresh, Rajiv Mathews, Adeline Wong, Cyril Allauzen, Françoise Beaufays, and Michael Riley. 2019. "Federated learning of n-gram language models." *arXiv preprint arXiv:1910.03432.*

Chen, Mingzhe, H Vincent Poor, Walid Saad, and Shuguang Cui. 2020. "Convergence time optimization for federated learning over wireless networks." *IEEE Transactions on Wireless Communications* 20 (4): 2457–2471.

Chen, Yiqiang, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. 2020. "Fedhealth: A federated transfer learning framework for wearable healthcare." *IEEE Intelligent Systems* 35 (4): 83–93.

Chen, Zhuo, Na Lv, Pengfei Liu, Yu Fang, Kun Chen, and Wu Pan. 2020. "Intrusion detection for wireless edge networks based on federated learning." *IEEE Access* 8:217463–217472.

Chismon, David, and Martyn Ruks. 2015. "Threat intelligence: Collecting, analysing, evaluating." *MWR InfoSecurity Ltd* 3 (2): 36–42.

Colombo, Armando W, Stamatis Karnouskos, Okyay Kaynak, Yang Shi, and Shen Yin. 2017. "Industrial cyberphysical systems: A backbone of the fourth industrial revolution." *IEEE Industrial Electronics Magazine* 11 (1): 6–16.

Coronavirus disease (COVID-19): How is it transmitted? https://www.who.int/news-room/questions-and-answers/item/coronavirus-disease-covid-19-how-is-it-transmitted. (Accessed on 06/02/2022).

Council, National Research, et al. 2007. *Toward a safer and more secure cyberspace.* National Academies Press.

Dalziel, Henry. 2014. *How to define and build an effective cyber threat intelligence capability.* Syngress.

Denning, Dorothy E. 2000. “Cyberterrorism: The logic bomb versus the truck bomb.” *Global Dialogue* 2 (4): 29.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. “Bert: Pre-training of deep bidirectional transformers for language understanding.” *arXiv preprint arXiv:1810.04805.*

Doshi-Velez, Finale, and Been Kim. 2017. “Towards a rigorous science of interpretable machine learning.” *arXiv preprint arXiv:1702.08608.*

Draper-Gil, Gerard, Arash Habibi Lashkari, Mohammad Saiful Islam Mamun, and Ali A Ghorbani. 2016. “Characterization of encrypted and vpn traffic using time-related.” In *Proceedings of the 2nd international conference on information systems security and privacy (ICISSP),* 407–414.

Du, Min, and Feifei Li. 2016. “Spell: Streaming parsing of system event logs.” In *2016 IEEE 16th International Conference on Data Mining (ICDM),* 859–864. IEEE.

Du, Min, Feifei Li, Guineng Zheng, and Vivek Srikumar. 2017. “Deeplog: Anomaly detection and diagnosis from system logs through deep learning.” In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security,* 1285–1298.

Ehrlicher, Damian. 2020. *Automation In The Cybersecurity World.* https://www.forbes.com/sites/forbestechcouncil/2020/11/05/automation-in-the-cybersecurity-world/?sh=242ee2113137. (Accessed on 06/02/2022).

Endsley, Mica R, and Daniel J Garland. 2000. *Situation awareness analysis and measurement.* CRC Press.

Eswaran, Sivaraman, Aruna Srinivasan, and Prasad Honnavalli. 2021. “A threshold-based, real-time analysis in early detection of endpoint anomalies using SIEM expertise.” *Network Security* 2021 (4): 7–16.

Feng, Shi, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. "Pathologies of neural models make interpretations difficult." *arXiv preprint arXiv:1804.07781.*

Frazão, Ivo, Pedro Henriques Abreu, Tiago Cruz, Hélder Araújo, and Paulo Simões. 2018. "Denial of service attacks: Detecting the frailties of machine learning algorithms in the classification process." In *International Conference on Critical Information Infrastructures Security,* 230–235. Springer.

Fu, Qiang, Jian-Guang Lou, Qingwei Lin, Rui Ding, Dongmei Zhang, and Tao Xie. 2013. "Contextual analysis of program logs for understanding system behaviors." In *2013 10th Working Conference on Mining Software Repositories (MSR),* 397–400. IEEE.

Goztepe, Kerim. 2012. "Designing fuzzy rule based expert system for cyber security." *International Journal of Information Security Science* 1 (1): 13–19.

Graf, Roman, Florian Skopik, and Kenny Whitebloom. 2016. "A decision support model for situational awareness in national cyber operations centers." In *2016 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (CyberSA),* 1–6. IEEE.

