

AI-Driven Log Forensics System For Intelligent Threat Detection

Dr. Neema George¹, Muhammed Ansif², Vivek M.V³, Muhammed Aslam⁴, Jithu Krishna N.R⁵

¹Department of Computer Science and Engineering Mangalam College of Engineering Kottayam, India

²Department of Computer Science and Engineering, Mangalam College of Engineering Kottayam, India .

³Department of Computer Science and Engineering Mangalam College of Engineering Kottayam, India.

⁴Department of Computer Science and Engineering Mangalam College of Engineering Kottayam, India

⁵Department of Computer Science and Engineering, Mangalam College of Engineering Kottayam, India .

Abstract— Fast, growing digital worlds plus smarter cyber attacks have made studying log files key to staying safe online. Because systems record what users do, how apps behave, and when something odd happens, those records help spot break, or misuse from inside. Even so, too much data coming too fast in too many formats overwhelms old, school tools, leading to slow responses, missed clues, or wrong alarms. To fix this mess, a new method uses artificial intelligence to gather, clean up, and examine different kinds of logs without constant human help. Hidden links and odd shifts in behaviour get spotted by smart pattern, finding tools built into the system. Instead of just following rules, it learns what usual activity looks like, then notices when something unfamiliar creeps in, even if it has never shown up before. While watching live data streams, it adjusts itself based on new threats, shifting its understanding without waiting for updates. Predictions about future risks come from trends pulled straight out of current signals, not guesses. Speed matters, so alerts fire quickly once a meaningful change appears.

Keywords—SVM, NER , BERT , LSTM, DNN , TF-IDF

Introduction

Nowadays companies depend heavily on digital systems, networked tools, online storage, apps working together, all making daily operations smoother but also opening more doors to cyber threats [1], [2]. Buried inside these systems are logs, quiet records tracking every click, message, program behaviour, and machine update. These traces form the backbone of how firms understand both their performance and potential dangers. From troubleshooting to spotting intrusions, such data quietly shapes decisions without drawing attention. From the start, these logs act like footprints left behind by system activity, each entry marking when something happened. They help spot what looks usual, plus catch actions that stand out unexpectedly. At times, they expose clear signs of harmful intent. Security experts rely on them daily. Their timeline format makes patterns easier to follow. Hidden shifts in behaviour often come to light only through careful review. Investigations gain direction because of this steady trail.

Even so, today's systems produce logs that are huge, mixed, and fast, moving. Old ways of checking them, done by hand or set rules, can't keep up when different kinds of logs pour in nonstop, leading to slow responses

Because threats often slip through, spotting them gets harder when alerts flood in. Systems miss warning signs while drowning in too many false alarms. Quick thinking tools are needed, ones that make sense of messy log records without delay. Machines now help sort this out, using smart pattern finding instead of old manual checks. Reading logs like a person does comes into play with language, aware models. Odd behaviors stand out clearer once routines learn what normal looks like. Gathering

scattered data points becomes smoother when automation handles the flow. These steps link together behind the scenes to track suspicious actions across different sources. Machines learn to tell harmless actions apart from harmful ones, using examples plus trial runs, spotting danger even if it looks new [12], [13]. Because threats change fast, these tools watch live data, adjust on the fly, while guessing what might happen next [14], [15]. Success now means spotting attacks early with sharp accuracy, not just cleaning up afterward. Logs stop being records of past events and start acting like shields, guarding complex online spaces before harm strikes [3], [4].

I. Related Work and Comparative Analysis

Across the globe, how people, businesses, and officials use technology has changed fast, creating tight links between systems that need data to stay available, private, and accurate [9]. Still, every new tech step opens more room for hackers, today's dangers go well past old, style viruses or fake emails, now including shape, shifting software, invisible programs hiding in memory, and smart tools built by machines [10]. Most standard methods used to investigate digital crimes started out focused only on what happened after a breach, often pulling full copies of hard drives, checking records, piecing together events, a slow process by design, always playing catch, up [4]. While such methods remain valuable in legal investigations, they are insufficient for contemporary cyber defense, as they lack the capability to provide timely intervention during active intrusions [13]. To address these limitations, recent research has proposed multi-layered analytical frameworks that begin with a conceptual understanding of digital forensics and associated cyber security challenges [9].

