Journal of Superintelligence and Computing

CoSci Press

Article

Adaptive Defense: Enhancing NIDS with Smart Honeypots and Attack Profiling

Rachel John Robinson

IU International University of Applied Sciences, Berlin 10247, Germany
Corresponding author: info@rachel-johnrobinson.com

Article History: Received 15 March 2026; Revised 27 July 2026; Accepted 27 August 2026; Published 27 August 2026

Abstract: The paper examines the integration of the clever honeypots with the attacker behaviour analysis to enhance the network intrusion detection in the contemporary cyberspace environment. Due to the rapid development of cyber threats, the target of traditional intrusion detection systems (that are primarily based on fixed rules and known signatures) is to identify zero-day exploits, polymorphism, malware, and advanced persistent threats. In an attempt to circumvent these constraints, the study employs a quantitative paradigm that incorporates the survey of experts, simulated honeypot logs, and machine-learning-based behaviour analysis. Honeypot data around the world show trends in the frequency of attacks and ports attacked as well as the geographic origin and time. The 4 machine-learning models, Logistic Regression, KNN, Random Forest and XGBoost, were trained and tested on the feature-engineered datasets. The ensemble techniques performed well as compared to their linear counterparts with XGBoost recording 99.96 0-percent accuracy and Random Forest recording a 100 percent accuracy in the dataset. The findings demonstrate that intelligent honeypots do not only collect valuable behavioural indications, but also offer highly predictive attributes to automated detecting mechanisms. The research concludes that combining honeypot intelligence with machine learning improves real-time identification, proactive defence, as well as reducing the false alarms.

Keywords: intelligent honeypots; intrusion detection; machine learning; cybersecurity; behavioural analysis; network security; threat intelligence

1. Introduction & Rationale

A drop in intrusion detection systems (IDS) has been an issue despite the modern developments in cybersecurity, making it hard to identify the advanced and rapidly changing attacks promptly. In comparison to zero-day attacks, polymorphic malware, and advanced persistent threats, conventional techniques, i.e. signature-based detection and static firewalls, are based on predetermined rules and known attack patterns, which is why they are not as effective [1]. Such a weakness exposes organisations to information attack, service interruptions and financial setbacks. In addition to that, although honeypots have delivered potential results in terms of deception-based system, the research on how intelligent honeypots can be applied together with behavioural analysis to formulate adaptive and real-time intrusion detection systems is not much. This is an important gap that needs to be bridged because organisations are increasingly relying on networked systems and are in need of proactive and automated solutions that would ensure that sensitive information is protected, and continuity of operations is maintained in dynamic and unstable cyber threat environments.

The research on improving network intrusion detection system using intelligent honeypots and examination of the attack behaviour has great academic and practical importance. In terms of academics, it bridges the gap between theoretical knowledge about cybersecurity and technical implementation of both as a means of tracing critical thinking and problem-solving. The new way of obtaining precise information concerning the attackers based on the implementation of intelligent honeypots provides the insights into the modified approaches of the intrusion which are disregarded by the conventional approaches [2]. In practise uses, the research accumulates network analysis, programming and machine learning, hence making it attainable to make adaptive detection models. The results will have a contribution to the enhancement of intrusion detection systems, which will provide proactive measures of mitigating threats. The research is also relevant to the global cybersecurity interests as it contributes to digital risk management efforts and assists organisations to protect sensitive information in more complicated and aggressive network structures. Unlike many existing studies that use honeypots primarily as passive deception tools or apply machine learning models only to static benchmark datasets, our approach uses honeypot-generated behavioural evidence to enrich NIDS detection capability and support dynamic attack profiling Specifically, the proposed framework introduces three novel aspects. First, it incorporates smart honeypots as active intelligence-gathering components that capture attacker behaviour, interaction patterns, and attack traces beyond conventional network traffic features. Second, the framework links these honeypot observations with NIDS outputs to build richer attack profiles, enabling improved contextual understanding of malicious activities. Third, the system is designed as an adaptive defence mechanism in which profiling results can inform detection refinement and future response strategies. These aspects distinguish the proposed work from prior approaches that treat honeypot deployment, intrusion detection, and attack classification as separate processes.

This research paper will attempt to examine how intelligent honeypots, together with studies on attacker behavior, can enhance network intrusion detection by improving real-time threats, enabling modifications, and the overall performance of the cybersecurity systems.

- ✧ To investigate the position of intelligent honeypots in identifying unauthorized network accessibility and intrusion efforts.
- ✧ To study the attacker practices and large patterns of intrusion obtained via honeypots.
- ✧ To determine the level of intrusion detection success using honeypot against the use of conventional network security mechanisms.
- ✧ To use Python to analyze data with the purpose of determining

trends and insights based on data gathered on network and attacker behavior.

✧ To make useful recommendations related to the need to enhance the network intrusion detection systems and threat response plans through research results.

The current study will be directed at improving the mechanism of network intrusion detection by making use of smart honeypots deployment and monitor the actions of attackers. It explores the technical aspects in terms of data collection, analysis the machine learning model assessment as well as the theoretical knowledge on intrusion patterns and strategies to combat cybercrime [3].