He, Pinjia, Jieming Zhu, Shilin He, Jian Li, and Michael R Lyu. 2016. "An evaluation study on log parsing and its use in log mining." In *2016 46th annual IEEE/IFIP international conference on dependable systems and networks (DSN),* 654–661. IEEE.

———. 2017. "Towards automated log parsing for large-scale log data analysis." *IEEE Transactions on Dependable and Secure Computing* 15 (6): 931–944.

He, Pinjia, Jieming Zhu, Zibin Zheng, and Michael R Lyu. 2017. "Drain: An online log parsing approach with fixed depth tree." In *2017 IEEE international conference on web services (ICWS),* 33–40. IEEE.

He, Shilin, Jieming Zhu, Pinjia He, and Michael R Lyu. 2016. "Experience report: System log analysis for anomaly detection." In *2016 IEEE 27th international symposium on software reliability engineering (ISSRE),* 207–218. IEEE.

———. 2020. "Loghub: a large collection of system log datasets towards automated log analytics." *arXiv preprint arXiv:2008.06448.*

Hevner, Alan R, Salvatore T March, Jinsoo Park, and Sudha Ram. 2004. "Design science in information systems research." *MIS quarterly,* 75–105.

Hindy, Hanan, David Brosset, Ethan Bayne, Amar Seeam, and Xavier Bellekens. 2018. "Improving SIEM for critical SCADA water infrastructures using machine learning." In *Computer Security,* 3–19. Springer.

Hodhod, Rania, Shuangbao Wang, and Shamim Khan. 2018. "Cybersecurity curriculum development using ai and decision support expert system." *International Journal of Computer Theory and Engineering* 10 (4): 111.

Houlsby, Neil, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. "Parameter-efficient transfer learning for NLP." In *International Conference on Machine Learning,* 2790–2799. PMLR.

Hu, Zhengbing, Sergiy Gnatyuk, Oksana Koval, Viktor Gnatyuk, and Serhii Bondarovets. 2017. "Anomaly detection system in secure cloud computing environment." *International Journal of Computer Network and Information Security* 9 (4): 10.

Huang, Shaohan, Yi Liu, Carol Fung, Rong He, Yining Zhao, Hailong Yang, and Zhongzhi Luan. 2020. "Hitanomaly: Hierarchical transformers for anomaly detection in system log." *IEEE Transactions on Network and Service Management* 17 (4): 2064–2076.

Hwoij, Abdalrahman, As' har Khamaiseh, and Mohammad Ababneh. 2021. "SIEM architecture for the internet of things and smart city." In *International Conference on Data Science, E-learning and Information Systems 2021,* 147–152.

ICS-CERT Review. 2016. https://www.cisa.gov/uscert/sites/default/files/Annual_Reports/Year_in_Review_FY2016_Final_S508C.pdf. (Accessed on 06/03/2022).

Jang-Jaccard, Julian, and Surya Nepal. 2014. "A survey of emerging threats in cybersecurity." *Journal of Computer and System Sciences* 80 (5): 973–993.

Jyothsna, VVRPV, Rama Prasad, and K Munivara Prasad. 2011. "A review of anomaly based intrusion detection systems." *International Journal of Computer Applications* 28 (7): 26–35.

Kaspersky. 2021. *Kaspersky Security Bulletin 2020-2021. EU statistics — Securelist.* https://securelist.com/kaspersky-security-bulletin-2020-2021-eu-statistics/102335. (Accessed on 06/02/2022).

Kim, Hyesung, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. 2018. "On-device federated learning via blockchain and its latency analysis." *arXiv preprint arXiv:1808.03949*

Kleij, Rick van der, Jan Maarten Schraagen, Beatrice Cadet, and Heather Young. 2022. "Developing decision support for cybersecurity threat and incident managers." *Computers & Security* 113:102535.

Kosek, Anna Magdalena. 2016. "Contextual anomaly detection for cyber-physical security in smart grids based on an artificial neural network model." In *2016 Joint Workshop on Cyber-Physical Security and Resilience in Smart Grids (CPSR-SG),* 1–6. IEEE.

Lah, Airull Azizi Awang, Rudzidatul Akmam Dziyauddin, and Marwan Hadri Azmi. 2018. "Proposed framework for network lateral movement detection based on user risk scoring in siem." In *2018 2nd International Conference on Telematics and Future Generation Networks (TAFGEN),* 149–154. IEEE.