These frameworks increasingly incorporate artificial intelligence techniques, including unsupervised learning, time-series modeling, and neural classifiers tailored for cyber security data [11]. Furthermore, studies have examined system architectures and presented case studies that demonstrate the practical application of AI-driven forensic systems across diverse industries [12].

Looking closely means checking how logs move through systems, how models learn, then how they respond instantly when needed. This work tackles issues, unclear model decisions, attacks that fool algorithms, risks to private information, challenges in scaling across big environments, offering paths for companies to apply artificial intelligence during security events without breaking rules or values. Next comes pulling out details, turning raw signals into organized formats machines can process. Think about things like how large a data packet is, how often someone tries to get in, what commands show up in terminals, how chaotic a file appears, or how long people stay logged in. Starting off, feature engineering helps models adapt their understanding across varied settings, so spotting oddities or sorting trends becomes possible. Instead of just combining inputs, it shapes them into signals that shift meaningfully with context. Detection engines take form during model training, built using methods that learn from stored data or update constantly over time. When examples of attacks come tagged, supervision guides the learning path clearly. In contrast, unknown threats push systems toward unguided or partly guided approaches instead. Structures such as auto encoders, LSTMs, or transformers now handle sequences better than before, revealing hidden shifts in behavior over time.

What stands out in the design is how threat data flows right into the system. Feeding it aren't just homegrown logs but outside sources too, like malicious file fingerprints or hacker communication spots. Because of this mix, alert details gain depth, linking local

actions to worldwide patterns. Picture internal activity paired with live attack methods being tracked somewhere overseas. Built in blocks, the setup grows without breaking stride, fitting neatly into service, based or container, run environments. That shape allows pieces to shift without bringing everything down. One moment it's handling heavy loads, the next scaling back when things quiet down, each module adjusts on its own. When one piece stumbles, others keep moving without delay, staying sharp under pressure. Found deep within Security Operations Centers, they run either behind the scenes or right at the front lines. These systems stitch together scattered signals into a clear map of danger online.

After the logs have been standardized, AI models perform a review of these records to find deviations and also check for any patterns that can signal a compromise. The employment of deep learning in the understanding of logs that are similar to natural language has been very popular recently, Transformer models such as BERT help grasp what events really mean. Because they understand structure along with meaning, spotting subtle signs of trouble becomes possible even in massive or tangled log records. Next comes linking clues across different layers. Here, artificial intelligence shines by revealing hidden ties, say, matching strange login attempts on a cloud server with a thwarted effort to sneak data out using disguised DNS traffic. This kind of insight is beyond traditional rule chaining as it employs temporal patterns, access graphs, and user-device interaction sequences.

Right away, digital proof gets locked down when something odd shows up. Captured instantly, its tagged with a timestamp, secured through hashing, plus tucked into protected storage. Because machines handle this step, courts can trust the data hasn't changed. Integrity matters most once things go legal. Altered logs? Not likely, crypto shields or blockchain trails stop that cold. Fakes get blocked before they start. Every move leaves a mark that can't be erased quietly. Trust builds not by saying but showing every handoff stayed clean. No gaps, no guesswork in who touched what and when. The system holds the line so people later won't have to argue. Reports get simpler thanks to smart tools. Instead of raw numbers, you see clear stories about how the breach happened, what systems were touched, who did what, plus steps to fix things, all written out like a person would explain it. These reports can be customized for various audiences—from technical responders to executive leadership—improving communication and accelerating decision-making.

This study looked at how artificial intelligence fits into digital crime investigation, highlighting faster spotting of strange activity plus handling risks in current network defenses. Because old investigative ways struggle and attacks keep shifting, smarter tools that adapt became essential. Different machine learning types, like labeled training, pattern finding without labels, and layered neural networks, showed skill in mapping tricky behaviors while catching unknown dangers. Building smart forensic setups involved several steps: pulling in information, picking out key traits, teaching prediction models, then generating reports automatically.

