



ISSN: 2321-2152

IJMECE

*International Journal of modern
electronics and communication engineering*

E-Mail

editor.ijmece@gmail.com

editor@ijmece.com

www.ijmece.com

NETWORK TRAFFIC IDENTIFICATION BASED ON MACHINE LEARNING AND DEEP PACKET INSPECTION

MOUNIKA BEDADAM¹, BHAVYASRI GAYAM², KEERTHANA PASUPULETTI³, BHAVYA PENTYALA⁴
G.SRINIVASA RAO⁵

UG students^{1,2,3,4} Associate Professor⁵

ABSTRACT: Accurate network traffic identification is an important basis for network traffic monitoring and data analysis and is the key to improving the quality of user service. In this paper, through the analysis of two network traffic identification methods based on machine learning and deep packet inspection, a network traffic identification method based on machine learning and deep packet inspection is proposed. This method uses deep packet inspection technology to identify most network traffic, reduces the workload that needs to be identified by the machine learning method, and deep packet inspection can identify specific application traffic, and improves the accuracy of identification. The machine learning method is used to assist in identifying network traffic with encryption, new applications, and unknown features, which makes up for the disadvantage of deep packet inspection that cannot identify new applications and encrypted traffic. Experiments show that this method can improve the identification rate of network traffic.

I. INTRODUCTION

As networking technology advances rapidly, customers' expectations for network speeds and quality continue to rise. Therefore, it has become one of the challenges in network operation and maintenance management to

effectively manage and control different types of network business traffic, differentiate between services, offer varying levels of quality assurance, and cater to the business requirements of users.

ANURAG ENGINEERING COLLEGE
AUTONOMOUS
(Affiliated to JNTU-Hyderabad, Approved by AICTE-New Delhi)
ANANTHAGIRI (V) (M), SURYAPETA (D), TELANGANA-508206

Application-specific traffic may be easily identified on a network using network traffic identification. Classifying, identifying, and distinguishing the application of network traffic allows for the traffic of various applications to be subdivided, allowing for the provision of individualized network services, which in turn increases both the quality of network services and user happiness.

For starters, the port-based traffic identification approach has a straightforward implementation concept and does not need complex computation and analysis. It's fast enough to satisfy the needs of identifying high-speed networks quickly. The port-based traffic identification method has been gradually phased out of use as a result of the proliferation of new network applications, particularly P2P applications, which rely on random ports and camouflage ports to conceal their network communications.

Unlike other methods, which depend on the port Settings of the programmer, the feature-field based identification approach may determine the effective fields in the load. High accuracy in detecting network activity and individual applications is achieved. By focusing on the first few carefully selected

packets of network data, this approach may identify activity fast. Unfortunately, this technique can only recognize preexisting apps and not create new ones since it is dependent on the feature field of the application protocol. Furthermore, load encryption network traffic cannot be identified using this technique.

Using identification technology from the field of data mining, the machine learning identification method based on the flow statistics features is able to realize traffic identification through this technique, thereby resolving issues that the first two approaches were unable to tackle, remaining immune to port and protocol feature changes, and allowing for the discovery of previously unknown applications. However, this type of approach, which relies on machine learning for both Bayesian identification and SVM (support vector machine) identification, is unable to distinguish between different applications' needs; it also lags behind in detecting traffic because it depends on the type of multiple packet flows and is easily affected by flow length; below a certain length, the misdiagnosis rate is high. Furthermore, the accuracy of this identification approach is readily influenced by dynamic network

changes and traffic attribute sets; furthermore, the computationally demanding nature of this method makes it unsuitable for real-time traffic identification of high-speed networks.

According to the principle of feature field-based identification method and flow statistics-based machine learning method, a network traffic identification method based on machine learning and DPI technology is proposed based on the analysis and comparison of the aforementioned traffic identification methods.