Lakhno, VA. 2019. "Algorithms for forming a knowledge base for decision support systems in cybersecurity tasks." In *International Conference on Computer Science, Engineering and Education Applications,* 268–278. Springer.

Lakhno, Valeriy, Alexander Petrov, and Anton Petrov. 2017. "Development of a support system for managing the cyber security of information and communication environment of transport." In *International Conference on Information Systems Architecture and Technology,* 113–127. Springer.

Lashkari, Arash Habibi, Gerard Draper-Gil, Mohammad Saiful Islam Mamun, and Ali A Ghorbani. 2017. "Characterization of tor traffic using time based features." In *ICISSp,* 253–262.

Lee, Jinhyuk, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining." *Bioinformatics* 36 (4): 1234–1240.

Lee, Jonghoon, Jonghyun Kim, Ikkyun Kim, and Kijun Han. 2019. "Cyber threat detection based on artificial neural networks using event profiles." *IEEE Access* 7:165607–165626.

Li, Beibei, Yuhao Wu, Jiarui Song, Rongxing Lu, Tao Li, and Liang Zhao. 2020. "DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems." *IEEE Transactions on Industrial Informatics* 17 (8): 5615–5624.

Li, Tian, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. "Federated learning: Challenges, methods, and future directions." *IEEE Signal Processing Magazine* 37 (3): 50–60.

Lif, Patrik, Magdalena Granåsen, and Teodor Sommestad. 2017. "Development and validation of technique to measure cyber situation awareness." In *2017 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA),* 1–8. IEEE.

Lim, Wei Yang Bryan, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. 2020. "Federated learning in mobile edge networks: A comprehensive survey." *IEEE Communications Surveys & Tutorials* 22 (3): 2031–2063.

Lin, Qingwei, Hongyu Zhang, Jian-Guang Lou, Yu Zhang, and Xuewei Chen. 2016. "Log clustering based problem identification for online service systems." In *Proceedings of the 38th International Conference on Software Engineering Companion,* 102–111.

Liu, Shaofeng, Alex HB Duffy, Robert Ian Whitfield, and Iain M Boyle. 2010. "Integration of decision support systems to improve decision support performance." *Knowledge and Information Systems* 22 (3): 261–286.

Liu, Yi, Sahil Garg, Jiangtian Nie, Yang Zhang, Zehui Xiong, Jiawen Kang, and M Shamim Hossain. 2020. "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach." *IEEE Internet of Things Journal* 8 (8): 6348–6358.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:1907.11692.*

Lou, Jian-Guang, Qiang Fu, Shengqi Yang, Jiang Li, and Bin Wu. 2010. "Mining program workflow from interleaved traces." In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining,* 613–622.

Mahdavifar, Samaneh, and Ali A Ghorbani. 2020. "DeNNeS: deep embedded neural network expert system for detecting cyber attacks." *Neural Computing and Applications* 32 (18): 14753–14780.

Malaiya, Ritesh K, Donghwoon Kwon, Jinoh Kim, Sang C Suh, Hyunjoo Kim, and Ikkyun Kim. 2018. "An empirical evaluation of deep learning for network anomaly detection." In *2018 International Conference on Computing, Networking and Communications (ICNC),* 893–898. IEEE.

Mármol, Félix Gómez. 2019. "BSIEM-IoT: A Blockchain-Based and Distributed SIEM for the Internet of Things." In *Applied Cryptography and Network Security Workshops: ACNS 2019 Satellite Workshops, SiMLA, Cloud S&P, AIBlock, and AIoTS, Bogota, Colombia, June 5–7, 2019, Proceedings,* 11605:108. Springer.

Mas'ud, Mohd Zaki, Aslinda Hassan, Wahidah Md Shah, Shekh Faisal Abdul-Latip, Rabiah Ahmad, Aswami Ariffin, and Zahri Yunos. 2021. "A Review of Digital Forensics Framework for Blockchain in Cryptocurrency Technology." In *2021 3rd International Cyber Resilience Conference (CRC),* 1–6. IEEE.

McMahan, Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. "Communication-efficient learning of deep networks from decentralized data." In *Artificial intelligence and statistics,* 1273–1282. PMLR.

McMahan, Brendan, and Daniel Ramage. *Google AI Blog: Federated Learning: Collaborative Machine Learning without Centralized Training Data.* https://ai.googleblog.com/2017/04/federated-learning-collaborative.html. (Accessed on 06/02/2022).