Several real, world examples showed how the technology works inside security hubs, essential services, and modern cloud setups. To check performance, numbers like precision, recall, and AUC, ROC got attention. Privacy concerns popped up. So did questions about resistance to attacks and whether decisions can be understood. These topics received review too. Ideas pointing ahead included shared model training without data exchange, tracking actions for identity checks, self, running threat responses, shaping what digital investigation might become.

II. METHODOLOGY

A. Log Collection

What runs behind the scenes in security tools? A piece often called the Log Collection Module, it pulls data from many spots to show what's happening across computers and networks. Most key moments an operating system goes through get written down: who signed in, programs starting up, crashes, adjustments to settings. These traces appear different depending on the platform. On Windows machines, they live inside something known as Event Logs. Over on Linux or Unix setups, everything flows through Syslog, handling alerts from the core, services talking, unusual happenings. When all those entries come together in one place, spotting odd behavior gets easier. So does catching mistakes made during setup. Trouble at the deepest layers can surface fast. Beyond just machine, level records, this part also grabs logs from software running on systems. Network traffic details join the mix too. Together, these paint a clearer picture of how people use apps, where slowdowns occur, and patterns in digital conversations between devices..

When an app runs, it leaves behind records, web servers, databases, and business software each log actions like sign, glitches, or data exchanges. Firewalls, IDS tools, and proxies supply another kind: info on incoming connections, stopped threats, or live network movement. Pulling together these separate trails builds a full picture, one that feeds into later stages where risks get spotted, events linked, or past incidents studied. This richer context helps those next steps work better.

B. Real-Time Monitoring & Adaptive Learning

A sudden alert pops up when odd activity slips into the stream, logs flow straight from origin to processor without pausing. This live feed lets the software spot trouble right away, tagging suspicious patterns before they spread. Updates sneak in quietly behind the scenes, sharpening the model's guesses over time through quiet observation. Instead of waiting, responses form during analysis, shaped by what just happened moments earlier. Detection shifts subtly each day, nudged by fresh examples no one had seen before.

Because attacks keep changing, the system learns on the fly. Instead of staying fixed, its brain, built around machine learning, refreshes itself using fresh logs along with up, to, date threat reports. This means new tricks used by hackers get recognized faster, while mistakes like false alarms drop off. On top of that, experts see what matters through dashboards showing odd patterns, shifts over time, and how serious each alert might be. These views move beyond clutter, letting analysts act fast when speed counts. As things unfold, clues turn into awareness without delay. Learning never stops here; every insight sticks, helping defenses grow sharper day after day. Clarity shapes response, not guesswork.

C. Predictive Modeling & Threat Forecasting

Predicting future risks begins by studying past logs, where unusual activity hides in plain sight. Instead of waiting, the software spots odd rhythms in old entries, patterns repeating just enough to matter. These signals, once pieced together, hint at problems before they show up live. Hidden shifts in network flow or user behavior stand out when viewed over months, almost like echoes. Warnings pop up not because something broke, but because what's happening now doesn't fit how things used to run. That gap, the space between normal and strange, is where alerts are born. Teams get notice earlier, sometimes days ahead, simply because history tends to repeat unless interrupted.

Adjustments happen quietly: permissions shift, systems isolate, connections freeze, all before damage spreads. It works less like chasing fire, more like sensing heat before flames rise. Security stops reacting. Instead, decisions unfold based on what the numbers suggest might come next. Protection grows sharper not by adding tools, but by listening closer to what already happened. Breaches lose momentum because responses start long before entry points crack open. Infrastructure stays steady since disruptions meet resistance early, often while still forming. Defenses evolve from catching harm to seeing its shadow first.