II. LITERATURE SURVEY

Shi Dong, Zhou Ding, Ding Wei Abstract: Network traffic identification is one of the hot research fields for network management and network security. This paper describes the current situation and common methods of network traffic identification; at the same time, this paper also states the currently popular Machine learning methods. We compared and evaluated the supervised and unsupervised classification and clustering algorithms. Introduction: Network traffic identification is an important application research direction for network management and measurement, the current network traffic identification methods roughly can be

classified into four categories. Port-based method; DPI (Deep packets inspection); host behavior method; flow-based method based on machine learning. Related Work: It has become hot research between domestic and foreign experts who take traffic identification as research direction, which proceeds distinguish, intrusion detection, traffic monitoring, billing, and, management. From the beginning of the study port-based method, this method used well-known port numbers to identify Internet traffic. Results and Discussion: In this paper, we use the routine evaluation standard for verifying the effectiveness of our identification algorithm. The effectiveness of the current flow identification algorithm has the following three concepts evaluation criteria: TA (True Positive), FP (False Positive), TN (True Negative), FN (False Negative) Conclusion: This paper has studied and analyzed the machine learning algorithm for network traffic identification. Through experiments on the classification algorithm of different datasets, comparing the classical unsupervised and supervised algorithms. By comparing several unsupervised machine learning algorithms (cluster algorithm), results show that the DBSCAN algorithm

has great potential and has more advantages than the other two kinds of algorithms in precision, the modeling time is between the K-Means method and DBSCAN method.

Zuleika Nascimento, Djamel Sadok, Stenio Fernandes Informatics Center Abstract: Considerable effort has been made by researchers in the area of network traffic classification, since the Internet grows exponentially in both traffic volume and number of protocols and applications. Task of traffic identification is a complex task due to the constantly changing Internet and an increase in encrypted data. There are several methods for classifying network traffic such as port-based and Deep Packet Inspection (DPI), but they are not effective since many applications use random ports and the payload could be encrypted. Introduction: In this context, identifying network traffic is a complex task, since access to the Internet is significantly increasing, bringing with it new users with different goals. Many P2P applications are increasingly popular and accessible, such as eMule, Ares, and BitTorrent. The user's behavior is also changing and the growth of streaming video services is notable. To bring to the expert's attention what is flowing through a network is an increasingly important activity. Related

Work: Recently, some methodologies have been investigated as network traffic classification tools. To classify the network traffic, the clustering k-means algorithm is used and compared to other model-based clustering methods along with rule-based classification models. Associations were found among flow parameters for several protocols and applications, such as HTTP, Mail, SMTP, DNS and IRC. Results and Discussion: The results of accuracy on the Skype application proved to be superior in the OHM (100%) against a 94.97% rate for the HM, as well as for the other metrics (except for FP). The results for correctness (CR) have improve significantly, which clearly indicates the improvement of the model when classifying applications, with less packets being classified as unknown data. Rates of FP were better tackled by the OHM regarding both applications. Conclusion: This work proposed an Optimized Hybrid Model (OHM) based on computational intelligence techniques, consisting of a rule-based model and a self-organizing map (SOM) model, both optimized by the Firefly Algorithm (FA). since the architecture has the ability of being in a constant learning process to extract patterns from new applications or protocols.

Geza Szabo, Zoltan Turanyi, Laszlo Toka

Abstract: We present an automatic application protocol signature generating framework for Deep Packet Inspection (DPI) techniques with performance evaluation. We propose to utilize algorithms from the field of bioinformatics. We also present preprocessing methods to accelerate our system. Moreover, we developed several postprocessing techniques to refine the accuracy of the results. Finally, we propose a DPI system, based on approximate string matching, and find it a viable, novel alternative for the refinement of exact string-matching algorithm's results. **Introduction:** In-depth understanding of the Internet traffic profile is a challenging task for researchers and a mandatory requirement for most Internet Service Providers (ISP). Deep Packet Inspection (DPI) can aid to ISPs in the profiling of networked applications. With this information ISPs may then apply different charging policies, traffic shaping and offer different quality of service guarantees to selected users or applications. **Related Work:** Three types of protocol signature generation methods can be found in the literature: a) worm signature generation e.g., [18, 8, 10, 16], b) spam rule generation [2] and c) application signature