2. Literature Base

2.1. Evolution of Network Security and Intrusion Detection Systems (IDS)

Network security has changed a lot over the years where cyber threats have become more complex. IDS has been developed step by step in order to improve the overall detection of attacks that protects system from harmful activities. In the early stage of computer networking, security methods are basic where the organizations are mainly using simple passwords, firewalls along with access control system. These tools have been focused on blocking several unwanted traffic from entering the network. When cyberattacks are increased then researchers have created the first intrusion detection system that looks at network traffic and also compares it with known attack patterns [4]. The early IDS is signature-based that means it could also detect attacks that had been seen before. It also made the security system both slow and reactive rather than proactive. Refer Figure1.

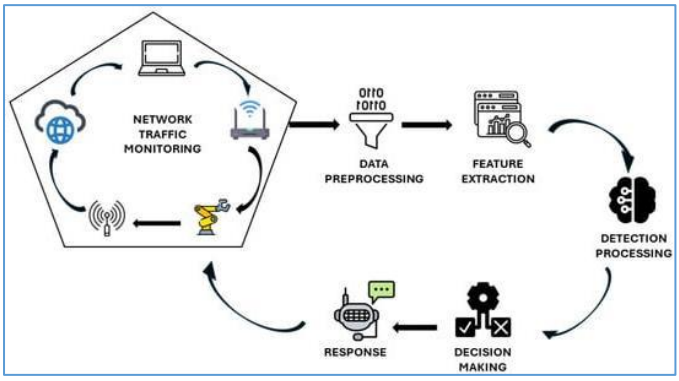


Figure 1. Architecture of a network intrusion detection system [4].

2.2. Concepts and Types of Honeypots

A honeypot is a computer system along with a network resource that is created in order to look real but is actually a controlled environment for monitoring attackers. It does not perform a real business function that instead it acts as a trap [5]. Two main types of honeypots have been discussed in the session below,

- Low-interaction honeypots: These imitate for the basic parts of the system including simple services and ports. It is easy to deploy, safe to manage and helps to detect automated attacks including scanning and brute force attempts [5].
- High-interaction honeypots: It provides attackers with more realistic environment, and it allows a deeper interaction along with collected detailed information about the attacker's behaviour [5]. For more effectiveness, it has been required a strong security control in order to prevent misuse.

Honeypots play several important roles, and they can capture attacker activities including commands, tools along with methods during intrusions. It helps organization in order to identify both new and unknown threats where that traditional security tools may miss. Honeypots help to reduce several false positives as any activity has been directed at them is usually considered suspicious [6]. In addition, they mainly support early warning systems through alerting defenders about several ongoing attacks. The honeypot strengthens several network security measures by providing deep insights into the attacker techniques that can be used to improve the overall intrusion detection, along with response strategies.

2.3. Machine Learning Techniques for Intrusion and Behaviour Detection

Machine learning plays an essential role for modern intrusion detection as it can learn patterns from the data that helps to identify threats more accurate rather than the traditional method. It helps to detect both known and unknown attacks through analysing network traffic and attack behaviour that is collected from honeypots and monitoring tools as in the Figure 2 [7].

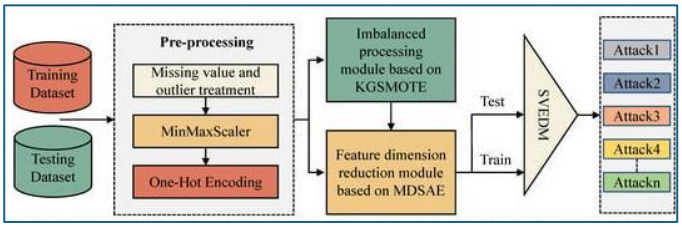


Figure 2. Machine learning-based intrusion detection system [7].

In supervised learning, models have been trained using labelled data where both normal along with malicious activities are identified. The common supervised techniques such as random forest, support vector machine along with logistic regression [8]. These models learn from past attacks where it can classify different new activities, and they work well where enough training data is present. The unsupervised learning does not require labelled data, and it also finds both unusual and abnormal patterns within the network. Several techniques such as K-Means clustering, Isolation Forest, and Auto encoders, helps to detect several unknown threats, Zero-day attacks along with strange behaviour. These methods are useful in an environment where the attacker's behaviour is constantly changing.

2.4. Cyber Deception and Proactive Defence Models in Modern Security Architecture and Gaps

The cyber deception works as a modern security strategy that has been focused on confusing attacks and also guides them away from the real system. Also, it helps to create fake system that real to the attacker. Multiple aspects such as honeypot, honey tokens, along with decoy network are common example. These tool helps organization to observe attacker behaviour, delay attacks along with collect valuable information without exposing critical system. The proactive defence model go beyond both traditional and reactive security approaches. The proactive defence aims to predict, detect and respond to threats [9]. It includes continuous monitoring, behaviour analysis, threat intelligence along with AI-based detection. The proactive defence model helps to identify unusual activity before it has become a serious issue. The modern security architecture often combines cyber deception with several proactive defences as in Figure 3. This combination makes the network environment more complex and difficult for attackers. When attackers interact with the deceptive elements, defender gains time

in order to analyse the attack and strengthen their security measures [10]. It helps to reduce the overall chances of a successful intrusion by forcing attacker in order to waste resources on the fake system. The cyber deception along with the proactive defence make the security system smarter and more adaptive that is handling the advanced cyber threats.