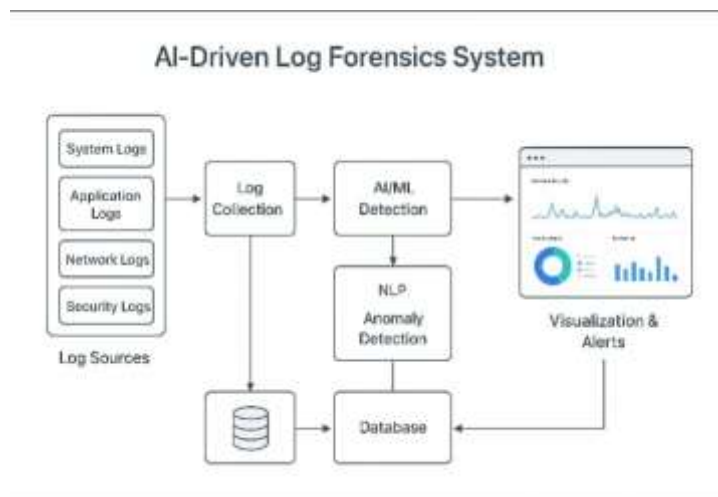
D. Forensic Evidence & Incident Response

When something unusual shows up, the moment it gets spotted matters just as much as what happens next. A clear path forms when every odd event leaves behind traces, exact times, machines involved, who did what, how data moved across networks, all saved without missing a single piece. Security folks then take these signals, slow down, look close, since not everything screams danger at the same volume. Sorting by impact helps decide which fires need water now versus those better studied later. Reports come after, built carefully so courts can follow facts, rules get met, teams inside the company understand what unfolded. Writing things down, weighing risks properly, sharing outcomes clearly, this rhythm turns chaos into control. Response becomes faster because steps repeat the same way each time. Trust grows when proof stays intact, audits go smoothly, defenses adapt quietly over time. The whole process holds itself together simply by never skipping pieces.

E. System Evaluation

What happens during testing shows if the AI log analysis tool works well enough. Not just speed but how often it catches real threats matters most. Instead of guessing, numbers tell whether alerts are correct or missed. Errors get counted so we know when warnings go wrong. Speed plays a role only after correctness proves stable. One way checks stand against older systems built on fixed rules. Another compares results next to similar smart tools already in place. Strengths appear where few others spot hidden attack patterns. Weak spots emerge whenever new logs look different than before. Coverage gaps reveal themselves under changing network behavior. Adaptation becomes clear once unknown breaches enter test runs. Results say what they say, no extra promises needed.

IV. SYSTEM ARCHITECTURE



V. SYSTEM MODULES

A. Log Collection

What keeps an eye on what happens across computers and networks. A piece of tech called the Log Collection Module, it pulls information from many spots so nothing slips through. This gathered info shows who signed in, when programs ran, if errors popped up, or settings got tweaked behind the scenes. On machines running Windows, those details live inside something named Windows Event Logs. Over on Linux or Unix boxes, a tool known as Syslog catches chatter from the core of the machine, services doing their jobs, and odd behaviour worth noting. When all these bits come together in one place, spotting trouble becomes easier, like wrong setups or silent breaches messing with the system's base layer. Another thing it does: pulling in

Logs from apps show what users do, how software runs, error events, plus login tries across web servers, databases, along with business tools. System records add deeper detail about each event happening inside machines. On top of those, network devices such as firewalls, IDS/IPS gear, and proxies keep track of data moving through, like attempted connections or denied packets that might hint at attacks. These streams together build a clearer scene than any single source could alone. Downstream tasks including spotting odd patterns, linking threats, or digging into past incidents rely heavily on this merged view. Without it, their work slows down or misses clues. The richer the log mix, the sharper the outcomes become over time. Each piece feeds insight others lack.

B. Preprocessing

Cleaning up raw log data happens first in the preprocessing stage. Through conversion into a common structure, timestamps and IP addresses gain consistency across varied inputs. Following that, formatting differences guide whether logs are seen as structured or not, shaping how they get handled later. Unwanted pieces like repeated records or meaningless actions vanish during filtering, clearing space for better results down the line. With those stages complete, the information turns tidy, reliable, ready to support deeper tasks without extra baggage.

