International Journal of Safety and Security Engineering
Vol. 14, No. 4, August, 2024, pp. 1293-1303
Journal homepage: http://iieta.org/journals/ijsse

Evaluation of Elasticsearch Ecosystem Including Machine Learning Capabilities

Deepali D. Ahir[1,2*], Nuzhat F. Shaikh[2]

[1] Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, S.P. Pune University, Pune 411041, India
[2] Department of Computer Engineering, MES Wadia College of Engineering, Pune, S.P. Pune University, Pune 411001, India

Corresponding Author Email: deepaliahir@gmail.com

https://doi.org/10.18280/ijsse.140426

Received: 9 June 2024
Revised: 5 August 2024
Accepted: 19 August 2024
Available online: 30 August 2024

Keywords:
Elasticsearch, Logstash, Filebeat, Kibana, machine learning, anomaly detection, data visualization, data analysis

ABSTRACT

Earlier methods of signature based threat detection are no longer enough to detect threats. Fencing your network and access is also ineffective in stopping malicious actors because the IT infrastructure is continuously being shifted to the cloud. Continuous data collection, monitoring and watching out for any malicious behaviors can detect zero day or unknown threats as well. This paper focuses on one of the most important and widely used collections of such tools which are built around Elasticsearch (ES). This paper explains Elasticsearch and its ecosystem of tools like Filebeat and Kibana. A test bed is set up consisting of Apache Web server, Elasticsearch, Filebeat and Kibana. Also the machine learning (ML) capabilities of Elasticsearch are demonstrated with manually injected anomalies in the metric data collected for the web server.

1. INTRODUCTION

Data collection is essential in many areas for effective business decision-making, quality assurance, research integrity, and cyber security as well. One of the most important things is that the data should be collected in the right way. A well-structured and efficient data collection process is important to get the right outcomes. Collecting data is the most comprehensive part of an information security risk assessment process [1]. Earlier security tools were signature-based but as the threat actors are getting sophisticated, the tools relying on signatures are rendered useless. New-age security tools must continuously monitor activities to detect deviation from usual behavior thereby catching the early indicators of an attempted compromise [2]. With the right data, these tools help security teams to catch zero-day attacks [3], and hence collecting the right data, in the correct schema, and removing noise from data becomes paramount.

Having the right data is one part of the puzzle because even if we have gathered the data, extracting any insight requires considerable effort. As security tools need all-round visibility, they need to collect a lot of data, and analyzing large data sets like these needs very efficient storage and querying capabilities. Analyzing data is a component that requires a significant amount of time and that too working with massive quantities of data, especially text, demands a considerable amount of effort [4]. At times, data needs to be preprocessed to optimize the storage, it may also need to be transformed and aggregated to make searching more efficient. Data analysis yields insights and these insights can be consumed either in textual, tabular, or visual form. Large datasets are easier to understand in visual form. Different charts, graphs, and images help us understand trends and insights in a much more intuitive way than churning numbers in tables or text.

There has been a lot of work already done in the field of data collection, processing, and visualization, both by researchers and commercial developers. As a result, there are many open-source as well as commercial tools available for data collection, analysis, and visualization. Various file formats and storage optimizations are developed to store data efficiently, whereas there has been the advent of ETL (extract, transform, and load) tools, query engines, and AI/ML frameworks to aid the analysis of the data. Numerous visualization frameworks and dashboarding tools are created to aid researchers in visualizing the data. One such tool, which has an entire ecosystem for all the data collection, analysis and visualization needs is Elasticsearch. The Elasticsearch ecosystem contains Elasticsearch (data storage and analysis), Logstash, Filebeat (Data collection), and Kibana (Visualization) [5-7].

In this paper, we have described and demonstrated the use of ELK (Elasticsearch, Logstash, Kibana) and Filebeat ecosystem to collect, analyze, and visualize data from web servers. Also demonstrated the use of built in machine learning capabilities of Elasticsearch. The rest of the paper is organized as: section 2 explains related work, section 3 presents a data collection ecosystem. Section 4 precisely explains data analysis. Section 5 describes data visualization in detail. Section 6 explains the experiment and results about machine learning features of Elasticsearch. Finally, section 7 ends with a concise overview of findings and a discourse.

2. RELATED WORK

The primary aim of this research is to describe and demonstrate the Elasticsearch ecosystem. This section will explore how Elasticsearch has been used in earlier research. Each approach detailed in the literature fulfills a distinct use case. Taylor et al. [4] have used indexing and searching capabilities of Elasticsearch while leaving machine learning for future work. Tsung et al. [5] applied Elastic Stack to improve effectiveness of searching and analyzing traffic control data in their research, they also did not exploit the full potential of Elasticsearch's machine learning capabilities. Papadimitriou et al. [7] have used ELK stack to ingest heterogeneous data and created Kibana dashboards to visualize this data and identify anomalies. They have also used a Logstash plugin to pull data from rabbitMQ. Zamfir et al. [8] have proposed the monitoring system based on Logstash, Elasticsearch and Kibana. Their goal was to check whether ELK stack satisfies the technological requirement of such a system, which they conclude in resounding yes. This system was not put to the test though as it was left for future work along with anomaly detection using machine learning capabilities. Elasticsearch was put to test in terms of read and write performance by Ansari et al. [9]. They compared Elasticsearch, Cassandra, MongoDB, Hbase and found Cassandra to be the best among all. However, this was purely based on a performance criterion of read and write operations while ignoring the overall advantage of the large ecosystem of Elasticsearch. Gutiérrez and Pérez Vera [10] integrated Elasticsearch with Big Query using Pub/Sub and Cloud functions. While doing so they highlighted the ability of Elasticsearch to handle large volumes of data and visualize it using Kibana dashboards. Hamilton et al. [11] used ingestion, aggregation and visualization tools of ELK stack to handle data from industrial control applications at CERN. Authors deferred the use of machine learning and alerting framework for the future work. Calderon et al. [12] demonstrated the deployment and assessment of an IoT platform by utilizing Elastic Stack and Apache Kafka for overseeing and supervising IoT networks. They also used Kibana for dashboarding. In the study by Shah et al. [13], the author presents insights into the standardization and configuration of Elasticsearch processes, which contribute to improved analytical efficiency. By ensuring a proper configuration of Elasticsearch and Kibana, organizations can achieve efficient real-time analysis of large volumes of data, allowing policymakers to access results immediately in a user-friendly format that aids in decision-making. Kononenko et al. [14] noted that the shift to Elasticsearch resulted in a substantial increase in performance, which rendered their tool ideal for real-time operations. In the study by Takaki et al. [15], authors have used ELK stack to create a log management system which anonymizes the sensitive data. ELK has been used by authors [16] as it is open source, easy to deploy and use. It satisfied their requirements as a data store, visualization and analysis tool.

3. DATA COLLECTION ECOSYSTEM

Data consists of facts, symbols, events, figures, and objects accumulated from diverse channels. Organizations rely on data to make informed decisions, which is why data is gathered from diverse sources at different intervals. To enhance decision-making, organizations gather data through a range of different data collection methods. While data holds significant value for organizations, it remains inactive until it is analyzed or processed to attain the desired outcomes.