McMahan, H Brendan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2017. "Learning differentially private recurrent language models." *arXiv preprint arXiv:1710.06963.*

Meng, Weibin, Ying Liu, Yichen Zhu, Shenglin Zhang, Dan Pei, Yuqing Liu, Yihao Chen, Ruizhi Zhang, Shimin Tao, Pei Sun, et al. 2019. "LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs." In *IJCAI,* 19:4739–4745. 7.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781.*

Miller, Bill, and Dale Rowe. 2012. "A survey SCADA of and critical infrastructure incidents." In *Proceedings of the 1st Annual conference on Research in information technology,* 51–56.

Mokalled, Hassan, Rosario Catelli, Valentina Casola, Daniele Debertol, Ermete Meda, and Rodolfo Zunino. 2019. "The applicability of a siem solution: Requirements and evaluation." In *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE),* 132–137. IEEE.

Mothukuri, Viraaji, Prachi Khare, Reza M Parizi, Seyedamin Pouriyeh, Ali Dehghantanha, and Gautam Srivastava. 2021. "Federated learning-based anomaly detection for IoT security attacks." *IEEE Internet of Things Journal.*

Moukafih, Nabil, Ghizlane Orhanou, and Said El Hajji. 2020. "Neural network-based voting system with high capacity and low computation for intrusion detection in SIEM/IDS systems." *Security and Communication Networks* 2020.

Mulyadi, Ferdy, Leela Aditya Annam, Ridnarong Promya, and Chalermpol Charnsripinyo. 2020. "Implementing Dockerized Elastic Stack for Security Information and Event Management." In *2020-5th International Conference on Information Technology (InCIT),* 243–248. IEEE.

Mustard, Steve. 2005. "Security of distributed control systems: The concern increases." *Computing and Control Engineering* 16 (6): 19–25.

Muthuraj, S, M Sethumadhavan, PP Amritha, and R Santhya. 2020. "Detection and Prevention of Attacks on Active Directory Using SIEM." In *International Conference on Information and Communication Technology for Intelligent Systems,* 533–541. Springer.

Nabil, Moukafih, Sabir Soukainat, Abdelmajid Lakbabi, and Orhanou Ghizlane. 2017. "SIEM selection criteria for an efficient contextual security." In *2017 International Symposium on Networks, Computers and Communications (ISNCC),* 1–6. IEEE.

Nedelkoski, Sasho, Jasmin Bogatinovski, Alexander Acker, Jorge Cardoso, and Odej Kao. 2020. "Self-attentive classification-based anomaly detection in unstructured logs." In *2020 IEEE International Conference on Data Mining (ICDM),* 1196–1201. IEEE.

Newman, Mark EJ. 2002. "Spread of epidemic disease on networks." *Physical review E* 66 (1): 016128.

Nguyen, Khoi Khac, Dinh Thai Hoang, Dusit Niyato, Ping Wang, Diep Nguyen, and Eryk Dutkiewicz. 2018. "Cyberattack detection in mobile cloud computing: A deep learning approach." In *2018 IEEE wireless communications and networking conference (WCNC),* 1–6. IEEE.

Nguyen, Kim Anh, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. "Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction." *arXiv preprint arXiv:1605.07766.*

Nicholson, Andrew, Stuart Webber, Shaun Dyer, Tanuja Patel, and Helge Janicke. 2012. "SCADA security in the light of Cyber-Warfare." *Computers & Security* 31 (4): 418–436.

Nilsson, Adrian, Simon Smith, Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. 2018. "A performance evaluation of federated learning algorithms." In *Proceedings of the second workshop on distributed infrastructures for deep learning,* 1–8.

Nvidia. 2022. *Train With Mixed Precision :: NVIDIA Deep Learning Performance Documentation.* https://docs.nvidia.com/deeplearning/performance/mixed-precision-training/index.html. (Accessed on 06/02/2022).

Onwubiko, Cyril. 2012. "Modelling situation awareness information and system requirements for the mission using goal-oriented task analysis approach." In *Situational awareness in computer network defense: Principles, methods and applications,* 245–262. IGI Global.

———. 2016. "Understanding Cyber Situation Awareness." *Int. J. Cyber Situational Aware.* 1 (1): 11–30.

Panaousis, Emmanouil, Andrew Fielder, Pasquale Malacaria, Chris Hankin, and Fabrizio Smeraldi. 2014. "Cybersecurity games and investments: A decision support approach." In *International Conference on Decision and Game Theory for Security,* 266–286. Springer.