Honeypot are valuable tools for intrusion detection where several research gaps still exist. Many current honeypots are limited in how much they are able to learn and adapt to new threats. They often have been required several manual configurations and fully respond in order to change attacker behaviour. Another major gap is the lack of strong integration between honeypots and machine learning models [11].

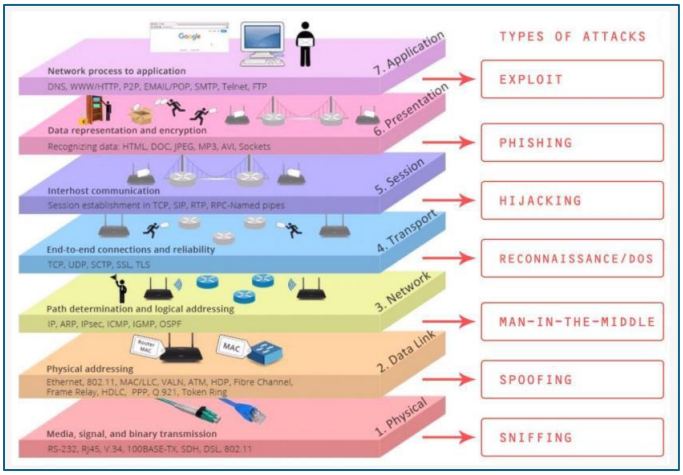


Figure 3. Modern system architecture [11].

Honeypot data has been collected that is used in order to train an intelligent detection system. There is limited research on how the honeypot performs and how it works. Also, the future trends show that the honeypots have become more intelligent, automated and connected.

3. Methodology

The research design is a quantitative and an empirical research design that best suits the analysis of measurability of trends in network intrusions as well as in estimating the effectiveness of detection systems. Quantitative methods allow the researcher to collect numerical data, compare the functioning of systems, and apply the statistical tests. These tests increase the validity of the findings, which is needed in the sphere of cybersecurity since the patterns of attacks, system response, and machine learning outputs are to be quantified in a reliable and objective way [12].

3.1. Data Collection Methods

First, the initial step of data collection was to design a structured questionnaire with the help of Google Forms. The survey focused on people having the appropriate know-how or expertise in cybersecurity, including security analysts, ethical hackers, network technicians and students enrolled in cybersecurity-related courses [13]. This was done to collect informed views on some of the most important aspects of intrusion detection. The questions concerned perceptions of honeypot use, problems commonly encountered when identifying intrusions, experience of attacker behaviour patterns, and familiarity with machine learning tools in a cybersecurity setting. Additional items were included to capture views on false positives, response speed, alert reliability, and the usefulness of behavioural indicators in supporting automated detection. These responses helped to contextualise the technical log analysis with expert judgement from practical cybersecurity environments. The structure of the survey was made in such a way that the questions were well-articulated, succinct and well-focused on the research objectives. To secure the privacy of participants and obtain genuine responses, all the submissions were made anonymously, and no personal and identifiable data were recorded [14]. The second element of data collection was to create behavioural logs where a simulated intelligent honeypot environment was used. This was a controlled environment which was meant to resemble real-life network situations and entice malicious activities without exposing any actual systems. Some of the data gathered by the honeypot pertaining to attacks included recurring bad connection attempts, suspect IP logs, bad authentication attempts, commands used by invaders, as well as the general patterns of attacks [15]. These logs were very useful in understanding the methods used by attackers and interaction behaviour. The foundation of the machine learning workflow in Python and based on the honeypot was the created dataset, which allowed the research to assess the patterns and determine the performance of various intrusion detection models.

3.2. Analytical Approach

The analysis of the data is provided in a systematic workflow in Python, which reflects the respective software processes presented in the implementation. The methodology involves data cleaning, machine-learning modelling, and visualisation to provide the full picture of the behaviour of attackers and their intrusion detection performance. This study will select three common machine-learning algorithms: Logistic Regression, Random Forest Classifier, and Support Vector Machine (SVM). Such models are typically used in cybersecurity studies and perform well when classifying. All models were trained using the created training data and tested using the reserved data. To assess the performance, some evaluation measurements were employed, and they included accuracy, precision, recall, F1-score and a confusion matrix [16]. There was the accuracy that involved the overall rate of correct prediction, precision, and recall that checked the manner in which the model would predict malicious activity without creating the high rates of false positive and false negative. The F1-score was a weighted score that incorporated both precision and recall [16]. Confusion matrices gave a graphical disaggregated information on prediction results and allowed establishing in which areas the models had strong or poor results.