C. Semantic Enrichment

Out of plain log entries, meaning begins to form when context gets added. Instead of just timestamps and numbers, details like where an event happened start to matter. One sign of something off? A login from a country never seen before. Think about IP addresses, they turn into clues once location and network owner are known. A sudden attempt at 3 AM makes more sense if you know who usually works then. Patterns emerge after watching how people act over time, what they open, when they show up. When someone accesses files in a different order than usual, alarms go quiet, but insights grow loud. Behind every odd click, there might be a mismatch between habit and action. Knowing the norm helps spot what does not fit, even if it looks harmless. From scattered lines in a file, behavior takes shape, piece by unnoticed piece.

Starting with messy log data, tools such as Word2Vec or BERT pull out meaning from words by learning how they're used. Because of this, patterns in event sequences start to show up more clearly. With those links visible, sorting entries into groups or spotting odd ones works far better. Instead of flat lines of text, each entry now carries background understanding. This shift turns raw output into something smarter, ready for deeper study, linking across systems, or digging into past issues.

D. Feature Extraction

Out of cleaned log data, numbers take shape, event tallies, how often errors pop up, sign, in tries, time spent per session, bits like those stand out when logs follow a pattern. Starting from messy lines of text, tools like Word2Vec or BERT turn words into tight number bundles where meaning sticks close even in code form. One part builds counts, another crafts vectors, but together they merge, feeding a unified snapshot for every recorded moment. That merged bundle becomes what machines later sort, scan for oddities, group, or link across streams. Every piece needed lands inside this format, lined up without gaps, ready to show what happened and how it connects.

E. Anomaly Detection

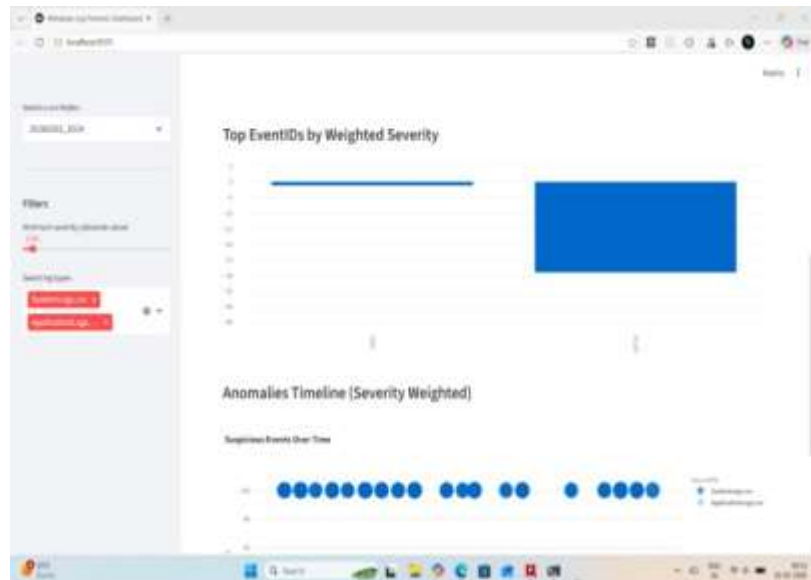
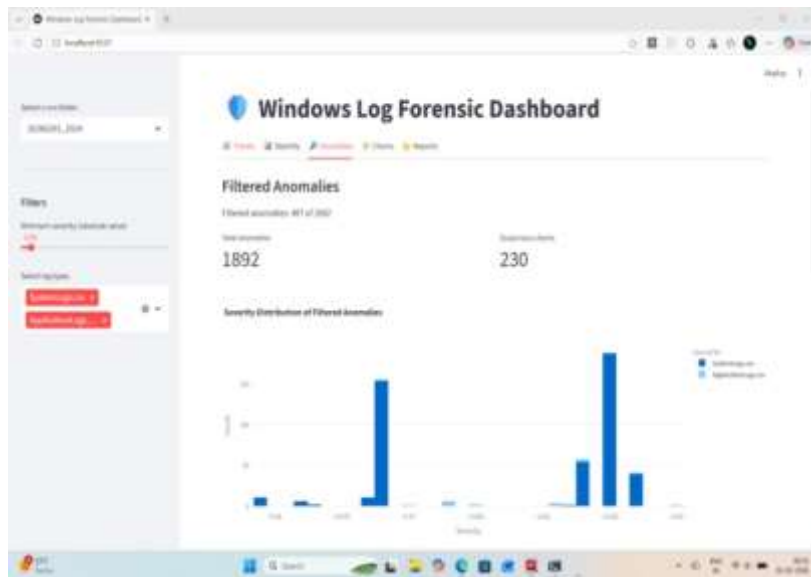
Something strange shows up when logs act out of line. Machine learning tools such as Random Forest or XGBoost learn from past data where each entry has been marked, good or bad, letting them sort fresh entries by what matches old danger signs. Instead of labels, some systems watch for odd habits without knowing ahead of time; Isolation Forests, Auto, encoders, and LSTM models fall into this group. They study how users and machines usually behave, building an idea of normal through repetition. When someone logs in at odd hours, traffic spikes with no reason, or actions stack in ways never seen before, alarms go off quietly. Not every risk follows a script, yet spotting these outliers helps catch unseen tactics lurking under the surface. Combining learned rules with outlier hunting makes it possible to flag threats whether they repeat history or try something entirely new. Speed matters because silence often means compromise.