Pande, Amit, and Vishal Ahuja. 2020. *Word embeddings for anomaly classification from event logs.* US Patent 10,530,795.

Panetta, Kasey. 2021. *Gartner Top Security and Risk Trends for 2021.* https://www.gartner.com/smarterwithgartner/gartner-top-security-and-risk-trends-for-2021. (Accessed on 06/02/2022).

Peffers, Ken, Tuure Tuunanen, Marcus A Rothenberger, and Samir Chatterjee. 2007. "A design science research methodology for information systems research." *Journal of management information systems* 24 (3): 45–77.

Pfeffers, K, Tuure Tuunanen, Charles E Gengler, Matti Rossi, Wendy Hui, Ville Virtanen, and Johanna Bragge. 2006. "The design science research process: A model for producing and presenting information systems research." In *Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), Claremont, CA, USA,* 83–106.

Pfeiffer, Jonas, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020. "AdapterFusion: Non-destructive task composition for transfer learning." *arXiv preprint arXiv:2005.00247.*

Pfeiffer, Jonas, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. "Adapterhub: A framework for adapting transformers." *arXiv preprint arXiv:2007.07779.*

Pfeiffer, Jonas, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. "Mad-x: An adapter-based framework for multi-task cross-lingual transfer." *arXiv preprint arXiv:2005.00052.*

Pokhrel, Shiva Raj, and Jinho Choi. 2020. "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges." *IEEE Transactions on Communications* 68 (8): 4734–4746.

Preuveneers, Davy, Vera Rimmer, Ilias Tsingenopoulos, Jan Spooren, Wouter Joosen, and Elisabeth Ilie-Zudor. 2018. "Chained anomaly detection models for federated learning: An intrusion detection case study." *Applied Sciences* 8 (12): 2663.

PV, Rajkumar, and Ravi Sandhu. 2016. "POSTER: security enhanced administrative role based access control models." In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security,* 1802–1804.

Rebuffi, Sylvestre-Alvise, Hakan Bilen, and Andrea Vedaldi. 2017. "Learning multiple visual domains with residual adapters." *Advances in neural information processing systems* 30.

Reddi, Sashank, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečn, Sanjiv Kumar, and H Brendan McMahan. 2020. "Adaptive federated optimization." *arXiv preprint arXiv:2003.00295.*

Rossetti, Giulio, Letizia Milli, Salvatore Rinzivillo, Alina Sırbu, Dino Pedreschi, and Fosca Giannotti. 2018. "NDlib: a python library to model and analyze diffusion processes over complex networks." *International Journal of Data Science and Analytics* 5 (1): 61–79.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams. 1985. *Learning internal representations by error propagation.* Technical report. California Univ San Diego La Jolla Inst for Cognitive Science.

Sadowski, Gorka, Toby Bussa, and Kelly Kavanagh. 2020. *Critical Capabilities for Security Information and Event Management.* https://scadahacker.com/library/Documents/White_Papers/Gartner-Critical-Capabilities-for-SIEM.pdf. (Accessed on 06/02/2022).

Salehkalaibar, Sadaf, and Stefano Rini. 2022. "Lossy Gradient Compression: How Much Accuracy Can One Bit Buy?" *arXiv preprint arXiv:2202.02812.*

Samtani, Sagar, Maggie Abate, Victor Benjamin, and Weifeng Li. 2020. "Cybersecurity as an industry: A cyber threat intelligence perspective." *The Palgrave Handbook of International Cybercrime and Cyberdeviance,* 135–154.

Saunshi, Nikunj, Sadhika Malladi, and Sanjeev Arora. 2020. "A mathematical exploration of why language models help solve downstream tasks." *arXiv preprint arXiv:2010.03648.*

Schuster, Mike, and Kaisuke Nakajima. 2012. "Japanese and korean voice search." In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP),* 5149–5152. IEEE.

Sekharan, S Sandeep, and Kamalanathan Kandasamy. 2017. "Profiling SIEM tools and correlation engines for security analytics." In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET),* 717–721. IEEE.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2015. "Neural machine translation of rare words with subword units." *arXiv preprint arXiv:1508.07909.*

Sharafaldin, Iman, Arash Habibi Lashkari, and Ali A Ghorbani. 2018. "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp* 1:108–116.

Singh, Vivek Kumar, Steven Perez Callupe, and Manimaran Govindarasu. 2019. "Testbed-based evaluation of siem tool for cyber kill chain model in power grid scada system." In *2019 North American Power Symposium (NAPS),* 1–6. IEEE.