The digital revolution has led to a significant increase in the amount of data generated in recent years. IDC's projections indicate that the Global Datasphere is projected to grow from 33 Zettabytes (ZB) in 2018 to 175 ZB by the year 2025 [17]. Cyber attackers are consistently drawn to data as their primary objective, making it the most enticing target in almost every instance of a cyber-attack. The incorporation of advanced data analytics, encompassing artificial intelligence (AI), statistical visualization, machine learning, and many similar tools, can expose companies to potential harm, as these techniques can be used to exploit their data. By analyzing modern and historical attack data, businesses can forecast upcoming attacks and take proactive measures to reduce the associated security threats. To uncover advanced details of different cyber-attacks, cyber security tools require data from multiple origins. By leveraging big data analytics, corporations are empowered to identify potential threats and attack patterns by thoroughly analyzing the data associated with the activities leading up to the attack.

The sheer volume of data being generated by today's businesses exceeds the capabilities of traditional data processing applications and databases. Efficient tools have been devised to handle the storage and processing of big data, effectively meeting the challenges and scaling as required. Among the tools available, there is Elasticsearch, a distributed search and analytics engine that serves its purpose effectively [8].

3.1 Elasticsearch

It is an open-source search engine, based on the Apache Lucene™ full-text search engine library. Elasticsearch (ES) serves as a clustered no-SQL data repository that is specifically utilized for housing vast amounts of data, and it offers the flexibility to expand horizontally to accommodate varying demands [6]. It is equipped with a built-in query language and provides AI/ML capabilities. Elasticsearch enables swift data analysis, facilitating the rapid execution of intricate statistical calculations on extensive datasets. The query API offered by ES is exceptionally adaptable, providing comprehensive support for sorting, filtering, aggregations, and pagination within a single query [9]. It will attempt to determine the field mappings, and will automatically add or remove new or existing fields. The automatic handling of unstructured data by Elasticsearch permits the indexing of JSON documents without the necessity of defining a schema in advance [9]. By refreshing the index every second as a default setting, Elasticsearch is capable of executing near real-time searches. Data can be searched by users using a specialized and adaptable search language called "Query DSL", which serves as a simplified version of the Lucene query syntax for user convenience [10]. Through its aggregation functionality, Elasticsearch is proficient in conducting complex analytics on the stored data [6]. As a member of the Elastic Stack, Elasticsearch is a crucial element within a suite of analytics and visualization tools that are purposefully built to effortlessly integrate. These tools are primarily employed to enhance the fundamental functionality of Elasticsearch, catering to the specific needs of the users. The subsequent subsections provide a brief description of

other tools within the stack that are applicable to this research.

3.2 Logstash

Logstash serves as a server-side data processing pipeline that is free and open-source. It is a component of ELK stack. Logstash effectively gathers data from multiple sources, applies necessary transformations, and subsequently transfers it to your desired storage destination [11, 18]. It is common for data to be fragmented or segregated across multiple systems in diverse formats. Logstash is equipped with the capability to handle diverse inputs, enabling the ingestion of events from a multitude of commonly used sources concurrently. Conveniently collect data from your logs, metrics, web applications, data stores, and multiple AWS services in a continuous, streaming format [18]. Logstash operates by deploying a shipper, which is a small agent installed on each monitored host. The shipper, in this case, is an instance of Logstash that has been appropriately configured to collect inputs from various sources such as stdin, stderr, and log files. These inputs are then transmitted to an indexer using ActiveMQ (active message queue). Within the Logstash framework, an indexer is an index that has been configured to efficiently parse, filter, and route logs and events that are received through AMQ [19].

Logstash filters play a crucial role in the journey of data from its source to storage. These filters meticulously parse each event, recognizing specific fields to construct a well-defined structure [18, 19]. By transforming the data into a unified format, Logstash enables more robust analysis and enhances the business value derived from it. It also offers a range of output options, allowing you to direct data to your desired destination and providing the versatility to enable numerous downstream applications. This way Logstash proves to be a vital tool in the data collection ecosystem of Elasticsearch.

3.3 Filebeat

Efficient log data management is a critical component of every contemporary organization. Given the rising volume of data produced from various origins, it is imperative to possess the appropriate tool for gathering, processing, and examining log data. Among the well-known open-source tools available for log data management, Filebeat stands out as a prominent choice. Developed by Elastic, Filebeat functions as a lightweight shipper, enabling the seamless forwarding and centralization of log data. The primary function of Filebeat is to acquire log data from diverse origins, such as files, Syslog, and third-party systems, and forward it to a preconfigured output location, such as Elasticsearch, Logstash, or Kafka [11].

By monitoring the log files or specified locations, Filebeat diligently collects log events and seamlessly transfers them to Elasticsearch, where they are indexed for further analysis and processing [12, 18]. Filebeat comes equipped with modules designed to streamline the process of collecting, parsing, and visualizing observability and security data from various log formats. With just a single command, you can effortlessly manage these data sources. Filebeat seamlessly integrates with the ELK stack, providing a user-friendly and efficient solution. This enables you to effectively handle your log files, ensuring fast and scalable log management while preserving data integrity.

Filebeat modules come with pre-configured settings for popular log formats like Apache, nginx, and MySQL logs. We can utilize them to streamline the configuration of Filebeat, parse the data, and analyze it in Kibana using pre-built dashboards. File beat modules consist of predefined file sets, including access logs and error logs. To find the different configurations for each module, navigate to the /etc/filebeat/module.d folder. This folder is available on both Linux and Mac operating systems. Modules must be enabled since they are initially disabled.

Filebeat's modular design empowers users to quickly and effortlessly onboard new data sources, whether it's web server logs, cloud infrastructure metrics, or specialized application-specific logs, without the need for extensive custom configurations or complex scripting. The versatility of Filebeat modules is further enhanced by their seamless integration with the broader Elastic Stack ecosystem, allowing for seamless data processing, enrichment, and visualization within the Elasticsearch, Logstash, and Kibana components. Moreover, Filebeat's modules can also be deployed in docker environments bringing in the telemetry even from your docker ecosystem [20].

The comprehensive list of Filebeat modules comprises Elasticsearch, coredns, apache, cisco, nginx, cef, mysql, aws, auditd, envoyproxy, zeek, googlecloud, traefik, haproxy, suricata, icinga, santa, ibmmq, redis, iptables, rabbitmq, iis, postgresql, kafka, panw, kibana, osquery, logstash mongodb, netflow, mssql and nats [21]. There are multiple methods to activate modules, with one approach being running a Filebeat command to enable specific modules. Another way to enable modules is to add a config file for that module under modules.d directory. By default, there are sample config files for all the modules under this modules.d directory with .disabled as extension. We can rename this apache.yml. disabled file to apache.yml and make necessary changes in the config to enable the Apache module. Figure 1 depicts the Elasticsearch ecosystem including Logstash and Filebeat along with Filebeat modules.