RESULTS

Outcomes from the AI, Driven Log Forensics System mean turning massive streams of unprocessed log entries into clear, useful insights that strengthen defenses against digital threats. What comes out most clearly is its ability to spot odd behaviors, like login attempts without permission, strange flow in network usage, or errors happening way too often, by measuring live data alongside known normal activity. With machine learning working together with natural language analysis, it tells apart safe actions from harmful ones, spotting danger even if the pattern has never shown up before.

Not just spotting issues, the system pieces together clues from various logs, linking events to expose hidden dangers like sneaky employee actions. What stands out shows up clearly on an intuitive screen, mapping odd behaviors alongside clear steps to fix them. By weeding out misleading alerts and adding detailed context, it lightens the load for security teams. Responses become faster, smarter, because the tool delivers focused insights without extra noise.





VI. CONCLUSION

A new tool built on artificial intelligence changes how we look at digital logs during investigations. Because it uses smart algorithms plus natural language understanding, spotting odd patterns becomes faster and more precise. Instead of relying only on old ways of analysis, this method learns as it goes, sorting normal behavior from harmful actions without prior examples. While most systems lag behind emerging attacks, this one adapts continuously through real, time observation. Even when faced with chaotic or mixed, up data streams, its structure holds up under pressure. As threats shift shape over time, the model updates itself silently in the background. One moment it watches quietly, next it flags something unusual before damage spreads. Its strength lies not in speed alone but in seeing connections others miss. Through constant adjustment, what once seemed invisible now comes into view clearly. Digital spaces gain an alert companion that never stops paying attention. Progress here doesn't shout, it simply

stays ahead without pause.

One step ahead involves checking how well the ai setup really works under pressure. Spreading the early version across real, world network conditions happens next, different setups, mixed environments. Large logs come into play here, filled with both recognized attacks and unseen threats popping up for the first time. Detection precision gets measured closely, alongside how often mistakes happen and delays in processing data. Another priority takes shape around making the models logic clearer to humans. Investigators must follow why an alert triggered, needing transparent paths through decisions made by machines. When those choices become visible, trust builds among forensic teams. Clear trails allow quicker moves from flagged events to usable courtroom proof.

One big leap comes when systems stop just spotting threats and start acting on their own. Instead of waiting, logs feed straight into tools that respond instantly, cutting off bad actors before they spread. This shift means hacked devices get locked down automatically, not after someone hits a button. What used to take hours now happens in seconds, quietly, behind the scenes. Intelligence built into logging turns alerts into actions, no meetings, no delays. Security begins fixing itself, like a door that slams shut the moment it senses danger. Responses unfold in sequence: block, verify, repair, all while humans sleep. Such setups reshape how investigations work, pulling answers out of patterns machines notice first. Speed changes everything, the faster you react, the less damage takes hold. Digital defenses begin looking ahead rather than chasing what already slipped through.