The results of the performance were then displayed in clear graphical representations developed by the visualisation libraries of Python, primarily Matplotlib and Seaborn, after training and evaluation. They consisted of confusion matrices of each model, charts of comparison of accuracy, and graphs of the importance of features of the random forest model. These images simplified the examination of behavioural patterns, comparing the performance of the model, and getting a sense of which aspects contributed most to realising malicious traffic [15,16].

3.3. Limitations of the Methodology

Despite the use of a systematic research design and data-driven approach, a number of limitations should be recognised. The answers in the survey are absolutely based on the self-report on the knowledge of the participant, and this might not be accurate depending on the personal experience and education. Also, the data of the honeypot utilised in the study was produced by a controlled simulation and therefore it cannot be as complete as the real-world Advanced Persistent Threats (APTs) can be [17]. With these types of attacks, they tend to be multi-stage, where they hard

to reproduce in a restrained condition. Lastly, the outputs of the machine learning models are directly related to the size and quality of the dataset. A more varied or bigger dataset has the potential to result in better model stability and more generalisable results.

3.4. Data Handling and Ethical Consideration

Data used in this study was generated by the honeypot and was generated in a controlled environment that was isolated from any type just to carry out research. There was no exposure to risk of any real systems, networks or users in the data generation process. This was done through a simulation environment, in which all the malicious activities, which are repeated logins, IP scans or command injection, were limited and could not be extended to an external testing environment [17]. This practice will be consistent with the ethical cybersecurity research practice since the experimental traffic of an attack should not interfere with or disrupt other systems or networks.

Respondents who participated in the survey were well put in touch with the scholarly conduct of the research that the data collection was intended to perform. They were made clear enough that they could take part voluntarily, and they could avoid responding to any such question, which they felt uncomfortable with. Participation, however, was not influenced by any incentives or pressure [18]. This informed consent proceeded to ensure that the informed consent assured that the participants knew their rights, the purpose of the research, and willingly donated their contribution to the study.

4. Results

4.1. Data Loading and Processing

The chapter starts with the presentation of the features and organisation of the cleaned data, the number of records, types of variables, and the elimination of the entries. After it, the findings of descriptive statistical analysis are presented, which point out the trend of the protocol usage, the commonly targeted destination ports, attack distribution per hour, the geographical sources of attackers, and the activity of honeypot nodes. Outputs obtained after the K-Means algorithm are also clustered to indicate geographic clustering tendencies among the attackers. The last part of this chapter shows the machine learning findings, such as the evaluation scores, the confusion matrices, and the comparisons of performance of the Logistic Regression, the Random Forest, the KNN, and the XGBoost models.

Figure 4 shows the first stage of the data analysis process, during which the honeypot dataset AWS_Honeypot_marx-geo.csv is loaded into Python, with the help of Pandas. The code loads the file, changes the column of dates and times into a correct format of date and prepares the dataset into the process of further analysis.

```
df = pd.read_csv("AWS_Honeypot_marx-geo.csv",
parse_dates=["datetime"], dayfirst=True)
```

Figure 4. Load the dataset.

Figure 5 shows the duplicate removal procedure used on the honeypot dataset. The dataset consisted of 451,581 rows before cleaning it; however, in an inspection of the dataset with the help of the df.duplicated() function, 108,030 instances were observed to be duplicates. These duplicate entries were possibly caused by having the same attack event being logged in, or the same system entry. After running drop duplicate, the new dataset was reduced to 343,551 rows of unique entries, a fact that all the found duplicates were actually removed. A much clearer representation of the statistical overview of some of the most important columns in the honeypot dataset is now seen in the Table 1. Such a cleaning measure is required because various entries can distort the statistical tendencies, inflate the number of attacks, and it can negatively affect the precision of machine learning. The management of special events reduces the size of the dataset and makes it more desirable and applicable in the behavioural analysis and learning the model. Table 1 shows numerical and categorical information and makes it understandable how attacker activity is distributed and what its features are. The field of data begins in January till December 2013, and there is continuous logging throughout the year. The host column shows that there were nine distinct honeypot servers, with Groucho-Tokyo getting the most attacks. The source values denoting the source IPs expressed in numbers depict such a varied set that the geographical source of such attacks is made by numerous nations. When it comes to fields like country geolocation, China has the highest number of results, indicating that there is a great concentration of attack attempts in that region. Quantitative spheres spt (source port), dpt (destination port), latitude and longitude exhibit various behaviour of the attackers and geographical distribution. It has complexity and regularity on the profiles of attackers. The large numerical spreads over the port fields are indicative that attackers are not targeting only one port and rather combining focused exploitation with random scanning. The fields of latitudes and longitudes have a significant geographic dispersion, which supports the fact that the attacks are perpetrated by infrastructures that are distributed all over the world, and not a single territory. These observations present an adequate perception of the complexity of the dataset. They validate that the honeypot collects heterogeneous attempts of intrusion hence capable of analyzing its behavior meaningfully and deriving threats based on machine-learning models.

```
print("Before removing duplicates:", df.shape)

# Count duplicates
duplicate_count = df.duplicated().sum()
print("Number of duplicate rows:", duplicate_count)

# Drop duplicates
df = df.drop_duplicates()

print("After removing duplicates:", df.shape)
print("Duplicates removed:", duplicate_count)

Before removing duplicates: (451581, 16)
Number of duplicate rows: 108030
After removing duplicates: (343551, 16)
Duplicates removed: 108030
```

Figure 5. Remove duplicates.