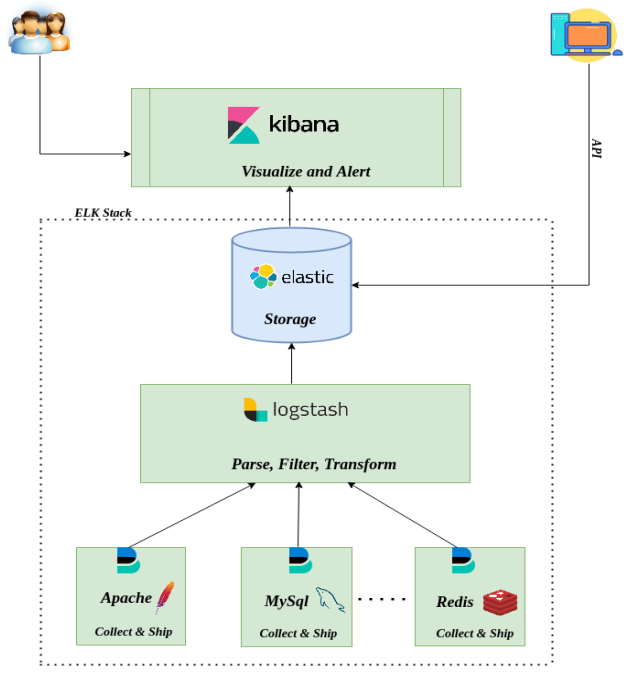

Figure 1. Elasticsearch ecosystem

4. DATA ANALYSIS

Data analysis involves examining, purifying, converting, and structuring data to uncover valuable information and aid in the process of making informed decisions. Analyzing data is essential in the current data-centric environment. Organizations can tap into the immense power of data, by utilizing it effectively, which can empower them to make strategic choices, streamline operations, and gain a competitive edge in the market. Analytics Maturity Models provide a useful framework for understanding the evolution of analytics capabilities within a business, ranging from basic reporting to more advanced predictive and prescriptive analytics [22].

By converting raw data into meaningful insights, data analysis empowers businesses to effectively identify opportunities, mitigate risks, and optimize their overall performance. This valuable process plays a pivotal role in enabling organizations to make informed decisions and drive success. The amount of data which needs to be stored and analyzed has gone up significantly in recent times, which has resulted in convergence in database/storage technologies and analytical capabilities like machine learning and artificial intelligence [23]. It is also important to note that data analysis is helpful when the user or the organization has a good understanding of data [23].

4.1 Built-in analysis in Elasticsearch

Elasticsearch serves as a distributed, RESTful search and analytics engine with the ability to handle a wide range of use cases. At the core of the Elastic Stack lies its ability to centrally store the data, facilitating swift search capabilities, highly tailored relevance, and scalable analytics that effortlessly adapt to your requirements [6-8, 17]. Elasticsearch can efficiently handle large volumes of time-series / real-time data [13, 14]. Data investigation is made more efficient and straightforward with Elasticsearch Query Language (ES|QL). The ES|QL engine enhances search capabilities, resulting in increased efficiency and faster resolution through streamlined workflows [24]. The analysis of data in Elasticsearch involves the utilization of advanced techniques such as custom scoring, machine learning, and aggregation.

4.1.1 Custom scoring

Elasticsearch employs a relevance score to arrange search outcomes according to their compatibility with the query [25]. Nevertheless, it might be necessary to personalize the scoring system to align it more effectively with your particular requirements. One common use case for custom scoring is in large-scale personalized search and recommendation systems [26]. The power of custom scoring in Elasticsearch lies in its ability to adapt to the specific needs of each application and user. By incorporating domain-specific signals, contextual ranking models, and advanced machine learning techniques into the scoring function, developers can create highly targeted and effective search experiences that cater to the unique requirements of their users. For example, in an e-commerce application, the scoring formula could incorporate signals like product price, inventory levels, user reviews, and past purchase history to provide a more relevant and personalized search experience [27]. Furthermore, the power of Elasticsearch's custom scoring capabilities extends to the integration of advanced machine learning techniques, such as neural network-based ranking models, trained on large datasets of user interactions and preferences, to provide even more personalized and accurate search results [27].

There exist numerous approaches to attain custom scoring:

- Scripted Similarity: The concept of scripted similarity in custom scoring has garnered significant attention, as it presents a unique approach to evaluating the quality and originality of written work. Similarity is a crucial aspect in various applications, such as document summarization, question answering, information retrieval, and document clustering and categorization. The effectiveness of a similarity metric may vary based on the specific application domains, such as text or image analysis, the types of feature formats used, like word count or tfidf, and the classification or clustering algorithms employed [28]. Painless, Elasticsearch's scripting language, allows for the creation of a personalized similarity algorithm. This feature proves valuable when implementing ranking strategies tailored to specific domains.
- Function Score Query: Function Score Query allows for the customization of the relevance scoring process by incorporating various factors into the ranking algorithm. It is a versatile tool that enables users to influence the relevance scoring of search results by incorporating various factors, such as distance, freshness, popularity, or custom-defined functions [6]. This feature is particularly useful in scenarios where the standard scoring mechanisms provided by Elasticsearch may not fully capture the nuances of a specific domain or use case [6]. One of the key advantages of the Function Score Query is its ability to fine-tune the relevance of search results based on the specific needs of the application or the user's preferences. By incorporating various scoring factors, developers can create more personalized and relevant search experiences, which is crucial in large-scale personalized search and recommender systems. By utilizing functions like field value factor, decay functions, or custom scripts, you can adjust the relevance score [25]. This adjustment empowers you to amplify or diminish the significance of documents based on specific criteria. Function score query is one of the most powerful features of Elasticsearch because of the extensive customization provided by it [29].

4.1.2 Machine learning

Elasticsearch, the powerful open-source search and analytics engine, has evolved beyond its traditional role as a robust text search solution, now offering a suite of integrated machine learning capabilities that have the potential to transform the way organizations extract insights from their data [30]. These machine learning features, seamlessly integrated into the Elasticsearch ecosystem, empower users to leverage advanced analytics without the need for complex external tools or extensive data engineering efforts. By incorporating machine learning directly into Elasticsearch, organizations can now harness the power of predictive modeling, anomaly detection, and other advanced analytics techniques within their existing data management infrastructure, eliminating the need to maintain separate analytical workflows and streamlining the process of transforming raw data into actionable insights [30]. By leveraging machine learning, Elasticsearch empowers users to identify anomalies, predict trends, and categorize data effectively. Several important characteristics of machine learning are:

- Model Inference: Employ pre-trained machine learning models to predict results on novel datasets.
- Anomaly Detection: Leverage unsupervised machine learning algorithms to identify unusual patterns in your data. This methodology can be highly advantageous in detecting fraudulent activities, monitoring the performance of your system, or pinpointing outliers within your dataset. There are various machine learning algorithms which can be used for anomaly detection, with applications to various fields including cyber security, IOT etc. [31, 32].
- Data Frame Analytics: Execute supervised machine learning operations, including classification and regression, to forecast outcomes or categorize data using historical samples. Classification and regression find its use in many use cases and diverse applications like healthcare [33] and cybersecurity [34].