Sinha, Koustuv, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. "Masked language modeling and the distributional hypothesis: Order word matters pre-training for little." *arXiv preprint arXiv:2104.06644.*

Smilkov, Daniel, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. "Smoothgrad: removing noise by adding noise." *arXiv preprint arXiv:1706.03825.*

Sornalakshmi, K. 2017. "Detection of DoS attack and zero day threat with SIEM." In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS),* 1–7. IEEE.

Steinwart, Ingo, Don Hush, and Clint Scovel. 2005. "A Classification Framework for Anomaly Detection." *Journal of Machine Learning Research* 6 (2).

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan. 2017. "Axiomatic attribution for deep networks." In *International conference on machine learning,* 3319–3328. PMLR.

Tao, Zeyi, and Qun Li. 2018. "{eSGD}: Communication Efficient Distributed Deep Learning on the Edge." In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18).*

Toneva, Mariya, and Leila Wehbe. 2019. "Interpreting and improving natural-language processing (in machines) with natural language-processing (in the brain)." *Advances in Neural Information Processing Systems* 32.

Tundis, Andrea, Samuel Ruppert, and Max Mühlhäuser. 2020. "On the automated assessment of open-source cyber threat intelligence sources." In *International Conference on Computational Science,* 453–467. Springer.

Turban, Efraim, and Paul R Watkins. 1986. "Integrating expert systems and decision support systems." *Mis Quarterly,* 121–136.

Vasilyev, Vladimir, and Rinat Shamsutdinov. 2020. "Security analysis of wireless sensor networks using SIEM and multi-agent approach." In *2020 Global Smart Industry Conference (GloSIC),* 291–296. IEEE.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. "Attention is all you need." *Advances in neural information processing systems* 30.

Vielberth, Manfred, Florian Menges, and Günther Pernul. 2019. "Human-as-a-security-sensor for harvesting threat intelligence." *Cybersecurity* 2 (1): 1–15.

Weng, Jiasi, Jian Weng, Jilian Zhang, Ming Li, Yue Zhang, and Weiqi Luo. 2019. "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive." *IEEE Transactions on Dependable and Secure Computing* 18 (5): 2438–2455.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. "Transformers: State-of-the-art natural language processing." In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations,* 38–45.

Xu, Feiyu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu. 2019. "Explainable AI: A brief survey on history, research areas, approaches and challenges." In *CCF international conference on natural language processing and Chinese computing,* 563–574. Springer.

Xu, Wei, Ling Huang, Armando Fox, David Patterson, and Michael I Jordan. 2009. "Detecting large-scale system problems by mining console logs." In *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles,* 117–132.

Yu, Xiao, Pallavi Joshi, Jianwu Xu, Guoliang Jin, Hui Zhang, and Guofei Jiang. 2016. "Cloudseer: Workflow monitoring of cloud infrastructures via interleaved logs." *ACM SIGARCH Computer Architecture News* 44 (2): 489–502.

Yuan, Ding, Haohui Mai, Weiwei Xiong, Lin Tan, Yuanyuan Zhou, and Shankar Pasupathy. 2010. "Sherlog: error diagnosis by connecting clues from run-time logs." In *Proceedings of the fifteenth International Conference on Architectural support for programming languages and operating systems,* 143–154.

Zenke, Friedemann, Ben Poole, and Surya Ganguli. 2017. "Continual learning through synaptic intelligence." In *International Conference on Machine Learning,* 3987–3995. PMLR.

Zhang, Tuo, Chaoyang He, Tianhao Ma, Lei Gao, Mark Ma, and Salman Avestimehr. 2021. "Federated learning for internet of things," 413–419.

Zhang, Xu, Yong Xu, Qingwei Lin, Bo Qiao, Hongyu Zhang, Yingnong Dang, Chunyu Xie, Xinsheng Yang, Qian Cheng, Ze Li, et al. 2019. "Robust log-based anomaly detection on unstable log data." In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering,* 807–817.

Zhao, Ying, Junjun Chen, Qianling Guo, Jian Teng, and Di Wu. 2020. "Network anomaly detection using federated learning and transfer learning." In *International Conference on Security and Privacy in Digital Economy,* 219–231. Springer.

Zhao, Ying, Junjun Chen, Di Wu, Jian Teng, and Shui Yu. 2019. "Multi-task network anomaly detection using federated learning." In *Proceedings of the tenth international symposium on information and communication technology,* 273–279.