generation [12, 14, 17, 26]. Authors of [26] presented AutoSys which extracts multiple common substring sequences from sample flows as application signature. **Results and Discussion:** The case when the usual DFA is substituted with sequence alignment and motifs are used. It has the highest coverage in the function of the number of signatures. Regarding the FP coverage in Figure 5, it has similar characteristics to the M+R case and has the lowest among all methods. The motifs evaluated via ASM represent a theoretical maximum of the motif expressiveness. **Conclusion:** In this paper we present a general framework of an automatic application protocol signature generation for Deep Packet Inspection (DPI) techniques. The proposed framework utilizes algorithms from the field of bioinformatics. In the preprocessing phase we applied a Rabin-Karp fingerprinting-based method to filter the once-occurring substrings and a prefix tree construction method to summarize the substrings with common pre- and postfixes.

III. PROPOSED SYSTEM

A network traffic Classification model based on machine learning and DPI technology is proposed. This method uses DPI Technology to identify most network traffic and reduces

workload. The workload is reduced by machine learning methods. DPI technology can identify specific application traffic and improves the accuracy of identification and also improves the identification of network traffic rate.

Traffic Identification Results Based on DPI Algorithm Flow identification algorithm in CentOS system implementation, using Wireshark capture data in the campus local area network, then carries on the processing, only keep BitTorrent, PPStream such type of P2P traffic, and belongs to the WWW HTTP traffic, finally the flow identification method based on DPI and traffic identification method based on this model to analyze traffic data. As shown in "TABLE 1", the dpi-based identification method is significantly less sensitive to PPStream traffic than to BitTorrent traffic. This is because BitTorrent P2P file sharing software is open source and its protocol features can be easily found through analysis of its programs and application protocols. DPI technology can be used to identify the corresponding network traffic. For private commercial applications of PPStream, only the protocol features can be obtained by analyzing network packets and decompiling. The accuracy is limited to some extent,

which leads to the reduction of identification recognition rate of these traffic. To identify the network flows through the machine learning method which cannot be recognized by DPI, the traffic of BitTorrent and PPStream is judged as P2P traffic, which makes up for the deficiency of DPI recognition. As shown in "TABLE II", the traffic generated by BitTorrent and PPStream is judged to be P2P traffic. The identification method adopted in this study has

significantly improved the identification of P2P traffic like BitTorrent and PPStream by combining machine learning algorithm and DPI technology to detect network traffic, thus improving the overall identification rate of network traffic.

Protocol name	Actual flow size	Identified traffic size	Traffic identification rate	Actual connection number	Number of connections identified	Connection identification rate
BitTorrent	307660238	36864749	90.9%	213	240	94.1%
WWW	42945	8026	96.9%	20	20	100%
PPStream	3075960	3686734	70%	236	181	72.4%

Table. 1 Traffic Identification Results Based on DPI Algorithm

Protocol name	Actual flow size	Identified traffic size	Traffic identification rate	Actual connection number	Number of connections identified	Connection identification rate
P2P	130024025	10998029	81.2%	41	47	94.0%
WWW	42945	8026	96.9%	20	20	100%

Table. 2 Traffic Identification Results Based on This Algorithm

Failing to meet non-functional requirements can result in systems that fail to satisfy user

needs.

IV SYSTEM DESIGN

The system development life cycle adopted in this project is the waterfall model which is based on a sequential design process. This methodology was adopted because of the flexibility it offers to researchers in terms of the sequential approach to solving problems. This methodology certifies that each stage of the design process has been successfully completed and thus guaranteeing that all stages are completely dealt with.

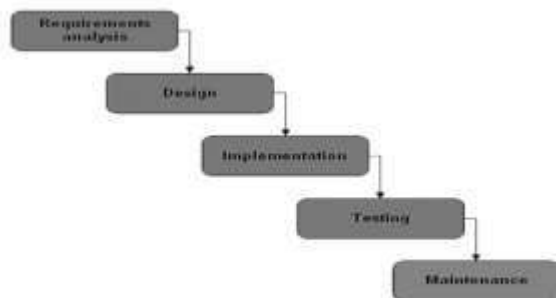


Figure 4.1: Waterfall model of the system development life cycle.