4.1.3 Aggregation

Elasticsearch, a powerful open-source search and analytics engine, offers a rich set of features for data analysis and visualization, one of which is the powerful aggregation framework. Aggregations in Elasticsearch provide a way to extract insights and metrics from data, allowing users to go beyond simple search and retrieve operations, and instead, gain a deeper understanding of their data by uncovering trends, patterns, and relationships. Aggregations in Elasticsearch are designed to be efficient, scalable, and flexible, enabling developers to perform complex data analysis tasks on large datasets with ease. Elasticsearch's aggregation framework can be used for a wide range of applications, from e-commerce product analytics [6] to scientific research data exploration. Utilizing aggregations in Elasticsearch offers a potent means of examining and summarizing data [13]. These functionalities empower you to categorize and derive statistical insights from your data using specified parameters.

- Metric Aggregations: The metrics being calculated encompass various statistical measures, including the sum, average, and count, for every individual bucket [25].
- Pipeline Aggregations: Additional calculations are carried out on the outcomes of other aggregations.
- Bucket Aggregations: Documents are categorized into buckets according to specific criteria, including terms, ranges, or filters [25].

Elasticsearch presents a broad selection of advanced analytics functionalities that can aid in extracting valuable insights from your data. By utilizing aggregations, machine learning, and customized scoring techniques, one can effectively carry out intricate data analysis tasks and enhance the significance of search outcomes.

4.2 Elasticsearch and third-party tools

Elasticsearch serves as a potent, open-source search and analytics engine that is specifically crafted to manage vast quantities of data in real-time, offering swift, trustworthy search results and valuable insights for diverse applications. The ready-made integrations provided by Elastic streamline the data ingestion and connection to different data sources, making it effortless to store, search, and analyze data from any source in your environment. There are Elasticsearch clients and APIs in various languages that can be used to interface with Elasticsearch. Apart from this, Elasticsearch functionality can be extended by using plugins.

4.2.1 Elasticsearch clients

Most of the applications provide REST APIs to interact with the users, these APIs can be used to configure, send inputs and get outputs from the applications One of the key features that contributes to Elasticsearch's popularity is its ability to provide near real-time search capabilities through a RESTful API, making it accessible to users across various domains. Elasticsearch clients are software libraries that provide a programmatic interface for interacting with the Elasticsearch cluster. These clients abstract away the underlying HTTP protocol, allowing developers to interact with Elasticsearch using the programming language of their choice [6]. Elasticsearch clients simplify the process of indexing, querying, and managing data within the Elasticsearch ecosystem, making it more available to a variety of developers and organizations. The selection of an appropriate Elasticsearch client depends on factors such as the programming language, project requirements, and the level of complexity needed. For instance, the Java High Level REST Client is a popular choice for Java-based applications, as it provides a more object-oriented and user-friendly interface compared to the low-level Java REST Client. Similarly, the Python Elasticsearch Client is a widely-used option for Python developers, offering a Pythonic API and support for advanced features like asynchronous operations. The choice of an Elasticsearch client can also be influenced by the specific needs of the project. Some clients may offer more advanced features, such as support for routing, filtering, and aggregations, while others may prioritize simplicity and ease of use. For example, the Go Elasticsearch Client is known for its lightweight and efficient design, making it a suitable choice for projects with strict performance requirements [6]. In contrast, the .NET Elasticsearch Client provides a rich set of features, including support for LINQ queries and seamless integration with the .NET ecosystem, making it a compelling choice for .NET-based applications [6].

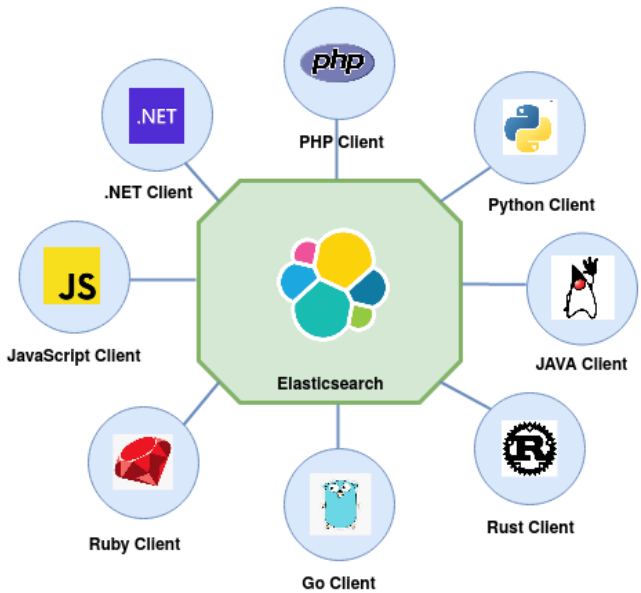

Figure 2. Elasticsearch clients

Elasticsearch provides official clients for multiple languages, ensuring a reliable and convenient solution for users. Java, JavaScript, Perl, PHP, Python, Ruby, and .NET have official Elasticsearch clients available as shown in Figure

2. Additionally, community versions provide support for a wide range of other programming languages.

4.2.2 APIs

Elasticsearch is a database that primarily relies on JSON format [15, 35] and is inclined towards accommodating unstructured data types, distinguishing itself from other databases in the market. It communicates via RESTful APIs and consolidates various datasets from logs, metrics, and application trace data into a central unit system. In a RESTful API, resources are distinguished by URLs, with a variety of HTTP methods being used to execute operations on these resources. For instance, data can be retrieved using a GET request, new data can be created with a POST request, existing data can be updated through a PUT request, and data can be deleted using a DELETE request. Through standard HTTP requests, users are able to engage with the engine using the Elasticsearch RESTful API [16]. Users have the capability to index and search data using this API, in addition to executing different administrative responsibilities like managing indices and nodes, configuring security settings, and monitoring cluster health. Elasticsearch offers a wide range of APIs, including document APIs, Connector APIs, Cluster APIs, and numerous others [36]. The official Elasticsearch documentation offers extensive and thorough documentation of the Elasticsearch API. With Elasticsearch's comprehensive API documentation and an array of tools, developers can effortlessly interact with Elasticsearch programmatically and develop bespoke applications.

4.2.3 Plugins

Plugins fall into two distinct categories: site plugins and code plugins. A site plugin does not add any new functionality; it simply serves a web page through Elasticsearch. Examples of site plugins include elasticsearch-head, elasticsearch-kopf, bigdesk, elasticsearch-hq, and whatson. A code plugin in Elasticsearch is a plugin containing JVM code that Elasticsearch can execute. These plugins can enhance Elasticsearch's functionality, like the AWS plugin for snapshotting indices to Amazon S3 or the ICU analysis plugin for language-specific text analysis [25]. Some plugins even replace internal Elasticsearch components, such as the shard distributor and discovery mechanisms. Among the code plugins available for Elasticsearch, we have elasticsearch-aws and elasticsearch-azure plugins. Additionally, there are elasticsearch-lang-* plugins, such as elasticsearch-lang-python and elasticsearch-lang-ruby, which enable support for different scripting languages. These plugins expand the capabilities of Elasticsearch by introducing various functionalities. Furthermore, there are plugins designed to enhance query capabilities, including additional highlighters and new types of aggregations. By utilizing a .jar file, developers can incorporate any desired functionality into Elasticsearch [25].