Zhu, Ziyun, and Tudor Dumitras. 2018. "Chainsmith: Automatically learning the semantics of malicious campaigns by mining threat intelligence reports." In *2018 IEEE European Symposium on Security and Privacy (EuroS&P),* 458–472. IEEE.

Zhuang, Peng, Talha Zamir, and Hao Liang. 2020. "Blockchain for cybersecurity in smart grid: A comprehensive survey." *IEEE Transactions on Industrial Informatics* 17 (1): 3–19.

APPENDIX A: Appendix A

A.1 Utilized adapter architectures

The base adapter structure includes a residual connection, a reduction factor (2,8,16,64) which is the bottleneck that makes able to down and up projections and a non-linearity layer (ReLU, LeakyReLU, Swish) (Pfeiffer, Kamath, et al. 2020), see A.1. This form of a base adapter is used in both LLA and LAA setups. Adapter structure variations and possible implementations are presented in Pfeiffer's work (Pfeiffer, Kamath, et al. 2020).

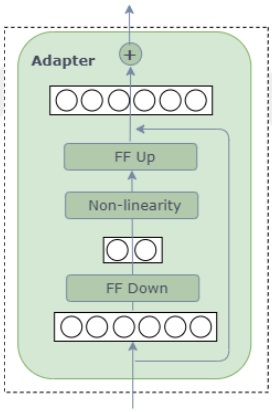

Figure A.1 Base adapter structure (Houlsby et al. 2019).

We presented LLA and LAA stack for complete view of anomaly detection infrastructure inside the transformer block. The type of an adapter structure, implemented for LAA, is shown in Figure A.2. In this architecture, the base adapter is added twice for each transformer block of ROBERTa model. One adapter is after multi-head attention and other adapter is added after feed-forward layer (Houlsby et al. 2019). For simplicity, we omitted the lower stack on LAA implementation in Section 3.3.2.

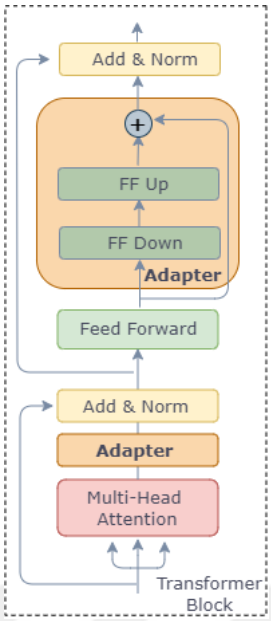

Figure A.2 Log anomaly adapter detailed implementation inside transformer block (Houlsby et al. 2019).

A.2 Training Configurations

In the training, ROBERTa pretrained language model is selected as a knowledge base which is transferred during adaptations. The model architecture's configuration is 12 transformer blocks, a hidden size of 768 and a vocabulary size of 50264 subword tokens. It generates approximately 120M parameters at start of the learning process and also, those are shared among adapter-tuning.

For the LLA training, we used the setup in Figure 3.5 with a reduction factor of 16 and ReLU as a non-linearity function. We have trained 3 epochs in MLM training objective. Same procedure applied for both Firewall and HDFS datasets. For the LAA training, we combined language and anomaly adapters as explained in Section 3.3.2. To achieve that, we used the setup in Figure A.2 with a reduction factor 16 and a non-linearity using Swish function. Differently, LAA does not have layer norm at the bottom. We have trained 3 epochs in binary classification objective. Same procedure is applied for both Firewall and HDFS datasets. For multi-anomaly task detection's training, we only optimized attention-based adapter selection module for one epoch using combination of Firewall and HDFS dataset.

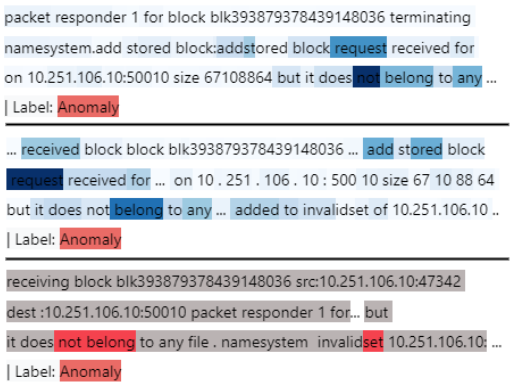

Figure A.3 Multi AAs decision on HDFS logs by Integrated/Smooth Gradients and Input Reduction methods.