Table 1. Data description.

	Date/time	host	src	proto	type	spt	dpt	srcstr	cc	Country	Locale	localeabbr	postalcode	Latitude
count	451581	451581	451581	451581	44811	466770	406770	451581	447985	447947	342112	331705	86478	448112
unique	NaN	9	NaN	3	NaN	NaN	NaN	69602	176	177	1179	612	2777	NaN
top	NaN	groucho-to-kyo	NaN	TCP	NaN	NaN	NaN	175.146.199.252	CN	China	California	CA	91789	NaN
freq	NaN	126189	NaN	327991	NaN	NaN	NaN	18472	191394	191394	37266	37278	9305	NaN
mean	2013-06-18 14:17:46	NaN	2154902000	NaN	7.514896	18685.46149	6684.258212	NaN	NaN	NaN	NaN	NaN	NaN	36.214115
min	2013-01-04 00:00:00	NaN	16777250	NaN	5	0	0	NaN	NaN	NaN	NaN	NaN	NaN	-43.5333
25%	2013-04-22 05:36:00	NaN	1169129000	NaN	8	6000	445	NaN	NaN	NaN	NaN	NaN	NaN	30.6667
50%	2013-06-21 15:05:00	NaN	2031191000	NaN	8	6000	1433	NaN	NaN	NaN	NaN	NaN	NaN	NaN
75%	2013-08-13 22:56:00	NaN	3164988000	NaN	8	33461	3389	NaN	NaN	NaN	NaN	NaN	NaN	NaN
max	2013-12-08 23:58:00	NaN	3758094000	NaN	13	65535	65500	NaN	NaN	NaN	NaN	NaN	NaN	NaN
std	NaN	NaN	1081011000	NaN	1.799866	19290.14175	14032.8398	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Duplicates are also eliminated by enhancing the statistical integrity of the given dataset. This will make sure that the future analyses like feature engineering, clustering and classification are based on real attacker behaviour as opposed to exaggerated repetitions. The logs might be duplicated, and this exaggerates some attack vectors resulting in misguided conclusions. As such, the purged dataset provides a more realistic picture of intrusion attempts, which enables machine learning models to be trained on patterns of real variations, as opposed to a product of false scheduling artifacts brought about by duplicate logging.

Figure 6 shows the 20 most frequently targeted destination ports in the honeypot data used by attackers. The bar chart is extremely skewed, with port 1433 (Microsoft SQL Server) holding a large portion of the attack scenario, with more than 100,000 attempts. This implies a high automated scan of database vulnerabilities. In line with the interest of attackers on the remote access services and file service, ports 3389 (RDP) and 445 (SMB) are next. Other popular ports attacked are 8080, 3060 and 22 (SSH) and 80 (HTTP) ports, similar to the common web-based attacks and brute force. The rest of the ports depict decreasing numbers. Overall, the chart shows the open preference of the attackers towards the broadly implemented and possibly exploitable network services.

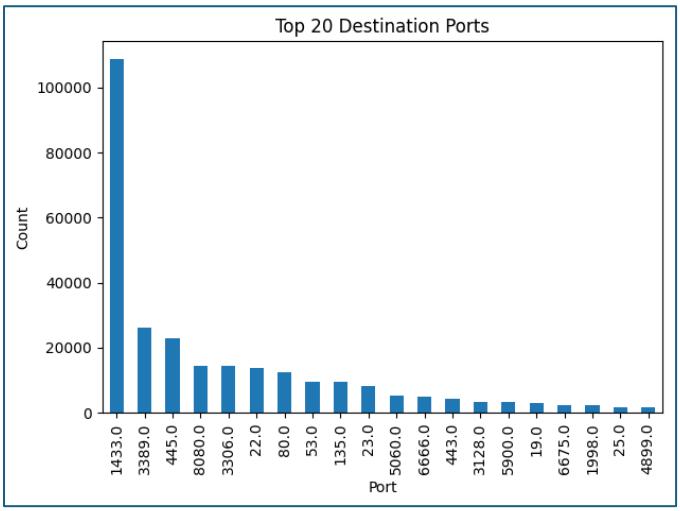


Figure 6. Top 20 destination ports.