4.3 ML frameworks

The Elasticsearch platform is equipped with integrated machine learning and AI features, allowing users to extract valuable insights and find necessary answers from their data using X-Pack [37]. From anomaly detection and supervised learning to vector search and natural language processing, Elasticsearch offers a wide range of machine-learning capabilities. The utilization of AI/ML features in Elasticsearch is simplified through user-friendly wizards, making it accessible even to inexperienced users.

Advantages of applying Elasticsearch machine learning to the data are:

- Seamlessly incorporate machine learning into a scalable and high-performing platform.
- Utilize unsupervised learning techniques along with preconfigured models to detect observability and security concerns without the need to be concerned about training an AI model.
- Utilize practical analytics to detect threats and irregularities in advance, speed up issue resolution, recognize patterns in customer behavior, and enhance your online interactions.

To use ML with Elasticsearch, you have multiple options:

- The most convenient choice is to utilize the preexisting models [38]. These models serve the purpose of identifying particular security risks, aiding in troubleshooting system problems, and handling data in languages other than English. Additionally, there are exclusive models, such as the Elastic Learned Sparse Encoder model, that are readily accessible.
- Users seeking additional or alternative options beyond the default models can utilize third-party PyTorch models available through platforms such as the HuggingFace model hub.
- One more choice is to import the model that you have trained on your own, this assistance is specifically for NLP transformers at the moment.

Elastic offers backing for diverse transformer models and numerous supervised learning libraries. The NLP and embedding models cover all transformers adhering to the standard BERT model interface and employ the WordPiece tokenization algorithm [39, 40]. When it comes to supervised learning scenarios, Elasticsearch is compatible with trained models from scikit-learn [41], XGBoost, and LightGBM libraries. In the case of generative AI requirements, Elasticsearch presents an API for LLM to manage queries, enhance them with context retrieved from Elasticsearch, and evaluate the outcomes.

Given below are the various use cases where Elasticsearch machine learning can be used:

- Anomaly detection to detect issues and threats early
- Root cause analysis
- Fraud detection using classification
- E-commerce product similarity search
- Job recommendation
- Patent search

5. DATA VISUALIZATION

The process of data visualization entails organizing data methodically to enable business users to easily interpret the information. Data visualization enhances the significance of data by weaving it into a compelling narrative. Every dataset holds a unique story waiting to be uncovered through pertinent analysis. By harnessing the power of visualization, one can effectively uncover hidden data patterns, swiftly extract meaningful information, and successfully address all decision-making dilemmas. Wide range of data visualization options, including charts, graphs, and dashboards, that can be customized to suit the unique needs of different organizations.

For individuals requiring immediate operational decisions, real-time visualization can be an invaluable tool. A range of alternatives exist for visualizing data, including Kibana, EMR notebooks, Elasticvue, Amazon QuickSight, and others [42, 43].

5.1 Kibana

Within the ELK stack, Elasticsearch is the backend which stores data and supports fast querying [44], while Kibana stands as a potent platform for visualization and querying, serving as the primary visual component [42, 43]. Kibana enables the visualization of data that is stored in an Elasticsearch cluster [20]. This encompasses a wide array of features, ranging from executing spontaneous queries, and generating visualizations like line charts and pie charts, to displaying data on dynamic dashboards. Kibana allows for seamless interaction with data, offering a superior experience compared to manually crafting Elasticsearch queries. Manipulating data is straightforward, and transitioning between various datasets can be achieved while maintaining context. Consequently, Kibana proves to be a valuable instrument for conducting data analysis, exploration, and investigation [43]. Dashboards play a pivotal role by empowering individuals and teams to access comprehensive data summaries. Kibana is a widely used tool for monitoring data, particularly in the field of observability. By utilizing Kibana and the Elastic Stack for observability, one can gain valuable insights into application performance, monitor service uptime, keep track of hardware and service utilization, and more. Additionally, Kibana is commonly employed for security analysis and the management of machine learning tasks.

5.2 Other user interface tools

While Kibana seamlessly integrates with Elasticsearch, it does not offer support for integrating with alternative data sources. Hence, the sole purpose of its usage is to visualize Elasticsearch data exclusively. In the present era, numerous organizations manage data from various sources such as NoSQL databases, SQL databases, REST APIs, and more. Consequently, the limitation of Kibana solely functioning with Elasticsearch might pose a challenge. Numerous alternatives exist for Kibana, such as Grafana, Splunk, and Knowi.

5.2.1 Grafana

An open-source data visualization tool called Grafana offers the capability to visualize diverse datasets [45]. It is commonly employed alongside InfluxDB, Graphite, and Elasticsearch. Grafana offers a range of visualization options that users can leverage to represent their data effectively. Users of Grafana can develop their plugins and seamlessly integrate them with diverse data sources. This software is particularly useful for time series analytics, providing users with the capability to explore, analyze, and monitor data trends over specific timeframes [46]. Through customization features, users can tailor data visualization to their liking by adjusting colors, sizes, labels, and other elements. The integration of Elasticsearch and Grafana provides a robust and scalable platform for monitoring and analyzing diverse data sources, from infrastructure metrics to application logs and beyond. This powerful duo has found widespread adoption in various industries, including IT operations, DevOps, and scientific research, where the need for efficient data management and visualization is paramount.

5.2.2 Knowi

The ability of BI systems to initiate problem articulation and dialogue, as well as facilitate data selection, is crucial in supporting organizational knowledge [47]. By enabling users to explore data, identify patterns, and engage in collaborative analysis, dashboarding tools empower decision-makers to make more informed choices; one such important tool is Knowi. Knowi is a cutting-edge data analytics platform that seamlessly integrates with Elasticsearch and various other data sources such as SQL, REST-API, and NoSQL. By utilizing data virtualization, it can establish real-time connections with any data source, eliminating the need for time-consuming ETL processes. With this advanced feature, users can effortlessly connect to their Elasticsearch indexes and perform efficient analytics on the data [48]. Knowi offers pre-built user management functionalities. It enables you to effortlessly create roles and users, and subsequently assign them permissions to access your dashboards. This feature empowers you to effectively regulate the individuals who can view your data and dashboards.

5.2.3 OpenSearch dashboards

OpenSearch Dashboards is a powerful and versatile data visualization and exploration platform that has become an increasingly essential tool for organizations seeking to gain meaningful insights from their data. Designed to seamlessly integrate with the OpenSearch ecosystem, this platform offers a wide range of features and functionalities that cater to the diverse needs of data analysts, developers, and decision-makers. One of the key strengths of OpenSearch Dashboards lies in its ability to transform complex data into intuitive and visually appealing dashboards. These dashboards empower users to interact with their data, uncover hidden patterns, and make informed decisions in record time.

OpenSearch Dashboards stands as the primary visualization solution within the OpenSearch ecosystem [49]. OpenSearch, an open-source search and analytics suite, is a collaborative endeavor that builds upon Elasticsearch and Kibana. By utilizing OpenSearch, users gain the ability to efficiently ingest, search, aggregate, visualize, and analyze data, catering to various needs such as log analysis. OpenSearch Dashboards further enhance the capabilities of Kibana 7.10.2 by incorporating a range of community-sourced add-ons and plugins, offering notable advantages in terms of compliance with SOC 2, CMMC, and NIST standards.