VII. FUTURE SCOPE

One way to look at this system is how it handles logs across many parts of technology setups. Not only does it gather event records from places like apps and networks, but it works through them without needing constant human input. Because it processes information from cloud services along with regular machines, spotting strange behavior becomes more consistent. Instead of relying on old methods, the approach learns patterns from varied sources over time. What stands out is its ability to catch harmful software actions never seen before in similar systems. Through structured steps behind the scenes, unusual activity gets flagged even when details differ each time.

After that comes a method built on advanced pattern recognition, mixing labeled training approaches together with open, ended analysis, tied to language understanding tools, this combo digs through clear, messy, or half, formed records, spotting odd shifts and hidden links regular systems miss. Then there's the part that watches live activity while quietly updating itself; as dangers shift shape, it reshapes too, swapping out old rules without help, peeking ahead just enough to block moves before they happen.

What stands out is how smoothly things run when fewer errors pop up, less hands, on work is needed, then responses happen quicker. Another point worth noting, it slots neatly into cyber security setups. Yet the ways it uses, like spotting oddities, matching behaviors, forecasting outcomes, might just work elsewhere too, places swamped with logs or events. Think checking system health, ticking audit boxes, managing daily risks. Even though, at its core, it sticks to finding strange patterns and dangers within recorded data alone. From here, the setup can't catch breaches hiding beyond logged data. Instead of swapping out skilled professionals, it makes their work sharper. What stands clear is a goal: build something that grows without breaking, learns like a person might, thinks ahead when needed. This thing must scan records closely, spot dangers better than before, much quicker too. Main idea? Craft a tool broad enough to fit many cases, smart in its choices, ready to shift when systems change. It watches inquiries

carefully. From those traces comes the ability to uncover risks accurately, fast, and reach into today's complex online spaces.

REFERENCES

- [1] A new method uses artificial thinking to catch odd behavior right away. Instead of waiting, alerts happen during attacks. This approach helps teams react before damage spreads. Work by Ogochukwu Susan Ndibe shows how timing changes outcomes. By 2025, such tools could reshape defense strategies.
- [2] Simran Keshari, Mitali Srivastava, Cyber Attack Prediction: Moving from Traditional Machine Learning to Generative Artificial Intelligence, 2025.
- [3] Log Poisoning Attacks in IoT: Methodologies, Evasion, Detection, Mitigation, and Criticality Analysis, 2024.
- [4] Meta Log: Generalizable Cross, System Anomaly.
- [5] Detection from Logs with Meta, Learning, 2024. The Evolution Of Automated Log Analysis.
- [6] Anomaly detection Algorithms for Large Scale Real, Time Log Data Analysis, 2022.
- [7] Deep Learning for Anomaly Detection in Log Data: A Survey, M. Landauer et al., 2023.
- [8] Deep Learning for Anomaly Detection in Log Data: A Survey (arXiv preprint), Landauer et al., 2022.
- [9] A Study of Machine Learning Techniques for Log, Based Anomaly Detection, S. Ali et al., 2025.
- [10] Impact of Log Parsing on Deep Learning, Based Anomaly Detection, Z.A. Khan et al., 2024.
- [11] Log Anomaly Detection via Evidential Deep Learning, Y. Duan et al., 2024.
- [12] Host, based intrusion detection with multi, data source and deep learning, R.H. Hwang et al., 2023.
- [13] Machine learning, based intrusion detection framework for infrastructure, V. Kantharaju et al., 2024.
- [14] Evaluating machine learning, based intrusion detection with Explainable AI, V.Z. Mohale et al., 2025.
- [15] Determining the best features in log files for anomaly detection through the use of optimization techniques, Shivaprakash Ranga, 2025.