Next in Figure 7 shows the correlation heatmap that illustrates the relationships between the key numeric features produced during the feature engineering step. The rest of the variables have weak or almost zero focus, which shows that they have an independent contribution to the data. It seems remarkable that there is a strong positive correlation (0.97) between latitude and latlon_mag as a result of the magnitude calculation involving the latitude value. The dpt (destination port) variable is also exhibiting moderate negative correlation (−0.31) with its common service, indicating that common service ports are not targeted as easily as random ports. Altogether, the heatmap assists with the process of redundant or impactful features detection prior to the machine learning model training.

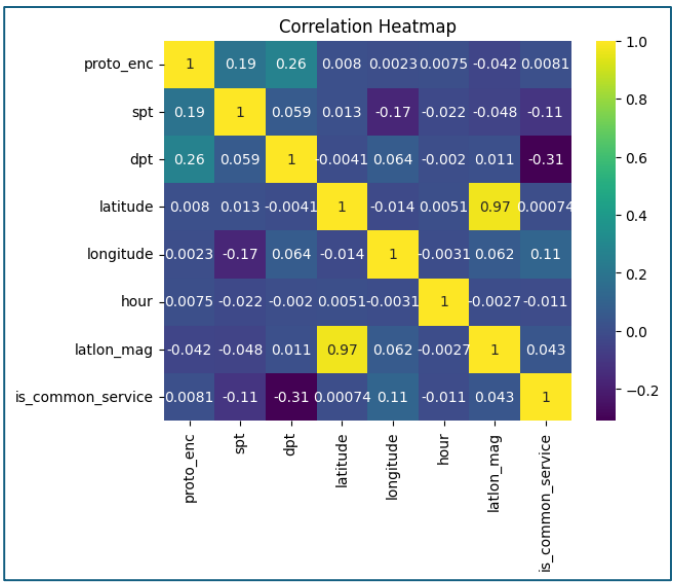


Figure 7. Correlation heatmap.

Figure 8 shows the trend of attacks per hour on the honeypot during the year 2013. The line plot indicates consistent changes in the number of attacks, with the majority of the hourly rates being between 50 and 150. Some noticeable spikes can be seen through the timeline, especially in April, June, October, and December, as the volume of attacks is extremely large, reaching 300 and even 500 in a few cases. Spores of free falls to zero display when the honeypot had no copies of attacks recorded at that time, as there may have been a system reset or network failures. In general, this chart indicates the enduring character of intrusion attacks throughout the year and their irregular character.

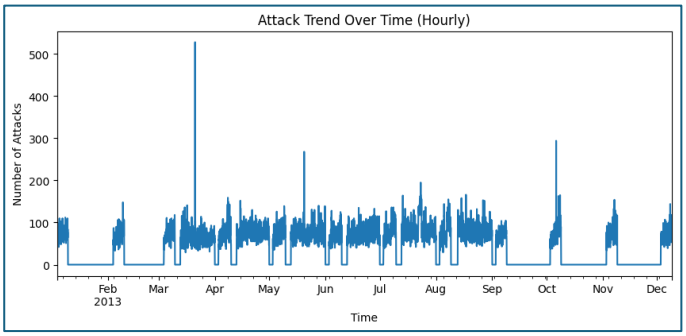


Figure 8. Attack trend over time.

Figure 9 indicates the number of attacks registered in the various honeypot nodes set up in various areas around the world. It can be seen in the chart that there is an apparent unbalance in the frequency of attacks between nodes, with Groucho-Oregon receiving the largest number of attacks at over 85,000, followed by Groucho-Singapore and Groucho-Tokyo receiving almost 65,000 events. These areas can be seen to form major targets, perhaps because of the network exposure or relative geographic location with respect to the active sources of threats. Other nodes like groucho-us-east, zeppo-norcal, groucho-eu, groucho-sa and groucho-sydney have much lower yet constant counts of attacks, around 20,000. On the whole, the number shows that the level of activity of attackers differs in different places, and geographic distribution determines the level of involvement in the honeypots.

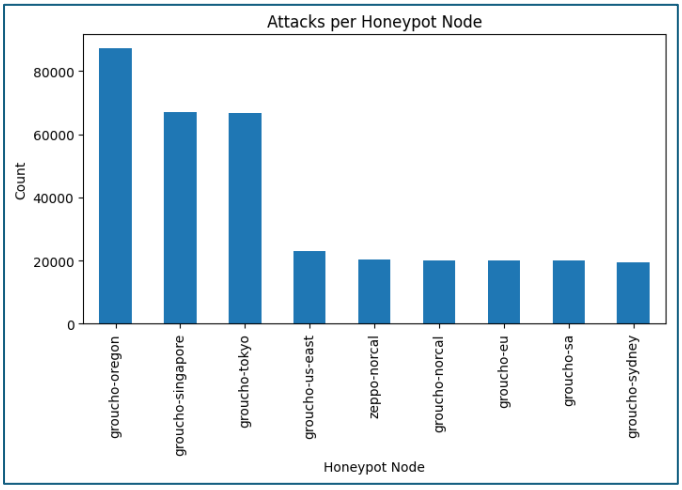


Figure 9. Attacks per honeypot node.