6. EXPERIMENT AND RESULTS

The Elasticsearch ecosystem is explained in detail in earlier sections. This section demonstrates the usage and the capabilities of Elasticsearch, Filebeat, and Kibana with real world use cases of anomaly detection using machine learning capabilities of Elasticsearch. Subsequent subsections detail out the experimental setup, the use cases, and the results. The results are properly explained along with the screenshots from Kibana.

6.1 Experimental SETUP

The developed architecture, as presented in Figure 3,

showcases a system dedicated to collecting, analyzing, and visualizing data from a web server. Within the proposed system, a client-server architecture is implemented, with the Apache web server being responsible for serving requests received from web applications. Load is generated on the Apache web server by hitting different URLs periodically. Filebeat service with the Apache module has been deployed and configured to retrieve both Apache access logs and error logs, which are then stored in Elasticsearch. Simultaneously, the data collection module gathers information about a machine's CPU utilization and memory usage. Kibana is utilized for data visualization through various graphs. The graphs depict CPU and Memory usage as a percentage, while the load on the Apache web server is represented by the number of requests served. Figure 4 illustrates the default dashboard provided by Filebeat. This dashboard has visualizations for access logs and error logs. This setup demonstrates how the ELK ecosystem, Filebeat, and the Apache plugin can be employed to gather data from the Apache web server. This experimental setup also highlights the ecosystem's ability to collect telemetry and logs from a variety of systems with Apache web server being a sample data source.

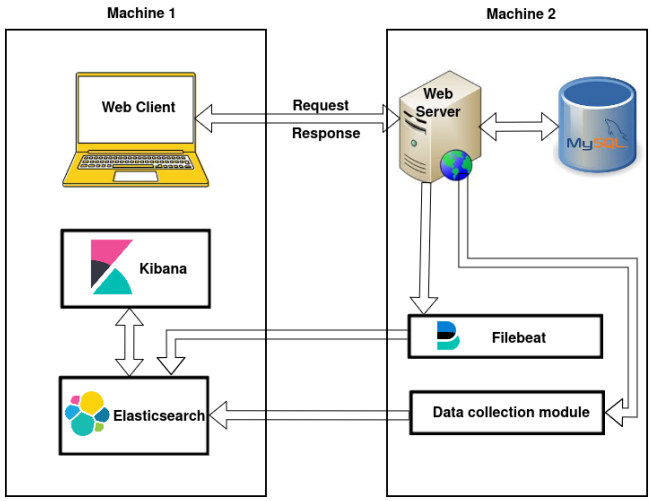

Figure 3. Architecture for data collection and visualization

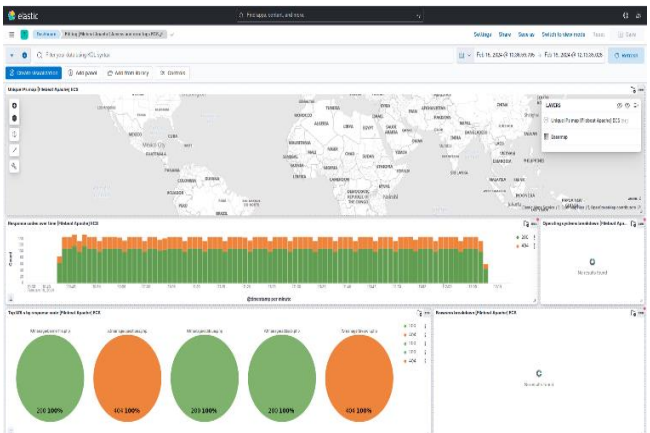

Figure 4. Filebeat dashboard

6.2 Dataset for anomaly and outlier detection

Elasticsearch comes with built-in machine-learning capabilities. Notably, it supports anomaly and outlier detection out of the box. A dataset consists of CPU utilization, memory utilization (both in percentages), the number of requests served in 5 seconds, and the number of bytes transferred in 5 seconds. Such data samples are collected for over a month and 2 data samples in every hour are altered by increasing CPU and/or memory utilization. Table 1 gives the details of data collection. While altering the values of CPU and memory utilization we took care to keep this alteration in a range such that the modified value is still a valid value i.e. CPU and memory utilization is always in the range of 10% to 100%. This way, there are approximately 2 anomalies per hour. Figure 5 shows this data in a Kibana dashboard.

Table 1. Details of dataset

Components	Details
Data collection frequency	Once every 5 seconds
Total Samples	5,50,000
Anomaly injection frequency	Once every hour (or one in every 12 samples)
Total Anomalies	1500

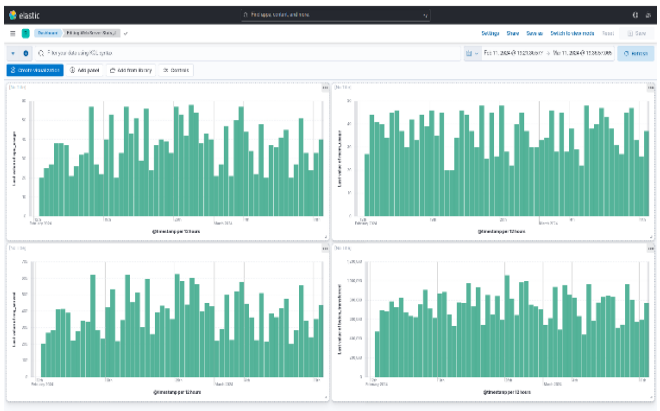

Figure 5. Kibana dashboard

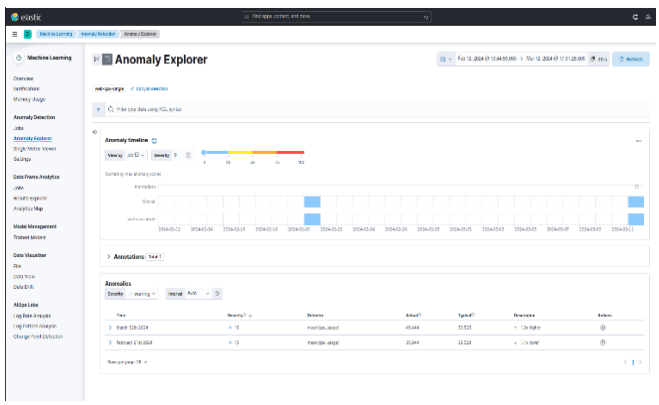

Figure 6. Anomaly explorer

6.3 Anomaly detection

In the experimental setup, a single metric anomaly detection job is created. Single Metric Anomaly Detection refers to a technique employed to observe a particular time series, which consists of a sequence of data points arranged chronologically, in order to identify any anomalies or deviations from expected patterns of behavior. We selected mean CPU usage as a metric because this is one of the attributes that was altered after data collection to inject anomalies. The anomaly detection job, on a single metric, could find only a couple of anomalies, the

results are shown below in Figure 6.

Even though there are approximately 1500 anomalies, the single metric anomaly detection could detect only 2. This could be because when we look at a single metric value like CPU usage, the values are all within the normal range. The anomaly can only be detected when CPU usage is correlated with other values like requests served and bytes transferred. So next we tried the outlier detection capabilities of Elasticsearch.