System Architecture:

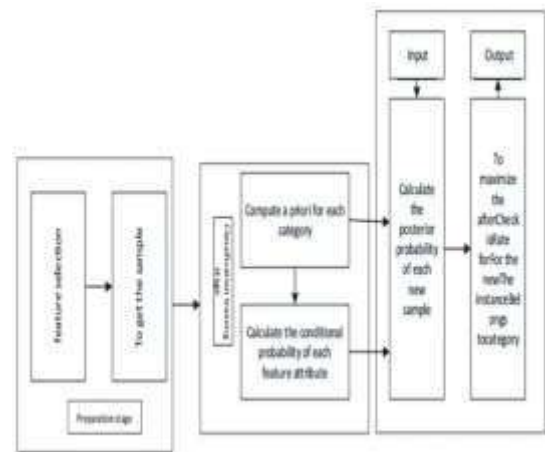


Figure 4.2: System Architecture

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users with a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modeling language.

5. Encourage the growth of the OO tools market.
6. Support higher-level development concepts such as collaborations, frameworks, patterns, and components.
7. Integrate best practices.

V SYSTEM TEST

AIM OF TESTING

The main aim of testing is to analyze the performance and to evaluate the errors that occur when the program is executed with different input sources and running in different operating environments. There are different types of approaches for testing a .NET framework-based application are;

- Unit testing
- Validation testing
- Integration testing
- User acceptance testing
- Output testing
- Black box and white box testing.

i. Unit Testing: This is the approach of taking a small part of testable application and executing it according to the requirements and testing the application

behavior. Unit testing is used for detecting the defects that occur during execution (MSDN, 2010). When an algorithm is executed, the integrity should be maintained by the data structures. Unit testing is made use for testing the functionality of each algorithm during execution. Unit testing reduces the ambiguity in the units in this project, we have developed an application using different phases like encryption, and decryption. So, for getting the correct output, all the functions that are used are executed and tested.

at least once so as to make sure that all the control paths, error handling, and control structures are in a proper manner.

Limitations of Unit Testing: This is limited to testing only the functionality of the units. It can't identify integration errors, performance problems, and system problems. Unit testing can show the errors which occur in the units when the testing runs. It may not display the errors that currently are absent. ii. Validation Testing: Validation is the process of finding whether the product is built correct or not. The software application or product that is designed should fulfill the requirements and reach the expectations set by the user.

Validation is done while developing or at the final stage of development process to determine whether it satisfies the specified requirements of user. Using validation test the developer can qualify the design, performance and its operations. Also, the accuracy, repeatability, selectivity, Limit of detection and quantification can be specified using “Validation testing” (MSDN, 2010).
Output Testing: After completion of validation testing the next process is output testing. Output testing is the process of testing the output generated by the application for the specified inputs. This process checks whether the application is producing the required output as per the user’s specification or not. The “output testing” can be done by considering mainly by updating the test plans, the behavior of application with different type of inputs and with produced outputs, making the best use of the operating capacity and considering the recommendations for fixing the issues (MSDN, 2010).

iii. Integration Testing: This is an extension to unit testing, after unit testing the units are integrated with the logical program. The integration testing is the process of examining the working behavior of the particular unit after embedding with

program. This procedure identifies the problems that occur during the combination of units. The integration testing can be normally done in three approaches.

- Top-down approach
- Bottom-up approach
- Umbrella approach

Testing/Predicting Stage:

In the testing/predicting stage of network traffic identification, the trained model is used to predict the protocol labels of unseen or incoming network traffic. Here's an overview of the steps involved:

1. **Data Preprocessing:** Similar to the training stage, the incoming network traffic data needs to be preprocessed to extract relevant features or attributes. This may involve extracting source and destination IP addresses, port numbers, packet size, time stamps, payload content, or other required information. Ensure that the preprocessing steps align with the preprocessing performed during the training stage.
2. **Feature Transformation:** Apply the same feature transformations and normalization techniques that were used during the training stage. This ensures consistency between the

features used during training and the features used during prediction. The transformed features should be in a suitable format to be passed into the trained model.