4.2. ML Models Comparison

In Figure 10 compares the four machine learning models, which are Logistic Regression, random forest, KNN, and XGBoost, based on their performance in terms of their accuracy, precision, recall, and F1 score. The weakest overall performance is made towards Logistic Regression, which has evidently lower precision and accuracy. KNN works well in all metrics, but a bit lower than XGBoost. Random Forest and XGBoost have almost perfect scores, with the former showing 100 in each category. The chart shows that ensemble-based models (Random Forest and XGBoost) are better as they can detect intrusion by complex attack patterns than simple classifiers.

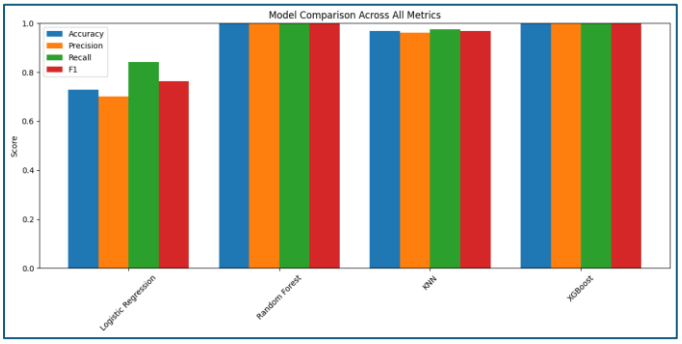


Figure 10. ML model comparison across all metrics.

In Figure 11 presents the rankings of feature importance produced by the Random Forest model, i.e. which variables played the biggest role in predicting attacks. The most common feature is the destination port (dpt), with over 60% importance, meaning that attackers are highly interested in particular ports. The source port (spt) and the encoded protocol (protoenc) are also meaningful, but with very trivial importance in the destination ports. Latitude, longitude, hour and latlon_mag are all features with only a modest contribution to feature classification. This figure demonstrates that port attributes are the most effective in the determination of malicious activity in the honeypot data.

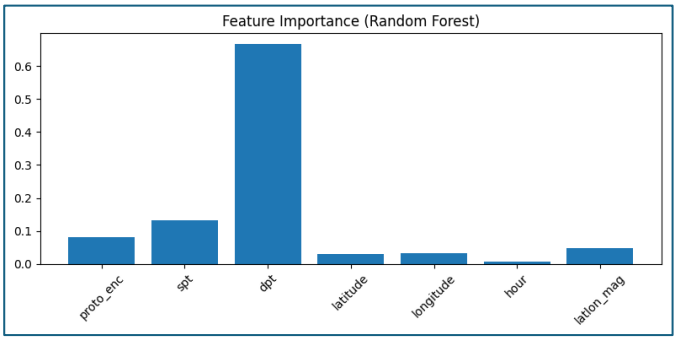


Figure 11. Feature importance (random forest).

Machine-learning results indicate that ensemble classifiers, i.e. Random Forest and XGBoost have higher predictive accuracy in comparison to the traditional classifiers; this point confirms the effectiveness of behavioural measurement based on the honeypot logs in the process of training highly predictive detection systems. Another issue highlighted in the manuscript is the necessity of incorporating the intelligence based on honeypots into intrusion detection systems to provide real-time responses and automation to counter emerging threats.

Though the use of honeypots at a strategic level, and behavioural-tracking and state-of-the-art artificial-intelligence models can significantly enhance the cybersecurity stance of organizations, thus turning an ancient reactive paradigm of defensive practice into a more active, rule-of-thumb.

5. Discussions & Conclusion

In summary of all outputs based on the dataset, survey responses, feature-engineering processes, and machine-learning models will be provided. There were 451,581 rows in the honeypot table initially, and using duplication, 343,551 unique entries were left after deduplication. The dataset consisted of sixteen variables, which were timestamps, IP data, ports, geolocation data, and protocol data types. The feature engineering resulted in more fields, including year, month, hour, protocol encoding, service classification, and the geolocation magnitude. The descriptive statistics reflected evident trends within the data. The protocol that was most common was TCP, with UDP and ICMP. The maximum ports destinations were 1433, 3389, 445, 8080 and 22. The number of days of the week showed equal distributions of the attacks analysed hourly. Different countries were registered as having great volumes of attack records on the geolocation data. There was differing traffic to the honeypot nodes, with certain nodes receiving a lot of traffic compared to others.

The results of the correlation indicated the existence of generally weak relationships among most engineered variables. K-Means clustering generated several visual geographic clusters of attack categories. The outcome of machine learning generated accuracy scores, precision, recall and F1-scores of each of the tested models. Logistic Regression demonstrated average results, whereas KNN demonstrated good results in all metrics. XGBoost resulted in almost flawless classification scores, whereas Random Forest scored equal scores in all categories of the evaluation.