6.4 Outlier detection

An outlier detection job was created with all four fields, and 512MB of memory was allocated to it. Upon successful completion, the outlier job writes data into another index, and along with the original fields in the dataset, it adds the information about the outlier score to it. The higher the outlier score, the more the chances of the record being an outlier/anomaly. Then the filter is applied on the outlier score to be greater than or equal to 0.99, and it correctly identifies exactly 2 outliers per hour. Figure 7 shows the outliers over time for a selected day.

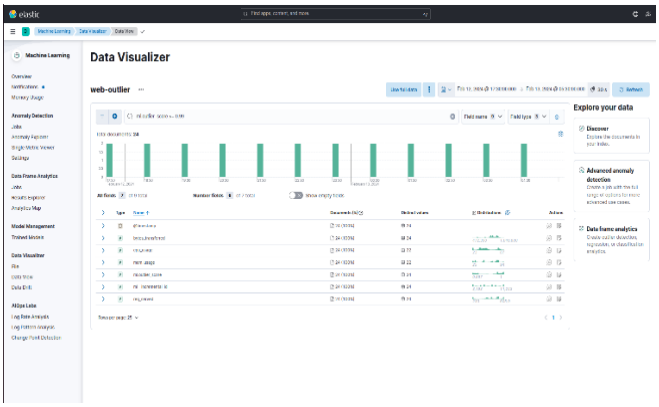

Figure 7. Outliner detected for given day

Table 2. Comparison of ML features in Elastic Stack

ML Feature of ES	Total Samples	Anomalies	Anomalies Detected	Accuracy
Single Metric detector	550000	1500	2	<1%
Outlier detector	550000	1500	1500	100%

6.5 Comparison of ML features

Table 2 shows the comparisons between the single metric anomaly detector and outlier detector. It is evident from the experiment that single metric anomaly detection failed miserably to detect injected anomalies. On the other hand, an outlier detector, which worked on all four parameters, detected all the anomalies. This could potentially be because a single metric doesn't have enough variation to indicate anomalies. As the injected anomalies have all the values in valid range for every parameter, the detector can detect anomalies only when it is fed all four parameters. As a future work, these algorithms should be tested with a variety of data samples, including the data where there is at least one parameter, which can show anomaly on its own.

7. CONCLUSION

This paper explained the entire ecosystem of elasticsearch which has tools to collect, store, visualize, and analyze data. These tools are used and the various capabilities of these arrays of tools are also demonstrated. It is found that Filebeat, Elasticsearch, Kibana and various data analytics capabilities are very easy to deploy and use. This ecosystem works very well for unstructured data and metric data, but may not be a good choice if there is structured data. RDBMSs like PostgreSQL and MySQL will be a better choice if the entire data is structured. Moreover, a couple of machine learning tools are used and it is found that in the experimental setup, the outlier detection algorithm detects outliers/anomalies with 100% accuracy but the single metric anomaly detection doesn't work well on the dataset which has multiple attributes which are correlated. With this research and analysis it is concluded that Elasticsearch, along with its ecosystem, is a formidable tool for any security analyst or a data scientist in general. The machine learning capabilities are also easy to use but they are not part of the free version, and there may be equally good open source ML frameworks present outside of Elasticsearch. In future, other complex datasets should be used with varying degrees of anomalous traffic/data to test machine learning capabilities of Elasticsearch. Work also needs to be done to use Elasticsearch to detect anomalies in non-metric data (like logs). It will also benefit the research community to compare Elasticsearch machine learning capabilities with other open source/free ML frameworks.

REFERENCES

[1] Talabis, M., Martin, J. (2013). Chapter 3 - Information security risk assessment: Data collection. Information Security Risk Assessment Toolkit, Syngress, 63-104. http://doi.org/10.1016/B978-1-59-749735-0.00003-8

[2] Continuous Security Monitoring, 2013. https://www.qualys.com/forms/whitepapers/continuous-security-monitoring/.

[3] Admass, W., Munaye, Y., Diro, A. (2024). Cyber security: State of the art, challenges and future directions. Cyber Security and Applications, 2: 100031. http://doi.org/10.1016/j.csa.2023.100031

[4] Taylor, R., Ali, M., Varley, I. (2018). Automating the processing of data in research. A proof of concept using elasticsearch. International Journal of Surgery, 55(1): S41. http://doi.org/10.1016/j.ijsu.2018.05.179

[5] Tsung, C.K., Yang, C.T., Yang, S.W. (2020). Visualizing potential transportation demand from ETC Log Analysis using ELK stack. IEEE Internet of Things Journal, 14(8): 6623-6633. http://doi.org/10.1109/JIOT.2020.2974671

[6] Gormley, C., Tong, Z. (2015). Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine. O'Reilly Media.

[7] Papadimitriou, G., Wang, C., Vahi, K., Silva, R., Mandal, A., Liu, Z., Mayani, R., Rynge, M., Kiran, M., Lynch, V., Kettimuthu, R., Deelman, E., Vetter, J., Foster, I. (2021). End-to-end online performance data capture and analysis for scientific workflows, Future Generation Computer Systems, 117: 387-400. http://doi.org/10.1016/j.future.2020.11.024

[8] Zamfir, V.A., Carabas, M., Carabas, C., Tapus, N.

(2019). Systems monitoring and big data analysis using the Elasticsearch system. In 22nd International Conference on Control Systems and Computer Science (CSCS), Bucharest, Romania, pp. 188-193. http://doi.org/10.1109/CSCS.2019.00039

[9] Ansari, M., Vakili, V., Bahrak, B. (2019). Evaluation of big data frameworks for analysis of smart grids. Journal of Big Data, 6: 109. http://doi.org/10.1186/s40537-019-0270-8

[10] Gutiérrez, S., Pérez Vera, Y. (2022). A cloud pub/sub architecture to integrate google big query with elasticsearch using cloud functions. International Journal of Computing, 21(3): 369-376. http://doi.org/10.47839/ijc.21.3.2694

[11] Hamilton, J.A.G., Gonzalez-Berges, M., Schofield, B., Tournier, J.C. (2018). SCADA statistics monitoring using the elastic stack (Elasticsearch, Logstash, Kibana). In 16th International Conference on Accelerator and Large Experimental Control Systems, Barcelona, Spain, pp. 451-455. http://doi.org/10.18429/JACoW-ICALEPCS2017-TUPHA034

[12] Calderon, G., del Campo, G., Saavedra, E., Santamaría, A. (2023). Monitoring framework for the performance evaluation of an IoT platform with elasticsearch and Apache Kafka. Information Systems Frontiers. http://doi.org/10.1007/s10796-023-10409-2

[13] Shah, N., Willick, D., Mago, V. (2022). A framework for social media data analytics using Elasticsearch and Kibana. Wireless Networks, 28: 1179-1187. http://doi.org/10.1007/s11276-018-01896-2

[14] Kononenko, O., Baysal, O., Holmes, R., Godfrey, M. (2014). Mining modern repositories with Elasticsearch. In Proceedings of the 11th Working Conference on Mining Software Repositories, pp. 328-333. http://doi.org/10.1145/2597073.2597091

[15] Takaki, O., Hamamoto, N., Takefusa, A., Yokoyama, S., Aida, K. (2023). Implementation of anonymization algorithms for log data analysis on a cloud-based learning management system. Procedia Computer Science, 225: 3774-3784. http://doi.org/10.1016/j.procs.2023.10.373

[16] Bagnasco, S., Berzano, D., Guarise, A., Lusso, S., Masera, M., Vallero, S. (2015). Monitoring of IaaS and scientific applications on the cloud using the Elasticsearch ecosystem. Journal of Physics: Conference Series, 608: 012016. http://doi.org/10.1088/1742-6596/608/1/012016

[17] The Digitization of the World from Edge to Core, 2018. https://www.seagate.com/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf.