3. Applying the Trained Model: Pass the preprocessed and transformed network traffic data through the trained model. The model leverages the patterns and associations learned during training to predict the protocol labels for the incoming traffic. The model's output can be a single predicted label or a probability distribution over multiple possible labels, indicating the confidence or likelihood of each label.

4. Post-processing and Decision Making: Depending on the application and requirements, postprocessing steps may be performed on the predicted results. This could involve filtering, aggregating, or analyzing the predicted labels to gain insights or trigger appropriate actions based on the identified protocols. For example, the system may use the predicted labels to enforce network security policies, prioritize traffic, or trigger alerts for specific protocols.

5. Performance Evaluation: Compare the predicted protocol labels with the ground truth labels, if available, to evaluate the

performance of the model during the testing/prediction stage. Performance metrics such as accuracy, precision, recall, or F1 score can be calculated to assess the model's ability to correctly identify and classify the network traffic.

6. Iterative Improvement: If the model's performance is not satisfactory, adjustments can be made. This may involve retraining the model with additional labeled data, refining the preprocessing steps, modifying the model architecture, or tuning hyperparameters to enhance the prediction accuracy.



VI RESULTS

Data Loading:

	0	1	2	3	4	5	6	7	8	9	...	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
--	---	---	---	---	---	---	---	---	---	---	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

0 1000 2000 3000 4000 5000 6000 7000 8000 9000 10000 11000 12000 13000 14000 15000 16000 17000 18000 19000 20000 21000 22000 23000 24000 25000 26000 27000 28000 29000 30000 31000 32000 33000 34000 35000 36000 37000 38000 39000 40000 41000 42000 43000 44000 45000 46000 47000 48000 49000 50000 51000 52000 53000 54000 55000 56000 57000 58000 59000 60000 61000 62000 63000 64000 65000 66000 67000 68000 69000 70000 71000 72000 73000 74000 75000 76000 77000 78000 79000 80000 81000 82000 83000 84000 85000 86000 87000 88000 89000 90000 91000 92000 93000 94000 95000 96000 97000 98000 99000 100000 101000 102000 103000 104000 105000 106000 107000 108000 109000 110000 111000 112000 113000 114000 115000 116000 117000 118000 119000 120000 121000 122000 123000 124000 125000 126000 127000 128000 129000 130000 131000 132000 133000 134000 135000 136000 137000 138000 139000 140000 141000 142000 143000 144000 145000 146000 147000 148000 149000 150000 151000 152000 153000 154000 155000 156000 157000 158000 1

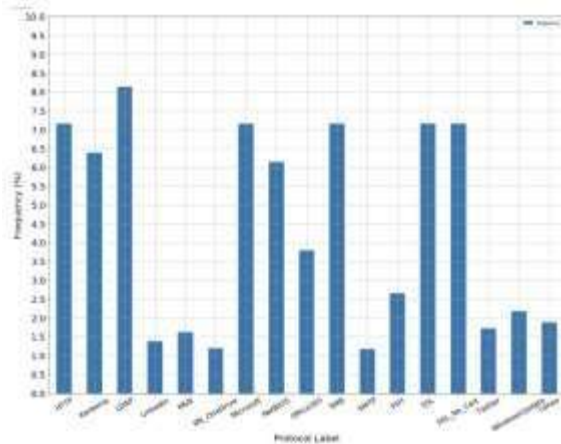


Figure 8.2: Protocol Frequency Distribution

Build a Sequential 1d-Convolutional Model:

Layer (type)	Output Shape	Param #
conv1d_2 (Conv1D)	(None, 1020, 2)	12
max_pooling1d_2 (MaxPooling1D)	(None, 510, 2)	0
dropout_2 (Dropout)	(None, 510, 2)	0
flatten_2 (Flatten)	(None, 1020)	0
dense_4 (Dense)	(None, 16)	16336
dense_5 (Dense)	(None, 8)	136
dense_6 (Dense)	(None, 24)	216
Total params: 16,700		
Trainable params: 16,700		
Non-trainable params: 0		