This study is highly indicative that smart honeypots and machine learning are effective at detecting an intrusion. Nevertheless, it also draws the attention to various valuable directions of work in future. A future acquisition that could help to curb the effect of adaptive honeypots is the development of adaptive honeypots that would respond dynamically to an interaction with the attacker [19] Such systems might automatically open ports, here imitate the user active, or place decoy vulnerabilities under specific attacker behaviour [19]. Such practices would bring a greater focus to behavioural profiling and allow to identify advanced attackers employing targeted or manual exploitation methods. With external intelligence combined with honeypot data, have a more accurate

idea of what is emerging and a clearer correlation of behavioural indicators. Further improvements in artificial intelligence, particularly in reinforcement learning and federated learning could enable intrusion-detection systems to learn automatically, with little requirement of human interference or central datasets [20]. The exploration of these methods would result in enhancing the flexibility, effectiveness, and strength of network security.

To conclude, the paper has indicated that honeypots, behavioural analysis, and machine learning as a combination tool provides a very effective way of enhancing intrusion detection. However, due to the speed at which cyber threats are changing, research, innovation, and adaption are the only effective ways of ensuring effective cybersecurity in the future.

Funding: This research received no external funding.

Ethical Approval: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: None.

Conflicts of Interest: The author declares no conflicts of interest.

References

[1] Altunay, H.C. Analysis of Cyber Attacks Using Honeypot. *Black Sea J. Eng. Sci.* **2024**, *7*, 954–959.
[2] Andrade, C. The inconvenient truth about convenience and purposive samples. *Indian J. Psychol. Med.* **2021**, *43*, 86–88.
[3] Robinson, R.J. Deciphering Generative AI Risk Trends to Integrate Compliance as an Industry Practice. *Int. J. Sci. Eng. Manag.* **2023**, *10*.
[4] Franco, J.; Aris, A.; Canberk, B.; Uluagac, A.S. A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 2351–2383.
[5] Gao, Y.; Zhang, G.; Xing, C. A multiphase dynamic deployment mechanism of virtualized honeypots based on intelligent attack path prediction. *Secur. Commun. Netw.* **2021**, *2021*, 6378218.
[6] Heidari, A.; Jabraeil Jamali, M.A. Internet of Things intrusion detection systems: a comprehensive review and future directions. *Cluster Comput.* **2023**, *26*, 3753–3780.
[7] Iyer, K.I. Adaptive honeypots: Dynamic deception tactics in modern cyber defense. *Int. J. Sci. Res. Arch.* **2021**, *4*, 340–351.
[8] Kocher, G.; Kumar, G. Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges. *Soft Comput.* **2021**, *25*, 9731–9763.
[9] Kumar, S.; Gupta, S.; Arora, S. Research trends in network-based intrusion detection systems: A review. *IEEE Access* **2021**, *9*, 157761–157779.
[10] Moric, Z.; Mršić, L.; Kunić, Z.; Đambić, G. Honeypots in cybersecurity: their analysis, evaluation and importance. *Preprints.org* **2024**, 202408.0946.v1 (preprint).
[11] Morrison, B.; Coventry, L.; Briggs, P. How do Older Adults feel about engaging with Cyber-Security? *Hum. Behav. Emerg. Technol.* **2021**, *3*, 1033–1049.
[12] Priya, V.D.; Chakkaravarthy, S.S. Containerized cloud-based honeypot deception for tracking attackers. *Sci. Rep.* **2023**, *13*, 1437.
[13] Rehnbäck, O. A case study of unauthorized login attempts against honeypots via remote desktop. Master's Thesis, Luleå University of Technology, Luleå, Sweden, 28 July 2023.
[14] Robinson, R.J. Cybersecurity Compliance Frameworks-a pragmatic view with an IT outsourcing company case study. *Soc. Lens* **2024**, *1*, 15–25.
[15] Survey of intrusion detection systems: techniques, datasets and challenges. Available online: https://cybersecurity.springeropen.com/articles/10.1186/s42400-019-0038-7 (accessed on 14 November 2025).
[16] Steingartner, W.; Galinec, D.; Kozina, A. Threat defense: Cyber deception approach and education for resilience in hybrid threats model. *Symmetry* **2021**, *13*, 597.
[17] Robinson, R.J. Digital Security: A Critical Evaluation Process for IT-Security Products. *Math. Comput. Sci. Contemp. Dev.* **2024**, *6*, 18–30.
[18] Yang, Y.T.; Zhu, Q. When to Deceive: A Cross-Layer Stackelberg Game Framework for Strategic Timing of Cyber Deception. *arXiv* **2025**, arXiv:2505.21244 (preprint).
[19] Zhekova, M. An Algorithm for Exploratory Analysis and Normalization of Big Data with Pandas. *Proc. Bulg. Acad. Sci.* **2023**, *76*, 1716–1723.
[20] Zhou, S.; Wang, Y. Modality-Independent Disentangled Neural Architecture for Enhanced Artificial Intelligence in Electronic Information Systems. *Int. J. Adv. AI Appl.* **2025**, *1*, 63–80.



Copyright: © 2026 by the author(s). *Journal of Superintelligence and Computing* supported by Wuchang University of Technology and published by CoSci Press. This is an open access publication under the Creative Commons Attribution 4.0 International License (CC BY 4.0).