[18] Turnbull, K.J. (2013). The Logstash Book, O'Reilly Media. https://www.oreilly.com/library/view/the-logstash-book/9780988820210.

[19] Ward, J., Barker, A. (2014). Observing the clouds: A survey and taxonomy of cloud monitoring. Journal of Cloud Computing, 3: 24. http://doi.org/10.1186/s13677-014-0024-2

[20] Boettiger, C. (2015). An introduction to Docker for reproducible research. ACM SIGOPS Operating Systems Review, 49(1): 71-79. http://doi.org/10.1145/2723872.2723882

[21] Filebeat modules. https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-modules.html, accessed on date May 2, 2024.

[22] Król, K., Zdonek, D. (2020). Analytics maturity models: An overview. Multidisciplinary Digital Publishing Institute, 11(3): 142-142. http://doi.org/10.3390/info11030142

[23] Garg, A., Goyal, D. (2019). Sustained business competitive advantage with data analytics. International Journal of Business and Data Analytics, 1(1): 4-15. http://doi.org/10.1504/IJBDA.2019.098829

[24] Kuć, R., Rogozinski, M. (2016). Elasticsearch Server. Packt Publishing Ltd. https://www.packtpub.com/en-us/product/elasticsearch-server-9781785888816.

[25] Hinman, M., Gheorghe, R., Russo, R. (2015). Elasticsearch in Action, 1st ed. Manning Publications Shelter Island: New York, NY, USA.

[26] Arya, D., Venkataraman, G., Grover, A., Kenthapadi, K. (2017). Candidate Selection for large scale personalized search and recommender systems. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. Association for Computing Machinery, New York, NY, USA. http://doi.org/10.1145/3077136.3082066

[27] You, G., Hwang, S. (2008). Search structures and algorithms for personalized ranking. Information Sciences, 178(20): 3925-3942. http://doi.org/10.1016/j.ins.2008.06.009

[28] Lin, Y., Jiang, J., Lee, S. (2014). A similarity measure for text classification and clustering. IEEE Transactions on Knowledge and Data Engineering, 26(7): 1575-1590. http://doi.org/10.1109/TKDE.2013.19

[29] Paro, A. (2015). [A2] ElasticSearch Cookbook - Second Edition. Packt Publishing. https://www.packtpub.com/product/elasticsearch-cookbook-second-edition/9781783554836.

[30] Hurwitz, J., Kaufman, M., Bowles, A. (2012). Applying advanced analytics to cognitive computing. Cognitive Computing and Big Data Analytics. http://doi.org/10.1002/9781119183648.ch6

[31] Inuwa, M., Das, R. (2024). A comparative analysis of various machine learning methods for anomaly detection in cyber attacks on IoT networks. Internet of Things, 26: 101162. http://doi.org/10.1016/j.iot.2024.101162

[32] Muñoz, L., Martínez, J., Pérez, F., Fonseca, I. (2024). Anomaly detection system for data quality assurance in IoT infrastructures based on machine learning. Internet of Things, 25: 101095. http://doi.org/10.1016/j.iot.2024.101095

[33] Ngiam, K., Khor, I. (2019). Big data and machine learning algorithms for health-care delivery. Lancet Oncol, 20(5): e262-e273. http://doi.org/10.1016/S1470-2045(19)30149-4

[34] Gaba, S., Budhiraja, I., Kumar, V., Makkar, A. (2024). Advancements in enhancing cyber-physical system security: Practical deep learning solutions for network traffic classification and integration with security technologies. Mathematical Biosciences and Engineering, 21(1): 1527-1553. http://doi.org/10.3934/mbe.2024066

[35] Ecma International, 2017. The json data interchange syntax (2nd edition). Standard ECMA-404. https://www.ecma-international.org/publications-and-standards/standards/ecma-404/.

[36] Shukla, P., Sharat, K. (2019). Learning Elastic Stack 7.0: Distributed Search, Analytics, and Visualization Using Elasticsearch, Logstash, Beats, and Kibana. Packt

Publishing Ltd.
[37] Collier, R., Azarmi, B. (2019). Machine Learning with the Elastic Stack: Expert Techniques to Integrate Machine Learning with Distributed Search and Analytics. Packt Publishing Ltd.
[38] Machine learning models in Elastic, https://www.elastic.co/search-labs/blog/may-2023-launch-machine-learning-models, accessed on date 10 May 2024.
[39] Dhakal, K. (2023). Log analysis and anomaly detection in log files with natural language processing techniques. MS Thesis. https://urn.fi/URN:NBN:fi:aalto-202310156380.
[40] Gorenstein, L., Konen, E., Green, M., Klang, E. (2024). Bidirectional encoder representations from transformers in radiology: A systematic review of natural language processing applications. Journal of the American College of Radiology, 21(6): 914-941. http://doi.org/10.1016/j.jacr.2024.01.012
[41] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Griselet, O. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12: 2825-2830. http://jmlr.org/papers/v12/pedregosa11a.html
[42] Azarmi, B. (2017). Learning Kibana 5.0. Packt Publishing Ltd. https://www.packtpub.com/en-us/product/learning-kibana-50-9781786463005
[43] Abbasi, A. (2020). Data visualization. AWS Certified Data Analytics Study Guide: Specialty (DAS-C01) Exam. http://doi.org/10.1002/9781119649489.ch5
[44] Taware, U., Shaikh, N. (2018). Heterogeneous database system for faster data querying using Elasticsearch. In F 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, pp. 1-4. http://doi.org/10.1109/ICCUBEA.2018.8697437
[45] Cruz, R., Guimarães, T., Peixoto, H., Santos, M. (2021). Architecture for intensive care data processing and visualization in real-time, Procedia Computer Science, 184: 923-928. http://doi.org/10.1016/j.procs.2021.03.115
[46] Hernández, R. (2022). Building IoT Visualizations Using Grafana: Power Up Your IoT Projects and Monitor with Prometheus, LibreNMS, and Elasticsearch. Packt Publishing. https://www.packtpub.com/en-in/product/building-iot-visualizations-using-grafana-9781803236124.
[47] Shollo, A., Galliers, R. (2016). Towards an understanding of the role of business intelligence systems in organisational knowing. Wiley-Blackwell, 26(4): 339-367. http://doi.org/10.1111/isj.12071
[48] Native Analytics on Elasticsearch With Knowi. https://www.knowi.com/blog/elasticsearch-analytics-tutorial/, accessed on date May 25, 2024.
[49] OpenSearch Dashboards, https://opensearch.org/docs/latest/dashboards/, accessed on date May 29, 2024.