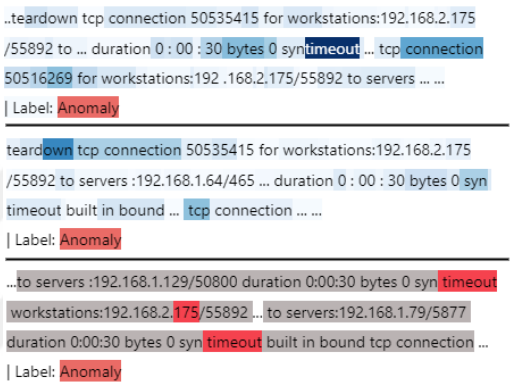

Figure A.4 Multi AAs decision making on Firewall logs by Integrated/Smooth Gradients and Input Reduction methods.

In all training phases, we implemented an early stopping criteria for controlling degradation in the F1-score and evaluated models in step-wise to prevent over-fitting.

A.3 Explainability: Multi-Anomaly Task Detection

Multi-anomaly task detection model fuses various AAs' architectures together. In Figure A.3 and A.4, we can interpret that different model decision mechanism is protected overall. We observe that the base model can be adapted to respond finding anomalies from different sources.

APPENDIX B: Appendix B

B.1 Network Topology

Real life example of a network topology is presented In Figure B.1. The network consist of one root node, three participating node and 19 non-attentive nodes. We assume that these node have less computing power and lack of data to participate in training.

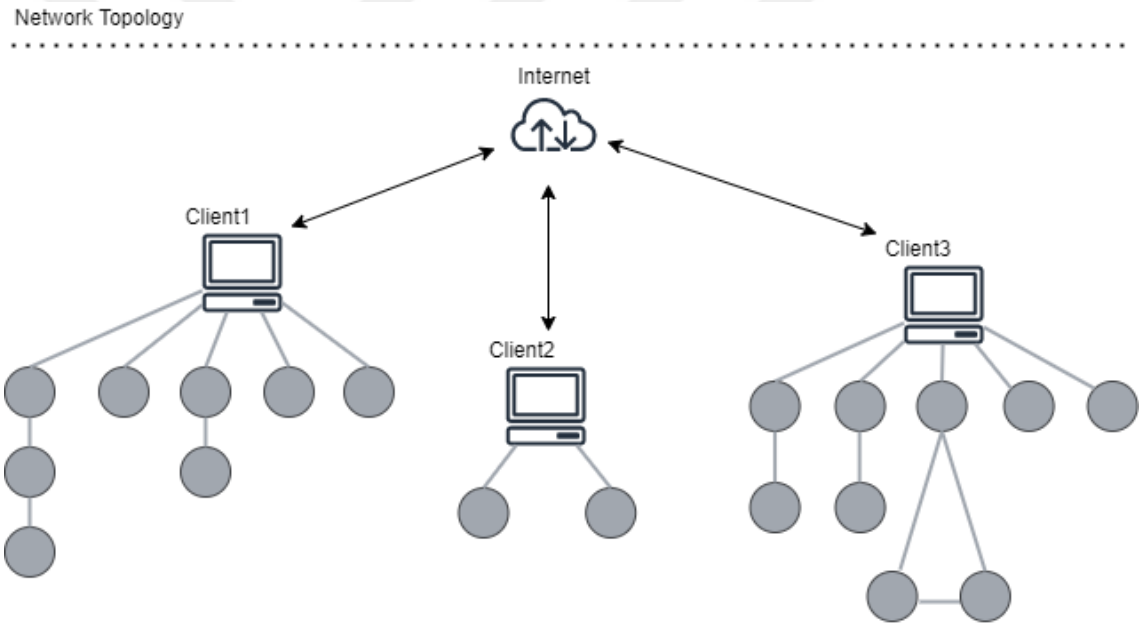

Figure B.1 Example network topology for simulation

CURRICULUM VITAE

Personal Information

Name Surname	:	UĞUR ÜNAL
Bachelor's Degree	:	Computer Engineering, Koç University Istanbul, Turkey 2006-2012
Master's Degree	:	Advanced Computing (Business Systems), Brunel University London, United Kingdom 2012-2014

Publications and Presentations Derived from the Thesis

Ünal, Uğur, and Hasan Dağ. 2022. "AnomalyAdapters: Parameter-efficient Multi-Anomaly Task Detection." *IEEE Access.*

Ünal, Uğur, Ceyda Nur Kahya, Yaprak Kurtlutepe, and Hasan Dağ. 2021. "Investigation of Cyber Situation Awareness via SIEM tools: a constructive review." In *2021 6th International Conference on Computer Science and Engineering (UBMK),* 676–681. IEEE.