Figure 8.3: Build a Sequential 1d-Convolutional Model

Predict: Measure the Performance of the Model on the Dataset

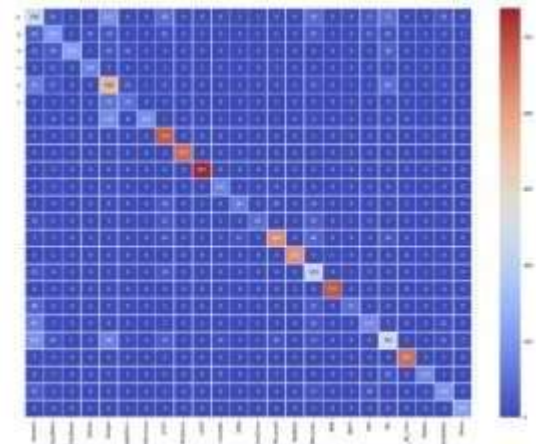


Figure 8.6: Predict: Measure the Performance of the Model on the Dataset

VII CONCLUSION

In conclusion, network traffic identification based on machine learning and deep packet inspection (DPI) is a rapidly evolving field that combines the power of advanced data analysis techniques with deep inspection of network packets. It aims to accurately classify and analyze network traffic for various purposes such as security, performance optimization, and policy enforcement. By analyzing two kinds of network traffic identification methods based on feature field and flow statistics, a network traffic identification method based on machine learning and DPI technology is proposed. This method uses DPI technology to identify most network traffic, reduces the workload that needs to be identified by the machine learning method, and improves the accuracy of identification. The machine

learning method based on the statistical characteristics of flow is used to assist the identification of network flows with encryption and unknown features, which makes up for the shortcomings of DPI technology in identifying new applications and encrypted traffic and improves the identification rate of network traffic. However, network traffic identification based on machine learning and DPI faces several challenges. These challenges include performance impact, handling encrypted traffic, dealing with protocol and application variations, addressing privacy concerns, mitigating evasion techniques, ensuring scalability, and complying with legal and regulatory requirements. To overcome these challenges and enhance network traffic identification, future advancements can focus on improving performance and scalability, effectively handling encrypted traffic, developing advanced feature extraction methods, exploring hybrid approaches, enabling real-time analysis and response, ensuring explain ability and interpretability, and building adaptive and self-learning systems.

REFERENCES

1. Kim, H., Han, S., & Lee, S. (2016). Deep packet: A novel approach to traffic classification. *IEEE Communications Magazine*, 54(10), 126-133.
2. Vilalta, R., & Ma, S. (2004). Learning and detecting anomalous network traffic. *IEEE Intelligent Systems*, 19(4), 42-49.
3. Wang, T., Li, B., & Xu, J. (2018). An overview of deep packet inspection for intrusion detection systems. *Security and Communication Networks*, 2018.
4. Baek, S. W., Kim, H., Han, S., Lee, S., & Hur, J. (2017). Deep packet inspection-based traffic classification using statistical flow information. *IEEE Transactions on Information Forensics and Security*, 12(1), 106-119.
5. Kong, Y., & Li, Q. (2019). Deep packet inspection and traffic analytics for improving network security. *IEEE Communications Surveys & Tutorials*, 21(1), 275-299.
6. Gao, Z., Han, S., & Kim, H. (2018). Traffic classification using deep learning: Systematic literature review. *IEEE Access*, 6, 23321-23334.

7. Kim, Y., & Shin, S. Y. (2020). A survey on deep learning-based network traffic identification. *Sensors*, 20(20), 5930.
8. Di Mauro, A., & Mazzocca, N. (2019). Machine learning techniques for traffic classification in network monitoring: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 21(3), 2668-2694.
9. Bernaille, L., & Teixeira, R. (2006). Traffic classification on the fly. *ACM SIGCOMM Computer Communication Review*, 36(2), 23-26.
10. Zhang, C., Jiang, Y., Chen, Y., & Zhang, Q. (2018). A novel network traffic identification method based on deep learning. In *2018 International Conference on Smart Internet of Things* (pp. 12-16). IEEE.